Utilization and Protection of Personal Data in Ubiquitous Computing Environments

George Drosatos
Department of Electrical and Computer Engineering
Democritus University of Thrace

Advisor: Pavlos S. Efraimidis

A thesis submitted for the degree of
Doctor of Philosophy
Xanthi, July 2013

Democritus University of Thrace
Department of Electrical and Computer Engineering
Building A, ECE, University Campus - Kimmeria, 67100 Xanthi, Greece

I would like to dedicate this thesis to my parents.

Acknowledgements

Since I started my PhD studies in November 2007, I have been fortunate to be surrounded, both at work and in life, by a number of extraordinary people, some of whom I knew from before, others whom I met in the course of my work or the places it took me. Without the direct or indirect support of these people I would never have completed this dissertation. Therefore I would like to express my gratitude to all of them.

I would like to express my heartfelt gratitude to my advisor Prof. Pavlos S. Efraimidis who tirelessly helped me to prepare my doctoral thesis. His guidance helped me during the time of research and writing of this thesis. Furthermore, I would like to thank Prof. Avi Arampatzis and Prof. Ioannis N. Athanasiadis for our excellent cooperation in the research we performed together which forms part of this dissertation. Also, I want to thank the other members of my examining committee, Prof. Alexandros S. Karakos, Prof. Sokratis Katsikas, Prof. Vassilis Tsaoussidis, Prof. Dimitrios Gritzalis, Prof. Vasilis Katos and Prof. Avi Arampatzis, for agreeing to read, comment on and judge my thesis.

I would also like to express my gratefulness to my colleagues Ellie D'Hondt, Fotis Nalbadis, Giorgos E. Stamatelatos, Matthias Stevens and to my friend and great fellow student Aimilia Tasidou for the excellent work we did together.

Most of all, I want to thank my girlfriend Chryssa, without whose constant encouragement, loyal support, patience and love I would probably not have found the courage to complete this challenging task.

Last but not least, I am grateful to my parents, Constantinos and Aggeliki, for their constant support in all its forms.

Abstract

The advances in information and communication technologies (ICT) and the wide acceptance of electronic transactions for everyday tasks of individuals have a strong impact on the use and protection of personal information. Desktop and mobile computing technology, the World Wide Web, sensors, and the advances in database and storage technologies have increased the amount of personal information that is generated and the potential for this information to be (permanently) stored and processed. Any kind of personal information that results as an outcome of electronic activities of individuals, either personal or professional, belongs to the category of personal data. Personal data is a critical and valuable resource that has to be protected in order to ensure the individual's privacy rights.

In this dissertation we investigate how this personal data can be managed at the user side and simultaneously can be used in privacy-preserving applications without violating individuals' privacy. Each individual has the right to protect his privacy by retaining control over his personal data and knowing who, when and why gets access to his data. At the same time, individuals, as well as the society as a whole, may obtain significant benefits if personal data can be used legitimately for beneficial purposes. We propose an approach for privacy-preserving computations and apply this approach to representative applications. These applications make use of cryptographic primitives and are based on secure multi-party computations (MPC's). Every privacy-preserving application is implemented by a prototype and experimental results are presented to illustrate the feasibility of our approach.

Apart from the management and usage of personal data we investigate how the web searches of individuals can be protected against search engine query-logs and simultaneously the target search results can be retrieved, without submitting the intended query. We model the problem theoretically, define a set of privacy objectives with respect to web search and investigate the effectiveness of the proposed solution with a set of real queries on a large web collection.

Extended Abstract in Greek (Περίληψη)

Η πρόοδος σε τεχνολογίες πληροφοριών και επικοινωνιών έχει οδηγήσει στην παραγωγή προσωπικών πληροφοριών και παρέχει τεράστιες δυνατότητες σε αναδυόμενες νέες εφαρμογές που μπορούν να χρησιμοποιήσουν τα προσωπικά δεδομένα προς όφελος των ατόμων. Μερικά παραδείγματα, είναι εξατομικευμένες διαδικτυακές υπηρεσίες που προσαρμόζονται αυτόματα στο προφίλ του ατόμου και location-based υπηρεσίες που συμπεριφέρονται σύμφωνα με τη τρέχουσα τοποθεσία του ατόμου. Ωστόσο, η χρήση των προσωπικών δεδομένων πρέπει να γίνεται με τρόπο που να εξασφαλίζει ταυτόχρονα την προστασία τους. Για την προστασία των προσωπικών δεδομένων, πολλοί οργανισμοί και χώρες έχουν εκδώσει κανονισμούς ιδιωτικότητας, οι οποίοι θα πρέπει να ακολουθούνται προκειμένου να διασφαλιστεί η προστασία των προσωπικών πληροφοριών. Συλλογικά αυτοί οι κανονισμοί αναφέρονται ως Fair Information Practices (FIP). Μερικά σημαντικά παραδείγματα τέτοιων FIP κανονισμών είναι το Data Protection Directive 95/46/EC και ακολουθούν κάποιοι άλλοι, όπως η καναδική PIPEDA και το Data Protection Act (DPA) του Ηνωμένου Βασιλείου. Με βάση τις διατάξεις περί απορρήτου, το κάθε άτομο έχει το δικαίωμα προστασίας της ιδιωτικότητας του, διατηρώντας τον έλεγχο πάνω στα προσωπικά του δεδομένα και να γνωρίζει ποιος, πότε και γιατί αποκτά πρόσβαση στα δεδομένα του. Επιπλέον, όταν ένα άτομο κάνει μια συναλλαγή, μόνο η ελάχιστη δυνατή ποσότητα προσωπικών πληροφοριών που απαιτούνται θα πρέπει να αποκαλύπτεται. Δηλαδή, η αποκάλυψη των προσωπικών δεδομένων θα πρέπει να γίνεται με τέτοιο τρόπο ώστε μόνο τα απολύτως απαραίτητα στοιχεία να αποκαλύπτονται και μόνο όταν πραγματικά χρειάζονται. Επιπλέον, η αποκάλυψη θα πρέπει να γίνεται με σαφείς όρους σχετικά με το πώς τα προσωπικά δεδομένα θα χρησιμοποιηθούν. Για το σκοπό αυτό, προτείνεται η αρχιτεκτονική Polis που έχει την δυνατότητα να διαχειρίζεται τα προσωπικά δεδομένα ενός ατόμου και να παρέχει ελεγχόμενη πρόσβαση σε αυτές τις πληροφορίες σε τρίτους.

Μια πολύ ενδιαφέρουσα κατηγορία προσωπικών δεδομένων είναι τα δυναμικά προσωπικά δεδομένα, όπως η τρέχουσα θέση ενός ατόμου. Η πρόσφατη πρόοδος στην τεχνολογία των κινητών συσκευών και γενικότερα σε περιβάλλοντα Ubiquitous Computing επιτρέπει στους χρήστες να

συλλέγουν και να επεξεργάζονται τέτοια δυναμικά προσωπικά δεδομένα. Αυτό ανοίγει το δρόμο για μια νέα κατηγορία σημαντικών εφαρμογών. Για το σκοπό αυτό, προτείνονται τέσσερις νέες εφαρμογές που διασφαλίζουν την ιδιωτικότητα και ταυτόχρονα χρησιμοποιούν δυναμικά προσωπικά δεδομένα για την παροχή χρήσιμων υπηρεσιών για τη κοινωνία. Πιο συγκεκριμένα, προτείνεται μια λύση ενισχυμένης ιδιωτικότητας στο πρόβλημα εύρεσης του πλησιέστερου γιατρού, μια αρχιτεκτονική για τη στατιστική ανάλυση ubiquitous ιατρικών δεδομένων παρακολούθησης, ένα σύστημα δημιουργίας περιβαλλοντικών χαρτών θορύβου από εθελοντές που αποστέλλουν τα δεδομένα τους στο cloud και μια νέα αρχιτεκτονική υπολογισμού των τηλεθεάσεων διασφαλίζοντας την ιδιωτικότητα των τηλεθεατών. Περισσότερες λεπτομέρειες σχετικά με αυτές τις εφαρμογές δίνονται στις περιγραφές των κεφαλαίων που ακολουθούν. Οι προτεινόμενες εφαρμογές αποδεικνύουν ότι είναι εφικτή η χρήση και ταυτόχρονα η προστασία των προσωπικών δεδομένων των ατόμων.

Το διαδίκτυο έχει γίνει σταδιακά η κύρια πηγή πληροφοριών για πολλούς ανθρώπους. Τις περισσότερες φορές, οι χρήστες υποβάλλουν ερωτήματα σε μηχανές αναζήτησης για να εντοπίσουν αυτό που αναζητούν. Οι αναζητήσεις αυτές είναι ένας εξαιρετικά σημαντικός μηχανισμός που λαμβάνει χώρα τόσο στους καθημερινούς κλασικούς υπολογιστές όσο και στην πλειοψηφία των σύγχρονων φορητών συσκευών. Λαμβάνοντας υπόψη το διαδίκτυο ως μια τεράστια βιβλιοθήκη, η διαδικτυακή αναζήτηση αντιστοιχεί σε μια αναζήτηση μέσα σε αυτήν τη βιβλιοθήκη. Ενώ τα συμβατικά αρχεία μιας βιβλιοθήκης είναι ιδιωτικά με βάση τη νομοθεσία, τουλάχιστον στις ΗΠΑ, οι χρήστες του διαδικτύου θα μπορούσε να εκτεθούν από τις αναζητήσεις τους. Κάθε φορά που ένας χρήστης υποβάλλει ένα ερώτημα σε μια μηχανή αναζήτησης στο διαδίκτυο, κάποιες προσωπικές πληροφορίες για το χρήστη και τα ενδιαφέροντά του θα μπορούσε να διαρρέουν μαζί με το ερώτημα. Το ερώτημα αντιπροσωπεύει τα ενδιαφέροντα ενός χρήστη και επομένως ανήκει στην κατηγορία των προσωπικών δεδομένων. Παρόλα αυτά, μπορεί να αποθηκεύεται στα logs της μηχανής αναζήτησης, μπορεί να υποκλαπεί από τον πάροχο διαδικτύου ή ακόμη και από οποιοδήποτε άλλο κόμβο στη διαδρομή μέσα από το δίκτυο. Για την προστασία της ιδιωτικής ζωής των χρηστών από τις μηχανές αναζήτησης προτείνονται δύο μέθοδοι που αντικαθιστούν το ιδιωτικό ερώτημα του χρήστη με ένα σύνολο από blurred (θολωμένα) ή scrambled (ανακατεμένα) ερωτήματα και στόχος είναι να προσεγγιστούν τα αρχικά αποτελέσματα της αναζήτησης που θα είχαμε κανονικά. Η μία μεθοδολογία χρησιμοποιεί σημασιολογικά scrambled ερωτήματα και η άλλη χρησιμοποιεί στατιστικά scrambled ερωτήματα. Ο στόχος και των δύο μεθόδων είναι η προστασία των ιδιωτικών ερωτημάτων και των ενδιαφερόντων του χρήστη.

Εδώ συνοψίζονται τα περιεχόμενα των κεφαλαίων της διατριβής:

Κεφάλαιο 2: Background

Σε αυτό το κεφάλαιο, παρέχεται το απαραίτητο υπόβαθρο για τη κατανόηση των βασικών εννοιών που χρησιμοποιούνται σε αυτή τη διατριβή. Πιο αναλυτικά, περιγράφονται έννοιες που αφορούν την ιδιωτικότητα, την ασφάλεια και την κρυπτογραφία.

Κεφάλαιο 3: Polis Framework

Προκειμένου να ενισχυθεί η προστασία της ιδιωτικότητας κατά τις ηλεκτρονικές συναλλαγές, προτείνεται, αναπτύσσεται και αξιολογείται μια αρχιτεκτονική διαχείρισης προσωπικών δεδομένων που ονομάζεται Polis. Αυτή η Polis αρχιτεκτονική είναι σύμφωνη με την ακόλουθη αρχή: Κάθε άτομο έχει τον απόλυτο έλεγχο των προσωπικών του δεδομένων, τα οποία βρίσκονται μόνο στη δική του πλευρά. Η νέα αυτή προσέγγιση μπορεί να είναι προς όφελος τόσο των ίδιων των ατόμων όσο και των επιχειρήσεων. Επιπλέον, έχουν εντοπιστεί αντιπροσωπευτικές ηλεκτρονικές συναλλαγές που αφορούν δεδομένα προσωπικού χαρακτήρα και προτείνονται πρωτόκολλα που βασίζονται στο Polis για την πραγματοποίηση αυτών των συναλλαγών. Η προσέγγιση αξιολογείται με ένα Polis πρωτότυπο τόσο ως μια απλή εφαρμογή όσο και ως μέρος ενός εμπορικού συστήματος διαχείρισης βάσεων δεδομένων. Στα πλαίσια της διατριβής αυτής μελετήθηκαν και αναπτύχθηκαν η διαχείριση των δημοσίων κλειδιών των χρηστών, οι πολικές ασφαλείας (policies) για τις συναλλαγές προσωπικών δεδομένων, η αλληλεπίδραση της Polis αρχιτεκτονικής με τα υπάρχοντα συστήματα διαχείρισης βάσεων δεδομένων και η υποστήριξη πρωτοκόλλων για την πραγματοποίηση υπολογισμών. Τα αποτελέσματα αυτής της δουλειάς δείχνουν ότι οι ηλεκτρονικές συναλλαγές μπορούν να παραμείνουν εφικτές και απλές, παραμένοντας τα προσωπικά δεδομένα μόνο στην πλευρά του ιδιοκτήτη τους.

Κεφάλαιο 4: Privacy-Preserving Solution for Finding the Nearest Doctor

Σε αυτό το κεφάλαιο, ορίζεται το Nearest Doctor Problem (NDP) για την εξεύρεση του πλησιέστερου γιατρού σε μία περίπτωση έκτακτης ανάγκης και παρουσιάζεται ένα πρωτόκολλο διασφάλισης της ιδιωτικότητας για την επίλυσή του. Η προτεινόμενη λύση βασίζεται στη χρήση κρυπτογραφικών εργαλείων και λαμβάνωντας υπόψιν τη τρέχουσα θέση του κάθε συμμετέχοντος γιατρού. Το πρωτόκολλο είναι αποδοτικό και προστατεύει το απόρρητο των θέσεων των ιατρών. Επιπλέον, παρουσιάζεται μια πρότυπη εφαρμογή που υλοποιεί την προτεινόμενη λύση για μια κοινότητα γιατρών που χρησιμοποιούν τις φορητές τους συσκευές με σκοπό να εντοπίσουν τη τρέχουσα θέση τους. Αυτή η πρότυπη εφαρμογή δοκιμάστηκε πειραματικά σε κοινότητες που τις αποτελούσαν εκατοντάδες agents εικονικών γιατρών.

Κεφάλαιο 5: Privacy-Preserving Management and Statistical Analysis of Ubiquitous Health Monitoring Data

Στο κεφάλαιο αυτό, προτείνεται μια αρχιτεκτονική επικεντρωμένη στο χρήστη για τη διαχείριση ubiquitous ιατρικών δεδομένων παρακολούθησης (UHMD) που παράγονται από wearable αισθητήρες σε ubiquitous ιατρικά συστήματα παρακολούθησης (UHMS), και εξετάζεται πώς αυτά τα δεδομένα μπορούν να χρησιμοποιηθούν για την πραγματοποίηση κατανεμημένης στατιστικής ανάλυσης ενισχυμένης ιδιωτικότητας. Ο σκοπός αυτής της προσέγγισής είναι να ενισχυθεί η ιδιωτικότητα των ασθενών και την ίδια στιγμή να αποσυμφορηθεί το Κέντρο Παρακολούθησης Υγείας (HMC) από το τεράστιο όγκο των βιοϊατρικών δεδομένων που παράγονται από τους wearable αισθητήρες των χρηστών. Στη προτεινόμενη λύση γίνεται χρήση προσωπικών agents που χρησιμοποιούνται για να λαμβάνουν και να διαχειρίζονται τα προσωπικά ιατρικά δεδομένα των ιδιοκτητών τους. Επιπλέον, οι προσωπικοί agents μπορούν να υποστηρίξουν κατανεμημένη στατιστική ανάλυση ενισχυμένης ιδιωτικότητας πάνω σε αυτά τα δεδομένα υγείας. Για το σκοπό αυτό, παρουσιάζεται ένα κρυπτογραφικό πρωτόκολλο που βασίζεται σε ασφαλείς υπολογισμούς (MPC) και δέχεται ως είσοδο τρέχουσες ή αρχειοθετημένες τιμές από τους wearable αισθητήρες των χρηστών. Επιπλέον, περιγράφεται μια πρότυπη υλοποίηση που εκτελεί στατιστική ανάλυση ενισχυμένης ιδιωτικότητας σε μια κοινότητα ανεξάρτητων προσωπικών agents και παρουσιάζονται πειραματικά αποτελέσματα από αρκετές εκατοντάδες agents που επιβεβαιώνουν τη βιωσιμότητα και την αποτελεσματικότητα της προσέγγισής.

Κεφάλαιο 6: Privacy-Preserving Computation of Participatory Noise Maps in the Cloud

Το κεφάλαιο αυτό παρουσιάζει ένα σύστημα ενισχυμένης ιδιωτικότητας για participatory sensing, το οποίο βασίζεται σε κρυπτογραφικές τεχνικές και κατανεμημένους υπολογισμούς στο cloud. Κάθε μεμονωμένος χρήστης αντιπροσωπεύεται από έναν προσωπικό agent που βρίσκεται στο cloud και συμμετέχει σε κατανεμημένους υπολογισμούς χωρίς να παραβιάζεται η ιδιωτικότητα ακόμη και από τους cloud παρόχους. Παρουσιάζεται μια γενική αρχιτεκτονική που περιλαμβάνει ένα κρυπτογραφικό πρωτόκολλο το οποίο βασίζεται στην ομομορφική κρυπτογράφηση των συγκεντρωτικών δεδομένων των χαρτών, και η ασφάλεια του στηρίζεται στο Honest-But-Curious μοντέλο τόσο για τους χρήστες όσο και για τους cloud παρόχους. Η προσέγγιση αυτή επιβεβαιώνεται χρησιμοποιώντας δεδομένα από το NoiseTube Project και παρουσιάζονται πειραματικά αποτελέσματα με πραγματικά και τεχνητά δεδομένα. Επιπλέον, παρουσιάζεται και ένα online demo που κάνει χρήση διαφόρων εμπορικών cloud παρόχων. Η προτεινόμενη αρχιτεκτονική είναι η πρώτη πρακτική εφαρμογή με ενισχυμένη ιδιωτικότητα στο χώρο του participatory sensing. Εάν και η προτεινόμενη λύση αφορά την δημιουργία χαρτών θορύβου, η προσέγγιση αυτή μπορεί να είναι εφαρμόσιμη σε οποιαδήποτε crowd-sourcing εφαρμογή που στηρίζεται στη γεωγραφική θέση των πολιτών, όπου οι χάρτες παράγονται με βάση συγκεντρωτικά δεδομένα και ανήκουν στην ερευνητική περιοχή της παρακολούθησης του περιβάλλοντος.

Κεφάλαιο 7: Privacy-Preserving Television Audience Measurement using Smart TVs

Τα συστήματα τηλεόρασης με σύνδεση στο διαδίκτυο, που συχνά αναφέρονται ως Smart TVs, είναι μια εξέλιξη της τηλεόρασης και των τεχνολογιών ψυχαγωγίας στο σπίτι. Σε αυτό το κεφάλαιο, προτείνεται μια νέα προσέγγιση για τη μέτρηση των τηλεθεάσεων (TAM) που σέβεται την ιδιωτικότητα, αξιοποιώντας τις δυνατότητες των Smart TV τεχνολογιών. Η νέα αυτή προτεινόμενη εφαρμογή για τον υπολογισμό των συνολικών μετρήσεων τηλεθέασης χρησιμοποιεί τις υπολογιστικές δυνατότητες των Smart TVs και τη μόνιμη πρόσβαση στο διαδίκτυο. Οι κρυπτογραφικές τεχνικές, συμπεριλαμβανομένων της ομομορφικής κρυπτογράφησης και των αποδείξεων μηδενικής γνώσης, χρησιμοποιούνται για να εξασφαλιστεί τόσο η προστασία της ιδιωτικότητας των συμμετεχόντων ατόμων όσο και η εγκυρότητα των αποτελεσμάτων. Επιπλέον, στους συμμετέχοντες δίνεται η δυνατότητα να αποζημιωθούν για τα τηλεοπτικά δεδομένα που έδωσαν. Τέλος, τα πειραματικά αποτελέσματα σε Android-based Smart TVs έδειξαν τη βιωσιμότητα της προσέγγισης.

Κεφάλαιο 8: Semantic Query Scrambling for Search Privacy on the Internet

Σε αυτό το κεφάλαιο προτείνεται μια μέθοδο για την προστασία της ιδιωτικότητας των αναζητήσεων στο διαδίκτυο, με έμφαση στην ενίσχυση της εύλογης δυνατότητας άρνησης (plausible deniability) έναντι στα logs των ερωτημάτων των μηχανών αναζήτησης. Στόχος της μεθόδου είναι η προσέγγιση των αποτελεσμάτων αναζήτησης, χωρίς να υποβάλεται το πραγματικό ερώτημα και αποφεύγοντας άλλα ερωτήματα που μπορεί εκθέτουν αναλόγως το χρήστη. Αυτό επιτυγχάνεται χρησιμοποιώντας ένα σύνολο ερωτημάτων που αντιπροσωπεύουν γενικότερες έννοιες από το πραγματικό ερώτημα. Πιο συγκεκριμένα, μοντελοποιείται το πρόβλημα θεωρητικά, και διερευνείται η πρακτική σκοπιμότητα και αποτελεσματικότητα της προτεινόμενης λύσης με μια σειρά από πραγματικά ερωτήματα, που έχουν θέματα ιδιωτικότητας, σε μια μεγάλη web συλλογή. Τα αποτελέσματα που παρουσιάζονται μπορεί να έχουν εφαρμογή και σε άλλους τομείς έρευνας της ανάκτησης πληροφοριών, όπως η επέκταση ερωτήματος (query expansion) και το fusion της μετα-αναζήτησης (meta-search). Τέλος, συζητούνται ιδέες για την ιδιωτικότητα, όπως το k-anonymity, και πώς αυτές μπορούν να εφαρμοστούν στη χρήση μηχανών αναζήτησης.

Κεφάλαιο 9: Statistical Query Scrambling for Privacy-Enhanced Web Search

Λαμβάνοντας υπόψη το πρόβλημα παραβίασης της ιδιωτικότητας που υπόκεινται οι χρήστες του διαδικτύου όταν πραγματοποιούν web αναζητήσεις, προτείνεται ένα framework για το μετριασμό αυτό του σημαντικού προβλήματος. Η προτεινόμενη προσέγγισή, η οποία βασίζεται και βελτιώνει την προηγούμενη προτεινόμενη λύση (αυτή που περιγράφεται στο Κεφάλαιο 8), έχει στόχο να πλησιάσει τα αποτελέσματα αναζήτησης αντικαθιστώντας το ιδιωτικό ερώτημα του χρήστη με ένα σύνολο θολωμένων (blurred) ή ανακατεμένων (scrambled) ερωτημάτων. Τα αποτελέσματα των ανακατεμένων ερωτημάτων (scrambled queries) στη συνέχεια χρησιμοποιούνται για να καλύψουν το αρχικό ενδιαφέρον του χρήστη. Πιο συγκεκριμένα, μοντελοποιείται το πρόβλημα θεωρητικά, ορίζονται μετρικές ιδιωτικότητας όσον αφορά την αναζήτηση στο διαδίκτυο και διερευνείται η αποτελεσματικότητα της προτεινόμενης λύσης με μια σειρά πραγματικών ερωτημάτων σε μια μεγάλη web συλλογή. Τα πειράματα δείχνουν σημαντικές βελτιώσεις στην αποτελεσματικότητα της ανάκτησης σε σύγκριση με τα προηγούμενα αποτελέσματα. Επιπλέον, η νέα μέθοδος είναι πιο ευέλικτη, έχει προβλέψιμη πλέον συμπεριφορά, μπορεί να είναι εφαρμόσιμη σε ένα ευρύτερο φάσμα των αναγκών πληροφόρησης, καθώς και η προστασία της ιδιωτικότητας που παρέχει είναι πιο κατανοητή για στον τελικό χρήστη.

Κεφάλαιο 10: Conclusion

Για την ολοκλήρωση της διατριβής το κεφάλαιο αυτό περιέχει τις κύριες συνεισφορές της δουλειάς αυτής και παρέχει μια επισκόπηση των εν εξελίξει και μελλοντικών εργασιών. Πιο αναλυτικά, σε αυτή τη διατριβή, προτάθηκε τη χρήση ubiquitous προσωπικών δεδομένων από αισθητήρες, φορητές συσκευές ή άλλες πηγές με σκοπό την δημιουργία χρήσιμων υπηρεσιών/εφαρμογών για την κοινωνία. Οι προτεινόμενες εφαρμογές χρησιμοποιούν τα προσωπικά δεδομένα των χρηστών, διασφαλίζοντας παράλληλα την ιδιωτικότητας τους. Η προστασία της ιδιωτικότητας επιτυγχάνεται με τη χρήση κρυπτογραφικών τεχνικών και πρωτοκόλλων που εκτελούν υπολογισμούς ενισχυμένης ιδιωτικότητας σε κοινότητες από προσωπικούς agents. Επιπρόσθετα, με την υλοποίηση αυτών των εφαρμογών αποδείχθηκε ότι είναι εφικτή η χρήση και ταυτόχρονα η προστασία των προσωπικών δεδομένων των ατόμων, και μάλιστα με αποδοτικό τρόπο. Τέλος, παρουσιάστηκαν και μέθοδοι για την προστασία των αναζητήσεων στο διαδίκτυο από τα logs των μηχανών αναζήτησης. Η σημαντικότητα της προστασίας αυτών των αναζητήσεων γιγαντώνεται όταν μάλιστα αυτές γίνονται από φορητές συσκευές που πιθανών κάνουν χρήση επιπρόσθετων πληροφοριών.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Research Context

The advances in ICT that cause the generation of personal information, provide a vast potential for emerging new applications that can use personal data in favor of the individuals' interests. Some examples are personalized web services that automatically adapt to the profile of an individual, and location-based services that behave according to the individual's current location or context. However, the use of personal data should be done in a way that simultaneously ensures their protection. In order to protect personal information, several organizations and countries have issued privacy regulations, which should be followed in order for personal information to be protected; the collectively referred to as Fair Information Practices (FIP). Examples of important FIP regulation frameworks are the Data Protection Directive 95/46/EC (henceforth referred to as The Directive) and follow-ups like the Canadian PIPEDA and UK's Data Protection Act (DPA). Based on the privacy regulations, each individual has the right to protect his privacy by retaining the control over his personal data and knowing who, when and why gets access to his data. Furthermore, when an individual makes a transaction, only the minimum possible amount of personal information that is needed to complete it should be disclosed. That is, release of personal data should be done in such a way that only the absolutely necessary items are disclosed and only when it is really needed. Moreover, the disclosure should take place with clear terms on how the personal data will be used. To this end, we propose the Polis framework that has the ability to manage the personal information of an individual and to provide controlled access to this information from other parties.

A very interesting class of personal data is dynamic personal data, such as the current location of an individual. The recent progress in mobile device technology and the advances in ubiquitous computing allow individuals to collect and process such dynamic personal data. This enables a new class of important applications. To this end, we propose four privacy-preserving applications that use dynamic personal data and provide useful services for the society. More specifically, we propose a privacy-preserving solution for finding the nearest doctor, a platform for statistical analysis of ubiquitous health monitoring data, a privacy-preserving cloud computing system for creating participatory noise maps and a privacy-preserving television audience measurement. More details about these applications you can find in the specific chapters of this dissertation. The proposed privacy-preserving applications prove that it is feasible to use and simultaneously protect the personal data of individuals.

The Internet has gradually become the primary source of information for many people. More often than not, users submit queries to search engines in order to locate content. The searches are an exceptionally important mechanism that takes place both in everyday classical computers and to the majority of modern mobile devices. Considering the Internet as a huge library, web-search corresponds to a search within this library. While conventional library records are private under law, at least in the U.S., Internet users might be exposed by their searches. Every time a user submits a query to a web search engine, some private information about the user and his interests might be leaked with the query. The query representing the interest will be saved in the engine's session-logs, or it may be intercepted by the Internet provider or any other node in the network path. To protect the users' privacy from the search engines we propose two methodologies that replace the private user query with a set of blurred or scrambled queries and approximate the target search results. The one methodology is based on semantic query scrambling and the other is based on statistical query scrambling. The goal of both methodologies is to protect the private query and the interests of user.

1.2 Motivation

Desktop, mobile computing, sensing technology and generally ubiquitous computing have greatly increased the amount of personal information that is generated, while recent advances of database technology enable the potential for this information to be (permanently) stored and processed. To give an indication on the volume of personal data, a case in point is that of Max Schrems, in September 2012. The 24-year-old Schrems asked Facebook for a copy of all the data the social network has on file for him and he got back a CD with 1,222 PDF files documenting his every move[1]. Furthermore, incidents of intentional or unintentional data breaches are unfortunately quite common and a reasonable worry is that a lot

[1] http://threatpost.com/twenty-something-asks-facebook-his-file-and-gets-it-all-1200-pages-121311

of them never reach the attention of the media. Some representative examples of such situations are the Choicepoint case, a data broker who sold private records of over 150,000 Americans to a group of criminals in 2005 [41], the incident that took place in the UK, where two computer discs containing the personal data of 25 million citizens were lost in the post [10], as well as the Deutsche Telecom incidents [158]. In September 2006, AOL released a collection with search query-log data containing about 21 million web queries collected from about 650 thousand users over three months [145]. To protect user privacy, each real IP address had been replaced with a random ID number. Soon after the release, the first 'anonymous' user had been identified from the log data. In particular, the user given the ID 4417749 in AOL's query-log was identified as the 62-old Thelma [16]. All these cases indicate that the personal data is a critical, valuable resource that has to be protected in order to ensure the individual's privacy.

At the same time, this personal data enables a new class of important applications. Consider, for example, the location of an expert, and in particular a doctor. In case of an emergency, the distance of the closest doctor could be live-saving information. In August 2007, in the area of Alexandroupolis, Greece, a 17-year old boy was seriously injured in his right leg. Vascular surgery was urgently needed. However, due to several administrative faults no specialized doctor was available. Even worse, it took a long time until it became clear that no specialized doctor could be found and only then the boy was transported to a hospital in Thessaloniki. Unfortunately, due to the long delay the injured leg had to be amputated. Even with the transport taking place, had the initial delay to find out where the nearest specialized doctor is been avoided, the consequences on the boy's health might have been less serious [181]. Another example is the use of statistical methods in medical research. A medical statistic may comprise a wide variety of data types, the most common of which are based on vital records, morbidity and mortality. Additional personal data items may needed for other well-known statistical computations like the demographic distribution of a disease based on geographic, ethnic and gender criteria. Another interesting example comes from the area of participatory sensing. Participatory sensing [28, 146] appropriates everyday devices such as mobile phones to acquire information about the physical world (and the people in it) at a level of granularity which is very hard to achieve otherwise. A crucial component of participatory sensing systems is *geolocation*, i.e., labeling data with geographic coordinates. This is particularly important in the context of *NoiseTube* [123, 175], a participatory sensing system and service[1] designed to monitor and map noise pollution. Indeed, it would be practically impossible to produce noise maps on the basis of sound level measurements, gathered quasi-continuously as contributors walk the streets, without automatic geolocation of

[1] http://www.noisetube.net

measurements by means of GPS. All these examples indicate that the usage of personal data is extremely important and opens a new category of innovative applications.

1.3 Methodology

The advances in mobile devices and network infrastructures, and in the same time the wide acceptance of these technologies for everyday tasks of individuals have increased the amount of personal information that is generated and have leaded to a new wide range of applications. Additionally, the advances in cryptographic algorithms and protocols have enabled the utilization and concurrently the protection of personal data. The goal of this dissertation is to exploit these advances and to propose new innovative applications that protect the individual's privacy and utilize his personal data. The methodology that was followed to achieve this goal is as follows:

- Following the Polis principle: keep data at the user's side.
- Detection and selection of the problem.
- Identification of critical personal data for protection and usage.
- Designing of the appropriate computation (distributed or not).
- Developing new cryptographic protocols or adapting existing protocols for the computational problem.
- Theoretical analysis of the proposed solution.
- Prototype implementation and experimental validation of the effectiveness of solution.

1.4 Results

We showed that the personal data can be protected and can concurrently be utilized successfully and efficiently for important innovative applications:

1. *NDP:* Privacy-preserving solution for finding the nearest doctor
    - **Personal Data:** Geographical location of doctors.
    - **Usage:** Health emergency.

2. *PrivStat:* Privacy-preserving statistical analysis on ubiquitous health monitoring data
    - **Personal Data:** Health data of patients.
    - **Usage:** Statistical analysis in medical research.
3. *NoiseTubePrime:* Privacy-preserving computation of participatory noise maps in the cloud
    - **Personal Data:** Geolocated, timestamped sound level measurements of volunteers.
    - **Usage:** Participatory noise map.
4. *PrivTAM:* Privacy-preserving television audience measurement using smart TVs
    - **Personal Data:** Viewing records of participants.
    - **Usage:** Television audience measurement.
5. *Query Scrambler:* Privacy-enhanced web search
    - **Personal Data:** Search queries of users.
    - **Usage:** Web search.

1.5 Structure of the Dissertation

Here we summarize each subsequent chapter of the dissertation:

Chapter 2: Background

In this chapter, we provide background to understand key concepts in this dissertation. Readers familiar with the concepts in privacy, security and cryptography may skip this chapter.

Chapter 3: Polis Framework

In order to enhance privacy protection during electronic transactions, we propose, develop and evaluate a personal data management framework called Polis, which abides by the following principle: Every individual has absolute control over his personal data, which reside only at his own side. This framework admits individuals to have their personal data stored only at their own side. The new approach can be of mutual benefit to both individuals and companies. Furthermore, we identify representative electronic transactions that involve personal data and proposes Polis-based protocols for them. The approach is evaluated on a Polis prototype both as a stand-alone application and as part of a commercial database management system. The results of this work indicate that electronic transactions can remain both feasible and straightforward, while personal data remain only at the owner's side.

Chapter 4: Privacy-Preserving Solution for Finding the Nearest Doctor

In this chapter, we define the Nearest Doctor Problem (NDP) for finding the nearest doctor in case of an emergency and present a privacy-preserving protocol for solving it. The solution is based on cryptographic primitives and makes use of the current location of each participating doctor. The protocol is efficient and protects the privacy of the doctors' locations. A prototype implementing the proposed solution for a community of doctors that use mobile devices to obtain their current location is presented. The prototype is evaluated on experimental communities with up to several hundred doctor agents.

Chapter 5: Privacy-Preserving Management and Statistical Analysis of Ubiquitous Health Monitoring Data

In this chapter, we propose a user-centric architecture for managing Ubiquitous Health Monitoring Data (UHMD) generated from wearable sensors in a Ubiquitous Health Monitoring System (UHMS), and examine how these data can be used within privacy-preserving distributed statistical analysis. The purpose of our approach is to enhance the privacy of patients and at the same time to decongest the Health Monitoring Center (HMC) from the enormous amount of biomedical data generated by the users' wearable sensors. In our solution personal software agents are used to receive and manage the personal medical data of their owners. Moreover, the personal agents can support privacy-preserving distributed statistical analysis of the health data. To this end, we present a privacy-preserving cryptographic protocol based on secure multi-party computations that accept as input current or archived values of users' wearable sensors. We describe a prototype implementation that performs a privacy-preserving statistical analysis on a community of independent personal agents and present experimental results with up to several hundred agents that confirm the viability and the effectiveness of our approach.

Chapter 6: Privacy-Preserving Computation of Participatory Noise Maps in the Cloud

This chapter presents a privacy-preserving system for participatory sensing, which relies on cryptographic techniques and distributed computations in the cloud. Each individual user is represented by a personal software agent, deployed in the cloud, where it collaborates on distributed computations without loss of privacy, including with respect to the cloud service providers. We present a generic system architecture involving a cryptographic protocol based on a homomorphic encryption scheme for aggregating sensing data into maps, and demonstrate security in the Honest-But-Curious model both for the users and the cloud service providers. We validate our system in the context of NoiseTube, a participatory sensing framework for noise pollution, presenting experiments with real and artificially generated data sets, and a demo on a heterogeneous set of commercial cloud providers. To the best of our knowledge our system is the first operational privacy-preserving system for participatory sensing. While our validation pertains to the noise domain, the approach used is applicable in any crowd-sourcing application relying on location-based contributions of citizens where maps are produced by aggregating data – also beyond the domain of environmental monitoring.

Chapter 7: Privacy-Preserving Television Audience Measurement using Smart TVs

Internet-enabled television systems, often referred to as Smart TVs, are a new development in television and home entertainment technologies. In this chapter, we propose a new, privacy-preserving, approach for Television Audience Measurement (TAM), utilizing the capabilities of the Smart TV technologies. We propose a novel application to calculate aggregate audience measurements using Smart TV computation capabilities and permanent Internet access. Cryptographic techniques, including homomorphic encryption and zero-knowledge proofs, are used to ensure both that the privacy of the participating individuals is preserved and that the computed results are valid. Additionally, participants can be compensated for sharing their information. Preliminary experimental results on an Android-based Smart TV platform show the viability of the approach.

Chapter 8: Semantic Query Scrambling for Search Privacy on the Internet

We propose a method for search privacy on the Internet, focusing on enhancing plausible deniability against search engine query-logs. The method approximates the target search results, without submitting the intended query and avoiding other exposing queries, by employing sets of queries representing more general concepts. We model the problem theoretically, and investigate the practical feasibility and effectiveness of the proposed solution with a set of real queries with privacy issues on a large web collection. The findings may have implications for other IR research areas, such as query expansion and fusion in meta-search. Finally, we discuss ideas for privacy, such as k-anonymity, and how these may be applied to search tasks.

Chapter 9: Statistical Query Scrambling for Privacy-Enhanced Web Search

We consider the problem of privacy leaks suffered by Internet users when they perform web searches, and propose a framework to mitigate them. Our approach, which builds upon and improves our previous work (is presented in Chapter 8) on search privacy, approximates the target search results by replacing the private user query with a set of blurred or scrambled queries. The results of the scrambled queries are then used to cover the original user interest. We model the problem theoretically, define a set of privacy objectives with respect to web search and investigate the effectiveness of the proposed solution with a set of real queries on a large web collection. Experiments show great improvements in retrieval effectiveness over a previously reported baseline in the Chapter 8. Furthermore, the methods are more versatile, predictably-behaved, applicable to a wider range of information needs, and the privacy they provide is more comprehensible to the end-user.

Chapter 10: Conclusion

To wrap up the dissertation this chapter lists the main contributions of our work and provides an overview of on-going and future work.

Chapter 2

Background

2.1 Privacy and Personal Data

2.1.1 About Privacy

Privacy is an elusive concept which can not be easily defined. According to [201], privacy is the ability of an individual or group to seclude themselves or information about themselves and thereby reveal themselves selectively. The boundaries and content of what is considered private differ among cultures and individuals, but share basic common themes. Privacy is sometimes related to anonymity, the wish to remain unnoticed or unidentified in the public realm. When something is private to a person, it usually means there is something within them that is considered inherently special or personally sensitive. The degree to which private information is exposed therefore depends on how the public will receive this information, which differs between places and over time. Privacy partially intersects security, including for instance the concepts of appropriate use, as well as protection of information.

In this thesis we emphasize on information privacy. Information or data privacy refers to the evolving relationship between technology and the legal right to, or public expectation of, privacy in the collection and sharing of data about one's self. Privacy concerns exist wherever uniquely identifiable data relating to a person or persons are collected and stored, in digital form or otherwise. In some cases these concerns refer to how data is collected, stored, and associated. In other cases the issue is who is given access to information. Other issues include whether an individual has any ownership rights to data about them, and/or the right to view, verify, and challenge that information.

Various types of personal information are often associated with privacy concerns. For several reasons, individuals may object to personal information such as their religion, sexual orientation, political affiliations, or personal activities being revealed, perhaps to avoid discrimination, personal embarrassment, or damage to their professional reputations.

It is generally agreed that the first publication advocating privacy in the United States was the article by Samuel Warren and Louis Brandeis [194]. In this article, the two American lawyers determined the privacy as "the right to be let alone". The reason for this article was a response to recent technological developments, such as photography, and sensationalist journalism, also known as yellow journalism. Photographs, for example, were used by the yellow press, in the view of the authors, as an attack to the personal privacy in accordance with the terms of the right to be let alone.

The most common definition of privacy that is used today, is that of Alan Westin: "The right of the individual to decide what information about himself should be communicated to others and under what circumstances" [195]. Another well known term for privacy that is used for the protection of personal data, and is also consistent with the definition of the Westin, is that of informational self-determination [198]. The term of informational self-determination was first used in the context of a German constitutional ruling relating to personal information collected during the 1983 census.

2.1.2 Privacy Laws

Privacy law [201] is the area of law concerning the protecting and preserving of privacy rights of individuals. While there is no universally accepted privacy law among all countries, some organizations promote certain privacy regulations, which should be followed by individual countries in order for personal information to be protected; the collectively referred to as Fair Information Practices (FIP). The most important privacy laws by county are as follows:

Europe. For Europe, Article 8 of the European Convention on Human Rights (ECHR) guarantees the right to respect for private and family life, one's home and correspondence. The European Court of Human Rights (ECtHR) in Strasbourg has developed a large body of jurisprudence defining this fundamental right to privacy. The European Union requires all member states to legislate to ensure that citizens have a right to privacy, through directives such as the *Data Protection Directive 95/46/EC* [67] on the protection of personal data. It is regulated in the United Kingdom by the Data Protection Act 1998 and in France data protection is also monitored by the CNIL, a governmental body which must authorize legislation concerning privacy before them being enacted. Based on the privacy regulations, each individual has the right to protect his privacy by retaining the control over his personal data and knowing who, when and why gets access to his data. Furthermore, when an individual makes a transaction, only the minimum possible amount of personal information that is needed to complete it should be disclosed. That is, release of personal data should be done in such a way that only the absolutely necessary items are disclosed and only when it is really needed. Moreover, the disclosure should take place with clear terms on how the personal data will be used.

United Kingdom. In the United Kingdom, it is not possible to bring an action for invasion of privacy. An action may be brought under another tort (usually breach of confidence) and privacy must then be considered under European Community law. In the UK, it is sometimes a defense that disclosure of private information was in the public interest. There is, however, the Information Commissioner's Office (ICO), an independent public body set up to promote access to official information and protect personal information. They do this by promoting good practice, ruling on eligible complaints, giving information to individuals and organizations, and taking action when the law is broken. The relevant UK laws include: *Data Protection Act 1998 (DPA)* [182]; *Freedom of Information Act 2000*; *Environmental Information Regulations 2004*; *Privacy and Electronic Communications Regulations 2003.*

United States. Concerning privacy laws of the United States, privacy is not guaranteed per se by the Constitution of the United States. The Supreme Court of the United States has found that other guarantees have "penumbra" that implicitly grant a right to privacy against government intrusion, for example in Griswold v. Connecticut (1965). In the United States, the right of freedom of speech granted in the First Amendment has limited the effects of lawsuits for breach of privacy. Privacy is regulated in the U.S. by the *Privacy Act of 1974* [183], and various state laws. Certain privacy rights have been established in the United States via legislation such as the *Children's Online Privacy Protection Act (COPPA)*, the *Gramm-Leach-Bliley Act (GLB)*, the *Health Insurance Portability and Accountability Act (HIPAA)* [1], the *Cable TV Privacy Act* [184], and the *Video Privacy Protection Act* [185].

Canada. Canadian privacy law is governed federally by multiple acts, including the *Canadian Charter of Rights and Freedoms*, and the *Privacy Act (Canada).* Mostly this legislation concerns privacy infringement by government organizations. Data privacy was first addressed with the *Personal Information Protection and Electronic Documents Act (PIPEDA)* [32], and provincial-level legislation also exists to account for more specific cases personal privacy protection against commercial organizations.

Australia. In Australia there is the *Privacy Act 1988* [12]. Privacy sector provisions of the Act apply to private sector organizations with a link to Australia, including: 1. individuals who collect, use or disclose personal information in the course of a business. For example, a sole trader's business activities will be regulated (unless it's a small business), but information gathered outside business activities won't be; 2. bodies corporate; and 3. partnerships, unincorporated associations and trusts - any act or practice of a partner, committee member or trustee is attributed to the organization. Organizations outside Australia must comply with the provisions in some circumstances. Sending information out of Australia is also regulated.

2.1.3 Criteria for Privacy Protection

2.1.3.1 k-Anonymity

The k-anonymity criterion is a non-probabilistic metric for anonymity concerning entries in statistical databases such as released by data holders for research purposes [178]. The author's interest in [178] is in re-identifiability of persons based on their entries in such databases, e.g. through inferences over multiple queries to the database or linking between different databases. A statistical database provides k-anonymity protection if the information for each person contained within cannot be distinguished from at least $k-1$ other individuals who appear in the database.

In [178] the author applies set-theory to formalize the notions of a table, rows (or 'tuples') and columns (or 'attributes'), and the quasi-identifier concept introduced by Dalenius [45]. A quasi-identifier is a set of attributes that are individually anonymous, but in combination can uniquely identify individuals.

Definition 1 (k-anonymity) *A simple definition of k-anonymity [39] in the context of this work is that no less than k individual users can be associated with a particular personal data value.*

The k-anonymity model assumes a global agent to calculate the metric. It also depends on the data holder's competence and willingness to correctly identify and work around quasi-identifiers. k-Anonymity protects against the 'oblivious' adversary targeting anyone (re-identifying anything he can, hoping to get lucky) as well as the adversary targeting a specific individual. One of the limitations of the original k-anonymity model is that it does not take into account the situation where the sensitive attribute has the same value for all k rows and is revealed anyway. l-Diversity was introduced to address this by requiring that, for each group of k-anonymous records in the data set, at least l different values occur for the sensitive column [122]. Further developments include t-closeness, m-invariance, δ-presence and p-sensitivity [31, 118, 137, 210].

2.1.3.2 Differential Privacy

Consider a trusted party that holds a dataset of sensitive information (e.g. medical records, voter registration information, email usage) with the goal of providing global, statistical information about the data publicly available, while preserving the privacy of the users whose information the data set contains. Such a system is called a statistical database. The notion of differential privacy formalizes the notion of privacy in statistical databases. Differential privacy aims to provide means to maximize the accuracy of queries from statistical databases while minimizing the chances of identifying its records. Also, the aim of differential privacy is to ensure that the ability of an adversary to inflict harm (or good, for that matter) – of any sort, to any set of people – should be essentially the same, independent of whether any individual opts in to, opts out of, the dataset [62, 61]. The formal definition of differential privacy follows.

Definition 2 (ϵ-Differential privacy [61]) *A randomized function $\mathcal{K}$ gives ϵ-differential privacy if for all data sets D_1 and D_2 differing on at most one element, and all $S \subseteq Range(\mathcal{K})$, the following holds:*

$$Pr[\mathcal{K}(D_1) \in S] \leq exp(\epsilon) \times Pr[\mathcal{K}(D_2) \in S]$$

The probability is taken is over the coin tosses of $\mathcal{K}$.

2.1.4 Privacy Types

There are two distinct problems that arise in the setting of privacy-preserving computations [119]:

(a) The first is to decide which functions can be safely computed, where safety means that the privacy of the participants is preserved if the result of the computation is disclosed. We will assume that the outcomes of the computations do not violate the privacy of the participants and will not further consider this problem in this work.

(b) The second is how, meaning with which algorithms and protocols, to compute the results while minimizing the damage to privacy. For example, it is always possible to pool all of the data in one place and run the computation algorithm on the pooled data. However, this is exactly what we don't want to do.

The focus in this dissertation is how to achieve privacy of type (b), that is, how to perform computations without pooling the personal data, and in a way that reveals nothing but the final results of the computation.

2.1.5 Personal Data

Personal data is any information relating and describing a person, such as: identifying information (name, age, residence, occupation, marital status, etc.), physical characteristics, education, work (experience, work habits, etc.), economic status (income, assets, economic behavior, etc.), interests, activities and habits. The person (natural person) to whom the data refer is called "data subject".

Beyond the general classification of some data as personal, there is a subset of those that is described by the term "sensitive personal data". Sensitive is called the personal data of a person that reveals racial or national origin, political opinions, religious or philosophical beliefs, membership in a trade union, health, social welfare, sexual orientation, criminal prosecutions and convictions, as well as participation in related to the above associations. The sensitive personal data is protected by the law with stricter regulations than the simple personal data.

Apart from these two categories of data that is described by the legislature [200], and institutionally falling under the supervision of responsible independent authority, there is a different category of data that is called "Personally Identifiable Information" (PII). The PII is information that can be used on its own or with other information to identify, contact, or locate a single person, or to identify an individual in context. This data can be single, e.g. the identity number, or combinations, e.g. the birth date and the zip code, and can accurately identify an individual.

It should be emphasized that the personal identifiable data is not limited only to personal data. An oft-cited example is that the 87% of the U.S. population can be identified by the combination of gender, birth date and zip code of residence.

From the above follows a critical conclusion. Knowing a group of information, such as the history of searches of an unknown user in a search engine it can lead us to extract relevant data, including sensitive personal data of the individual. Since the identification of the initially unknown user can possibly be done from a group of personally identifiable information that may be entered in searches, we can accordingly extract from the form and the content of the searches information about his actions and also his thoughts. As a result, form an anonymous set of information, in the first level we may recognize the creator of the questions, and then, in the second level we may retrieve sensitive data that concern him.

Finally, due to recent advances in the development of mobile devices and generally Ubiquitous Computing technologies, it is now possible to collect "dynamic personal data". Examples of this special category of personal data are the geolocation (geographic position) and information relating to the physical condition and health of an individual, such as blood pressure, heart rate and mood. The television viewing records of individuals also belong to this category. In this dissertation, we examine how to protect such personal data while at the same time making use of it.

2.2 Basic Concepts of Cryptography

2.2.1 Main Idea of Cryptography

The need to send a message from a sender to a recipient, without risking to be read by someone else, so that this message to be sent with security, is a main subject of cryptography [166]. This initial message is the plaintext (or cleartext). The process by which the message is hidden from the content of the information is called encryption. The encrypted message is the ciphertext. The process of changing the ciphertext back to the plaintext is called decryption. (Note: According to the standard ISO 7498-2, the terms "encipher" and "decipher" instead of the terms "encrypt" and "decrypt" are used, respectively.) All these are presented in Figure 2.1.

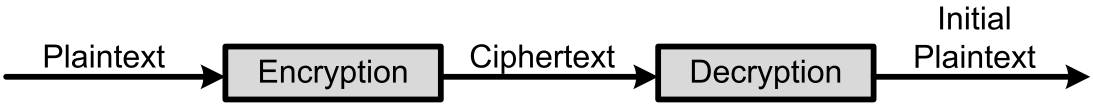


Figure 2.1: Encryption and decryption.

The science that has as its object the security of messages is the cryptography, and is performed by cryptographers. The cryptanalysts are engaged in the cryptanalysis, the breaking of ciphertext, so that to be able to see what is hidden behind the encrypted data. The branch of science that deals with cryptography and cryptanalysis is the cryptology and those who make this profession are the cryptologists. The modern cryptologists are generally trained in theoretical mathematical level. In practice with the term cryptography we usually refer overall in cryptology [166].

Definition 3 (Cryptography) *Cryptography is the practice and study of techniques for secure communication in the presence of third parties (called adversaries). More generally, it is about constructing and analyzing protocols that overcome the influence of adversaries and which are related to various aspects in information security such as data confidentiality, data integrity, authentication, and non-repudiation.*

The plaintext is indicated by the letter M (message) or the letter P (plaintext). The plaintext can be a stream of bits, a text file, an image file, a stream of digital voice, a digital video image etc. With regard to a computer, the M is simply binary data. The plaintext may be either the transport or storage of data. Essentially, the M is the message that is encrypted.

Let us now assume that the ciphertext is indicated by the letter C. The ciphertext is also binary data, sometimes the same size with the M and others may be greater. (The encryption in combination with the compression can lead so that the C to be less than M, without the encryption can be performed directly.) The encryption function E acts on the M to produce the C. The mathematical expression of this is:

$$E(M) = C$$

In the reverse process, the decryption function D acts on the C to produce the M:

$$D(C) = M$$

Since the essence of the encryption and then the decryption of a message is to regain the initial plaintext, the following equation shows this:

$$D(E(M)) = M$$

2.2.2 Cryptographic Goals

Cryptography is often used to provide services that are related to various aspects in information security such as:

- **Confidentiality** is a service used to keep the content of information from all but those authorized to have it. Privacy is sometimes synonymous with confidentiality.

- **Integrity** is a service which addresses the unauthorized alteration of data. To assure data integrity, one must have the ability to detect data manipulation by unauthorized parties.

- **Authentication** is a service related to identification. This function applies to both entities and information itself. Two parties entering into a communication should identify each other.

- **Non-repudiation** is a service which prevents an entity from denying previous commitments or actions.

These concepts are critical for the social interaction with the computer usage, and reflect with the interpersonal interactions. Some examples that can apply to the above services are:

– That someone is the one who says he is (Authentication).

- That a driver's license, medical degree, and the passport is valid (Signature).
- That a document is absolutely certain that it have come from a particular person (Non-repudiation).
- That a document has not been altered or modified (Integrity).

2.2.3 Algorithms and Keys of Cryptography

A cryptographic algorithm (known as cipher), is the mathematical function that is used for encryption and decryption [166]. Generally, there are two related functions: one for encryption and another for decryption.

If the security of an algorithm is based on the secret operating mode of the algorithm, then such an algorithm is restricted. Such restricted algorithms have only historical interest and are considered inadequate for today's standards. A large or changing group of users can not use them, because every time which a user changes group he must use a different algorithm. Also if someone accidentally reveals the secret, then everyone will have to change the algorithm.

Furthermore, these restricted algorithms do not allow any quality control or standardization. Each user group must have a unique algorithm. Such a group can not use ready products of hardware or software, because anyone could buy the same product and learn the algorithm, so they have to write themselves the algorithms and their applications. If no one in the group is not good cryptographer, they will not be able to know if their algorithm is safe.

Despite these significant drawbacks, the restricted algorithms are very popular for low security applications. In these cases the users neither realize nor care about the security problems that exist in their system.

The modern encryption systems solve this problem by using a key, that is indicated by the letter K. This key can be any of a large number of values. The range of possible values of the key is called keyspace. Therefore, the procedures of encryption and decryption utilize this key (i.e., depend on this key, and this fact is indicated by the indicator K), so the functions now become (see Figure 2.2):

$$E_K(M) = C$$
$$D_K(C) = M$$
$$D_K(E_K(M)) = M$$

Some algorithms use different keys in the encryption and the decryption (see Figure 2.3). Where the encryption key, K_1, is different from the corresponding decryption key, K_2. In this case, the functions become:

$$E_{K_1}(M) = C$$
$$D_{K_2}(C) = M$$
$$D_{K_2}(E_{K_1}(M)) = M$$

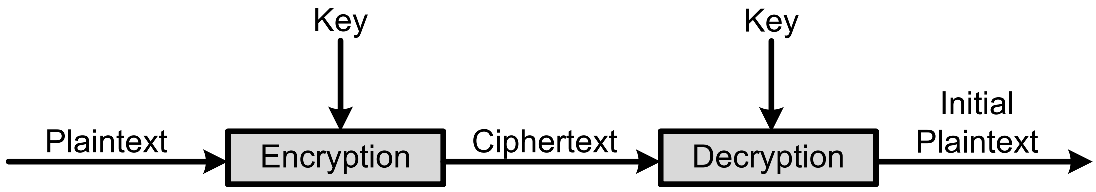


Figure 2.2: Encryption and decryption with one key.

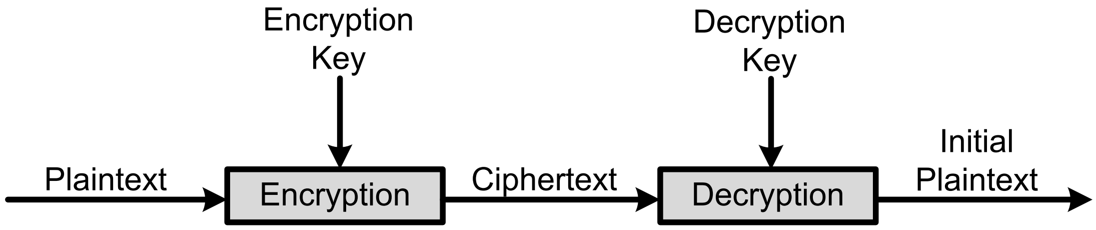


Figure 2.3: Encryption and decryption with two different keys.

The whole security of these algorithms is based on the key (or keys) and not in the details of the algorithm. This means that the algorithm can be published and analyzed, and as a result the products that use the algorithm can be mass-produced. It also does not matter if someone knows your algorithm, you can not read your messages because he does not know the special key. Thus, a modern cryptographic system now consists of the algorithm, all the possible plaintexts, the ciphertexts and the keys.

2.2.4 Symmetric Algorithms

There are two general types of key-based algorithms: the symmetric algorithms and the public-key algorithms. The symmetric algorithms, which are sometimes called conventional algorithms are algorithms where the encryption key can be calculated from the decryption key and vice versa. In most symmetric algorithms, the encryption key and the decryption key are the same. These algorithms are also called single-key algorithms and require the sender and the receiver to agree on a key before they can communicate securely. The security of a symmetric algorithm is based in the key, with no need to hide the key points of encryption and decryption. If it is necessary to keep the communication secret, the key must remain secret [166].

The encryption and decryption with a symmetric algorithm are shown by the equations:

$$E_K(M) = C$$
$$D_K(C) = M$$

The symmetric algorithms can be divided into two categories:

1. Those that the plaintext is a bit (or sometimes a byte) at a time and are called stream algorithms or stream ciphers.
2. And those that the plaintext is a group of bits. The group of bits is called block, and these algorithms are called block algorithms or blocks ciphers. For the modern computer algorithms, a common size of block is 64 bits, large enough to prevent breakdown and small enough to be practicable.

2.2.5 Public-key Algorithms

The public-key algorithms (or asymmetric algorithms) are designed so that the key that is used for encryption is different from the key that is used for decryption. Furthermore, the decryption key can not be calculated (at least to a reasonable amount of time) from the encryption key. These algorithms are called as "public-key" because the encryption key can be made public: A stranger can use the encryption key to encrypt a message, but only the person with the corresponding decryption key can decrypt the message. In these systems, the encryption key is often called the public key and the decryption key is often called the private key. The private key is also sometimes called secret key, but to avoid confusion with the symmetric algorithms, this phrase is generally not used [166]. The encryption that uses the public key K can be described by the following equation:

$$E_{K_{pub}}(M) = C$$

The decryption with the private key is described by the equation:

$$D_{K_{pri}}(C) = M$$

Sometimes, the messages are encrypted with the private key and decrypted with the public key. Specifically, this is used in digital signatures. Despite the potential confusion, these processes are described by the equations:

$$\begin{gathered} E_{K_{pri}}(M) = C \\ D_{K_{pub}}(C) = M \end{gathered}$$

2.2.6 Homomorphic Encryption

Homomorphic encryption [197] is a form of encryption which allows specific types of computations to be carried out on ciphertext and obtain an encrypted result which decrypted matches the result of operations performed on the plaintext. For instance, one person could add two encrypted numbers and then another person could decrypt the result, without either of them being able to find the value of the individual numbers. This property can have both positive and negative effects on a cryptographic system. So the model of homomorphic encryption is vulnerable to malicious attacks from its design, making it unsuitable for secure data transmission. But the homomorphic properties of various cryptographic systems can be used to create secure voting systems, collision-resistant hash functions, private information retrieval schemes and enable widespread use of cloud computing by ensuring the confidentiality of processed data.

Definition 4 (Homomorphic Encryption) *Homomorphic encryption [159] is a form of encryption where one can perform a specific algebraic operation on the plaintext by performing a (possibly different) algebraic operation on the ciphertext. Particularly, an encryption algorithm* $E()$ *is homomorphic if given* $E(x_1)$ *and* $E(x_2)$*, one can obtain* $E(x_1 \circ x_2)$ *without decrypting* x_1*,* x_2*, for some operation* $\circ$*.*

Homomorphic cryptosystems can be separated into two main classes. The first are the partially homomorphic cryptosystems that can support only one operation, for example addition or multiplication, and the second are the fully homomorphic cryptosystems [74] that support both addition and multiplication operations. In this thesis, we only focus on the partially homomorphic cryptosystems that are more practical and efficient. More details about partially homomorphic cryptosystems we will see in the Sections 2.4, 2.5 and 2.6. Moreover, there are recent results on "somewhat"[1] homomorphic cryptosystems [75], i.e., cryptosystems which support a limited number of homomorphic operations including both additive and multiplicative operations. During the last years fully homomorphic cryptosystems supporting any number of additions and multiplications have been published, starting with the seminal work of [74]. However, so far fully homomorphic cryptosystems are not efficient enough to be used in practical applications, though one could probably use "somewhat" homomorphic cryptosystems for some appropriate functions. A discussion of the efficiency and the practical relevance of current fully homomorphic and somewhat homomorphic cryptosystems, and their applications in cloud computing, can be found in [136].

2.3 Cryptographic Hash Algorithms

2.3.1 One-way Hash Functions

A hash function is a function that receives as input data of arbitrary length and turns that on one-way data (with no possibility of return in the original

[1] The term "somewhat" is used by Gentry [75] himself to refer to an encryption scheme that can support a limited number of arithmetic operations on the encrypted data before the accumulated noise makes the resulting ciphertext indecipherable.

form). In fact, the one-way hash functions are based on the idea of a compression function. The result of this one-way function is a hash variable with n length, for given input greater length of m. The inputs of the compression function is a block message and the output of the previous block of text [166] (Figure 2.4). The output of this function is the hash of all blocks up to that point. The hash of block M_i will be:

$$h_i = f(M_i, h_{i-1})$$

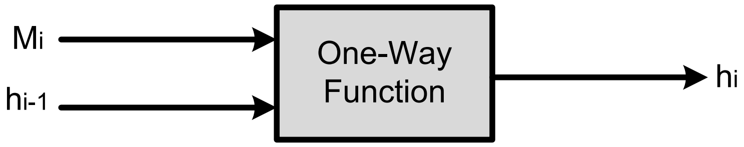


Figure 2.4: One-way hash function.

This hash variable, together with the next block message, becomes the next input in the compression function. The hash of entire message is the hash of the last block.

The pre-image (i.e., before beginning the process) must contain a binary representation of the length of the entire message. This technique overcomes a possible security problem in messages that may have different lengths hashing for the same variable. This technique is sometimes referred as MD-strengthening.

Several researchers have theoretically examined that if the compression function is secure, then the hashing method arbitrary length with pre-image is also safe, but nothing is proved.

2.3.2 Secure Hash Algorithm

The family of secure hash algorithm (SHA) is a set of related cryptographic hash functions. Until now, the SHA family consists of four members, the SHA-0, the SHA-1, the SHA-2 and the SHA-3. The most commonly used function of this family, the SHA-2, is adopted in a wide variety of popular security applications and protocols, including TLS, SSL, PGP, SSH, S/MIME, and IPSec. The SHA-1 is considered the successor of MD5, an earlier, widely used hash function. The SHA algorithms were designed by the U.S. National Security Agency (NSA) and published by the National Institute of Standards and Technology (NIST).

The first member of the family was published in 1993 and is officially called SHA. However, as it is often called SHA-0 to avoid confusion with its successors. Two years later it was published the SHA-1, which is the first successor of SHA. In 2001, four additional hash functions in the SHA family, named after their digest lengths (in bits), were published: SHA-224, SHA-256, SHA-384, and SHA-512 (sometimes collectively referred as SHA-2). In 2012, the SHA-3 that consists of a 5 × 5 array of 64-bit words, 1600 bits total, was published. In Table 2.1 the different types of SHA [205] are shown.

Algorithm		**Output size**	**Internal size**	**Block size**	**Max message**	**Word size**	**Rounds**	**Collisions**
SHA-0		160	160	512	$2^{64}-1$	32	80	Yes
SHA-1		160	160	512	$2^{64}-1$	32	80	Yes
SHA-2	SHA-224	224	256	512	$2^{64}-1$	32	64	Yes
	SHA-256	256						
	SHA-384	384	512	1024	$2^{128}-1$	64	80	Yes
	SHA-512	512						
	SHA-512/224	224						
	SHA-512/256	256						
SHA-3		224/256/384/512	1600			64	120	None

Table 2.1: Comparison of SHA functions [205].

2.4 RSA Cryptosystem

One of the most popular cryptographic systems is the RSA cryptosystem [160], that was coined in 1978 by Ronald Rivest, Adi Shamir and Leonard Adelman and was named from the initials letters of their surnames. The RSA is a public-key cryptosystem and until now is considered impossible to crack using modern computers, but if sometime the quantum computers are constructed this may radically change. Today is widely used especially in financial and banking transactions.

2.4.1 RSA Algorithm

The RSA algorithm [203] consists of three main parts: the key generation, the encryption algorithm and the decryption algorithm.

Key Generation. The procedure that should be followed to create a pair (public and private) of keys is:

1. We choose two distinct prime integers the p and q and calculate their product $n = p \cdot q$.
2. We choose a random number d, which is prime as to the $(p-1)$ and $(q-1)$, so that the greatest common divisor of d, $(p-1)$ and $(q-1)$ to be the one.
3. We calculate the number e from the equation: $(e \cdot d) \bmod (p-1)(q-1) = 1$. So, the e is the inverse of d: $d^{-1} \bmod (p-1)(q-1)$.
4. The public key is the pair of numbers (e, n).
5. The private key is the pair of numbers (d, n).

Encryption Algorithm. The algorithm that is followed to encrypt a message is as follows:

1. The public key is sent to the sender of the message.
2. The sender encrypts the message with the public key (e, n).
3. The encrypted message $E_k(m) = m^e \bmod n$ is sent to the recipient.

Decryption Algorithm. The algorithm that is followed to decrypt the initial message is:

1. The private key (d, n) is held by the recipient.
2. In order to decrypt the message, it is only needed the private key.
3. The result of the decryption is $D_k(c) = c^d \bmod n$, where $c = E_k(m)$.

2.4.2 Homomorphic Property of RSA

The multiplicative homomorphic encryption property of the RSA cryptosystem means that multiplication of encrypted values corresponds to product of decrypted ones. Concretely, if the x_1 and x_2 are two plain integers, where $x_1, x_2 \in \mathbb{Z}_n$, and the notation $\mathcal{E}(x)$ is used to denote the encryption of the message x, then the RSA homomorphic property is shown by the following equation:

$$\begin{aligned} \mathcal{E}(x_1) \cdot \mathcal{E}(x_2) &= x_1^e \, x_2^e \bmod n \\ &= (x_1 \, x_2)^e \bmod n \\ &= \mathcal{E}(x_1 \cdot x_2) \end{aligned}$$

2.4.3 Security of RSA

In order to better understand the security of RSA cryptosystem we will firstly mention how we can crack it (what we have to do to decrypt a encrypted message without to have the private key). Since, we know the number n from the public key (e, n), we only have to analyze this number n in product of two prime numbers in order to find the numbers p and q. Once we find them, the decryption is immediately done since the method of the RSA system is known.

While it is very easy to multiply two prime numbers in order to find their product, it is very difficult to analyze a number into a product of two primes and it is practically impossible if the number has many digits. So the primes p and q should be large enough so that the best known factoring algorithm requires time greater than in which the data must be protected. In Table 2.2, it is presented illustrative key sizes and corresponding cases in which should apply these sizes.

p, q	**n**	**protection time**	**data type**
256 bits	512 bits	few weeks	information that shortly affects the stock exchange (e.g. decision to merge two companies)
512 bits	1024 bits	50 - 100 years	personal secrets
1024 bits	2048 bits	> 100 years	commercial secrets, personal data
2048 bits	4096 bits	age of the Universe	military secrets

Table 2.2: RSA key sizes and indicative data types for protection.

In order to prove that the RSA cryptographic system can not crack, Rivest, Shamir and Adelman have requested who is think that he can analyze an integer with 129 digits into a product of two prime numbers. After 17 years, the number was analyzed by a network of 1600 computers. So with today's technological advances, the problem of analyzing a number into a product of two prime numbers can not be solved using modern computers and when the number has many digits.

2.5 ElGamal Cryptosystem

The ElGamal cryptosystem [126] is a probabilistic asymmetric public-key encryption algorithm that is based on the basic idea of Diffie-Hellman [50]. It was firstly described by Taher Elgamal in 1984. The security of ElGamal encryption is based on the discrete logarithm problem. In the following subsections, there is a detailed description of the algorithm.

Definition 5 (Probabilistic Encryption) *Probabilistic encryption is the use of randomness in an encryption algorithm, so that when encrypting the same message several times it will, in general, yield different ciphertexts.*

2.5.1 ElGamal Algorithm

The ElGamal algorithm [196] consists of three main parts: the key generation, the encryption algorithm and the decryption algorithm.

Key Generation. The procedure that should be followed to create a pair (public and private) of keys is:

1. We choose a random large prime number p and a prime generator g from the set ${Z_p}^*$, where ${Z_p}^*$ denotes the set of all integers $\{1, 2, \ldots, p-1\}$, i.e. $g^k \neq 1 \bmod p$ for all k smaller of $p-1$.
2. We choose a random number a in the interval $1 \leq a \leq p-1$ as the private key.
3. We calculate the $y = g^a \bmod p$.
4. The public key is the (p, g, y) and the private key is the a.

For finding the generator g should note the following:

- If the $g^k = 1 \bmod p$ is true for a integer in $1 \leq k \leq p-1$, then the number k necessarily divides the $p-1$.
- So, if we want to check if a number g is a generator in mod p, we do not need to raise to all the powers $\{1, 2, \ldots, p-1\}$, it is more sufficient to raise to the divisors of the $p-1$.

Encryption Algorithm. The algorithm that is followed to encrypt a message is as follows:

1. We choose a random number r in the set $\{1, 2, \ldots, p-1\}$.
2. We express the message with an integer m in the set $\{1, 2, \ldots, p-1\}$.
3. We calculate the $\gamma = g^r \bmod p$ and the $\delta = m\,y^r \bmod p$.
4. So the ciphertext is the $E_k(m, r) = (\gamma, \delta)$.

Decryption Algorithm. The algorithm that is followed to decrypt the initial message is:

1. We calculate the γ^{-a}, from the moment that we know the private key a.
2. The initial message is the $m = (\gamma^{-a})\, \delta \bmod p$.
3. With other words, the recovery of the initial message is the operation $\frac{\delta}{\gamma^a}$.
4. So the initial plaintext is the $D_k(\gamma, \delta) = m \pmod p$.

2.5.2 Homomorphic Property of ElGamal

The multiplicative homomorphic encryption property of the ElGamal cryptosystem means that multiplication of encrypted values corresponds to product of decrypted ones. Concretely, if the x_1 and x_2 are two plain integers, where $x_1, x_2 \in \mathbb{Z}_p^*$, and the notation $\mathcal{E}(x)$ is used to denote the encryption of the message x, then the ElGamal homomorphic property is shown by the following equation:

$$\begin{aligned}\mathcal{E}(x_1) \cdot \mathcal{E}(x_2) &= (g^{r_1}, x_1 \cdot y^{r_1})(g^{r_2}, x_2 \cdot y^{r_2}) \\ &= (g^{r_1+r_2}, (x_1 \cdot x_2)y^{r_1+r_2}) \\ &= \mathcal{E}(x_1 \cdot x_2)\end{aligned}$$

2.5.3 Security of ElGamal

The adversary will attempt to attack on the ElGamal cryptosystem, he must recover the private key a, by:

$$y = g^a \bmod p$$

with knowledge of the public key (p, g, y). So, the adversary should solve the discrete logarithm with base g. However, we believe that the security of ElGamal cryptosystem is based on discrete logarithm, because the solution of discrete logarithm can makes the cryptosystem unsafe, but it has not proved the opposite, i.e. that that the security of the cryptosystem is exclusively based on the discrete logarithm problem. The discrete logarithm problem is also known and as decisional Diffie-Hellman (DDH) assumption. If the decisional Diffie-Hellman assumption (DDH) is held, then the ElGamal achieves semantic security [82], that is, it is infeasible for a computationally bounded adversary to derive significant information about a message (plaintext) when given only its ciphertext and the corresponding public encryption key.

The existence of the random number r has as result the possibility of matching of plaintext with $p-1$ ciphertexts. The process, where the plaintext is mixed with a random variable, is called randomization process. This step, which is not in the RSA, makes the ElGamal cryptosystem more resistant to attacks such as those presented in RSA. Of course, the use of random number introduces an additional risk that leads to an additional requirement. For each message that is encrypted, you should choose a different random number r. If two messages m and m' are encrypted with the same random number r, then the corresponding ciphertexts will be the (γ, δ) and (γ', δ'), where the knowledge of a message allows the recovery of the other as follows:

$$\frac{\delta}{\delta'} = \frac{m \cdot y^r}{m' \cdot y^r} = \frac{m}{m'}$$

Finally, as regards the size of the p, the lower threshold is proposed to be 1024 bits. Generally, during the encryption with the ElGamal cryptosystem, this size is important implementation criterion, due to the increased time that is required for the encryption (two exponentiation operations versus one in RSA) and the expansion of ciphertext. These disadvantages have as result the reduced size of the preferred modulus.

2.6 Paillier Cryptosystem

The Paillier cryptosystem [141], invented and implemented by Pascal Paillier in 1999, is a probabilistic asymmetric algorithm (see Definition 5 in Section 2.5) for public-key cryptography. The problem of computing n-th residue classes is believed to be computationally difficult. The decisional composite residuosity assumption is the intractability hypothesis upon which this cryptosystem is based. In the following subsections, there is a detailed description of the algorithm.

2.6.1 Paillier Algorithm

The Paillier algorithm [199] consists of three main parts: the key generation, the encryption algorithm and the decryption algorithm.

Key Generation. The procedure that should be followed to create a pair (public and private) of keys is:

1. We choose two large prime numbers p and q randomly and independently of each other such that $\gcd(pq, (p-1)(q-1)) = 1$.

2. We calculate the $n = pq$ and the $\lambda = \operatorname{lcm}(p-1, q-1)$.

3. We select a random integer g where $g \in \mathbb{Z}^*_{n^2}$.

4. We ensure that n divides the order of g by checking the existence of the following modular multiplicative inverse: $\mu = (L(g^\lambda \bmod n^2))^{-1} \bmod n$, where function L is defined as $L(u) = \frac{u-1}{n}$.

5. The public key is the (n, g) and the private key is the (λ, μ).

Encryption Algorithm. The algorithm that is followed to encrypt a message is as follows:

1. Let m be a message to be encrypted where $m \in \mathbb{Z}_n$.

2. We select a random number r where $r \in \mathbb{Z}^*_n$.

3. The encrypted message is computed as: $E_k(m, r) = g^m \cdot r^n \bmod n^2$.

Decryption Algorithm. The algorithm that is followed to decrypt the initial message is:

1. Let c be the ciphertext of the encrypted message m where $c \in \mathbb{Z}^*_{n^2}$.

2. In order to decrypt the message, it is only needed the private key (λ, μ).

3. The result of the decryption is $D_k(c) = L(c^\lambda \bmod n^2) \cdot \mu \bmod n$.

2.6.2 Homomorphic Property of Paillier

The additive homomorphic encryption property of the Paillier cryptosystem means that multiplication of encrypted values corresponds to sum of decrypted ones. Concretely, if the x_1 and x_2 are two plain integers, where $x_1, x_2 \in \mathbb{Z}_n$, and the notation $\mathcal{E}(x)$ is used to denote the encryption of the message x, then the Paillier homomorphic property is shown by the following equation:

$$\begin{aligned}
\mathcal{E}(x_1) \cdot \mathcal{E}(x_2) &= (g^{x_1} \cdot r_1^n) \cdot (g^{x_2} \cdot r_2^n) \\
&= g^{[x_1 + x_2 \bmod n]} \cdot (r_1 r_2)^n \bmod n^2 \\
&= \mathcal{E}([x_1 + x_2 \bmod n])
\end{aligned}$$

2.6.3 Security of Paillier

The original Paillier cryptosystem [199] as shown above does provide semantic security [82] against chosen-plaintext attacks (IND-CPA). The ability to successfully distinguish the challenge ciphertext essentially amounts to the ability to decide composite residuosity. The so-called decisional composite residuosity assumption (DCRA) is believed to be intractable.

Because of the aforementioned homomorphic properties however, the system is malleable, and therefore does not enjoy the highest echelon of semantic security that protects against adaptive chosen-ciphertext attacks (IND-CCA2). Usually in cryptography the notion of malleability is not seen as an “advantage”, but under certain applications such as secure electronic voting [17] and threshold cryptosystems [46], this property may indeed be necessary.

Paillier and Pointcheval [142] however went on to propose an improved cryptosystem that incorporates the combined hashing of message m with random r. The hashing prevents an attacker, given only c, from being able to change m in a meaningful way. Through this adaptation the improved scheme can be shown to be IND-CCA2 secure in the random oracle model.

2.7 Public Key Certificate

In cryptography, a public key certificate [202] (or identity certificate) is a certificate which uses a digital signature to bind together a public key with an identity (ie, information such as the name of a person or organization, the address, etc.). This certificate may be used to check whether a public key belongs to an individual.

In a typical public key infrastructure (PKI) scheme, the signature will be of a certificate authority (CA). In a web of trust scheme, the signature is of either the user (a self-signed certificate) or other users (”endorsements”). In either case, the signatures on a certificate are attestations by the certificate signer that the identity information and the public key belong together. The most common certification standard is the ITU-T X.509. Several PKI international standards have been reviewed in [86].

A certificate may be revoked if it is found that the related private key has been lost, or if the relationship (between an entity and a public key) that is incorporated in the certificate is found to be inaccurate or has changed (e.g., if a person changes work or name). If and the revocation is a rare occurrence, the user should always check the validity of the certificate. This can be done by comparing the certificate with a certificate revocation list (CRL), that is, a list of revoked or canceled certificates. Another way to check the validity of certificates is also to question the certificate authority (CA) using the online certificate status protocol

(OCSP). Furthermore, a systematic and robust framework for the evaluation of the certificate revocation mechanisms is presented in [94].

The contents of a typical digital certificate X.509 are:

- **Serial Number:** Used to uniquely identify the certificate.
- **Subject:** The person, or entity identified.
- **Signature Algorithm:** The algorithm used to create the signature.
- **Signature:** The actual signature to verify that it came from the issuer.
- **Issuer:** The entity that verified the information and issued the certificate.
- **Valid-From:** The date the certificate is first valid from.
- **Valid-To:** The expiration date.
- **Key-Usage:** Purpose of the public key (e.g. encipherment, signature, certificate signing).
- **Public Key:** The public key.
- **Thumbprint Algorithm:** The algorithm used to hash the public key certificate.
- **Thumbprint:** The hash itself, used as an abbreviated form of the public key certificate.

2.8 Transport Layer Security

Transport Layer Security [206] (TLS) and its predecessor, Secure Sockets Layer (SSL), are cryptographic protocols that provide communication security over the Internet. They use asymmetric cryptography for authentication of key exchange, symmetric encryption for confidentiality and message authentication codes for message integrity. Several versions of the protocols are in widespread use in applications such as web browsing, electronic mail, instant messaging and voice-over-IP (VoIP).

The TLS protocol provides certified and isolated communication over the Internet using cryptography. In a typical use, only the server is certified (i.e. its identity is guaranteed) while the client remains anonymous. In a mutual certified communication it is required an additional scheme of public key infrastructure (PKI) to the clients. The TLS protocol allows client-server applications to communicate across a network in a way designed to prevent eavesdropping[1] and tampering[2]. Since protocols can operate either with or without TLS (or SSL), it is necessary for the client to indicate to the server whether it wants to set up a TLS connection or not. There are two main ways of achieving this; one option

[1] *Eavesdropping* is the act of secretly listening to the private conversation of others without their consent.

[2] *Tamper-evident* describes a device or process that makes unauthorized access to the protected object easily detected.

is to use a different port number for TLS connections (for example port 443 for HTTPS). The other is to use the regular port number and have the client request that the server switch the connection to TLS using a protocol specific mechanism (for example STARTTLS for mail and news protocols).

2.9 Secure Multi-Party Computation

In cryptography, the secure multi-party computation [204] (also known as secure computation or multi-party computation (MPC)) is a problem that was firstly proposed by Andrew C. Yao in 1982 [214]. The example that he used to describe the secure multi-party computation is the millionaire's problem: Alice and Bob are two millionaires that want to find out who is richer without revealing the precise amount of their wealth. In this problem, Yao proposed a solution that allows Alice and Bob to satisfy their curiosity while at the same time respecting the above limitation.

The millionaire problem and its solution gave way to a generalization to multi-party protocols. In an MPC, a given number of parties $P_1, P_2, \ldots, P_N$ each have a private data, respectively $D_1, D_2, \ldots, D_N$. The parties want to compute the value of a public function $\mathcal{F}$ on N variables at the point $(D_1, D_2, \ldots, D_N)$. An MPC protocol is secure if no party can learn more from the description of the public function and the result of the global calculation. More analytically, the multi-party protocol executed by $P_1, P_2, \ldots, P_N$ securely if the following conditions hold:

1. *Completeness:* If all parties $P_1, P_2, \ldots, P_N$ honestly follow the protocol then they obtain as output the correct computation of $\mathcal{F}$ on $D_1, D_2, \ldots, D_N$.

2. *Input/Output Privacy:* Any party behaving dishonestly in the protocol does not gain any information about the private inputs/outputs of the other parties (except the information that can be inferred by the output of the protocol and the own private input).

In the following subsection we define what is a dishonest behavior and how to prove that a protocol securely implements a functionality $\mathcal{F}$.

2.9.1 Dishonest Behavior/Security Notions

The dishonest behavior [78, 4] of the parties models the possible real-world attacks from adversarial machines. According to how much power is given to the adversary (corrupting honest machines, controlling schedule of/mauling the messages over the network, controlling the activation of the protocols) different security notions are defined. In the following we classify the dishonest behavior and therefore the security notions starting from the weakest one (that considers a very restricted real-world adversary) to the strongest one (that gives to the adversary full control over the network and the machines executing the protocol).

- **Honest-But-Curious Adversary:** The dishonest party must follow the protocol but can arbitrarily analyze the protocol transcript off-line in order to infer some additional information.
- **Malicious Adversary:** The dishonest party can arbitrarily deviate from the protocol and corrupt (i.e. obtain the entire state).
  - ***Adaptive/Static Corruption:*** Adaptive adversaries are allowed to decide the parties to corrupt during the protocol execution therefore depending upon the transcript and the state of the parties corrupted so far. Static adversaries instead corrupt the parties before the protocol execution starts.
  - ***Parallel/Concurrent Composability:*** One can require that the security of the protocol holds even when a dishonest party executes several instances of the same protocol (resp. different protocols) concurrently or in parallel. In this case we say that the protocol securely realizes a functionality under parallel/concurrent (resp. general concurrent) composition.
- **Computationally Bounded/Unbounded Adversary:** According to the computational capability of the real-world adversary each notion of security shown above is denoted as computational or unconditional. The computational setting assumes that the running time of the dishonest party is bounded by a polynomial. Unconditional setting puts no restriction on the running time of the adversary, i.e. it can be exponential.

Proving that a protocol satisfies a security notion - Ideal world/Real World paradigm: Input/Output Privacy is formally proved using the ideal/real world paradigm. Consider an ideal world in which there exists a trusted third party (TTP) who computes the functionality $\mathcal{F}$. In this world parties do not communicate with each other but they send their inputs to the TTP and receive the output of $\mathcal{F}$. Since the TTP is trusted in this world the privacy of the parties' inputs is guaranteed by definition. In order to prove Input/Output privacy it is required to show that whatever a dishonest party (the dishonest behavior depends of the security notion that one wants to prove) can infer about the inputs/outputs of the honest parties by exploiting the protocol execution in the real world, it can be inferred also by an adversary (called simulator) playing in the ideal world and interacting only with the TTP.

2.10 Zero-Knowledge Proof

A zero-knowledge proof [81] (ZKP) or zero-knowledge protocol is a method by which one party (the prover) can prove to another party (the verifier) that a given statement is true, without conveying any additional information apart from the fact that the statement is indeed true. For cases where the ability to prove the statement requires some secret information on the part of the prover, the definition implies that the verifier will not be able to prove the statement to anyone else. Notice that the notion only applies if the statement being proven is the fact that the prover has such knowledge. This is a particular case known as zero-knowledge proof of knowledge, and it nicely illustrates the essence of the notion of zero-knowledge proofs: proving that one possesses a certain knowledge is in most cases trivial if one is allowed to simply reveal that knowledge; the challenge is proving that one has such knowledge without revealing it or without revealing anything else.

For zero-knowledge proofs of knowledge, the protocol must necessarily require interactive input from the verifier, usually in the form of a challenge or challenges such that the responses from the prover will convince the verifier if and only if the statement is true (i.e., if the prover does have the claimed knowledge). This is clearly the case, since otherwise the verifier could record the execution of the protocol and prove it to someone else, contradicting the fact that proving the statement requires knowledge of some secret on the part of the prover.

Some forms of non-interactive zero-knowledge proofs of knowledge exist, but the validity of the proof relies on computational assumptions (typically the assumptions of an ideal cryptographic hash function).

2.10.1 Properties of a ZKP

A zero-knowledge proof [207] must satisfy three properties:

1. **Completeness:** If the statement is true, the honest verifier (that is, one following the protocol properly) will be convinced of this fact by an honest prover.

2. **Soundness:** If the statement is false, no cheating prover can convince the honest verifier that it is true, except with some small probability.

3. **Zero-knowledge:** If the statement is true, no cheating verifier learns anything other than this fact. This is formalized by showing that every cheating verifier has some simulator that, given only the statement to be proven (and no access to the prover), can produce a transcript that "looks like" an interaction between the honest prover and the cheating verifier.

The first two of these are properties of more general interactive proof systems. The third is what makes the proof zero-knowledge.

Zero-knowledge proofs are not proofs in the mathematical sense of the term because there is some small probability, the soundness error, that a cheating prover will be able to convince the verifier of a false statement. In other words, they are probabilistic rather than deterministic. However, there are techniques to decrease the soundness error to negligibly small values.

2.10.2 Variants of Zero-Knowledge

Different variants of zero-knowledge [207] can be defined by formalizing the intuitive concept of what is meant by the output of the simulator "looking like" the execution of the real proof protocol in the following ways:

- We speak of *perfect zero-knowledge* if the distributions produced by the simulator and the proof protocol are distributed exactly the same.
- *Statistical zero-knowledge* means that the distributions are not necessarily exactly the same, but they are statistically close, meaning that their statistical difference is a negligible function[1].
- We speak of *computational zero-knowledge* if no efficient algorithm can distinguish the two distributions.

[1] A negligible function is a function $\mu(x) : \mathbb{N}\to\mathbb{R}$ such that for every positive integer c there exists an integer N_c such that for all $x > N_c$, $|\mu(x)| < \frac{1}{x^c}$.

Chapter 3

Polis Framework

3.1 Introduction

As the use of computers and the Internet becomes more popular by the minute, the issue of protecting one's personal data is more essential than ever. The way electronic transactions are conducted nowadays, makes it necessary for the customer to give away his personal data to the service provider and hope that the latter will not use them in a malicious way. In order to protect personal information, several organizations and countries have issued privacy regulations, which should be followed in order for personal information to be protected; the collectively referred to as Fair Information Practices (FIP). Examples of important FIP regulation frameworks are the Data Protection Directive 95/46/EC (henceforth referred to as The Directive) and follow-ups like the Canadian PIPEDA and UK's Data Protection Act (DPA).

In this work, we assert that electronic transactions can be feasible, whilst personal data resides at the individuals' side. To support this claim, we design, build and evaluate the prototype system Polis, which implements the above principle. We show that Polis can satisfy important data protection principles in a natural and efficient way and describe how Polis can be integrated into online transactions to manage personal data. The results of this work indicate that the Polis approach can lead to a simple, scalable solution that can be beneficial to both individuals and service providers.

3.2 Related Work

The idea that individuals should own their personal information themselves and decide how this information is used, is discussed in [114]. A point made in [164] is that, although considering personal data the owner's private property is a very appealing idea, it would be rather difficult to practically apply it and legally enforce it. Our approach proposes an idea that has the same practical effect as considering personal data the owner's private property, but withdraws the legal objections involved with this idea. The economic aspects of privacy are examined in [188] where the following point is made: "It is worth observing that the Fair Information Practices principles would automatically be implemented if the property rights in individual information resided solely with those individuals". The argument that personal data would be safer at the user's side is also examined in [131].

Different kinds of frameworks that are related to personal data have recently been proposed or are in progress. In particular, privacy sensitive management of personal data in ubiquitous computing is discussed in [90], storing personal data in an individual's mobile device is examined in [95]. Protecting personal data that is stored within a company is considered in [163, 104]. More related to Polis is a rich but also complicated framework for privacy protection, proposed in [120]. This framework is built on the principle that personal data is kept inside a "Discreet Box", located at the service provider's side. An agent-based solution to address usability issues related to P3P (Platform for Privacy Preferences Project) is presented in [117]. Other results in this field, less related to Polis, can be found in [76, 15, 87]. General surveys on privacy enhancing technologies are given in [77, 88, 85]. To our knowledge, Polis is the first general framework for managing personal data only at the owner's side.

3.3 The Polis Approach

The Polis approach is based on the following principle:

> *"Polis-users are prohibited from storing any*
> *personal data but their own."*

Polis is meant to be employed by privacy concerned internet users which fulfill the requirements of having:

- A reliable, always-on access to the Internet, in order for his agent to be always accessible.

- A certificate from an approved Certification Authority.

We design, implement and evaluate a Polis prototype and show that the above simple and straightforward assumptions suffice to build a personal data management framework that works seamlessly with online transactions. The Polis prototype and its evaluation are described in Section 3.7.

3.3.1 Polis Concepts and Architecture

At this point we consider it necessary to introduce a few terms that will be used in this work:

- In Polis, personal data refers to primitive personal information of individuals like name, birth date, address, etc. Personal data corresponds to what is called *off-line identity* in [3]. Our focus is on privacy-enhanced management of the off-line identity.

- An individual Internet user is a potential customer who can purchase either goods or services. This user can be called *individual*, *customer* or *data subject (according to The Directive)*. We will use the terms *individual* and *customer*, interchangeably.

- An entity that provides the aforementioned goods or services can be called *shop*, *company*, *service provider* or *data controller (The Directive)*. We will use the terms *shop*, *company* and *service providers*.

- Both individuals and companies can become *Polis-users*.

Every Polis-user is represented by a dedicated entity. This entity can be used to instantiate a corresponding Polis-agent, which is the main architectural component of Polis. The agent is used to manage the personal data of the entity and to provide controlled access to it. Service providers use the agent to retrieve personal data from affiliated users. The general architecture of Polis, as well as the constituents of a customer agent and a shop agent are presented in Figure 3.1. We would like to emphasize the following characteristics of the Polis architecture:

- From the service provider's point of view, Polis provides a decentralized approach for the storage and management of personal data.

- On the contrary, from the customer's point of view, Polis is a fully centralized system, in the sense that personal data is located and managed locally by the owner's agent.

3.3.2 Schemes for Personal Data and Policies

Critical components for a personal data management framework like Polis are the schemes for representing personal data and policies. Some known schemes for personal data are P3P [191] and CPExchange [22]. Approaches for policies related to personal data are also discussed in [103, 104], while related work on personal data and policy schemes is performed in [52].

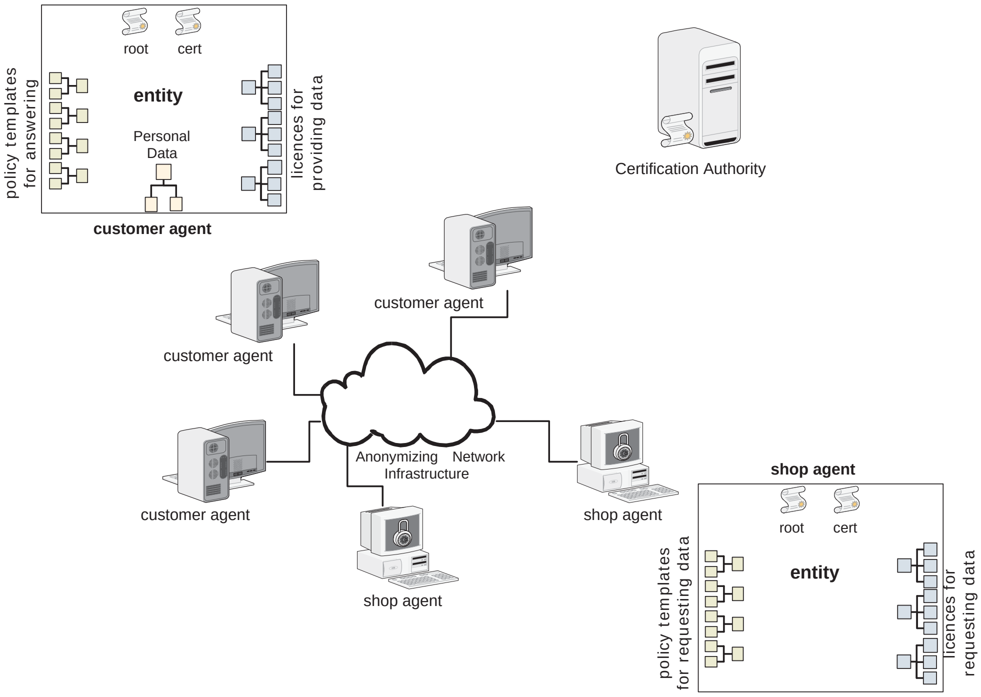

root
cert
entity
Personal Data
policy templates for answering
licences for providing data
customer agent
Certification Authority
customer agent
customer agent
customer agent
Anonymizing Network Infrastructure
shop agent
shop agent
shop agent
root
cert
entity
policy templates for requesting data
licences for requesting data

Figure 3.1: The Polis architecture.

We currently use schemes that are simple, yet powerful enough, for the current needs of the Polis prototype. Examples of a personal data scheme and a policy, as used in Polis, are shown in Figure 3.2.

There are eight general categories of personal data in Polis, organized hierarchically, namely Name, BDate, Cert, Skill, Characteristic, Home-Info, Business-Info and CreditCard. Each of them has one or more subcategories. The terminology used is based on P3P for the user information part, with the addition of the financial information (CreditCard) taken from CPExchange, plus the extra personal information fields (Skill and Characteristic). Each entity stores its personal data in a local XML document.

The components of a policy are the following:

- *Principals*: The Polis-entities.
- *Data*: Every single item of a user's (Polis-entity) personal data.
- *Purposes*: The set of purposes that entitle principals to retrieve data.
- *Usage restrictions*: Additional restrictions exist that limit access rights to a specific number of accesses or a specific time interval, or both.

```
Policy

<User Enabled="true" Entity=" eshop">
 <Name>
  <Given>
    <Permissions>
      <License Purpose="shipping">
        <GrantAccess>true</GrantAccess>
        <DateTime>
          <Start>2008-01-01 00:00:00</Start>
          <End>2008-12-31 23:59:59</End>
        </DateTime>
      </License>
      <License Purpose="billing">
        <Count>3</Count>
        <DateTime>
          <Start>2008-01-01 00:00:00</Start>
          <End>2008-12-31 23:59:59</End>
        </DateTime>
      </License>
        ...
    </Permissions>
  </Given>
 </Name>
</User>
```

```
Personal Data

<User>
 <Name>
   <Given>John</Given>
   <Family>Doe</Family>
 </Name>
 <Home-Info>
   <Postal>
     <Street>Nowhere Street 001</Street>
     <City>Deadend</City>
     <StateProv>Ouitcy</StateProv>
     <PostalCode>11111</PostalCode>
     <Organization>DIPH</Organization>
     <Country>Neverland</Country>
   </Postal>
   <Telecom>
      ...
   </Telecom>
 </Home-Info>
</User>
```

Figure 3.2: Examples of a personal data scheme and a policy.

Other important concepts of Polis are the licence and the contract. A *licence* comprises of the data involved, the valid purposes that allow data retrieval, as well as the rules to provide either full or restricted access. The use of licences to protect personal data is discussed in [34, 110, 70]. A *contract* concerns two principals and an arbitrary set of licences. An agent can sign any number of contracts with an arbitrary number of entities.

3.4 In Defense of the Polis Approach

The central assumption in the described personal data management approach is that personal data can only be stored at the owner's side. One may dismiss this as an unrealistic hypothesis and contend that we cannot count on users to abide by the Polis principle. One may also doubt that there are any incentives to adopt an approach like Polis, especially for the service providers.

We are well aware that Polis specifies an extreme approach for the management of personal data. However, Polis is meant to provide proof of concept that personal data management can be fair, privacy respecting and more effective than current practices. What makes the Polis approach possible is that the recent scientific and technological developments and especially the universal acceptance of the Internet have prepared the ground for citizen-centric applications. At the same time, the powerful surveillance and data management tools have contributed to making privacy threats and personal data misuse one of the most important problems in the electronic world.

For the above reasons, we believe that the conditions are mature for investigating alternative paradigms in the management of personal data. The new paradigms should enhance the individuals' control over their personal data. In this context we designed, implemented and evaluated the Polis platform. There is no doubt, that much more has to be done for a solution like Polis to find its way to practice. However, this work constitutes a confirmation that such a solution is feasible.

With Polis we aim at providing a consumer-led solution for personal data management to be used instead of the current company-centric approach. We would like to point out an interesting analogy of the Polis-proposed switch in the current practices in the field of personal data management with another proposed switch in the field of identity management. According to the Crosby report [44] we should focus on identity assurance instead of identity management:

> At an early stage, we recognized that consumers constitute the common ground between the public and private sectors. And our focus switched from "ID management" to "ID assurance". The expression "ID management" suggests data sharing and database consolidation, concepts which principally serve the interests of the owner of the database, for example the Government or the banks. Whereas we think of "ID assurance" as a consumer-led concept, a process that meets an important consumer need without necessarily providing any spin-off benefits to the owner of any database [44].

The Crosby report [44] was published in March 2008 after we had designed and built the Polis prototype. We consider it very encouraging that positions about consumer-led solutions are expressed within a very applied context, like the Forum on Identity Management which prepared the report.

3.5 Incentives and Objections

In this Section we discuss incentives and objections for the Polis approach and provide arguments that the Polis solution can be beneficial not only to individuals but also to (well meaning) shops.

The fact that a Polis-user's personal data must be retrieved from the owner's side every time it is needed, automatically fulfills many critical requirements found in FIP-like regulations. Moreover, the way the Polis framework can be integrated into database management systems automatically fulfills the requirements of Hippocratic databases [9]. Besides individuals, shops can also obtain important benefits from the adoption of Polis, like a more privacy-friendly profile, simplified data maintenance and data cleansing, as well as significantly reduced responsibility for the safety of customers' personal data.

3.5.1 Incentives for Individuals

1. *Polis-users maintain maximum control over their personal data.* They are able to monitor, at any point in time, all the contracts they have signed, as well as the time, purpose and principal of each data item access has taken place. Unlike the Polis approach, current practices result in inability to keep track of where one's personal data reside and how often they are being accessed.

2. *Individuals can trivially exercise their right to up-to-date personal data.* A user has simply to update his locally stored record. Each time a company wants to access it, it will be retrieved on-the-fly from the person's agent and not from the company's outdated database. Consider an individual who changes his address or telephone number. With current practices, the individual has to recall every peer that rightfully possesses this information and go through a record update procedure for each such peer.

3. *Individuals can handle all kinds of privacy-related rights and preferences through their Polis-agent.* To this end, a unified user interface is being used, while personal data disclosure takes place through a clear data flow. These attributes are absolute requirements for the effective privacy-enhanced management of personal data [116].

4. *The risk that the privacy of the individual is violated due to data breaches from company databases is significantly reduced.* Just like credit cards, Polis-contracts can be canceled to become useless to invaders. Even if a company does not realize that its database has been compromised, invaders will have to acquire the company's private key, in order to be able to use the stolen contracts. Even in that case, the invaders will only have access to the particular data that the contracts authorize this company for. Furthermore, the data owner will be able to know what data has leaked and when this happened.

5. *Privacy-concerned individuals will no longer have to choose between either giving away their personal data or not conducting an electronic transaction.* Nowadays individuals suffer the coercion that occurs when there is only one reasonable way for them to receive certain needed services or information [128], i.e., by giving away their personal data. Furthermore, according

to Acquisti in [3]: "... as merchants decide against offering anonymizing technologies to their customers, the privacy concerned customers choose not to purchase on-line, or to purchase less. A latent, potentially large market demand remains therefore unsatisfied". We believe that approaches like Polis can offer a viable alternative to current practices for personal data management.

3.5.2 Incentives for Service Providers

1. *The customer's personal data in the shop's database remains always up-to-date.* In addition, this is accomplished without any maintenance costs for the shop.

2. *The use of Polis contributes to improved data quality and can simplify the data cleansing task.* Data cleansing is the act of detecting and removing and/or correcting a database's dirty data (i.e., data that is incorrect, out-of-date, redundant, incomplete, or formatted incorrectly). Data quality is a critical factor for the success of enterprise intelligence initiatives and can incur costs and delays to company operations [154, 189].

3. *Polis releases the shop from the burden and responsibility of keeping customer data safe.* The shop is freed from a set of serious responsibilities for protecting customers' personal data and the risk of being considered liable for serious data breaches. Incidents of intentional or unintentional data breaches are unfortunately quite common and a reasonable worry is that a lot of them never reach the attention of the media. Some representative examples of such situations are the Choicepoint case, a data broker who sold private records of over 150,000 Americans to a group of criminals in 2005 [41], the incident that took place in the UK, where two computer discs containing the personal data of 25 million citizens were lost in the post [10], as well as the recent Deutsche Telecom incidents [158].

4. *Polis promotes a more privacy-friendly image for the service providers that adopt it.* The commitment that the shop does not store any personal data locally is an appealing argument for privacy-sensitive customers.

5. *Polis can be integrated into a company's existing information system.* As we illustrate in Section 3.7, Polis can naturally handle heterogeneous sets of customers, consisting of both Polis and conventional ones. This fact removes the counterincentive of companies having to go through a demanding transition process in order to integrate Polis into their systems. The com-

pany does not get tied down by Polis into having only Polis users in its database.

3.5.3 Potential Objections for Individuals

1. *Managing a personal agent is by definition a critical task, prone to errors and omissions by the user.* However, being in charge of the data disclosure process through a unified procedure like Polis, is much more convenient and effective compared to current practice, as described in incentive for individuals 3.

2. *Considerations about the agent's security.* An individual's Polis-agent contains critical personal data and digital agreements for data access. Consequently, a production-ready Polis-agent should satisfy high security levels. We believe that this is a viable task, since the Polis-agent has a precise, well-defined functionality and can be operated behind firewalls on a user-controlled computing platform. Moreover, the decentralized approach of Polis for personal data can also contribute to improved data security, since invaders find large collections of personal data much more inviting than an individual's personal data [131].

3. *Polis does not protect individuals from malicious shops that misuse personal data.* Nevertheless, a malicious shop in Polis cannot cause more damage than it could cause with current practices.

3.5.4 Potential Objections for Service Providers

1. *Loosing control of customer data.* This objection does not really apply to the Polis approach since service providers will still have access to the data they are entitled to. Well-meaning parties will not loose control over their customer's data. Internet connection reliability could also be an issue for Polis, but as already mentioned, it is widely accepted that soon enough, reliable Internet connectivity will be considered a given. Besides, Polis does not restrain companies from keeping records of customer's profiles. These records will not contain any data of the customer's offline identity and will resemble pseudonymous data processing.

2. *The adoption of Polis can cause significant overheads to company processes.* The possible delays in data retrieval caused by the employment of Polis should not be a hindering factor for its adoption. Retrieval of personal data is neither a task that is carried out frequently, nor a time critical

process, therefore these delays will not affect the efficiency of the company procedures.

3. *Service providers could be scammed from malicious Polis-users.* Polis-contracts and licences constitute proof that a service provider has the right to access the specified Polis-user data. Therefore, when needed, a company can resort to the appropriate actions. The CA or some other designated trusted third party could be used to settle such cases.

3.5.5 Enforcement and Detection

An important aspect of every (electronic) contract is the ability to verify and enforce that the parties will not violate its terms. Polis can handle detectable privacy breaches, i.e., breaches for which data released to the shop finds its way back to the individual who submitted that information [83]. In this case a Polis compliant shop must be able to present evidence that those data were rightfully obtained for the specific purpose, at the specific time, using data licences [34]. A more challenging task would be to detect Polis-shops that leak customer's personal information. A relevant problem is discussed in [83].

Due to the very nature of personal data, it seems that once a service provider possesses some data, there is no technically feasible way for absolute abuse prevention. Consequently, apart from technical measures, we will have to rely on market, legal and social dynamics for handling personal data properly ([83, 8] and [90, Section 5.8.5]).

As far as violations from the user's side are concerned, if the terms of an agreement are violated and the individual refuses to fulfill his contract defined obligation of providing personal information, then the service provider can use his customer-signed license to prove entitlement to access the data.

3.6 Polis Applications

In this Section, we discuss how Polis can be used within common electronic transactions and present indicative higher level applications that can be built on top of a decentralized personal data management framework like Polis.

3.6.1 Polis in Common Transactions

Polis can be potentially employed in any transaction where a user has to enter (some of) his personal data. The overall procedure is outlined below:

Polis in a transaction. When the user has to fill in a form with his personal data, he instead provides the contact details of his agent. The agents of the shop and the user/customer establish an agreement. A successful agreement grants access to the customer's private data for the specific data items and the amount of time needed to complete the transaction. In Figure 3.3 the procedure of a Polis-transaction with an e-shop is detailed.

This procedure can be used for registrations at e-shops, portals and other online services. In general, any application that involves personal data, like identity management systems [150] and e-government platforms can be supported. The need for privacy protection in e-business applications is stressed in [106]. The ease of employing Polis lies in the fact that it can work as middleware, which takes care of the personal data exchange between parties in higher level applications.

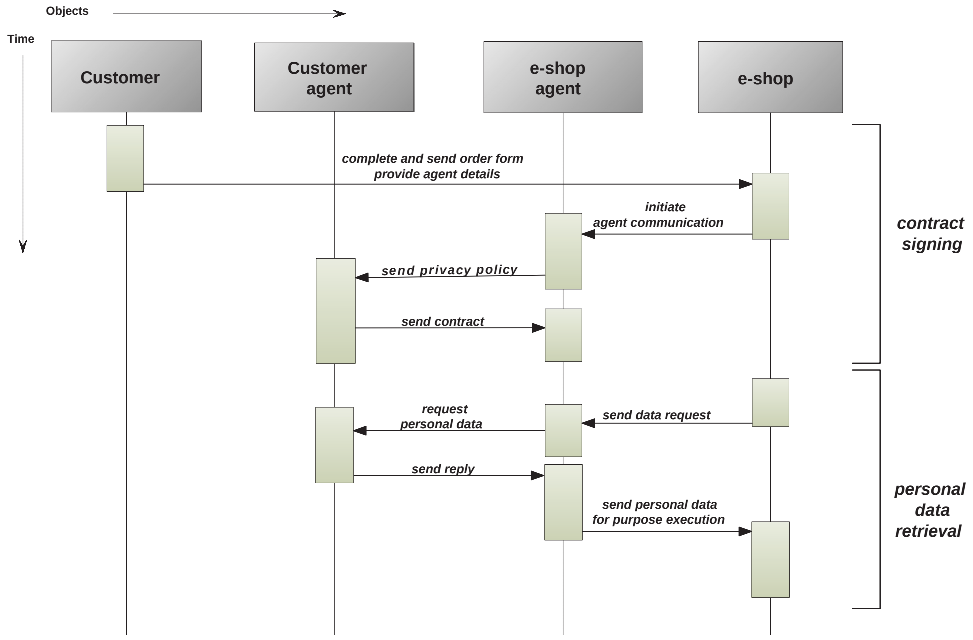


Figure 3.3: Polis in transactions with an e-shop.

3.6.2 Prospective Applications for Polis

An infrastructure like Polis can be a realistic step in the direction of effectively controlling personal information. Apart from the direct gains of using Polis in every day electronic transactions, there are some interesting possibilities for higher level applications that could utilize it.

Microtrades and Information Markets. The financial aspects of privacy are studied in several works like [114, 188, 109, 3]. Polis could be utilized to facilitate personal data exchange in personal-level microtrades between Polis-agents. Such an application is examined in [179]. Polis-users can give permission to information gathering companies to access (some of) their personal data, for an agreed price. Each time a company needs to regain access to them, the agreed amount of money should be paid. Furthermore, Polis could provide the ground for more advanced financial applications for personal data. The market for personal data described by Laudon in [114] is an example of such applications. In particular, Laudon proposes the so called National Information Market (NIM), where personal information can be traded in a National Information Exchange. The adoption of a framework like Polis would simplify the evolution of NIM-like infrastructures.

Privacy-enhanced ubiquitous computing. Online data of an individual can be conveyed through his Polis-agent. In this case, Polis could work as an open architecture for ubiquitous computing applications. For example, dynamic location information could be retrieved from the individual's Polis-agent, like the rest of his personal data.

3.7 The Polis Prototype

We designed and implemented a Polis prototype with the main objective to demonstrate that electronic transactions are feasible while personal data remain only at the owner's side. Another technical objective of the development of the Polis prototype was to make its deployment simple and friendly to contemporary information management practices. We believe that we have fulfilled the above goals adequately. Furthermore, a fully developed Polis platform should be able to satisfy the general properties that a privacy technology must have in order to be considered useful according to [77]. The freely available Polis binaries and online demos of Polis can be found at the Polis project site [149].

3.7.1 Technologies of the Prototype

The basic technologies used to develop and employ the Polis prototype are:

- The Eclipse IDE and the Java programming language to create portable, platform independent tools.

- A Pubic Key Infrastructure (PKI) is used for creating trusted certificates according to the X.509 standard. For demonstration purposes, an elementary Polis CA has been developed to be used in experiments. In a real world application, a commercial CA could be utilized.

- User data, policies and contracts are represented as XML documents.

- The Tor anonymizing infrastructure is used optionally to achieve anonymity for the clients and/or implement agents as hidden services [51].

- The Derby embedded database server is employed internally by the agent for its data storage needs.

- Bouncy Castle's security provider is used for cryptographic primitives.

- The database case study has been implemented on an Oracle database management server (DBMS). Similar integrations of Polis should be feasible with other popular DBMSs like IBM DB2 or Microsoft SQL Server.

3.7.2 Deploying the Polis Prototype

In order to deploy Polis:

- Customers install the Polis-agent, store their personal information and prepare the necessary policy templates.

- Companies install the Polis-agent, prepare policy templates and integrate the agent with the company's information system. *Polis-customers can coexist with normal customers at a company side.*

3.7.3 Polis Collaborating with a Database Management System

Polis can be incorporated into the back-office of a company and take care of the personal data management. This is accomplished by integrating Polis with the company's database management system. The basic idea is that personal data fields do not contain the actual data, instead, a ticket (represented by an appropriate object) is used to retrieve the data value on the fly. We tested Polis with an Oracle database server. The approach is illustrated in Figure 3.4.

The integration was straightforward. Two Java Stored Procedures (JSP's) and a small set of triggers and database views were sufficient to implement the connectivity between Polis and the database server. It is noteworthy that, using simple object-relational features, as well as views and triggers, the Polis enhanced

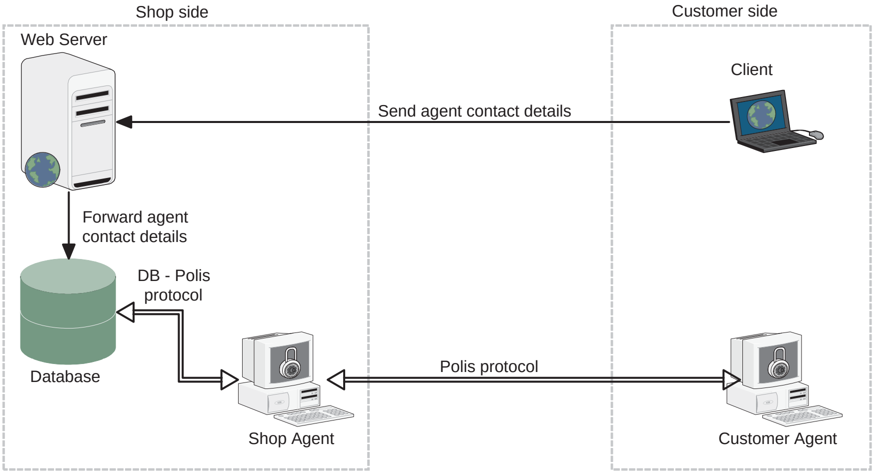
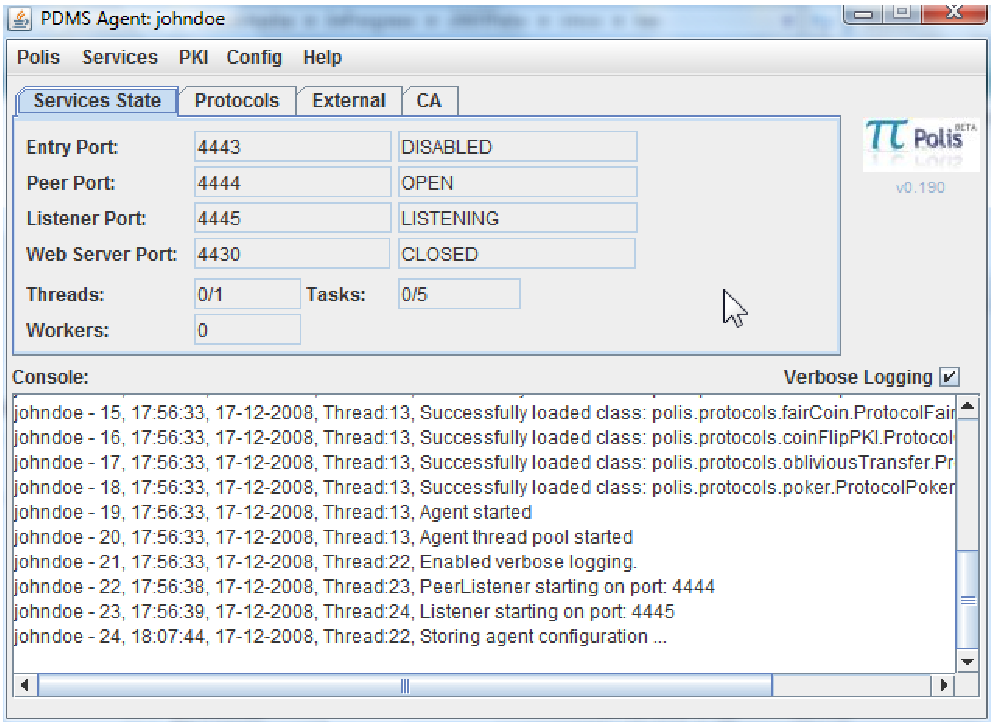


Figure 3.4: A Polis-entities interaction example. The shop uses a Polis-enabled database for customer registration and for retrieving personal data of customers.

database can be operated as a normal one, while the Polis related operations are transparent to the database user.

3.7.4 Experimental Evaluation

We prepared an elementary Polis environment with a set of Polis-agents installed on the local network of our laboratory. A snapshot of a Polis-agent's GUI is shown in Figure 3.5. A set of web pages, including web forms and dynamic web pages, were used to support experiments. The customer database contained 27 customers in total; 11 conventional customers and 16 Polis-customers (4 of which used Tor hidden services [51] for their agents). We performed an extensive set of experiments within the above Polis environment. The experiments involved database operations on tables with Polis data to verify their integration into the database. In particular, we executed some representative *insert* and *select* operations on the customer table, a *join* operation between two tables and created some *views* in the database. Both Tor-enabled Polis-agents (the agent is accessed through a Tor hidden service) and conventional Polis agents were tested. As expected, all the operations were accomplished successfully. Moreover, the Tor-enabled agents operated indistinguishably from the conventional agents with the exception of occasional timeouts, due to the Tor network itself.



Entry Port: 4443 DISABLED

Peer Port: 4444 OPEN

Listener Port: 4445 LISTENING

Web Server Port: 4430 CLOSED

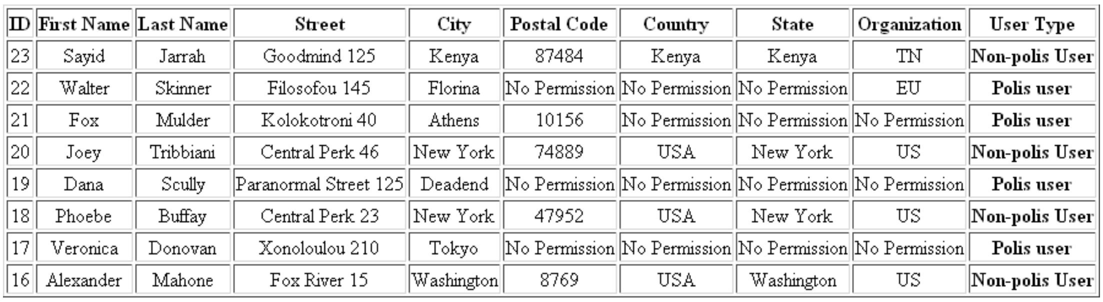
Threads: 0/1 Tasks: 0/5

Workers: 0

Console: Verbose Logging ☑

```
johndoe - 15, 17:56:33, 17-12-2008, Thread:13, Successfully loaded class: polis.protocols.fairCoin.ProtocolFair
johndoe - 16, 17:56:33, 17-12-2008, Thread:13, Successfully loaded class: polis.protocols.coinFlipPKI.Protocol
johndoe - 17, 17:56:33, 17-12-2008, Thread:13, Successfully loaded class: polis.protocols.obliviousTransfer.Pr
johndoe - 18, 17:56:33, 17-12-2008, Thread:13, Successfully loaded class: polis.protocols.poker.ProtocolPoker
johndoe - 19, 17:56:33, 17-12-2008, Thread:13, Agent started
johndoe - 20, 17:56:33, 17-12-2008, Thread:13, Agent thread pool started
johndoe - 21, 17:56:33, 17-12-2008, Thread:22, Enabled verbose logging.
johndoe - 22, 17:56:38, 17-12-2008, Thread:23, PeerListener starting on port: 4444
johndoe - 23, 17:56:39, 17-12-2008, Thread:24, Listener starting on port: 4445
johndoe - 24, 18:07:44, 17-12-2008, Thread:22, Storing agent configuration ...
```

Figure 3.5: A Polis-agent's GUI snapshot.

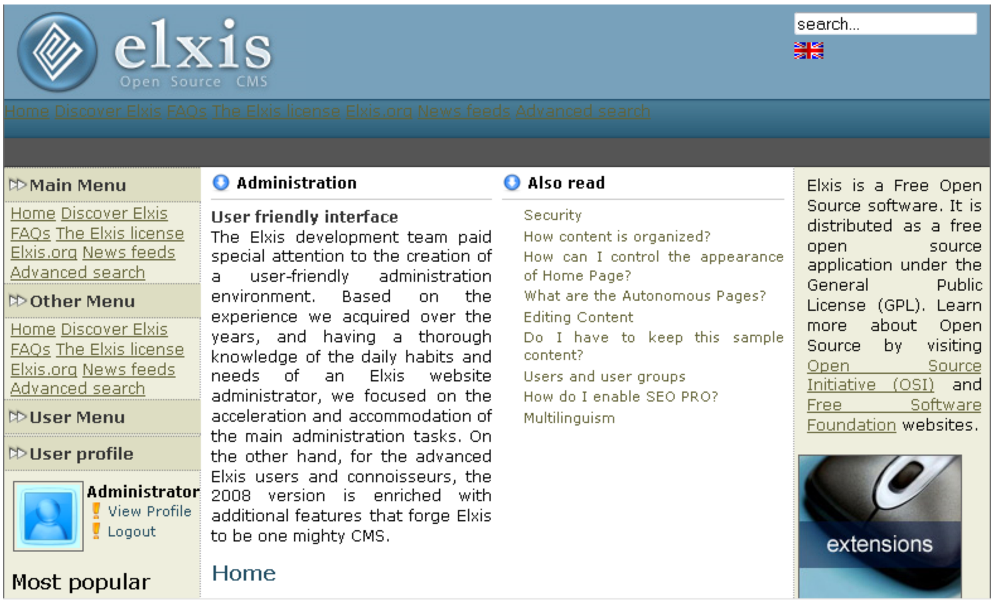


ID	First Name	Last Name	Street	City	Postal Code	Country	State	Organization	User Type
23	Sayid	Jarrah	Goodmind 125	Kenya	87484	Kenya	Kenya	TN	**Non-polis User**
22	Walter	Skinner	Filosofou 145	Florina	No Permission	No Permission	No Permission	EU	**Polis user**
21	Fox	Mulder	Kolokotroni 40	Athens	10156	No Permission	No Permission	No Permission	**Polis user**
20	Joey	Tribbiani	Central Perk 46	New York	74889	USA	New York	US	**Non-polis User**
19	Dana	Scully	Paranormal Street 125	Deadend	No Permission	No Permission	No Permission	No Permission	**Polis user**
18	Phoebe	Buffay	Central Perk 23	New York	47952	USA	New York	US	**Non-polis User**
17	Veronica	Donovan	Xonoloulou 210	Tokyo	No Permission	No Permission	No Permission	No Permission	**Polis user**
16	Alexander	Mahone	Fox River 15	Washington	8769	USA	Washington	US	**Non-polis User**

Figure 3.6: Report from the customer table of a Polis-enabled database. The table contains both, Polis and non-Polis, users.

3.7.5 A First Case Study

We performed a preliminary case study on the integration of Polis with a content management system (CMS). More precisely, we integrated Polis with the Elxis CMS, an open source CMS released under the GNU/GPL license. We selected Elxis for our case study because it is a fully functional CMS, it is open source and it supports the Oracle DBMS. A number of extensions exist for Elxis that enrich its functionality; one of them, the IOS eshop component, turns Elxis into an e-shop.



elxis
Open Source CMS

search...

Home Discover Elxis FAQs The Elxis license Elxis.org News feeds Advanced search

Main Menu

Home Discover Elxis
FAQs The Elxis license
Elxis.org News feeds
Advanced search

Other Menu

Home Discover Elxis
FAQs The Elxis license
Elxis.org News feeds
Advanced search

User Menu

User profile

Administrator
View Profile
Logout

Most popular

Administration

User friendly interface
The Elxis development team paid special attention to the creation of a user-friendly administration environment. Based on the experience we acquired over the years, and having a thorough knowledge of the daily habits and needs of an Elxis website administrator, we focused on the acceleration and accommodation of the main administration tasks. On the other hand, for the advanced Elxis users and connoisseurs, the 2008 version is enriched with additional features that forge Elxis to be one mighty CMS.

Home

Also read

Security
How content is organized?
How can I control the appearance of Home Page?
What are the Autonomous Pages?
Editing Content
Do I have to keep this sample content?
Users and user groups
How do I enable SEO PRO?
Multilinguism

Elxis is a Free Open Source software. It is distributed as a free open source application under the General Public License (GPL). Learn more about Open Source by visiting Open Source Initiative (OSI) and Free Software Foundation websites.

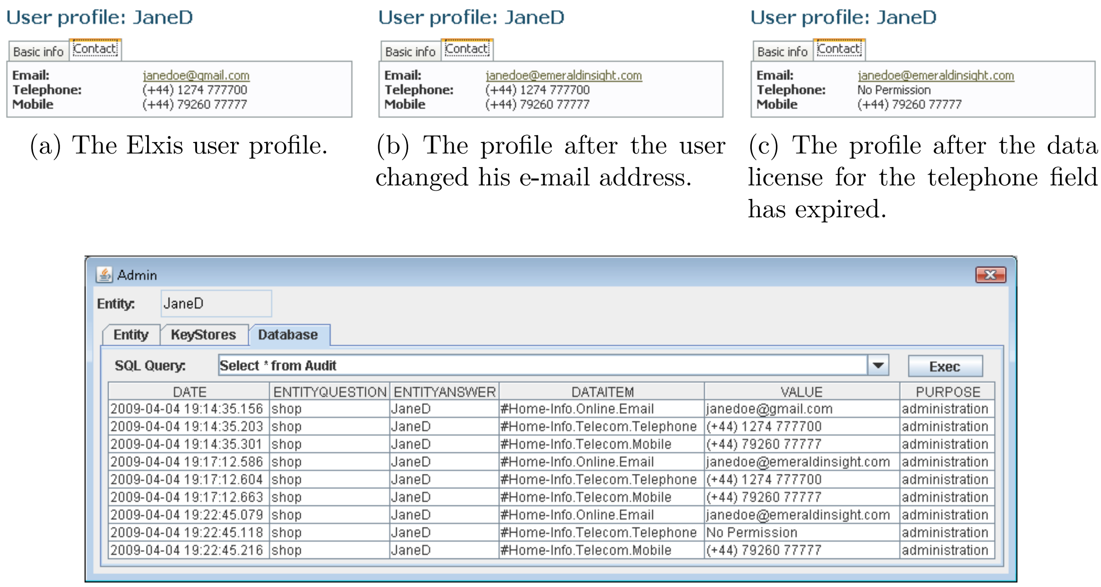
Figure 3.7: The Polis-enabled Elxis CMS instance.

The integration process was straightforward. After less than a man week of work, a first version of a Polis-enabled Elxis CMS instance was working in beta status (Figure 3.7). Figure 3.8 shows how the profile of a specific user appears in the Elxis application, at different time instances. Auditing the user's Polis-agent reveals each access to the user's personal data.

3.8 Conclusions

The evaluation of the Polis prototype and the Polis case study proved the feasibility of the main Polis approach and confirmed the features of the Polis approach discussed in Section 3.5. A comparison of the features that are available to Polis-enabled individuals in contrast to conventional/non Polis-enabled individuals of the Elxis CMS-based application is shown in Table 3.1. The table compares the Polis approach to the current practice in personal data management. To this end it highlights a set of important advantages/disadvantages for Polis-users of the Polis-enabled Elxis CMS. The comparison should be valid for a wide range of possible Polis-enabled e-business applications.

As noted in Section 3.1, to the authors knowledge, there is no approach comparable to Polis for managing personal data wholly at the owner's side. The closest work is the approach of [120] where the data of individuals resides in a



User profile: JaneD

Email: janedoe@gmail.com
Telephone: (+44) 1274 777700
Mobile (+44) 79260 77777

(a) The Elxis user profile.

User profile: JaneD

Email: janedoe@emeraldinsight.com
Telephone: (+44) 1274 777700
Mobile (+44) 79260 77777

(b) The profile after the user changed his e-mail address.

User profile: JaneD

Email: janedoe@emeraldinsight.com
Telephone: No Permission
Mobile (+44) 79260 77777

(c) The profile after the data license for the telephone field has expired.

Entity: JaneD

SQL Query: Select * from Audit

DATE	ENTITYQUESTION	ENTITYANSWER	DATAITEM	VALUE	PURPOSE
2009-04-04 19:14:35.156	shop	JaneD	#Home-Info.Online.Email	janedoe@gmail.com	administration
2009-04-04 19:14:35.203	shop	JaneD	#Home-Info.Telecom.Telephone	(+44) 1274 777700	administration
2009-04-04 19:14:35.301	shop	JaneD	#Home-Info.Telecom.Mobile	(+44) 79260 77777	administration
2009-04-04 19:17:12.586	shop	JaneD	#Home-Info.Online.Email	janedoe@emeraldinsight.com	administration
2009-04-04 19:17:12.604	shop	JaneD	#Home-Info.Telecom.Telephone	(+44) 1274 777700	administration
2009-04-04 19:17:12.663	shop	JaneD	#Home-Info.Telecom.Mobile	(+44) 79260 77777	administration
2009-04-04 19:22:45.079	shop	JaneD	#Home-Info.Online.Email	janedoe@emeraldinsight.com	administration
2009-04-04 19:22:45.118	shop	JaneD	#Home-Info.Telecom.Telephone	No Permission	administration
2009-04-04 19:22:45.216	shop	JaneD	#Home-Info.Telecom.Mobile	(+44) 79260 77777	administration

(d) The user's Polis-agent audit.

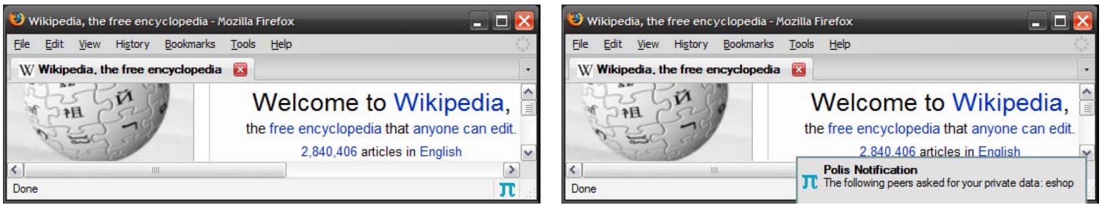
Figure 3.8: The profile of a Polis-enabled Elxis user at different time instances and the corresponding entries in the user's Polis-agent audit.

		Polis-enabled user(s)	Conventional user(s)
1	awareness	the individual is aware of any access to his data	the individual receives no information
2	data opt-out	trivial	the individual has to contact the Elxis shop
3	specifying policies	the individual can specify his own data access policies or adopt proven policy templates	the individual has to rely on the company specified privacy policy
4	security	the individual is not affected by most types of attacks on the Elxis shop	any data leakage from the Elxis shop may affect the individual
5	effort	the individual has to manage his own data	the individual simply gives away his data
6	delay	the data is retrieved from the Polis-agent of the individual	data is retrieved from the company database
7	data cleansing	trivial (the data is retrieved from the Polis-agent)	data entry errors may occur
8	data update	trivial (happens implicitly)	the individual has to go through a proprietary procedure

Table 3.1: Advantages/disadvantages for Polis-users of the Polis-enabled Elxis CMS instance.

Discreet-box located at the shop side. However, this case differs from Polis in that data is not at the premises of the individual. Furthermore, it is more complicated than Polis and unfortunately we were unable to obtain an implementation for evaluation/comparison purposes. Other technologies, like IBM's Tivoli Privacy Manager, concern the management of customer data within a company, but do not hand over the control to the actual customer. Such technologies can complement the Polis approach (by increasing the accountability of the company's data practices) but are not a substitute for it.

Finally, the evaluation of Polis also revealed important improvements that are possible for the Polis-agent and the accompanying tolls. One improvement concerns the usability of Polis. In the relevant literature it is pointed out that the use of an appropriate Graphical User Interface (GUI), that clearly demonstrates concepts for expressing privacy preferences, is very supportive for effective use of privacy protection systems [116, 143, 96, 2]. A first tool that we have created is a Polis Add-on for Firefox, which, amongst other features, alerts the user each time his personal data are requested (Figure 3.9).


Wikipedia, the free encyclopedia - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Wikipedia, the free encyclopedia
Done
Wikipedia, the free encyclopedia - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Wikipedia, the free encyclopedia
Done

Welcome to Wikipedia,
the free encyclopedia that anyone can edit.
2,840,406 articles in English

Welcome to Wikipedia,
the free encyclopedia that anyone can edit.
2,840,406 articles in English

Polis Notification
The following peers asked for your private data: eshop

(a) Internet browsing moment of a Polis-user. (b) The user's data have been requested.

Figure 3.9: The Polis Add-on for Firefox alerts the user when his personal data are requested.

3.9 Discussion

In this work, we design, implement and evaluate Polis, a personal data management framework which embodies a fundamental privacy principle: Personal data of individuals reside only at their side. Polis aims at making storage of personal data unnecessary for contemporary online transactions to work efficiently. This way, users will be able to monitor and limit the distribution of their personal data, according to their needs and preferences. Furthermore, the safety of stored personal data is enhanced and personal data accuracy is ensured.

In conclusion, this work demonstrates the fact that it is possible to deploy a privacy-enhancing prototype like Polis, in order to achieve significant privacy protection, in the current electronic world. We cannot expect Polis to become a panacea for all kinds of privacy problems. However, we believe that Polis has more advantages than disadvantages compared to current practices for personal data management. Finally, it is very encouraging that given one basic assumption, the transition to a personal data protecting way of conducting online transactions, can be natural and smooth.

Chapter 4

Privacy-Preserving Solution for Finding the Nearest Doctor

4.1 Introduction

A very interesting class of personal data is dynamic personal data, such as the current location of an individual. The recent progress in mobile device technology and the advances in ubiquitous computing allow individuals to collect and process such dynamic personal data. At the same time, the protection of this ubiquitous personal data is extremely critical for the privacy of individuals [56, 55]. This enables a new class of important applications. Consider, for example, the location of an expert, and in particular a doctor. In case of an emergency, the distance of the closest doctor could be live-saving information. In August 2007, in the area of Alexandroupolis, Greece, a 17-year old boy was seriously injured in his right leg. Vascular surgery was urgently needed. However, due to several administrative faults no specialized doctor was available. Even worse, it took a long time until it became clear that no specialized doctor could be found and only then the boy was transported to a hospital in Thessaloniki. Unfortunately, due to the long delay the injured leg had to be amputated. Even with the transport taking place, had the initial delay to find out where the nearest specialized doctor is been avoided, the consequences on the boy's health might have been less serious [181].

In this work, we focus on dynamic personal data and examine the possibility of development of innovative applications that exploit this kind of data, while ensuring the privacy of individuals. To this end, we propose the following problem, called the Nearest Doctor Problem (NDP), to find the nearest doctor in case of an emergency. In a hypothetical but feasible scenario, each doctor has a personal agent where his current location is always stored. In case of an emergency, the agents of all doctors interact to identify the doctor who happens to be closer than any other to the emergency location. We assume that the doctors may be off duty and thus the current location of each doctor is sensitive personal data that should not be revealed to anyone, including other doctors.

The NDP problem is an example of a privacy-preserving application based on dynamic personal data; the location of each individual doctor. We propose a privacy-preserving solution that solves the problem without revealing the location of any doctor. The individual who is anonymously identified as the closest doctor can then reveal his identity and offer his services to the emergency event. The privacy guarantees of this work concern the current location of each doctor, which is the only personal information of the participants (doctors) used in the computation. Our approach solves the NDP problem without any doctor location being disclosed; only a small amount of aggregate or anonymous information about participants distances is leaked.

The solution that we propose for NDP makes use of cryptographic primitives and decentralized computation technologies. A basic assumption is that all doctors have at their disposal a personal data management agent where their current location is stored. Each agent is under the control of its owner and all personal agents are permanently connected to the Internet. We consider this assumption feasible because, on the one hand, most modern smartphones (even low cost smartphones) are equipped with a GPS which allows the users to monitor their current location and, on the other hand, the doctor agent can exist in a cheep, low-energy consuming nettop or possbily even in the ADSL router.

In case of an emergency, the agents of all doctors execute a distributed computation to identify in a cryptographic safe way who is the closest doctor to the incident. For performance, scalability and fault tolerance reasons and additionally for enhancing privacy, the computation is executed in a fully decentralized way. The agents/nodes are (self-)organized in a distributed topology. To achieve this we employ techniques from the field of Peer-to-Peer (P2P) networks. The use of P2P techniques allows us to satisfy the requirements of high scalability of the system and to reduce the risk for privacy breaches. We apply techniques that have been developed in [173] and are based on the well known Chord [176] architecture for P2P networks.

The NDP problem is an illustrative example of an application where personal data can be used for a common good (public health) whereas at the same time the privacy of all involved individuals is preserved. We believe that many new applications can emerge from the same principle of simultaneously using and protecting personal data. For example:

- First aid in case of a car emergency. The European Union has launched the eCall project [68] for dealing with the ability of providing assistance in case of car emergencies. The project's goal is to deploy a hardware black box installed in vehicles that will send an emergency request in case of an accident on the road. The request will be transmitted over wireless communication technologies like GSM and will include information like the GPS coordinates of the emergency location, airbag deployment and impact sensor information. An additional action could be to search if anyone in the nearby cars could offer first aid (who would be entitled to offer help in such cases is an issue that is out of the scope of this work). However, the location of a vehicle is private information and so the search for nearby cars has to be done in a privacy-preserving way. An approach like the NDP solution presented in this work could be used to identify a nearby car. A different problem, probably easier than NDP, would be to warn all nearby cars to slow down.

- Police or fire emergency. In case of a police or fire department emergency, a policeman or a fireguard who is not on duty and happens to be near the event location, might be able to provide critical services if he is informed about the emergency. At the same time, since the individual (policeman or fireguard) is not on duty, the exact location of a person is sensitive personal data and nobody has the right to know it. A solution like NDP could identify such an individual (with his consent). The individual would be contacted by his own agent only if he is the closest person and if he is close enough to be able to help in such an emergency.

4.2 Related Work

The Active Badge Location System [192] was the first indoor location system for contacting people in an office environment. The system raised issues on location privacy at work. Extensions of the initial system and follow-up projects like [193] offered enhanced features to the users for controlling the way their location data is accessed. However, all these systems assume a trusted server that manages the location data. A system that assumes a decentralized control of personal data is the Cricket Location-Support System [151]. Cricket describes an approach that offers an individual the option to learn his physical location within a building (that offer the Cricket service). The user can then decide to whom he discloses his location. This approach offers a better control over who obtains the location information of the individual. However, if the user wants to actively use his location information to perform some task, he has to disclose it. An approach like Cricket could be used to allow individuals to learn their location when they are within buildings where GPS cannot be used. All the above location systems are for indoor applications.

The privacy concerns for applications like NDP are even more critical since they apply to individuals who may be in their private time and not only at their office but at any location. Therefore, we present a privacy-preserving solution for NDP. The solution is based on secure multi-party computations (MPC's), i.e., computations that receive input from two or more parties and calculate the output without revealing the input of any participant. General models for MPC have been proposed in the seminal work of Yao [214] and in follow-up works. However, the general models are practically inefficient. More efficient approaches are being developed for specific applications, like for example [216, 19]. A first large-scale and practical application of multi-party computation took place in Denmark in January 2008 [21]. A centralized approach using four separate servers was used to implement an electronic double auction that enabled Danish farmers to trade contracts for sugar beet production on a nation-wide market. The NDP solution presented in this work is a decentralized, efficient privacy-preserving scheme for the NDP problem.

4.3 The NDP Problem

In this section we define the Nearest Doctor Problem (NDP). The main goal of NDP is to find the nearest doctor without violating the privacy of doctors. The personal data which is needed for the NDP computation are the exact locations of all doctors. An instance of the NDP problem consists of:

- **N doctors** $D_1, D_2, ..., D_N$.
- For $i = 1, 2, \ldots, N$, let L_i be the current location of doctor D_i. For instance, the location L_i may be the exact GPS location of the doctor, obtained from a portable GPS device.
- **The NDP lookup function:** In case of an emergency, (the agents of) all doctors perform a distributed privacy-preserving computation.
  - **Input:** The location L_{em} of an emergency.
  - **Output:** At the end of the computation, the doctor who is the nearest one to the location of the emergency becomes aware of this fact and can offer his services.

4.4 A Solution for NDP

We describe a distributed privacy-preserving computation for solving the NDP problem. An overview of the architecture of the solution is presented in Figure 4.1.

The communication between entities in our architecture is performed over secure sockets (SSL/TLS) with both server and client authentication enabled. At the heart of our approach is a cryptographic protocol for a secure distributed computation.

4.4.1 Assumptions

We make the following plausible assumptions:

- Every doctor has a personal data management agent with permanent access to the Internet.
- The current location of each doctor is stored in his personal agent.

4.4.2 Security Model and Privacy

We first define the security model and the kind of privacy that is achieved and then proceed with the description of the distributed computation in the next Section. We will show that the proposed protocol is safe in the Honest-But-Curious (HBC) model, i.e., the doctors are assumed to follow the protocol steps but also may try to extract additional information (see details in Section 2.9.1). The HBC model is commonly used in cryptographic protocols and is well suited for the NDP problem, since the participants are certified doctors. In Section 4.5.2 we go beyond the HBC model and examine how to handle some cases of malicious user behavior.

Regarding privacy, the question we will address is how to achieve privacy of type (b) (see details in Section 2.1.4), that is, how to identify the nearest doctor without pooling the location data, and in a way that reveals (almost) nothing else about the distributed computation.

4.4.3 The NDP Service

At the application level, the NDP solution is offered as a service through a dedicated node, called the NDP Service Gateway (NSG). When an individual is in an emergency, the following steps take place (Figure 4.2):

- The individual or some authority submits a request with the emergency incident to the NDP Service Gateway (NSG). The request contains the current location of the individual (for example, the exact geographic coordinates) and possibly additional information about his identity, his current condition etc. Note that in the NDP problem, the location of the emergency is not considered private.

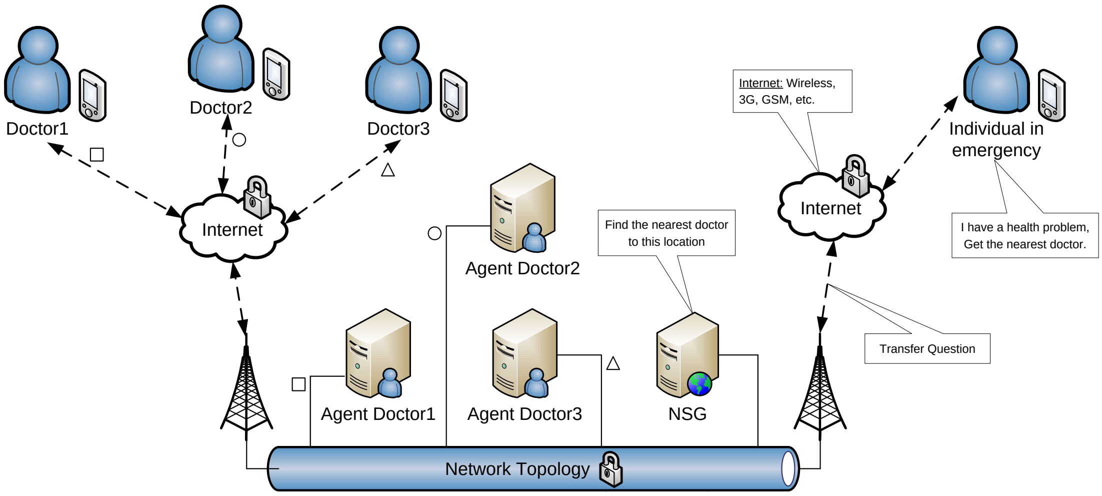


Figure 4.1: The architecture of NDP solution.

- The NSG is an access point that accepts the request and forwards it to an agent of the doctor's community. The agent which receives the request from the NSG takes the role of the root-node for the particular computation.
- The root-node coordinates a distributed computation that calculates the distance of the nearest doctor.
- At the end of the distributed computation, the agent of the doctor who is the nearest to the location of the emergency becomes aware of this fact and contacts the NSG to declare his readiness to offer help.

4.4.4 Outline of the Distributed Computation

We present a protocol for a secure distributed computation that solves the NDP problem. The protocol does not disclose the location of any doctor; only a small amount of aggregate or anonymous information is leaked. The computation consists of three main phases (Figure 4.2).

- In Phase 1, the closest interval containing at least one doctor is found.
- In Phase 2, the distance of the nearest doctor and an associated random ID are found.
- Finally, in Phase 3, the doctor who owns the random ID realizes that he is the nearest doctor and contacts the NSG to offer his help.

We provide a summary of each phase of the computation.

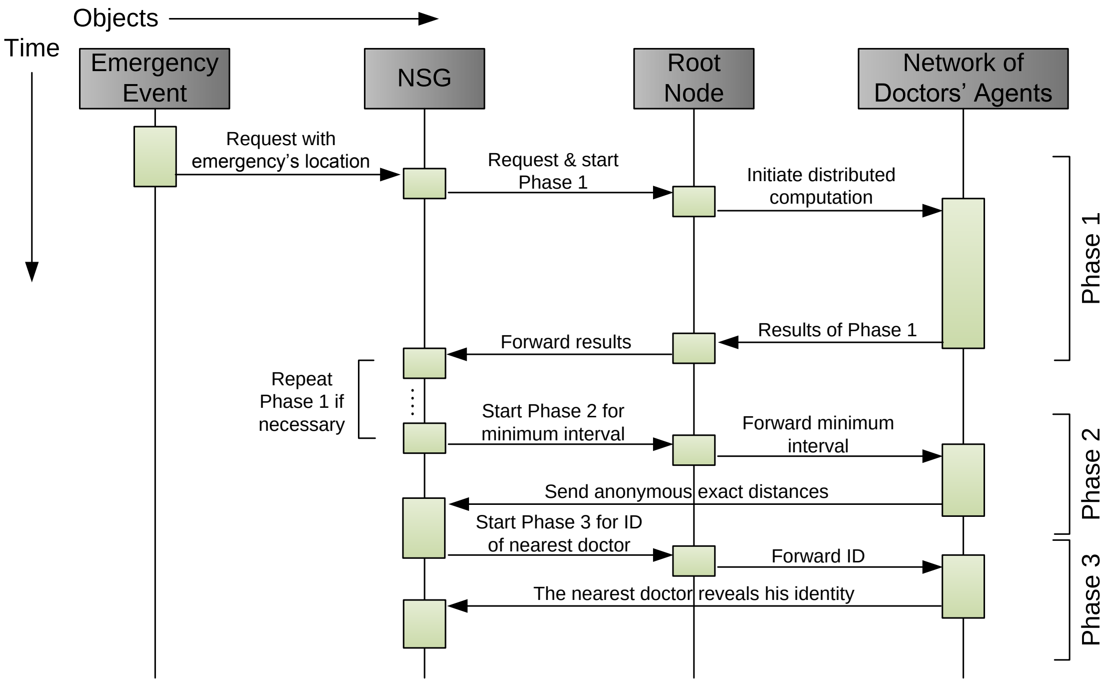

Objects
Time
Emergency Event
NSG
Root Node
Network of Doctors' Agents
Request with emergency's location
Request & start Phase 1
Initiate distributed computation
Phase 1
Results of Phase 1
Forward results
Repeat Phase 1 if necessary
Start Phase 2 for minimum interval
Forward minimum interval
Phase 2
Send anonymous exact distances
Start Phase 3 for ID of nearest doctor
Forward ID
The nearest doctor reveals his identity
Phase 3

Figure 4.2: Interaction diagram of a nearest doctor calculation.

- **Phase 1**
  - **Input:** The location L_{em} of the emergency.
  - **Output:** An interval I containing the minimum distances in which there is at least 1 doctor and at most K doctors, where K is a given constant (e.g. $K = 5$).
  - **Description:** The NSG chooses a node as the root-node for the particular computation and sends the location L_{em} of the emergency to it. Then, the root-node sends a broadcast message to the agent community that starts the distributed protocol and initiates Phase 1. The protocol is executed on a logical binary tree topology that contains all doctor agents (nodes) as leaves (Figure 4.3) and some of them also act as intermediate nodes (see Section 4.4.7 for details).

    Phase 1 may last for several rounds. In each round, the root-node collects the (intermediate) result of the computation as an encrypted message and sends this message to the NSG. The message is encrypted with the public key of the NSG, which has to be known to all nodes. Let $Count_D$ be the number of distances that belong to the interval of minimum distances. The NSG decrypts the result and obtains $Count_D$. If $Count_D > K$, then the computation is repeated in a new round, this

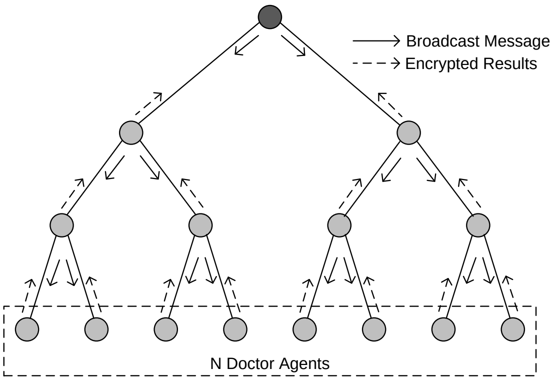


Figure 4.3: A binary tree topology.

time within the interval that has been found. This procedure continues until an interval that contains the closest $Count_D$ doctors, where $1 \leq Count_D < K$, is found. An example of this procedure where the appropriate interval is found in two rounds is shown in Figure 4.4.

In subsection 4.4.5 we describe in detail the protocol of Phase 1 and show that it ensures k-anonymity (see Section 2.1.3.1), where $k = N$ and N is the number of all nodes in the network, for the participants of the protocol.

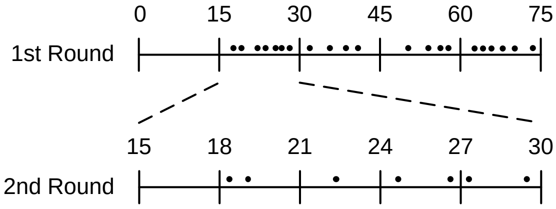


Figure 4.4: Example of Phase 1.

- **Phase 2**
  - **Input:** The interval I from Phase 1.
  - **Output:** The - anonymously collected - exact distances of the $Count_D$ nearest doctors and the associated random ID's.
  - **Description:** In this phase, the NSG sends the interval I of Phase 1 to the root-node which in turn sends a broadcast message to announce the interval I to the agents. Each agent whose distance is in the interval I responds by anonymously sending a message to the NSG. The message is encrypted with the public key of the NSG and contains

the exact distance of the agent and a random ID (a nonce, i.e., a number used once). For example, by an universally unique identifier (UUID) the probability of collisions is practically negligible. Moreover, even if a collision would occur, the NSG would detect it and repeat this phase. The anonymous transmission is achieved with onion routing [156] techniques. More information about onion routing is given in Section 4.4.6. The NSG collects all anonymous messages and finds the distance of the nearest doctor and the associated random ID. Since the messages are anonymously sent, k-anonymity is preserved in this step; assuming that no background information about the participants of the computation is available, the privacy of doctors is preserved.

- **Phase 3**
  - **Input:** The random ID associated with the distance of the nearest doctor.
  - **Output:** The owner of the random ID realizes that he is the nearest doctor D_n^* and can contact the NSG.
  - **Description:** The NSG sends a message containing the random ID of the distance of the nearest doctor to the root-node. The root-node broadcasts the ID to the agents' network. The doctor who generated the ID becomes aware of the fact that he is the nearest doctor and contacts directly the NSG.

4.4.5 A Privacy Preserving Protocol for Phase 1

We present a cryptographic protocol that finds the first interval of distances in which there is at least one doctor. The protocol uses a trick to encode distance values which has been applied in [216] for a secure dynamic programming protocol. Moreover, the cryptographic protocol uses the ElGamal public key cryptosystem [126] and its homomorphic encryption property which supports multiplicative homomorphism (see details in Section 2.5). Other homomorphic public key cryptosystems like the Paillier cryptosystem [141] can be used in our protocol in place of ElGamal.

4.4.5.1 The Protocol

The protocol accepts three parameters: the minimum distance $minDist$, the maximum distance $maxDist$ and the number n of subintervals. These parameters are used to partition the interval of distances $(minDist, maxDist)$ into a set of n consecutive subintervals. For simplicity, we use subintervals of equal size, but it is straightforward to adapt the approach for example to geometrically increasing subintervals. The outcome of the protocol is the first subinterval that contains (the distance of) at least one doctor. In the protocol, each subinterval is represented with a ciphertext and the whole set of subintervals is represented with the ordered list (or tuple) of the corresponding ciphertexts. Overall, each message has n encrypted numbers, as many as the subintervals into which the initial interval is partitioned. Such a message containing the ordered list of n ciphertexts passes through each agent.

Each agent prepares its own ordered list of ciphertexts as follows: For doctor D_i, where $i = 1, 2, \ldots, N$, let $\ell_i \in 1, 2, \ldots, n$ be the number of the subinterval that contains the distance of the doctor. Then, the ciphertext for the ℓ_i first subintervals are encryptions of the number "1" and the $\ell_i + 1$ subinterval is encryption of a number z, where $z > 1$ is a fixed value known to all agents. For example z might be $z = 2$. For the rest of the $n - (\ell_i + 1)$ subintervals the ciphertexts are encryptions of uniformly chosen random powers of z. An example of a local message is shown in Figure 4.5.

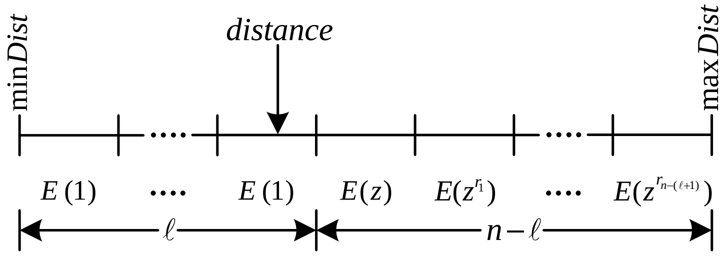


Figure 4.5: The local message of a doctor D_i.

When the agent receives the accumulated message, it calculates the new accumulated message as the product of the respective ciphertext of the local message and the accumulated message. The outcome, i.e., the new accumulated message, is then forwarded to the next node or nodes.

The distributed computation is performed on a logical binary tree topology in which the leaves of the tree are the N doctors' agents. The depth of the tree is $\lceil \log_2 N \rceil$ and so the accumulated outcome is computed with a reduce operation that requires $\lceil \log_2 N \rceil + 1$ parallel computation steps.

The general form of the final accumulated message is shown in Figure 4.6. Let L be the index of the last ciphertext that is an encryption of the number "1". Then, the value of L indicates that the first $L - 1$ subintervals are empty (no doctor is located at a distance within these intervals) and subinterval L is the first non-empty subinterval. The exponent k of the number in the $(L + 1)$-th ciphertext reveals the number of doctors in this subinterval. The ciphertexts of higher subintervals are encryptions of some random powers of z and are ignored.

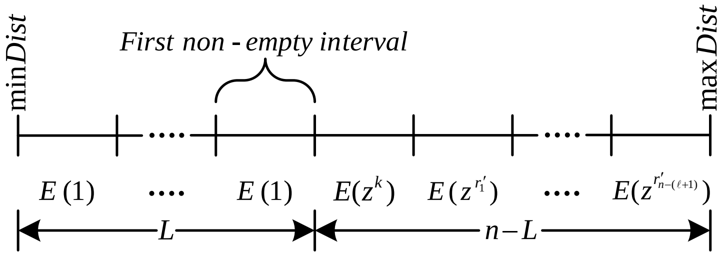


Figure 4.6: The final accumulated message.

The NSG decrypts the final-message and obtains the first non-empty interval and the number of doctors in it.

4.4.6 Onion Routing

In Phase 2 of the distributed computation we use onion routing [156], a popular technique for anonymous communication over a network. A simplified description of onion routing is: A node that wants to send a message to another node does not send the message directly to its destination. Instead, the sender chooses a random path that passes through intermediate nodes and terminates at the destination node. Moreover, the sender encrypts the message repeatedly with the keys of the intermediate nodes. So the message is packed with multiple layers of encryption and looks like an "onion". Each intermediate node that receives the message, takes away a layer of encryption to reveal routing instructions, and sends the message to the next router where this process is repeated. This prevents intermediary nodes from knowing the origin, the destination and the contents of the message.

The advantage of onion routing is that it is not necessary to trust each cooperating onion router (intermediate node); if one or more, but not all, routers are compromised, the anonymity of the communication is preserved. This is because each router in an onion routing network accepts messages, re-encrypts them and then forwards them to another onion router. An attacker with the ability to monitor every onion router in a network might be able to trace the path of a message through the network, but an attacker with more limited capabilities will have difficulty even if he controls one or more onion routers on the message's path.

In order to accomplish the anonymity for sending a message like we described in the Phase 2 (subsection 4.4.4), one option is to use the Tor network [51], a widely used, general purpose platform for onion routing. Another option, is to implement an onion-like or some other anonymity providing mechanism within the agent community, where each agent forwards its messages through other random agents of the agents community. In the current implementation of the NDP solution we applied the latter approach (see Section 4.6).

4.4.7 Network Topology

A critical component of the NDP solution is the logical network topology of the agents. The network topology must be scalable, reliable, and should support privacy-preserving communications and computations of the agents. We address the above requirements by employing networking technologies from the field of Peer-to-Peer (P2P) networks. In particular, we apply techniques that have been developed in [173] and are based on the well known Chord [176] topology for P2P networks.

The network topology has the following features: The agents are organized into a logical ring that serves as the backbone of the topology. Each node in the ring, knows its predecessor and its successor. Actually, for increased tolerance to node changes/failures, each node keeps links to a set of successors. In addition, each node maintains a set of links, called fingers, to nodes at geometrically increasing distances in the ring. These links allow the network to behave as a logical binary tree topology. An example of the embedding of a binary tree topology into the logical ring of the Chord-like architecture is shown in Figure 4.7. Similar approaches to embed a virtual tree topology on a chord P2P network are used for example in [102].

The proposed network architecture provides a fully decentralized and scalable network topology for the doctors' agents. The links of existing nodes and the establishment of the links of new nodes are accomplished with stabilization procedures that are similar to the typical stabilization procedures of Chord P2P networks.

4.5 Preserving Privacy

In this Section, we examine the security properties of the proposed NDP protocol. We first consider the Honest-But-Curious (HBC) model and show that the protocol preserves the location privacy of the doctors in this model; only a small amount of aggregate or anonymous information is leaked. Then, we examine scenarios with malicious users and discuss how these can be handled. More details about the meaning of security models you can find in Section 2.9.1. To prove that the protocol preserves privacy we show that it satisfies the criterion of k-anonymity (see Section 2.1.3.1).

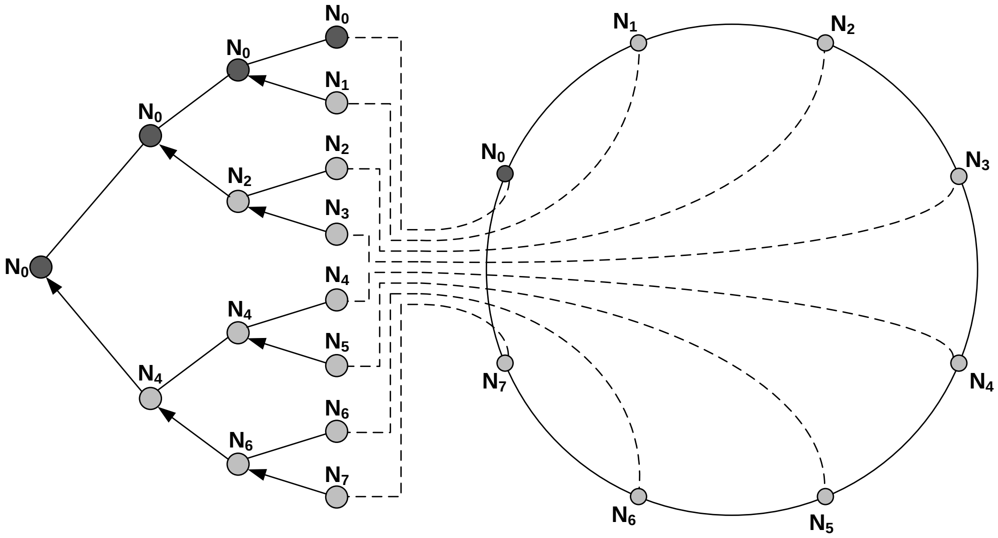


Figure 4.7: Embedding of a binary tree into the logical ring.

4.5.1 Privacy in the HBC Model

We first note that the doctors do not use their location in the protocol but only their distance to the location of the emergency. Then, the security of the ElGamal cryptosystem and its homomorphic property ensure that the distances cannot be associated with any particular doctor. Finally, the security of onion routing protects the anonymity of the nearest doctors that disclose their distances in Phase 2. Below, we discuss in detail the preservation of privacy in each Phase of the distributed computation.

- **Phase 1**
  - Each doctor uses his private location and the location of the emergency to calculate his distance to the emergency event. Consequently, only the distance is used in the distributed computation, not the private location itself. Moreover, the doctor does not use the exact value of his distance but only the subinterval in which this distance belongs.
  - Each agent cannot obtain information from the accumulated message that it receives, because the contents of the message are encrypted with the public key of the NSG using ElGamal encryption.
  - All ciphertexts of the accumulated message are altered by each node, even the ones that are multiplied with an encryption of the number “1”.
  - At the end of each round, the final accumulated message reveals the number of doctors in the first interval that contains at least one doctor.

No exact location and not even any exact distance is disclosed. Moreover, since no individual doctor can be associated with the doctors in this first interval, Phase 1 preserves k-anonymity, where k is equal to N, the total number of agents in the network. Thus, the aggregate information disclosed in Phase 1 does not violate the location privacy of the doctors.

  - Note, that each round of Phase 1 is a secure multi-party computation that reveals to the NSG the first non-empty interval and the number of doctors in it. This is an immediate consequence of the security of the ElGamal cryptosystem. Moreover, the first non-empty interval of each round is announced to all doctors (either in Phase 1 or in Phase 2). In conclusion, the outcomes of each round of Phase 1 leaks a small amount of aggregate information about the distances of the participating doctors (Table 4.1).

- **Phase 2**

  - We make the plausible assumption that Onion Routing works reliably. More details on the security of Onion Routing can be found in [51]. Then, the security features of Onion Routing ensure that the exact distances of the doctors in the first interval are anonymously sent to the NSG. Hence, k-anonymity for k = N is preserved in this phase too.

  - We consider the anonymous disclosure of exact distances an acceptable tradeoff between efficiency and privacy protection. However, there is the possibility that even an honest NSG may attempt to combine background knowledge with the specific distances to try to identify doctors who live at such a distance from the emergency location. This leakage could be mitigated or avoided if we used less accurate distances in Phase 2 or a more complex protocol among the closest $Count_D$ doctors' agents (see Section 4.5.2.2).

- **Phase 3**

  - In this phase, the random ID associated with the closest distance is announced to the network. The agent that recognizes that it is the owner of this ID can now directly contact the NSG and reveal its identity. The ID does not leak information to any other node.

A summary of the critical data items of our solution and to which of the participating entities these items are disclosed, is given in Table 4.1.

			Doctors	
Participants Data Items		NSG	All	$Count_D$-closest
Doctors' Location		✗	✗	✗
Emergency Location		✓	✓	✓
Phase 1 (each round)	Closest Interval	✓	✓	✓
	Number of $Count_D$	✓	✗	✗
Phase 2	Exact $Count_D$ Distances	✓	✗	✗
Phase 3	Nearest Doctor D_n^*	✓	✗	✗

Table 4.1: The scope (columns) of the critical data items (rows).

4.5.2 Malicious Users

In this section, we examine scenarios with malicious users and discuss how and to what extend the NDP protocol can handle them. The classic results [80, 79] on secure multi-party computation show how to convert a secure multi-party computation of the semi-honest model into a computation in the malicious model. However, the conversion introduces a significant computational overhead since it requires each participant to validate every message by supplying an appropriate zero-knowledge proof that the message is consistent with the protocol specification. In general, a zero-knowledge proof [153] is an interactive method for one party to prove to another that a statement is true, without revealing anything other than the veracity of the statement. This overhead caused by the zero-knowledge proofs raises important practical issues, especially for distributed computations with a large number of nodes as in the case of the NDP problem.

Instead of using a heavy general conversion technique to handle malicious users, one may consider deriving some proprietary approach for the NDP protocol. In the context of homomorphic encryption there are examples of practical problem-specific solutions that can tolerate malicious behavior. For example, in [42, 47] zero-knowledge based proofs are used within a secure multi-party computation with homomorphic encryption to handle malicious users. However, the above approaches require each node to publish the encryptions of its values to all other nodes of the doctors' network and to execute all the products of ciphertexts. In our case, this is impractical due to the large number of agents.

A straightforward approach could be to handle the complexity of the computationally demanding protocol for malicious nodes by executing the protocol within smaller user groups of NDP nodes (Figure 4.8). Such a hybrid solution could be used to assure the tolerance of the NDP protocol against a small number of malicious users. We will describe such an approach in Section 4.5.2.1. Moreover, in Section 4.5.2.2 we present an improvement of Phase 2 to avoid the anonymous disclosure of the smallest exact distances. However, we first examine three specific examples of malicious behavior and then we consider the hybrid approach as a more general solution to handle malicious users in the NDP protocol.

We shortly discuss the case of a malicious NSG and then proceed to scenarios with malicious nodes. A malicious NSG could collude with any malicious agents to uncover data of neighboring (in a particular computation) agents. Fortunately, such a malicious NSG behavior can be effectively handled by employing a threshold decryption model [73, 40] for the ElGamal Cryptosystem. Using threshold decryption is a classic defense in e-voting systems with purpose to protect their voting process against malicious coordinators, and can be applied within our solution too. In such a case, the NSG instead of merely decrypting the encrypted result, uses n parties (the NSG could be one of them) with their secret keys, so that at least t parties, where $t \leq n$, are required to decrypt the final result. We will not further address malicious NSG behavior in this work. Instead, we focus on the community of the doctor agents and examine the following cases of malicious agent behavior.

- **Case 1:** A user (doctor agent) maliciously or unintentionally reports an erroneous close distance to the emergency incident. As a consequence, this doctor might be wrongly chosen as the nearest doctor and the actual nearest doctor might not be informed to offer his help.

  A possible solution for this case is to modify the NDP computation of Phase 1 to find a larger number of nearest doctors. For example, the protocol could search for the $Count_D$ nearest doctors, where m is a fixed number and $m \leq Count_D < K$. That is, if the current closest interval contains less than m doctors, then Phase 1 will continue by extending the interval to include larger distances until $Count_D$ is within the specified range. Note that a malicious node would presumably not show up in Phase 3 of the NDP protocol where it has to reveal its identity. Thus, when there are at most $m - 1$ malicious users, the NDP protocol will succeed in finding the actual nearest doctor.

- **Case 2:** A malicious user incorrectly modifies or replaces ciphertexts in the accumulated message of distance intervals of Phase 1. In other words, the user maliciously modifies the accumulated data from the inputs of the preceding users. Note that none of the doctors' agents can see the contents of the accumulated message but it can change the accumulated message or replace any of its items when the accumulated message is in the possession of the agent. Consequently, the NSG may obtain at the end of the round a wrong number of doctors in the nearest distance interval or even a wrong nearest distance interval.

The impact of the malicious modification of the ciphertexts depends on many parameters, including the location of the malicious user in the virtual tree of the distributed computation and the difference between the correct and the falsified contents of the ciphertext. The most likely consequence is that the NSG may wrongly decide to proceed or not to Phase 2. In Phase 2 it will become evident that something is wrong if the number of doctors that anonymously reveal their exact distances or the distances themselves do not satisfy the results of Phase 1.

- **Case 3:** A node of the virtual tree topology does not correctly execute the communication operations of the protocol. By definition, a malicious node may not follow the communication steps of the protocol. For example, the node may receive the accumulated message during the execution of protocol and then not forward this message to next nodes of tree topology. As a result, the distributed computation will not run correctly.

  This type of misbehavior can be handled at the network topology level. A fault-tolerant network topology can detect if the communication is delayed at some nodes and handle these cases as node failures. In the current prototype, we assume that all nodes of the network topology are reliable.

4.5.2.1 A Hybrid Approach

In the hybrid approach the agents are organized into groups, where each group consists of R agents. As shown in Figure 4.8, each group of agents corresponds to a vertex of the virtual tree topology. The value of R determines the level of tolerance that we wish to achieve against malicious users. We present the hybrid approach in order to give some hints about how to address malicious user behavior in the context of NDP; we do not provide the full details of such a solution.

Within each group of nodes, we can apply a zero-knowledge proof that an encrypted message lies in a given set of messages against malicious behavior. This type of zero-knowledge proofs have been successfully applied in secure e-voting systems [43, 26], and it is straightforward to apply them within our protocol. With such an approach, the R nodes of each group would exchange the encrypted representations of their distance and each agent would verify that the encrypted values of subintervals lie in the public set $S = \{1, z\}$. No information about the actual value of the encrypted messages would be revealed. We note that we may have to adapt the form of the local encrypted messages in order to simplify the zero-knowledge proofs within the groups. For example, an appropriate form for the local message could be $(1, \ldots, 1, z, 1, \ldots, 1)$. Such an approach will cause

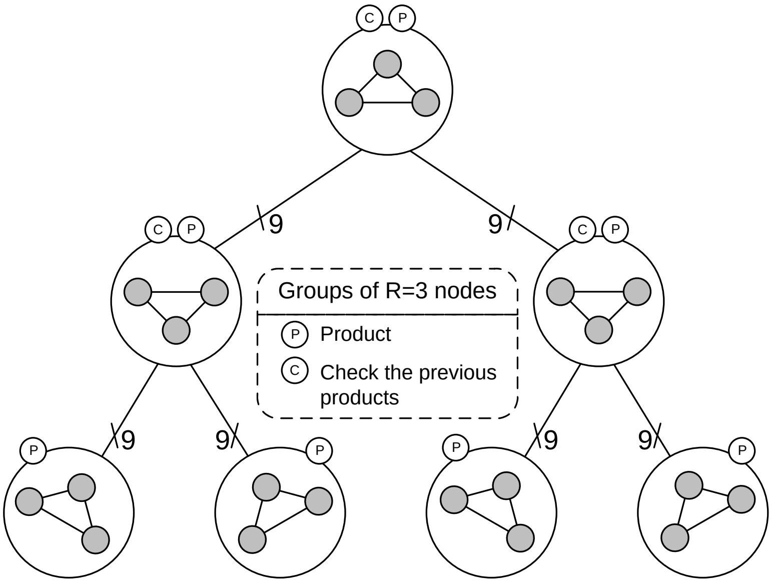


Figure 4.8: The execution of Phase 1 on groups of $R = 3$ nodes.

higher leakage of aggregate data to the NSG but allows us to simplify the zero-knowledge proof. Since each group of nodes is a clique graph with $R(R-1)/2$ edges and each pair has to exchange a constant number of messages, the overall verification will require $O(R^2)$ communications between the nodes.

Next, each node of the group independently calculates the accumulated message of the group-node and sends its output to each of the nodes of the parent vertex in the virtual topology. This step requires R^2 messages. Finally, each of the agents of the parent vertex independently verifies the correctness of the received messages by checking if the received partial results from the R nodes of the child vertex are identical. Overall, each node of a group will receive messages from $3R-1$ nodes and send messages to $2R-1$ nodes.

If the results that a node receives from the nodes of the child vertex are not identical, this indicates the existence of malicious behavior. The hybrid scheme will successfully detect such malicious behavior if there are at most $R-1$ malicious nodes. Actually, even a much larger number of malicious nodes will be detected as long as at most $R-1$ malicious nodes belong to the same group. Moreover, if at least $R/2+1$ nodes of a child vertex give the same results, then the nodes of the parent vertex may resolve this issue with a simple majority criterion and proceed with the calculations of the NDP protocol.

4.5.2.2 An Improvement for Phase 2

As noted earlier, there is a leakage of data to the NSG in Phase 2, namely the disclosure of the exact distances of the $Count_D$ nearest doctors. Even though the distances are reported anonymously, if the NSG combines them with background knowledge for example on the addresses of the doctors, it may potentially identify some of the doctors. We describe a sketch of a solution to the above problem. The solution that we propose is based on a cryptographic protocol – ciphertext comparison [147] – that can compare two ciphertexts without revealing the two encrypted messages. The protocol is valid both in the Honest-But-Curious (HBC) and the Malicious model. The main idea is that a (small) set of independent parties (instead of a single NSG) share the responsibility to coordinate the comparison process and that the correctness of the comparison can be publicly verified.

In the improved Phase 2, the only modification would be that the doctors' agents would have to encrypt their exact distances in an appropriate form given in [147]. This encryption is performed with the common public key of the independent parties. In our case where we have $Count_D$ encrypted distances, in total $Count_D - 1$ secure comparisons are required to find the encryption of the smallest distance or $Count_D \cdot \log(Count_D)$ secure comparisons to sort the encrypted distances. Finally, the random ID of the doctor corresponding to the encrypted smallest distance can then be used in Phase 3. The above comparison process remains efficient for a small number of $Count_D$ distances.

4.6 Experimental Results

To confirm the feasibility of the NDP solution and examine its practical efficiency we developed and evaluated an NDP prototype. The application is developed in Java and for the cryptographic primitives the Bouncycastle [91] library is used. In the prototype, the current location of each doctor is stored in his personal data management agent, which is an adaptation of the personal agents of the Polis platform (see Chapter 3). The personal agents use production-ready cryptographic libraries and employ 1024 bits RSA X.509 certificates. The communication between agents is performed over secure sockets (SSL/TLS) with both client and server authentication (see details in Sections 2.7 and 2.8).

In the next subsections, we first describe a toy-case example with four doctor agents to illustrate the process of execution of the NDP protocol. Then, we present a set of larger scale experiments with up to 300 doctor agents. In both experiments, we implemented an anonymity mechanism for Phase 2. For simplicity, we used a simple Crowds-like [157] approach where each message is initially passed to a random agent which in turn randomly decides either to forward the message to the receiver or to another randomly chosen agent. More precisely, each doctor agent whose distance belongs to the closest interval identified in Phase 1, prepares a message with its exact distance and encrypts it with the public key of the NSG. The agent then sends this message to another, random agent of the agent community. Let's call the recipient, an intermediate agent. Each intermediate agent uses the multiplicative homomorphic property to multiply the encrypted message with the number "1". This changes the ciphertext but leaves the encrypted plaintext unchanged. Then, with probability "1/3" the intermediate agent forwards the message to the NSG and with probability "2/3" to another, random, intermediate agent. Since we assume Honest-But-Curious nodes this Crowds-like algorithm seems to protect the anonymity of the original sender. However, we do not claim strong security properties of the above algorithm; it simply serves the experimental evaluation of the NDP solution.

4.6.1 A Toy-Case Experiment

In this experiment, the NSG submits an emergency event to a community of four doctor agents. The NDP protocol is then used to identify the doctor who is the closest to the emergency. The doctor agents are assumed to be Honest-But-Curious (HBC) and there are no network or agent failures. The agent community is organized into a simple ring topology.

The experiment covers a hypothetic square area of 10000 km^2. The locations of the doctors and the emergency are chosen independently and uniformly at random in the above area. Each agent chooses its random location and the NSG chooses a random location for the emergency event. The NDP solution tries to find the nearest doctor within a distance of at most $75km$ from the emergency location. The values of the internal parameters of the NDP solution are $K = 2$ and $z = 2$, where K is the upper bound of the closest doctors of Phase 1 and z is a fixed number used in the encryptions of the intervals.

The NSG chooses an agent node, in this case "$Agent\_1$", as the root-node, and forwards the location of the emergency event to this node. The coordinates of the location of the emergency of the experiment are $L_{em} = [41.140110, 24.913660]$ and the exact distances of the 4 agents from this emergency location are:

$$\begin{array}{l}
Agent\_1 \Rightarrow 17.544817\,km \\
Agent\_2 \Rightarrow 53.157742\,km \\
Agent\_3 \Rightarrow 25.797003\,km \\
Agent\_4 \Rightarrow 66.221868\,km
\end{array}$$

The cryptographic protocol starts with Phase 1. In the first round the interval of distances (in km) $[0, 75]$ is partitioned into 5 equal intervals. As a result, the encrypted representation of agents' distance (in km) is:

	$0-15$	$15-30$	$30-45$	$45-60$	$60-75$
Agent_1	$E(1)$	$E(1)$	$E(2)$	$E(2^5)$	$E(2^3)$
Agent_2	$E(1)$	$E(1)$	$E(1)$	$E(1)$	$E(2)$
Agent_3	$E(1)$	$E(1)$	$E(2)$	$E(2^6)$	$E(2^9)$
Agent_4	$E(1)$	$E(1)$	$E(1)$	$E(1)$	$E(1)$

The final accumulated message in round 1 is shown below:

	$0-15$	$15-30$	$30-45$	$45-60$	$60-75$
result	$E(1)$	$E(1)$	$E(2^2)$	$E(2^{11})$	$E(2^{13})$

The decryption of the ciphertexts of the final message reveals that the first non-empty interval is $[15, 30)$ and that this interval contains two doctors. Since the number of doctors in the first interval is equal or less than K, Phase 1 terminates. In Phase 2, the root-node broadcasts the first interval to all nodes. Every node with a distance in this interval, anonymously sends its exact distance together with a random ID nonce (number used once) to the NSG.

The NSG receives the following two exact distances and the associated random ID numbers:

$$[Dist = 17.544817, ID = 56770656]$$
$$[Dist = 25.797003, ID = 45413392]$$

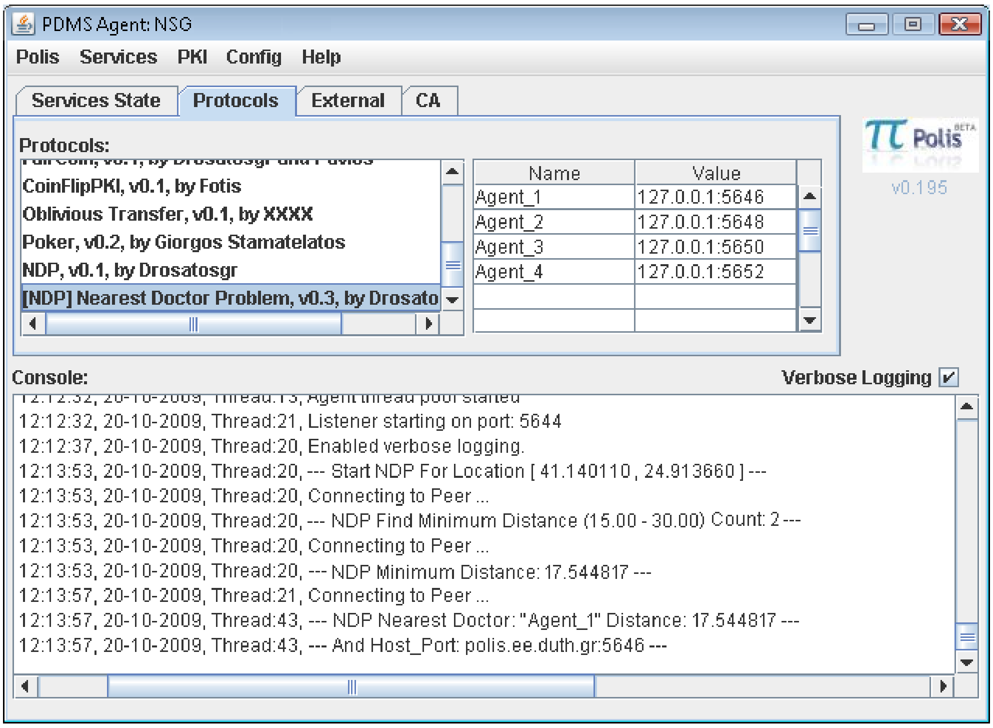
The NSG finds that the minimum distance is 17.544817 km. In Phase 3, the NSG sends the random ID nonce that is associated with the minimum distance to the root-node, which in turn broadcasts this ID to the doctors' network.

$$ID : 56770656$$

Finally, "*Agent_1*" realizes that it is the nearest doctor and directly contacts the NSG to offer its services. A snapshot of the application GUI during the experiment is shown in Figure 4.9.

4.6.2 The Large-Scale Experiment

We also conducted a set of experiments with a gradually increasing number of agents from 50 to up to 300 agents. Also in this case, we assume that the doctor agents are Honest-But-Curious (HBC) and there are no network or agent failures. The experiments were executed on a 100 Mbps network with 30 workstations, each with a CPU Intel Core 2 Quad Q8300 processor at 2.5 GHz and 2 GB



Protocols:

Full Coin, v0.1, by Drosatosgr and Favios
CoinFlipPKI, v0.1, by Fotis
Oblivious Transfer, v0.1, by XXXX
Poker, v0.2, by Giorgos Stamatelatos
NDP, v0.1, by Drosatosgr
[NDP] Nearest Doctor Problem, v0.3, by Drosato

Name	Value
Agent_1	127.0.0.1:5646
Agent_2	127.0.0.1:5648
Agent_3	127.0.0.1:5650
Agent_4	127.0.0.1:5652

Console:

```
12:12:32, 20-10-2009, Thread:13, Agent thread pool started
12:12:32, 20-10-2009, Thread:21, Listener starting on port: 5644
12:12:37, 20-10-2009, Thread:20, Enabled verbose logging.
12:13:53, 20-10-2009, Thread:20, --- Start NDP For Location [ 41.140110 , 24.913660 ] ---
12:13:53, 20-10-2009, Thread:20, Connecting to Peer ...
12:13:53, 20-10-2009, Thread:20, --- NDP Find Minimum Distance (15.00 - 30.00) Count: 2 ---
12:13:53, 20-10-2009, Thread:20, Connecting to Peer ...
12:13:53, 20-10-2009, Thread:20, --- NDP Minimum Distance: 17.544817 ---
12:13:57, 20-10-2009, Thread:21, Connecting to Peer ...
12:13:57, 20-10-2009, Thread:43, --- NDP Nearest Doctor: "Agent_1" Distance: 17.544817 ---
12:13:57, 20-10-2009, Thread:43, --- And Host_Port: polis.ee.duth.gr:5646 ---
```

Figure 4.9: A snapshot of the NSG (NDP Service Gateway).

RAM. The agents were distributed evenly on the available workstations so that each workstation ran at most 10 agents. Every measurement was averaged on 10 independent executions of the experiment, each with different random values for the location of emergency and the locations of doctors.

The network topology is a logical binary tree with root the root-node of the particular NDP computation. At this stage of the prototype development, the tree topology is build by the NSG or by a Directory Service (DS) where every active agent registers itself. Consequently, we do not rely on the Chord-like network to build the tree topology as a full-blown implementation of the NDP solution would do.

The measurements concern the execution time of each phase of the NDP protocol and the total run time. We present the results and derive conclusions about the efficiency and the scalability of the solution. In Figure 4.10, the execution times of a single run of Phase 1 are shown. Recall that Phase 1 may be executed more than once, depending on how many doctors are found in the closest non-empty interval. In this figure we focus on the time for a single round of Phase 1. The total run-time for all repetitions of Phase 1 in an NDP computation is taken into account in Figure 4.12 that presents running times of the complete NDP computation.

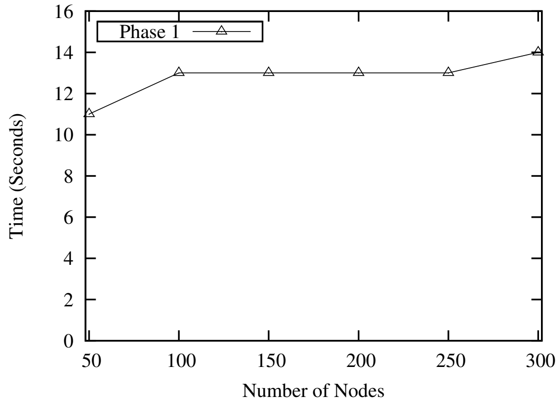


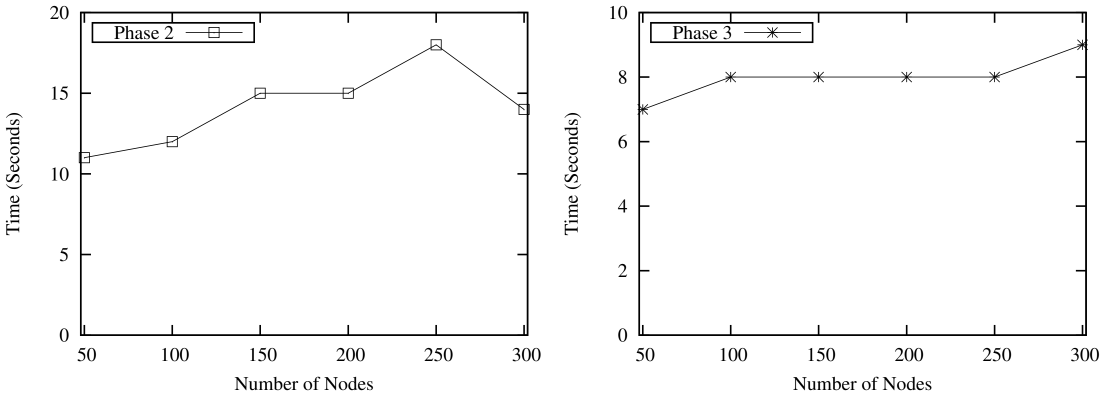
Figure 4.10: Execution times of Phase 1 with respect to the number of agents.

From Figure 4.10 we conclude that the execution time of Phase 1 depends almost linearly on the depth of tree topology of the distributed computation. For example, for an NDP computation with 100 agents the depth of the underlying tree topology is $\lceil \log_2 100 \rceil = 7$, whereas for computations with 150, 200 or 250 agents the depth is $\lceil \log_2 150 \rceil = \lceil \log_2 200 \rceil = \lceil \log_2 250 \rceil = 8$. The results show that Phase 1 scales well as the number of agents increases.

Figures 4.11a and 4.11b show the execution times of Phases 2 and 3 respectively. For Phase 2, the results show that its execution time does not seem to depend on the number of agents. This is rather expected, since the workload of Phase 2 mainly depends on the number of nearest doctors accrued in Phase 1 and how fast these doctors can anonymously send their exact distances to the NSG. The results of Phase 3 show that the execution time of this phase is dominated by a broadcast operation, which in turn depends on the depth of the virtual tree topology.

Finally, in Figure 4.12 the overall results are presented. In particular, the execution time of Phase 1 (of a single execution of the phase), Phase 2, Phase 3, the sum of the previous times and the total time of the whole NDP computation are presented. Note that the total time may differ from the sum of the three phases. This happens in cases where Phase 1 has to be executed more than once within the same computation.

The general conclusions of this large scale experiment is that the NDP protocol can be successfully performed on a virtual tree topology and that, as expected, the execution times seem to increase linearly with the depth of tree topology and thus logarithmically with the number of nodes. Based on the design of the NDP protocol and the above experimental results, we are confident that the NDP solution can scale well to handle much larger communities with thousands of nodes.



(a) Execution times of Phase 2.

(b) Execution times of Phase 3.

Figure 4.11: Execution times of Phase 2 and Phase 3 with respect to the number of agents.

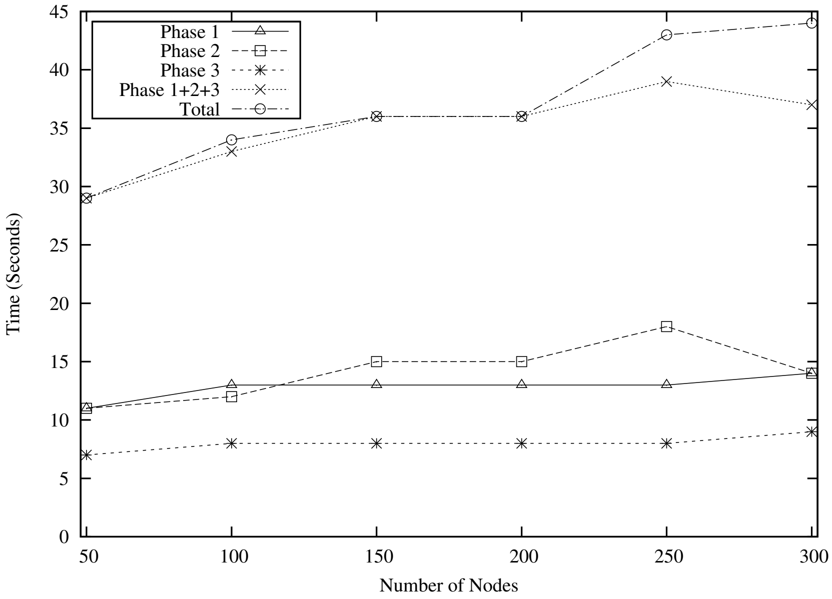


Figure 4.12: Execution times of Phase 1, Phase 2, Phase 3, sum of 3 Phases and total time with respect to the number of agents.

4.7 Discussion

The development, analysis and evaluation of the NDP prototype confirmed the feasibility and the effectiveness of the NDP solution. However, there are still important open questions, technical and non-technical. A critical technical issue is the robustness of the network topology against failures. In a community with a large number of nodes, the short-term or long-term failure of individual nodes or network links will be a common event, and the P2P-based network topology should be able to handle these failures. A lookup of the nearest doctor may take too long if the size of the doctors' community is large and the network is currently recovering from some failure. Furthermore, a temporary node or link failure, which becomes more likely as the network size increases, can disrupt the whole distributed computation. We believe that a fully developed network platform based on Chord-like peer to peer networking techniques can address these technical issues of the network topology.

At the service level, the NDP solution could be improved with a more appropriate distance metric. In the prototype, we used the great-circle distance. However, for inhabited areas and not only, a much better distance metric could be the time that each doctor needs to reach to the emergency location. For example, a navigation software tool running at the agents' side could use the GPS location, updated maps and possibly the current traffic conditions to calculate an estimate of the time that the doctor will need to arrive at the location of the emergency. Such a distance metric would be much more effective for the NDP problem.

Another important point is that, even though wireless communication options like 3G, Wi-Fi and Satellite communications are now widely available, there are still technical and economic issues. For instance, a personal mobile device will have to regularly update the doctor's location at his personal data management agent. This may cause the energy consumption of the mobile device and the cost for the wireless data transfer to become prohibitive. However, the current momentum of mobile device technology and telecommunications services predispose that these issues will soon be overcome.

Finally, we would like to emphasize an issue, which may not be technical, but is of equal importance for the acceptance of a solution like NDP by real doctor communities or other communities that could use such a system. Clearly, many individuals may not be eager to adopt a technology like the NDP solution in their everyday life. However, such difficulties commonly exist for every new technology. We believe that the doctors' community should not feel threatened in any way by an application like the NDP solution, because:

1. The privacy of each doctor's location is preserved and under the absolute control of the doctor.

2. The solution is simple and cheap enough to be feasible even with current information and communication technologies (ICT).

3. The benefits of an application like NDP for the public well-being are practically immeasurable.

4.8 Conclusion

In this work, we proposed the use of the current location of each doctor for supporting services for the public well-being. For this reason, we define the Nearest Doctor Problem (NDP) and make a first attempt to present an efficient privacy-preserving protocol for solving it. The proposed scheme utilizes the doctors' personal data (location) while ensuring their privacy. The protection of privacy is achieved by using cryptographic techniques and performing a distributed computation within a network of personal agents. Furthermore, we studied how our proposed approach can be applied under malicious users and suggested possible countermeasures. For the feasibility of our approach, we developed a prototype implementation and confirmed the viability and the efficiency of the proposed solution by conducting a set of experiments with up to several hundred doctor agents.

In our view, the NDP solution for offering help in case of an emergency should be considered a complement to existing emergency handling services. The NDP solution would probably make a difference only in some cases of emergencies. However, even a small number of successful applications of NDP, justifies, at least in our view, the approach.

A future direction for the improvement of our solution could be to give a more precise security and privacy analysis of our protocol. Even though only a small amount of aggregate or anonymous information is leaked by the intermediate results of the computation, a formal estimation of this leakage would be an important step for this work. Even more interesting would be to obtain a solution with no side-leakage at all; essentially a secure multi-party computation for the NDP problem. Finally, an interesting extension of the NDP problem would be to require the location of the emergency to be private, too.

Chapter 5

Privacy-Preserving Management and Statistical Analysis of Ubiquitous Health Monitoring Data

5.1 Introduction

The requirement to provide health care to special groups of people who have the need of continuous health monitoring is an integral part of today's society. Moreover, the number of people who need such health monitoring services is increasing. An important reason for this is the aging of the populations, which constitutes a social and economical challenge for the whole world. Related researches which have been carried out both in the European Union [217] and the United States [93] indicate that the number of people over the age of 65 is increasing; a similar increase is expected to take place throughout the developed world. Many elderly people suffer from chronic diseases that require health care and frequent visits to hospitals. For people of this category, it is important to continuously monitor the state of their health. Effective monitoring of the health state can improve the quality of the patients' life or even save their life, while simultaneously reducing the cost of health care.

The rapid development of the wearable sensors technology led to the appearance and the implementation of prototype Ubiquitous Health Monitoring Systems (UHMS's) [140, 60, 211]. Moreover, there is a plethora of researches in the area of ambient assistive living services [190, 208, 11, 84] and controlled access to ubiquitous hospital information [115]. The objective of a UHMS is to provide continuous health monitoring, both at home and outside of it. People need to have their health condition under control not only when at home, but wherever they are. One of the main features of a UHMS is to automatically generate alerts to notify the family or the patient's doctor about a possible health emergency so that they should rush to their help to him. Examples of the data used for the detection of a possible health incident, as they are reported in [30], are: heart rate, blood pressure, galvanic skin response, skin temperature, heat flux, subject motion, speed and the covered distance.

An important issue of UHMDS and health-related applications in general, is that health data are sensitive personal data of patients. Privacy-related legislation like the European Data Protection Directive [67] and the HIPAA (Health Insurance Portability and Accountability Act) [1] explicitly define the rules for protecting the privacy of patients. The so far general architecture of a UHMS requires that all personal medical data (such as those reported above) which are produced by the patients' wearable sensors are collected and stored in a central service, specifically at the Health Monitoring Center (HMC) [140, 60]. The HMC is responsible not only for the collection and storage, but also for the control of these critical personal data. However, this technique runs significant risks for the security of the actual data, for the privacy of the monitored people, and, moreover, has an enormous computational and storage cost for the HMC.

At the same time, the use of statistical methods is an integral part of medical research. A medical statistic may comprise a wide variety of data types, the most common of which are based on vital records (birth, death, marriage), morbidity (incidence of disease in a population) and mortality (the number of people who die of a certain disease in relation with the total number of people). Additional personal data items may needed for other well-known statistical computations like the demographic distribution of a disease based on geographic, ethnic, and gender criteria, the socioeconomic status and education of health care professionals, and the costs of health care services.

In this work, we deal with the privacy-enhanced management of ubiquitous health monitoring data as well as how this data can be used within privacy-preserving distributed statistical analysis. Regarding the first deal, we suggest the decentralization of the collection of medical data at the users' side. This is achieved by the use of personal agents that will be continuously online and collect the medical data of their owners. In addition to the data that are obtained by wearable sensors, the agents may also have other data, such as demographic elements about the patient and further information about his health records, as well. The additional data can be used to support filtering of the results within distributed computations. Apart from the management of the personal data, the patient agent's automatically monitors the different changes in medical data with a dedicated health component. As soon as the health component detects aberrations in the raw health data, it informs the HMC by giving it access to the user's data so as to decide itself for the danger of the situation. In our approach, the usage of the agents does not block the remote monitoring of the patient's health by an authorized doctor; it only ensures the controlled, user-aware, access to these sensitive data.

For the statistical analysis, we propose a privacy-preserving cryptographic protocol based on secure multi-party computations that accept as input current or archived values of users' wearable sensors. This distributed computation is performed by a community of the patients' personal software agents. We describe a prototype implementation of the proposed solution and present experimental results that confirm the viability and the effectiveness of our approach.

The personal data management approach proposed in this work achieves a number of advantages in comparison with the existing architecture of a UHMS, and simultaneously enhances the privacy of the patients in such a system. The main advantages are:

- Only controlled access to the health data is provided and every data access is logged by keeping who retrieved which data items and when this happened.
- The whole history of medical data, including the raw sensors' data, can be kept in the agent, whereas this might not be possible on the HMC for practical reasons. At the same time, decongestion of the HMC from the large amount of data. This can make the computational requirements of the central servers more tolerable.
- Less risk of massive theft of personal data since they are distributed at the users' side.
- Option for usability of these data by authorized third independent services or for performing distributed computations.

On the other hand, important advantages of our privacy-preserving statistical analysis approach in comparison to traditional statistical analysis techniques are:

- Utilizing valuable, sensitive, up-to-date personal data while ensuring privacy.
- Simplifying the process and significantly reducing the time and cost for conducting a statistical analysis.

A prerequisite for our approach is that each patient must have a personal software agent at his disposal and permanent access to the Internet. The computational requirements for the personal agent can be fulfilled with commodity hardware and hence its cost is not high. Thus, it is plausible to assume that patients with a UHMS can afford the extra cost for such an agent.

5.2 Related Work

Personal data of users are commonly stored in central databases at the service provider's side. In this way, the users have essentially no control over the use of their personal data. To address privacy concerns, different kinds of frameworks that are related to personal data have recently been proposed. See for example [120] and the references therein. Moreover, general surveys on privacy enhancing technologies are given for example in [90, 77]. Of particular importance for the management of health data in this work is the Polis platform presented in Chapter 3. In this work, we extend Polis agents with additional features and adapt the decentralized, agent-based approach of Polis for the management of the patients' personal data.

In the second and main part of this work, we present a solution for distributed privacy-preserving statistical analysis of personal health data. Our approach is based on secure multi-party computations (MPCs). The general model of a MPC was firstly proposed by Yao [214] and later was followed by many others [170, 147]. In general, a MPC problem concerns the calculation of a function with inputs from many parties, where the input of each participant is not disclosed to anyone. The only information that should be disclosed is the output of the computation. The general solution for MPC presented in [214] is powerful but commonly leads to impractical implementations.

A secure two-party computation (S2C) for the calculation of statistics from two separate data sets is presented in [57]. Each data set is owned by a company and is not disclosed during the computation. Similar results are shown in [58], this time focusing on linear regression and classification and without using cryptographic techniques. Some indicative works from the related field of privacy-preserving data mining are [100, 59, 125]. A major difference of our work from the above is that in our approach every participant is in control of his health data and that the distributed computation is performed by the community of the personal software agents. Using software agents as building blocks for software systems is an established practice; see for example [72] and for a recent survey [13].

Another approach for statistics on personal data is anonymization, i.e., the sanitization of a data collection by removing identifying information. The data anonymization approach and some of its limitations are discussed for example in [7, 133, 218]. Data anonymization applies to data collections in central databases and is not directly comparable to our decentralized approach. Finally, an example of an efficient privacy-preserving distributed computation is given in Chapter 4, where personal agents of doctors execute a distributed privacy-preserving protocol to identify the nearest doctor to an emergency. The focus of the present work is on privacy-preserving distributed statistical analysis using a massive number of participants.

5.3 Privacy-Enhanced Management of UHMD

In this section, we describe the proposed architecture for privacy-enhanced management of UHMD and show how it fulfills the goal of protecting the personal data and enhancing the privacy of patients.

5.3.1 Management Architecture

An overview of the proposed architecture for a UHMS is presented in Figure 5.1. The emphasis of the description is on the part of personal agents. The biomedical data that are produced by the patients' wearable sensors are wirelessly collected through a local wireless network in the patient's body into a personal mobile device, such as a smart phone. Afterwards, the measured biomedical data are transmitted via multiple complementary wireless networks (GPRS, 3G, Wi-Fi), through the Internet, towards the patient's personal agent. The personal agents that are used for this task are the Polis agents and have been suitably modified for this purpose. The features which have been added to the Polis agents so as to be used in a UHMS are:

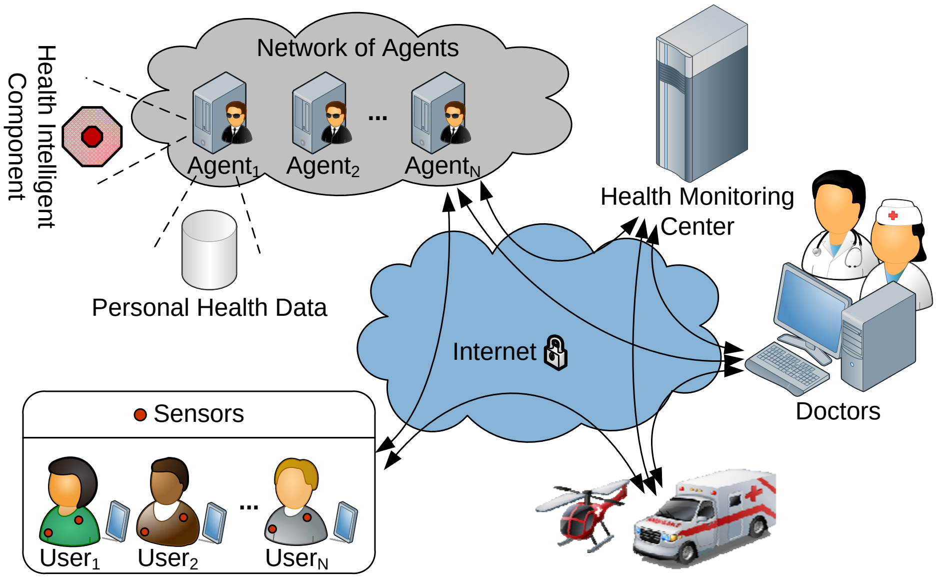
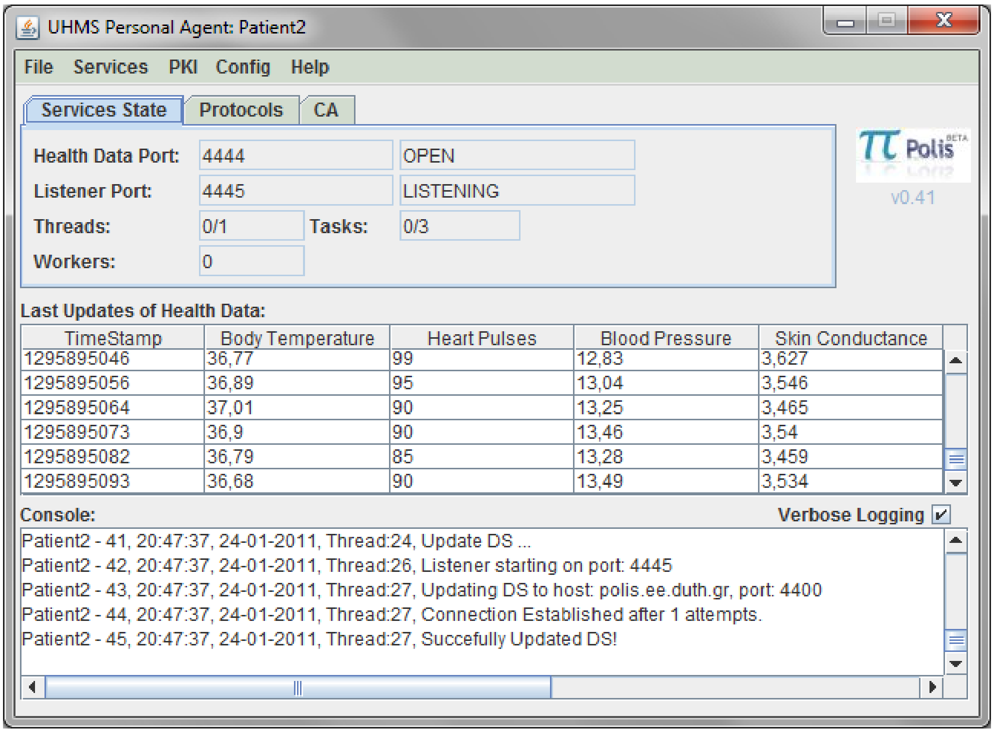


Figure 5.1: The proposed management architecture for a UHMS.

1. Ability to collect dynamic personal data, such as the biomedical data of the patients' wearable sensors.

2. Ability to control the values of the biomedical data for the detection of some indicative cases of emergency.

A snapshot of a patients' personal agent is shown in Figure 5.2. On the other hand, the patients' personal agents are self-organized into an appropriate virtual network topology that can provide easy organization and identification of the agents. This network topology can be used as a tool to conduct privacy-preserving distributed computations.



Health Data Port: 4444 OPEN

Listener Port: 4445 LISTENING

Threads: 0/1 Tasks: 0/3

Workers: 0

Last Updates of Health Data:

TimeStamp	Body Temperature	Heart Pulses	Blood Pressure	Skin Conductance
1295895046	36,77	99	12,83	3,627
1295895056	36,89	95	13,04	3,546
1295895064	37,01	90	13,25	3,465
1295895073	36,9	90	13,46	3,54
1295895082	36,79	85	13,28	3,459
1295895093	36,68	90	13,49	3,534

Console: Verbose Logging ☑

```
Patient2 - 41, 20:47:37, 24-01-2011, Thread:24, Update DS ...
Patient2 - 42, 20:47:37, 24-01-2011, Thread:26, Listener starting on port: 4445
Patient2 - 43, 20:47:37, 24-01-2011, Thread:27, Updating DS to host: polis.ee.duth.gr, port: 4400
Patient2 - 44, 20:47:37, 24-01-2011, Thread:27, Connection Established after 1 attempts.
Patient2 - 45, 20:47:37, 24-01-2011, Thread:27, Succefully Updated DS!
```

Figure 5.2: A snapshot of UHMS personal agent.

Our architecture can support an intelligent health component which can make a first check of the health data in real time. We provide an overview of the functionality of such a component; a real implementation of such a tool is outside of the scope of this work. The health component of the personal agent checks automatically the incoming vital signs with the purpose to address for further thorough check in HMC if there are indications of an emergency (see Figure 5.3). An example of rules/decisions that a health component can apply in order to decide about an emergency can be found in [190]. If necessary, the HMC can be consulted by the personal doctor of the patient. The personal doctors are shown as "Doctors" in Figure 5.1. Depending on the situation, the HMC can coordinate the immediate medical service at the closest or most appropriate local medical facility using the best available transportation service (e.g.: ambulance). Finally, an additional responsibility of the HMC is to inform the family of the patient about his condition so that they could rush to provide their help.

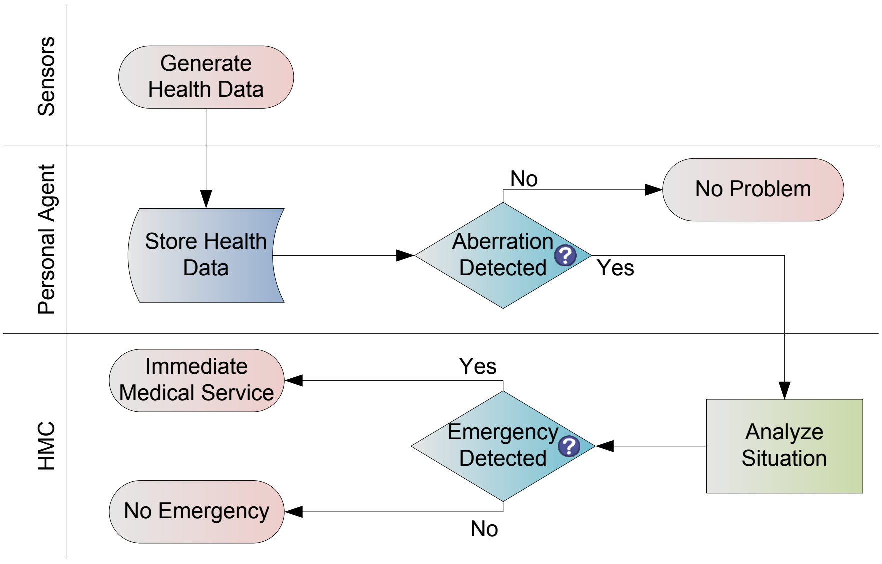


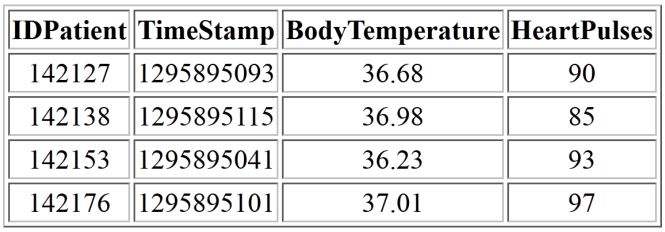
Figure 5.3: A system flowchart of the biomedical information.

5.3.2 Benefits of the Architecture

The idea of a decentralized architecture for storage and control of the patients' medical data into their personal agents, as it has already been mentioned provides the advantage of enhanced control on the user's personal data. Moreover, this decentralized approach can also contribute to improved data security, since invaders find large collections of personal data much more inviting than an individual's personal data [131]. The decentralized approach grants to the patient the right to control the disclosure of his health data and mitigates its feeling of being under permanent surveillance. In addition to enhancing privacy, the decongestion of HMC from the huge amount of data, including raw sensors' data, that would be accepted if the patients sent their data directly to it, is achieved. Even in the case that the data would be collected at the HMC, these would be much less in volume than those that would actually be produced by the sensors, thus the analysis would not be as effective as the one that would be made by the agents themselves by having the complete data. With the proposed health data management approach of this work, the HMC has now to handle only those cases which may be at a certain risk. In case that the patient's agent is out of operation, the patient's data which are collected by his smart phone could be transmitted for storage and control directly to the HMC until the failure is restored. This will ensure fault-tolerance against possible agent failures.

It is noteworthy that storing health data at the patients' side does not exclude the possibility to access the data from a central database as long as the database is entitled to do so. As shown in Chapter 3, the personal agents of Polis can be interconnected with mainstream database servers to provide transparent access to the personal data fields. The basic idea is that personal data fields in the central database do not contain the actual data; instead, a ticket represented by an appropriate data object is used to retrieve the data value on the fly. With this approach, which has been tested with an Oracle database server, a query submitted to the database may transparently retrieve – on the fly – personal data items from the associated personal software agents and present the personal data within the recordset (the answer of the database) of the query. An example query and the corresponding recordset are given in Figure 5.4. The data fields TimeStamp, BodyTemperature and HeartPulses are personal data fields and their content are – transparently for the database user – dynamically retrieved from the corresponding personal agents.

```
SQL> Select IDPatient , TimeStamp , BodyTemperature ,
     HeartPulses From CurrentBiomedicalData
     Where IDPatient Between 142120 And 142180;
```



IDPatient	TimeStamp	BodyTemperature	HeartPulses
142127	1295895093	36.68	90
142138	1295895115	36.98	85
142153	1295895041	36.23	93
142176	1295895101	37.01	97

Figure 5.4: SQL access to remote health data.

The choice to store the patient's data in an agent enables the possibility to utilize these data for the common wealth. The Nearest Doctor Problem (NDP) that was presented in Chapter 4 is a typical example. The NDP is a privacy-preserving protocol, which uses a network of the doctors' agents aiming to find the nearest doctor in case of an emergency, by using dynamic data such as their location. In our case, the data of the patients could be used for a similar privacy-preserving distributed computation. Such an example is the monitoring progress/spread of a pandemic in a region. Data such as the location and the body temperature of the patients would be required for this example. Another example is a medical statistical research on the biomedical data of the wearable sensors as well as on the medical records of patients. In the following section we use our distributed data management approach for a privacy-preserving distributed statistical analysis application.

5.4 Privacy-Preserving Statistical Analysis on UHMD

In this section, we present a method for privacy-preserving statistical analysis of ubiquitous health monitoring data (UHMD). The core of our approach is a privacy-preserving distributed computation that is collaboratively executed by the participating personal software agents. Regarding privacy, the question we will address is how to achieve privacy of type (b) (see details in Section 2.1.4), that is, how to compute the statistic results without pooling the data, and in a way that reveals nothing but the final results of the distributed computation.

5.4.1 Architecture of the Distributed Computation

Our solution is build on top of the privacy-enhanced UHMS presented earlier in this work. An overview of the architecture of the statistical analysis system including the extra components that are required for a distributed statistical analysis computation, i.e., the Network Community of Personal Agents and the Statistical Analysis Service (SAS), is shown in Figure 5.5.

The personal agents are organized into a virtual topology, which may be a simple ring topology or a more involved topology for time-critical computations. On the other hand, the SAS is a server that initiates the distributed computation on the users' medical data and collects the aggregate results. Each researcher who wishes to carry out a statistical research and is entitled to do so, can submit his task to the SAS.

5.4.2 The Main Steps of the Distributed Computation

The main steps of the proposed distributed computation for the statistics calculation are:

- Initially, the researcher submits the request to conduct a specific statistical analysis to the SAS.

- The SAS accepts the request after verifying the credentials of the researcher.

- The SAS picks one of the personal agents to serve as the root-node for the particular computation and submits the request to it.

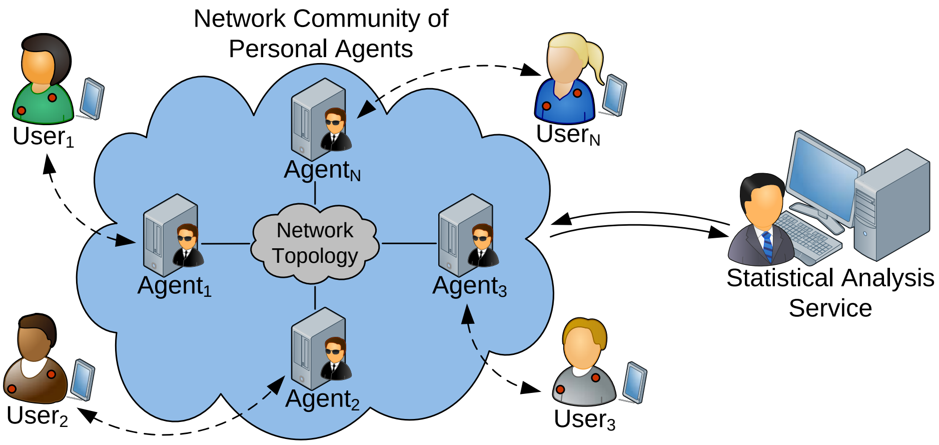


Figure 5.5: The architecture for performing privacy-preserving statistical analysis.

- The root-node coordinates a distributed computation that calculates the specified statistical function.
- At the end of the distributed computation, the SAS and the researcher will only learn aggregate results of the computation without any additional information of the personal data of individual participants.

5.4.3 The Secure Distributed Protocol

In this section, we present the main idea of the cryptographic protocol that is used in the privacy-preserving statistical computations. The protocol is secure in the Honest-But-Curious (HBC) model (see Section 2.9.1), where the users' agents participating in the computation follow the protocol steps but may also try to extract additional information. During the calculation the actual users' personal data are not disclosed in any stage of the process but only the aggregate results are revealed at the end. An instance of a statistical computation problem consists of:

- **N patients** $P_1, P_2, \ldots, P_N$ and their personal data.
- **N personal software agents:** The agents of all patients that will participate in the distributed privacy-preserving computation.
  - **Input:** The type of the statistical function and its parameters. In addition, selectivity constraints for the data set may also be specified. Note that more than one statistical functions on the same dataset can be calculated with a single computation.

- **Output:** The necessary aggregate values (e.g. w_x, u_x, z_{xy} and n, which are defined later) that are needed to calculate the given statistical function.

Consider the following statistical computation instance: Computing the average of the female patients' age in a city. First, we assume that the results of the specific query are not considered a threat against the users' privacy, that is, privacy type (a) of Section 2.1.4 is preserved. Then, given the computation instance, the SAS chooses a node from the network of the users' agents as the root-node for the particular computation. The SAS sends the type of the requested computation and its parameters to the root-node. The parameters of the computation, i.e., the female gender and the city name, are used to filter the data set. Each personal agent, decides privately to provide data or not to the statistical research.

A simple topology for the personal agents is a virtual ring topology that contains all agents as nodes (Figure 5.6b). For time-critical computations, more complex topologies like a virtual tree can be used (Figure 5.6a). The tree topology for example has been used in privacy-preserving computation of Chapter 4. At the end of the execution, the root-node collects the results of the calculation as an encrypted message and sends it to the SAS. The message is encrypted with the public key of the SAS, which is assumed to be known to all nodes. In this way, the protocol ensures k-anonymity (see Section 2.1.3.1), where $k = N$ and N is the number of all the nodes in the network. We use the Paillier public key cryptosystem [141] for the proposed cryptographic protocol. An important feature of the Paillier cryptosystem is its homomorphic property which supports additive homomorphism (see details in Section 2.6).

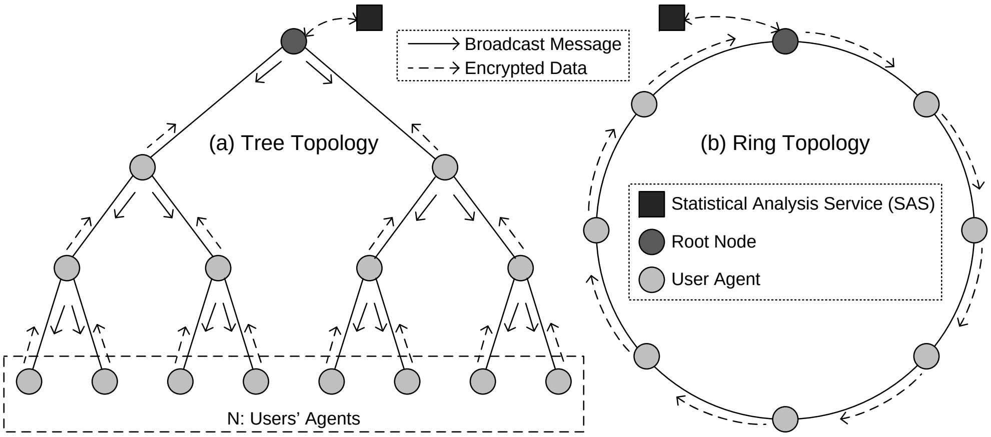


Figure 5.6: Possible network topologies.

5.4.4 The Computations

In this section, we use our approach to calculate representative statistical functions with a distributed privacy-preserving computation. Wherever it is necessary, the expression of the statistical function is brought to a form that is appropriate for the distributed computation.

5.4.4.1 Arithmetic Mean

The arithmetic mean of a variable X (with sample space $\{x_1, \ldots, x_n\}$) is given by the following equation:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

We use the additive homomorphic property of Paillier to calculate the value of the terms $u_x = \sum_{i=1}^{n} x_i$ and n. The calculation is privacy-preserving; no single x_i information is disclosed. Once the SAS learns the values of the terms u_x and n, it can compute the arithmetic mean. More analytically, using the homomorphic property of Paillier, the two terms u_x and n can be transformed into the following form:

$$E_{pk}(u_x) = \prod_{i=1}^{n} E_{pk}(x_i) \quad \text{and} \quad E_{pk}(n) = \prod_{i=1}^{n} E_{pk}(1) ,$$

where the E_{pk} indicates that the message is encrypted with the current public key of SAS for the specific statistical analysis. Each agent i that participates in the statistical analysis, prepares its own encryptions $E_{pk}(x_i)$ and $E_{pk}(1)$. These encrypted messages are used to calculate the above two global products. Agents that do not participate in the statistical computation (because for example they do not satisfy some selection criterion) multiply each of the above two products with an independent encryption of zero $E_{pk}(0)$.

5.4.4.2 Frequency Distribution

The frequency distribution is a tabulation of the values that one or more variables take in a sample. Each entry in the table contains the frequency or count of the occurrences of values within a particular group or interval; in this way, the table summarizes the distribution of values in the sample. The graphical representation of the frequency distribution is the well known histogram. Figure 5.7 shows how the frequency distribution would become by using ciphertext as counters in each range, where each ciphertext is given by the following equation:

$$E_{pk}(n_v) = \prod_{i=1}^{n} E_{pk}(m), \text{ where } m = \begin{cases} 1, & x \in [x_{v-1}, x_v) \\ 0, & x \notin [x_{v-1}, x_v) \end{cases}$$

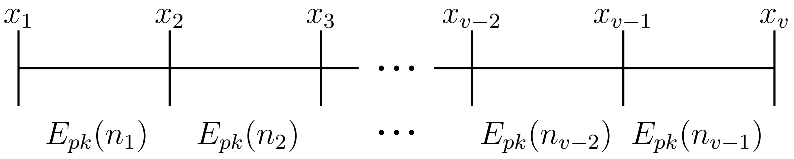


Figure 5.7: Representation of a frequency distribution.

5.4.4.3 Variance

The variance $var(X)$ of a variable X is used as a measure of how far a set of numbers are spread out from each other and is defined as:

$$var(X) = \frac{1}{n}\sum_{i=1}^{n} x_i^2 - \left(\frac{1}{n}\sum_{i=1}^{n} x_i\right)^2$$

The unknown terms that are required to calculate the variance with the help of the homomorphic property of Paillier are $w_x = \sum_{i=1}^{n} x_i^2$, $u_x = \sum_{i=1}^{n} x_i$ and n, by taking the following form:

$$E_{pk}(w_x) = \prod_{i=1}^{n} E_{pk}(x_i^2), \quad E_{pk}(u_x) = \prod_{i=1}^{n} E_{pk}(x_i) \quad \text{and} \quad E_{pk}(n) = \prod_{i=1}^{n} E_{pk}(1)$$

5.4.4.4 Linear Regression

The linear regression of a dependent variable Y of the regressors X is given by the equation $y = a + bx$, where a and b are parameters. The determination of a and b gives an approximate line, which connects the values of Y with the corresponding values of X. This line can be constructed by using the method of least squares and the parameters a and b are given by the following equations:

$$b = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2} \quad \text{and} \quad a = \frac{1}{n}\sum_{i=1}^{n} y_i - b\frac{1}{n}\sum_{i=1}^{n} x_i$$

The unknown terms that are required to calculate the parameters of line y with the help of the homomorphic property of Paillier are the $w_x = \sum_{i=1}^{n} x_i^2$, $u_x = \sum_{i=1}^{n} x_i$, $u_y = \sum_{i=1}^{n} y_i$, $z_{xy} = \sum_{i=1}^{n} x_i y_i$ and n, by taking the following form:

$$E_{pk}(w_x) = \prod_{i=1}^{n} E_{pk}(x_i^2), \quad E_{pk}(u_x) = \prod_{i=1}^{n} E_{pk}(x_i),$$

$$E_{pk}(u_y) = \prod_{i=1}^{n} E_{pk}(y_i), \quad E_{pk}(z_{xy}) = \prod_{i=1}^{n} E_{pk}(x_i y_i) \quad \text{and} \quad E_{pk}(n) = \prod_{i=1}^{n} E_{pk}(1)$$

5.4.4.5 Linear Correlation Coefficient

The linear correlation coefficient $corr(X, Y)$ of two random variables X and Y is a measure of the strength and the direction of a linear relationship between two variables and is defined as:

$$corr(X,Y) = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sqrt{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}\sqrt{n\sum_{i=1}^{n} y_i^2 - \left(\sum_{i=1}^{n} y_i\right)^2}}$$

The unknown terms that are required to calculate the linear correlation coefficient with the help of the homomorphic property of Paillier are $w_x = \sum_{i=1}^{n} x_i^2$, $u_x = \sum_{i=1}^{n} x_i$, $w_y = \sum_{i=1}^{n} y_i^2$, $u_y = \sum_{i=1}^{n} y_i$, $z_{xy} = \sum_{i=1}^{n} x_i y_i$ and n, by taking the following form:

$$E_{pk}(w_x) = \prod_{i=1}^{n} E_{pk}(x_i^2), \quad E_{pk}(u_x) = \prod_{i=1}^{n} E_{pk}(x_i),$$

$$E_{pk}(w_y) = \prod_{i=1}^{n} E_{pk}(y_i^2), \quad E_{pk}(u_y) = \prod_{i=1}^{n} E_{pk}(y_i),$$

$$E_{pk}(z_{xy}) = \prod_{i=1}^{n} E_{pk}(x_i y_i) \quad \text{and} \quad E_{pk}(n) = \prod_{i=1}^{n} E_{pk}(1)$$

5.4.4.6 Covariance

The covariance $cov(X, Y)$ of two random variables X and Y is a measure of the strength of the correlation between the two variables and is defined as:

$$cov(X,Y) = \frac{1}{n}\sum_{i=1}^{n} x_i y_i - \frac{1}{n}\sum_{i=1}^{n} x_i \cdot \frac{1}{n}\sum_{i=1}^{n} y_i = \frac{1}{n}\sum_{i=1}^{n} x_i y_i - \frac{1}{n^2}\sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i$$

The unknown terms that are required to calculate the covariance with the help of the homomorphic property of Paillier are $u_x = \sum_{i=1}^{n} x_i$, $u_y = \sum_{i=1}^{n} y_i$, $z_{xy} = \sum_{i=1}^{n} x_i y_i$ and n, by taking the following form:

$$E_{pk}(u_x) = \prod_{i=1}^{n} E_{pk}(x_i), \quad E_{pk}(u_y) = \prod_{i=1}^{n} E_{pk}(y_i),$$

$$E_{pk}(z_{xy}) = \prod_{i=1}^{n} E_{pk}(x_i y_i) \quad \text{and} \quad E_{pk}(n) = \prod_{i=1}^{n} E_{pk}(1)$$

Comments. From the analysis of the above statistical functions, we conclude that apart from the frequency distribution, all other function can be simultaneously calculated by computing once the required aggregate terms. Moreover, it is clear that the proposed solution can also be used to calculate other statistical functions, such as the polynomial regression and so on. We discuss such issues in the next section.

5.4.5 The Protocol's Security

In this section, we show that the proposed protocol of a distributed statistical analysis in a UHMS does not violate the privacy of participants. The security holds for the model of Honest-But-Curious (HBC) users (see details in Section 2.9.1).

In the cryptographic protocol described above, the information exchanged by agents is encrypted with the Paillier cryptosystem [141], which is known to offer Semantic Security [82], that is, it is infeasible for a computationally bounded adversary to derive significant information about a message (plaintext) when given only its ciphertext and the corresponding public encryption key. Consequently, assuming honest-but-curious parties and that users' agents do not collude with the SAS party outside of the protocol, our approach is semantically secure. In Section 5.4.6, we show that the case where some user agents collude with the SAS outside of the protocol can be handled with a threshold decryption model.

From the above, we conclude that the computation with the homomorphic encryptions does not leak personal information of participating individuals (privacy type (b) in Section 2.1.4). As noted earlier, the (decrypted) outcomes of the statistic computation are also assumed to preserve privacy of type (a). We can now discuss the privacy guarantee of the whole approach. A common criterion for privacy protection is k-anonymity (see Section 2.1.3.1), which requires that data of the outcome cannot be associated with any particular patient. The proposed solution offers k-anonymity in the sense that the result computed at the end of the protocol cannot be attributed to any of the N participated agents, i.e., $k = N$ even if the list of participating users is known (assuming no background information on specific users is available).

In summary, the key security features of our protocol are:

- Each agent that receives a message from the previous node cannot obtain information about the contents of the message, because the ciphertexts are encrypted with the Paillier cryptosystem.

- Each node alters the ciphertexts of the computation. Even the nodes that do not participate in the statistical function multiply the ciphertexts with an encryption of number “0”, which is the neutral element of the additive homomorphic property of Paillier. Thus, the ciphertext is modified at every node, even if the corresponding node does not give any input to the computation.

- At the end of the protocol, only the variables that are needed for a particular statistical function are revealed. As a result, no individual can be associated with the value that he had used in the computation. Consequently, the proposed protocol preserves k-anonymity for $k = N$, where N is the number of all agents in the network.

Another criterion for evaluating privacy protection is the concept of differential privacy (see details in Section 2.1.3.2). Loosely speaking, the aim of differential privacy is to ensure that the ability of an adversary to inflict harm (or good, for that matter) – of any sort, to any set of people – should be essentially the same, independent of whether any individual opts in to, opts out of, the dataset [62, 61].

If privacy of type (a) (Section 2.1.4) is preserved, for example, no queries or sequences of queries addressing a very small number of individuals are permitted etc., then it is plausible to assume that our approach achieves a satisfactory level of differential privacy. Note that the outcomes of the statistical computations are sums or aggregate results computed from a large number of sensor measurements and demographic values of a large population. One may also consider of adding Laplace noise [63] to the statistical results in order to further enhance the differential privacy criterion, even though there is some recent criticism of such an approach [165].

5.4.6 Security Discussion

In this section, we identify some representative threats against our application and discuss how they are or can be addressed within our approach. The threats concern either the correctness of the aggregated results or the privacy of the involved participants.

- *Incorrect sensor measurements.* This case refers to the case where one or more sensors generate erroneous data of values large enough to significantly influence the aggregate result. Such incidents could disrupt a statistical analysis and would be difficult to be noticed in the statistical results. However, such incorrect measurements could be detected by the intelligent health component or some dedicated filter of the patient's agent and excluded from the current statistical analysis. This solution is acceptable in the HBC model. Moreover, even for the case where such incorrect measurements could be maliciously submitted in order to skew the statistical result, we could use more advance techniques of the area of electronic voting [111]. In this case, each node would have to run a zero-knowledge proof with its predecessor/s with purpose to verify that the measurements are within an acceptable range.

- *Dedicated queries with purpose to reveal personal biomedical data of a particular patient.* One query or a set of queries may be chosen and submitted to target specific patients, by using background information on the set of participating individuals. Such dedicated queries may cause leakage of personal data of the selected patients. As noted earlier, such an attack is a threat against privacy of type (a) and the participants have to be protected with respect to such attacks. The problem is well known in the area of statistical databases [5] and it is not something new. A possible solution could be to use a second authority which will check if there are enough patients who cover the query's criteria before the SAS performs the specific statistical analysis.

- *Collusion among some patients and the SAS.* In this case, the SAS will try to collaborate with at least two patients (in the simple ring topology) with purpose to reveal the private values of a patient. These two patients have to be the predecessor and the successor of the particular patient. More specifically, the colluding predecessor creates neutral ciphertexts and forwards them to the intermediate node. This node would then encrypt its private values and forward the result to its colluding successor (according to the topology). The successor would then immediately return the values to the SAS which now gets to decrypt these private values. Such malicious behaviors can be effectively handled by deploying threshold decryption model [46] for the decryption of the encrypted values. Threshold decryption model requires a number of designated parties exceeding an appropriate threshold to cooperate for the decryption to be possible.

5.4.7 Experimental Results

To evaluate our solution, we developed a prototype that carries out distributed statistical analysis on medical data. The application is implemented in Java and for the cryptographic primitives the Bouncycastle [25] library is used. The personal agents of the Polis platform (Chapter 3) are used as the personal data management agents of the patients. For this approach, the Polis agents were suitably modified so as to be able to manage both health records and health data that would actually be collected through a secure communication channel by the patients' wearable sensors. The community of the personal agents is organized as a Peer-to-Peer network. At this stage of development of the prototype, the backbone of the topology is a virtual ring topology. The ring offers a simple and reliable solution for the interconnection of the agents. For time-critical calculations of statistics a more involved topology like a virtual tree should be used.

The personal agents use production-ready cryptographic libraries and employ 1024 bits RSA X.509 certificates. The communication between agents is performed over secure sockets (SSL/TLS) with both client and server authentication (see details in Sections 2.7 and 2.8). Below we describe an experiment of a distributed statistical analysis with 6 agents and the SAS. The requested statistic is:

- *The arithmetic mean of the current body temperature of patients who are aged between 55 and 65 years old and their gender is female.*

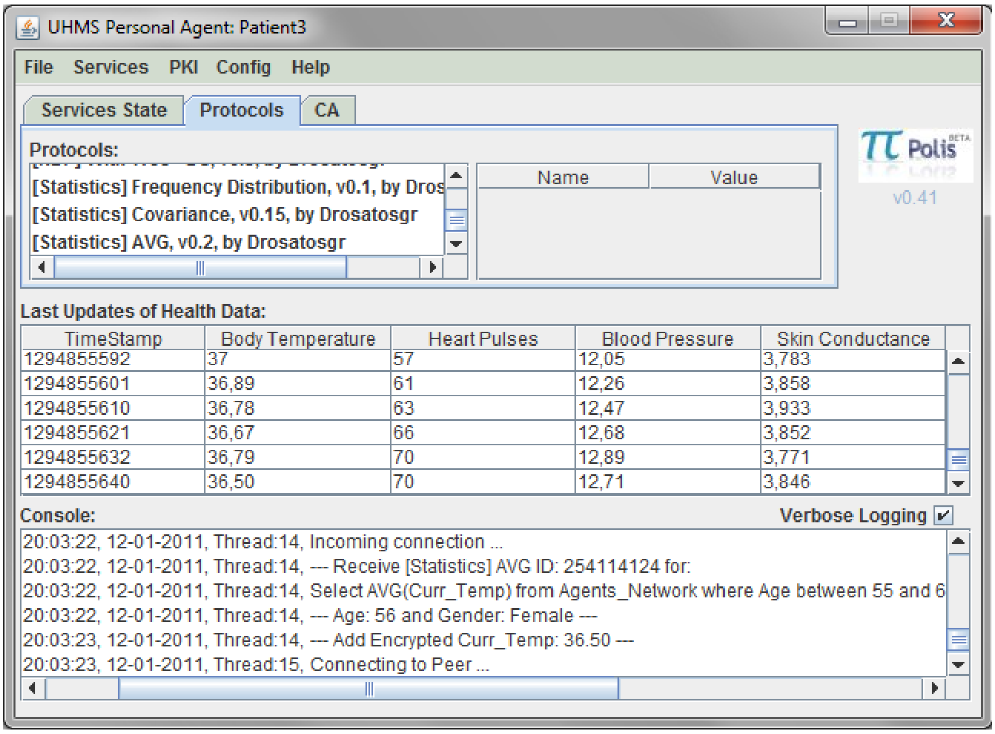
For the needs of the experiment, each agent generates random values for the age, the gender, and the current body temperature. We assume that the selectivity of the query criteria is high enough to preserve privacy of type (a) (Section 2.1.4). Then, in brief, the statistical computation works as follows. Initially, the SAS randomly chooses a node from the agents' network, in this case agent 'Patient2', as the root-node, and forwards the description of the statistical computation to it. The values of each agent which are related to the computation are shown in Table 5.1. The last two columns show the aggregate values that are encrypted after the corresponding agent applies its values to the results. Since the homomorphic property of Paillier applies to integers, decimal values like the body temperature have also to be represented with integers. In our example, the temperature is rounded to a number with at most two decimal digits and then multiplied by 100 to become an integer.

At the end of the computation, the agent 'Patient2' as the root-node collects the results and sends them back to the SAS. Finally, the SAS decrypts the results and finds that the average of the question which was submitted is 37.125 ^oC. A snapshot of the application during the execution of the experiment is shown in Figure 5.8.

We also performed a set of large-scale experiments with up to 300 agents. More precisely, we evaluated the efficiency of our solution with a series of experiments on a gradually increasing number of up to 300 agents. For this experiment, a network of 30 computer workstations with Intel Core 2 Quad Q8300 CPU's at 2.5 GHz, 2 GB RAM and a 100 Mbps network, were used. The workstations

Agent	Curr. Temp.	Age	Gender	$E_{pk}(u_x) = \prod_{i=1}^{n} E_{pk}(x_i)$	$E_{pk}(n) = \prod_{i=1}^{n} E_{pk}(1)$
Patient2	36.68 ^oC	51	Female	$E(0)$	$E(0)$
Patient3	36.50 ^oC	56	Female	$E(3650)$	$E(1)$
Patient4	37.70 ^oC	60	Female	$E(7420)$	$E(2)$
Patient5	38.10 ^oC	65	Female	$E(11230)$	$E(3)$
Patient6	37.12 ^oC	59	Male	$E(11230)$	$E(3)$
Patient1	36.20 ^oC	63	Female	$E(14850)$	$E(4)$

Table 5.1: Example of computation, where the agents in gray rows did not take part in computation.



Protocols:

[Statistics] Frequency Distribution, v0.1, by Dros
[Statistics] Covariance, v0.15, by Drosatosgr
[Statistics] AVG, v0.2, by Drosatosgr

Name	Value

Last Updates of Health Data:

TimeStamp	Body Temperature	Heart Pulses	Blood Pressure	Skin Conductance
1294855592	37	57	12,05	3,783
1294855601	36,89	61	12,26	3,858
1294855610	36,78	63	12,47	3,933
1294855621	36,67	66	12,68	3,852
1294855632	36,79	70	12,89	3,771
1294855640	36,50	70	12,71	3,846

Console:

```
20:03:22, 12-01-2011, Thread:14, Incoming connection ...
20:03:22, 12-01-2011, Thread:14, --- Receive [Statistics] AVG ID: 254114124 for:
20:03:22, 12-01-2011, Thread:14, Select AVG(Curr_Temp) from Agents_Network where Age between 55 and 6
20:03:22, 12-01-2011, Thread:14, --- Age: 56 and Gender: Female ---
20:03:23, 12-01-2011, Thread:14, --- Add Encrypted Curr_Temp: 36.50 ---
20:03:23, 12-01-2011, Thread:15, Connecting to Peer ...
```

Figure 5.8: A snapshot of the agent 'Patient3'.

were running a 32-bit operating system and the agents were executed in 32-bit Java virtual machines. Each computer was shared by at most 10 agents, to ensure an even workload distribution and avoid single overloaded workstations; an overloaded workstation would become a bottleneck that could significantly delay the execution of the whole protocol.

The running times of our experiments are shown in Figure 5.9. In this figure, we present the execution times for the computation of the arithmetic mean, the variance and the frequency distribution (for 10 subintervals) functions. As expected, the execution times depend practically linearly on the number of agents which take part in the computation and on the number of encryptions and multiplications in every statistical function. The overall running time is more than satisfactory for batch execution of statistical computations. In case of large numbers of statistical computations, the rate of computations can be substantially improved by using a pipeline of independent computations. For cases where the run-time of the computations is important, the distributed computation can be executed on a virtual tree or some other – low depth – topology, instead of the ring topology. In this case one would expect, and we actually have such preliminary measurements albeit within a different context (Chapter 4), that the total running time will depend only logarithmically on the total number of nodes.

Finally, the execution times of the computations could be significantly reduced by simply using 64-bit Java virtual machines for running the experiments. This change would greatly improve the execution times especially of the heavy encryption operations which involve BigInteger[1] variables. In a comparable, independent, experiment we noticed an almost four-times improvement of the execution times when 64-bit Java was used in place of 32-bit Java. The use of the 64-bit virtual machine seems to effectively exploit the bigger registers of the AMD64 architecture for the cryptographic operations.

5.5 Conclusions

The tendency of the society towards increasing numbers of elderly people and generally people who need continuous health monitoring makes the need of Ubiquitous Health Monitoring Systems (UHMS) imperative. At the same time the concerns of the public about privacy are also rising. In this work, we presented an architecture for privacy-enhanced UHMS and proposed the use of the ubiquitous health data that are obtained by the wearable sensors in a UHMS for caring out statistical researches. The proposed architecture allows the patients to have enhanced control over their personal data, so as not to have the feeling of being continuously under surveillance. The enhanced control on their personal data was achieved by using personal software agents for the management of the patients' personal data. Putting personal agents in charge of personal health data can open the way for the definition and implementation of new services which utilize personal data to contribute to public well being, while at the same ensuring the privacy of the involved individuals.

[1] BigInteger is an immutable arbitrary-precision integer.

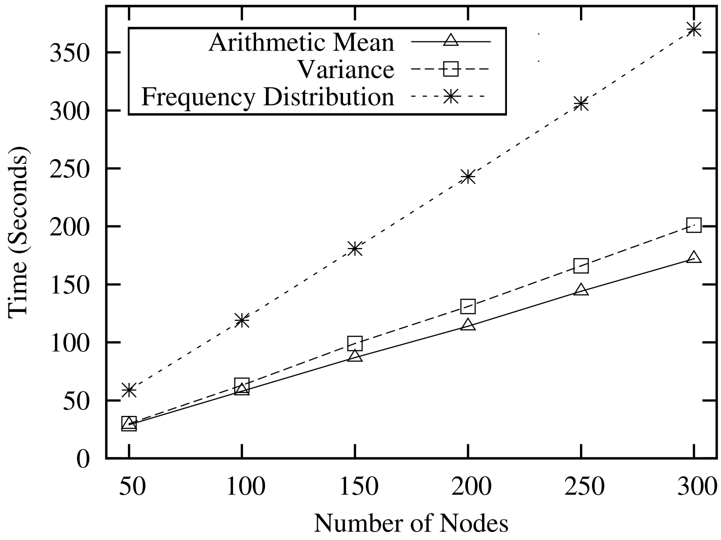


Figure 5.9: Computation times of arithmetic mean, variance and frequency distribution (for 10 subintervals) statistical functions with respect to the number of agents.

In this direction, we presented a solution for privacy-preserving statistical analysis on ubiquitous health data. The protection of privacy is achieved by using cryptographic techniques and performing a distributed computation within a network of patients' personal agents. We described how representative statistical functions can be executed distributedly by using the proposed cryptographic protocol. Finally, we developed a prototype implementation and performed an experimental evaluation that confirmed the viability and the efficiency of our approach.

Chapter 6

Privacy-Preserving Computation of Participatory Noise Maps in the Cloud

6.1 Introduction

Many people are reluctant to entrust today's computer systems with their personal information, thus in [132] the authors have identified privacy as a pillar of trustworthy software systems. Specifically, the authors consider trustworthy systems to respect *privacy* if *the customer is able to control data about themselves, and those using such data adhere to fair information principles*. This work contributes to this end, by presenting an architecture and implementation for incorporating privacy-preserving techniques in participatory sensing applications.

Participatory sensing [28, 146] appropriates everyday devices such as mobile phones to acquire information about the physical world (and the people in it) at a level of granularity which is very hard to achieve otherwise. A crucial component of participatory sensing systems is *geolocation*, i.e., labeling data with geographic coordinates. For example, in the context of *NoiseTube* [123, 175], a participatory sensing system and service[1] designed to monitor and map noise pollution, it would be practically impossible to produce noise maps on the basis of sound level measurements, gathered quasi-continuously as contributors walk the streets, without automatic geolocation of measurements by means of GPS (Global Positioning System). The same situation applies more generally, as the potential of high measurement granularity essential to participatory sensing frameworks is only manageable if this data can be automatically organized, e.g. through location.

[1] http://www.noisetube.net

However, location traces constitute sensitive personal information. In small-scale deployments, in which individual contributors know or trust each other, the disclosure of such information may be acceptable. However, in larger-scale deployments, involving more contributors and possibly coordinated by some authority, trust relationships tend to be much weaker and contributors may be uncomfortable about the type of information that is collected, and with whom it is shared. Hence, scaling up a participatory sensing project inherently increases privacy concerns [113, 38], which in turn can severely hamper the project from reaching its goals.

In this work, we present a privacy-preserving solution for participatory sensing frameworks where location-based data aggregation is used to produce maps involving measurements of groups of users. Our system, called *NoiseTubePrime* or *NTPrime* for short, relies on privacy-preserving distributed computation in the cloud and is oriented towards coordinated mapping campaigns set up by citizens and/or authorities. NoiseTubePrime differs from earlier work on privacy-preserving mobile sensing systems [101, 18, 168], as it is at the same time simple, safe, verified, and transparent to end-users. The core of the NoiseTubePrime architecture is a privacy-preserving cryptographic protocol (see for example [20]) implementing a large scale distributed computation.

The novelty of our approach is first, that by thinking in terms of campaigns rather than when thinking in terms of privacy of stand-alone users one can deal with privacy in a distributed way. This allows us to avoid to compensate privacy with the accuracy of the resulting maps (as e.g. data obfuscation does). Campaigns are focused sensing efforts of groups of users, where geographical, temporal and/or contextual concerns are put forward. The outcome of a campaign typically entails aggregating individual user data into a composite map, and it is precisely this property that allows us to rely on a distributed cryptographic protocol which ensures privacy of users and at the same time precise noise maps (in terms of the contributing measurements).

Second, our approach is the first to incorporate cloud computing, essential to ensure transparency and efficiency. The main reasons for resorting to computation in the cloud are high availability, ease of deployment, and scalability. The need for high availability is essential because we need to ensure that all the data collected by different campaign contributors is available every time an aggregated map is to be generated. Concretely this means that a piece of software representing each contributor (i.e. an agent) must be *online* and able to respond to outside requests at all times. While in principle this is feasible with a smartphone application (cf. chat applications), mobile data connectivity can be intermittent and local computational resources are limited. Also the fact users would need to run a “server-like” application on their personal phone could raise additional privacy concerns. For these reasons it seemed desirable to host the software agents on an infrastructure with near-permanent availability and vast computational resources. Essentially this means we decouple the role of collecting data on a mobile phone (using the NoiseTube Mobile app) from the role of managing the data and taking part in distributed computations (through the NoiseTubePrime software agent). Also, data management directly in a mobile device contains security threats that are described in [135]. Ease of deployment is an important concern because you just cannot expect campaign contributors to install and configure complicated software on their personal computers, let alone run their own servers. Hence low-cost or even free cloud computing services that allow people with relatively moderate computing skills to set-up and manage their own software agent with a few clicks (using a deployment package provided to them) offer a suitable alternative. Cloud computing services are also extremely scalable, meaning that sensing campaigns can grow without the coordinators having to worry about things like server load and network bandwidth.

6.2 Background

6.2.1 Participatory Monitoring Campaigns

Participatory platforms, are typically client-server systems that consist of a mobile application used by contributors on the one hand, and a *community memory* [174, 175] system running on a central server on the other. The former enables users to sense environmental parameters (e.g. sound) whenever and wherever they please. When data collected by a user is uploaded to the server, either automatically or manually, typically a map (and, optionally, a statistical analysis) is produced showing how the measured parameter is distributed geographically. In the concrete example of NoiseTube, the map is a location trace of a user's measurements, shown as green-to-red colored dots depending on the sound levels measured [123, 175].

NoiseTube is one of the first participatory sensing platforms to endeavour the transition from a tool used by individuals to one that can serve as a basis for coordinated measurement campaigns, be it grassroots or authority-led. Indeed, recent work [48, 49, 175] shows that, when coordinated properly, NoiseTube campaigns can produce collective noise maps that are of comparable quality to simulation-based maps produced by governments today. To do this a statistical component was introduced which produces a single aggregate noise map from a collection of measurement tracks contributed by groups of users. The basic procedure is this: divide the surveyed area into smaller areas using a regular grid, partition the set of measurements over those areas based on their geographic coordinates, make a statistical analysis per unit area, and finally, map the color coded averages on each pertaining area. While such aggregated map-making has been carried out before in individual instances, we are currently extending the NoiseTube Web application with this collective noise mapping functionality so that it enables community-driven environmental sensing. The idea is that larger and more diverse groups of people may use NoiseTube to define and coordinate their own campaigns with little to no involvement of experts. Concrete examples are citizens that wish to map the noise in their commune or city while construction works are going on, or a commune which wants to investigate how rescheduling of a bus line effects Monday morning traffic.

In small-scale campaigns, such as the one reported in [48, 49], privacy issues tend to be of little concern. The reason is that participants typically already know and trust each other (e.g. because they are members of a citizen activist group), who consciously take part in a scientific experiment or community effort and got time to get acquainted with the researchers and/or coordinators in person. However, in campaigns that cover larger areas, last longer, and involve larger numbers of more diverse contributors and coordinators, this kind of mutual confidence could easily break down.

The issue of privacy is thus an important hurdle for the adoption of NoiseTube as a tool for larger-scale (e.g. city-wide) noise mapping campaigns, a situation which holds more generally. Hence there is a clear need for a privacy-preserving extension of participatory sensing platforms. This is why, in this work, we design a privacy-preserving extension of participatory sensing frameworks, introducing privacy-preserving functionalities at several levels, and implement it in the context of the NoiseTube platform. As a proof of concept, as well as a validation of correctness, we use data from the above-mentioned earlier experiments to demonstrate that NoiseTubePrime can produce exactly the same maps in a privacy-preserving way.

6.2.2 Personal Data and Privacy

Desktop, mobile computing and sensing technology have greatly increased the amount of personal information that is generated, while recent advances of database technology enable the potential for this information to be (permanently) stored and processed [99, 98]. To give an indication on the volume of personal data, a case in point is that of Max Schrems, recently on the news. The 24-year-old Schrems asked Facebook for a copy of all the data the social network has on file for him and he got back a CD with 1,222 PDF files documenting his every move[1]. This personal data is a critical, valuable resource that has to be

[1] http://threatpost.com/twenty-something-asks-facebook-his-file-and-gets-it-all-1200-pages-121311

protected in order to ensure the individual’s privacy rights: to protect his privacy by retaining the control over his personal data and knowing who, when and why gets access to these data. At the same time, the wide acceptance of electronic transactions for everyday tasks resulted in an abundance of applications that rely on the processing of this personal data. Thus, locking all personal data away is not a solution. Instead, it should be possible to process such data in a way that is both efficient and ensures its protection. Furthermore, when an individual makes a transaction, only the minimum amount of personal information that is needed to complete it should be disclosed, with clear terms on how the personal data will be used.

Personal data of individuals who contribute to a participatory sensing project typically exists in the form of location traces. To preserve user privacy in this setting we follow an approach similar to one described in the Polis platform (Chapter 3). Concretely, each user is represented by a personal software agent, who manages his personal data and controls access to this data. Third-party applications, other agents or services direct requests at these personal software agents rather than at the user himself. The agents respond to these requests according to a corresponding license agreement or policy. For the needs of this work, we adapted the Polis approach to the management of personal data of participatory sensing. However, the most important difference of this work with respect to previous applications of the Polis approach, is that in NoiseTubePrime, the personal software agents are outsourced to the cloud.

6.2.3 Cloud Computing

The past few years have seen a shift towards support for cloud computing technology, both by industry and governments. Computing infrastructure, instead of being offered *as a product*, rather is offered *as a service*. Instead of running on machines owned or controlled by the user or client these services run “in the cloud”[1]. In this way platforms, storage, computational power, as well as software is designed, managed and delivered as cloud-based services. Today, there exist several operational cloud platforms, offering services though APIs either for free or under certain cost models. While cloud technology significantly changed how computing infrastructure is offered, there are two major issues that have arisen: one is interoperability, and the second is privacy. Indeed, cloud computing comes in different flavours from various vendors, and as standardised APIs are still largely lacking, deploying similar services in each one of them can require significant efforts. The other issue is that of privacy, as cloud technology has been

[1] I.e. on servers in vast, remote data centres with high bandwidth and high reliability, operated and maintained by cloud service providers.

criticized in terms of the potential for cloud service providers to gain access over personal data.

An important novelty of the NoiseTubePrime approach is that personal software agents, guarding each user's privacy concerns, are outsourced to existing commercial cloud infrastructures. This relieves the user from the trouble to run and manage his own software agent. In NoiseTubePrime, personal agents are implemented as Web services, deployed in the cloud. Note that our architecture deals with both privacy and interoperability issues, as it does not disclose any personal data to the cloud service providers (all data is encrypted), while we demonstrate it over a heterogeneous environment of different service providers.

6.3 The NoiseTubePrime System

In order to remedy privacy concerns when creating collective maps, we propose a solution relying on a privacy-preserving distributed computation algorithm for generating grid-based maps for a target area and time-frame. Here we are inspired by the procedure used in the existing NoiseTube service [49, 175], though we stress that our ideas hold more generally for any participatory sensing framework where maps are produced in terms of aggregated measurements. However, for the sake of convenience we phrase our explanations below in terms of noise. At the basis of this algorithm lies a privacy-preserving cryptographic protocol for secure multi-party computations [214], which inputs are current or archived datasets of geolocated sound level measurements gathered by multiple users. Computation is executed by software agents running in the cloud.

Each user is represented by a personal, cloud-based software agent which acts as a mediator. Such an agent temporarily stores encrypted user data, takes part in the generation of participatory maps on the user's behalf, while also crucially preserving his privacy. All data transmitted by users to NoiseTubePrime agents is encrypted. In this way we overcome privacy issues related to how the cloud service provider might treat the data. Cloud deployment ensures that agents are online continuously and have adequate computational resources. In this way users do not need to operate agents on their mobile phones or personal servers.

An architectural diagram of the NoiseTubePrime system is shown in Figure 6.1. A typical scenario proceeds as follows. Suppose a particular entity, be it an authority or a citizens' organization, is interested to map a local area during a time span of interest (e.g. Friday night in a pub area). The initiative taker(s) then organize a measurement campaign in which a group of citizens use the NoiseTube system to gather geolocated sound level measurements in the specific geographical region and time period. The campaign proceeds through the following steps (Figure 6.2):

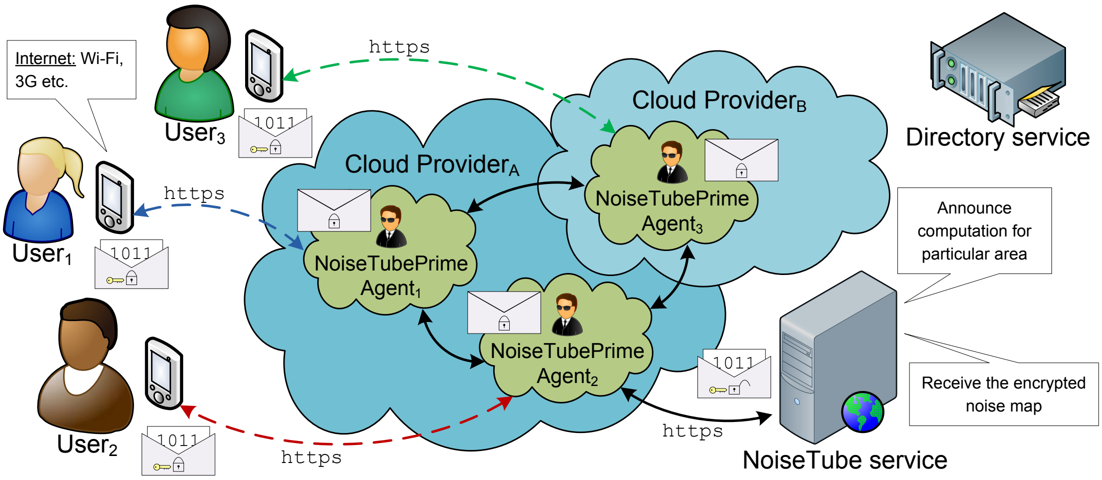


Figure 6.1: The general architecture of our system.

(a) To collect data about noise pollution users download the NoiseTube client application for their mobile device (e.g. from the Android Market)[1]. By default measurement data is stored locally on each user's mobile phone. Users set up their personal cloud agent, which registers to a *Directory Service* (DS) for the virtual network topology we deploy. Each user mandates his NoiseTubePrime agent to take part in existing or future campaigns, following a user-specified privacy policy.

(b) At some point in time the NoiseTube service announces a new campaign. Users are invited to participate through their agents, where agent policy dictates how agents should respond to such requests. For instance, agents may choose to participate to campaigns based on whether their owners plan to collect data in the specific region or not, or have collected relevant data before[2]. A deadline is set for all agents interested in contributing to register via the DS.

(c) When a user agrees to join a campaign (through the mediating agent), his mobile device inspects the user's local dataset for measurements that satisfy the given constraints, and uses this data to generate and encrypt the contribution of the user. The encrypted contribution is then handed over to the user's NoiseTubePrime agent in the cloud as soon as connectivity is available. This data upload operation takes place once per user and campaign/computation, and from that point on the user's mobile device is no

[1] Note that the NoiseTubePrime functionality is not yet incorporated in NoiseTube app that is currently available for download.

[2] Hence the computations may involve both past, current or future data.

longer involved in the computation.

(d) Each NoiseTubePrime agent manages a user's private data in the form of an encrypted map for the area of interest. Maps are encrypted with a public key that is either used across the system, is specific for the campaign in question, or for a specific time period. This public key is one of a public-private key pair that was generated by the NoiseTube service for this purpose; the private key is kept only by the NoiseTube service itself. Agents only use the encrypted data to participate in the generation of collective maps, when allowed to do so by user policy.

(e) After the announced deadline has passed the NoiseTube service initiates the distributed computation in the cloud. Note that agents from different users can be hosted on different cloud services. A list of the participating agents is retrieved from the DS. Agents are organized into a virtual network topology in which distributed computations take place. This may be a simple ring topology or something more sophisticated such as a tree for time-critical computations. One of the agents is selected to operate as the root-node for the specific computation via an appropriate request.

(f) The root-node coordinates a distributed computation that generates the specified noise map. This algorithm is detailed in the following Section 6.4.

(g) When agent interactions for the distributed computation are over, the NoiseTube service receives an encrypted aggregate noise map without any trace of the personal data of individual users. The NoiseTube service, using its private key, decrypts the received data to obtain the requested noise map, which is then made available accordingly. Interested parties can log on to the service to visualise and explore the resulting noise maps. A user's private information is not disclosed at any stage of the participatory noise mapping process.

6.4 The Privacy-Preserving Computation

Privacy-preserving computation is a large field which encompasses many challenges and tools. In short, a major direction in this field is to make efficient privacy-preserving applications for real applications. The theoretical foundations have been set since the seminal work of Yao on secure multi-party computations [214]. It is known that theoretically any distributed computation can be converted to a secure multi-party computation (which can be used within a privacy-preserving computations). The problem is that the general theoretical

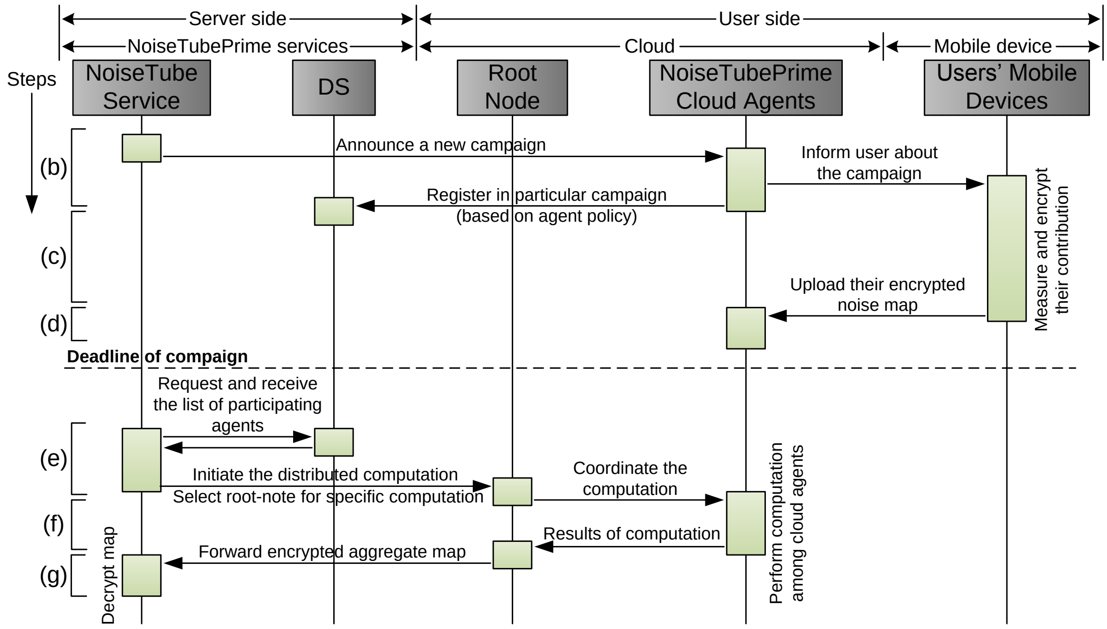

Server side
User side
NoiseTubePrime services
Cloud
Mobile device
Steps
NoiseTube Service
DS
Root Node
NoiseTubePrime Cloud Agents
Users' Mobile Devices
(b)
(c)
(d)
(e)
(f)
(g)
Announce a new campaign
Inform user about the campaign
Register in particular campaign (based on agent policy)
Measure and encrypt their contribution
Upload their encrypted noise map
Deadline of compaign
Request and receive the list of participating agents
Initiate the distributed computation
Select root-note for specific computation
Coordinate the computation
Perform computation among cloud agents
Results of computation
Forward encrypted aggregate map
Decrypt map

Figure 6.2: Interaction diagram of the NoiseTubePrime system.

solutions are computationally very demanding to such an extent that they are considered impractical for almost any practical application. The challenge is to build efficient solutions to real problems. Indeed, it is possible to derive optimized specialized solutions for particular problems. Such an example is the NoiseTubePrime application presented in this work. Moreover, NoiseTubePrime is also a new approach for outsourcing computations to the cloud in a privacy-preserving manner.

In the rest of this section, we describe the cryptographic protocol for calculating noise maps in a privacy-preserving way, which is implemented by the NoiseTubePrime agents. The communication between agents in our protocol is performed over secure sockets (SSL/TLS). The protocol is secure in the *Honest-But-Curious* (HBC) model (see Section 2.9.1 for details). We also assume that the cloud providers are honest-but-curious, and that they do not collude with NoiseTube to reveal users' data. In any case, the later threat can be addressed by deploying a threshold decryption scheme [46] .

6.4.1 The PrimeNoiseMap Problem Definition

The main goal of our work is to generate aggregate noise maps without violating the privacy of participants. The personal data which is needed for the computation are sound level measurements, associated with the user location and time-stamp, compatible with a particular campaign. To formalize the problem addressed in this work, we define the abstract *PrimeNoiseMap* problem for the privacy-preserving computation of participatory noise maps related to a particular measurement campaign. NoiseTubePrime is then an approach and associated system that solves the PrimeNoiseMap problem. An instance of the problem consists of:

- **N users** $u_1, u_2, \ldots, u_N$ and their geolocated, timestamped sound level measurements, where N is the number of participants that have expressed interest in the campaign.
- **Input:** The geographic area of interest (defined by minimum and maximum latitudes and longitudes) together with the cell dimensions (e.g. 40 m×40 m) of a grid covering that area, the time intervals of interest, the deadline for the distributed computation and a public encryption key.
- **Output:** The aggregate noise map with the required statistical information per grid cell. No personal data is disclosed during the computation.

We should note that the PrimeNoiseMap problem can be easily generalised to pertain to different kinds of measurements (e.g. temperature instead of sound level) and is thus relevant to other participatory sensing systems and scenarios.

6.4.2 The Distributed Protocol

We present a protocol for a privacy-preserving computation that solves the PrimeNoiseMap problem. The protocol does not disclose any locations, timestamps or sound level measurements of any participants; only the final aggregate noise map is revealed at the end of the computation.

Initially, the NoiseTube service announces that a specific campaign is planned. The announcement includes the campaign name, the area and time period of interest, the public encryption key and the response deadline.

When the campaign's deadline is reached each NoiseTubePrime agent, registered with the DS for that specific campaign, receives a request for the distributed computation as well as a corresponding deadline. Within the deadline, each agent communicates with the user's mobile device, and asks for any data that is relevant to the specific computation instance. The user is involved in the particular computation according to his privacy policy[1]. In case the user participates, the campaign proceeds according to the steps of Section 6.3. Data relevant for the

[1] User privacy policy can be quite sophisticated: User may contribute to all campaigns, even if there is no data, or only to certain ones selected manually with care. While such a broad spectrum of strategies for user policy can be supported by our system, it is not the focus of this work.

$$\begin{array}{cc} & \\ \text{Longitudes} & \begin{bmatrix} \begin{bmatrix} E_c(0) \\ E_s(0) \end{bmatrix} & \begin{bmatrix} E_c(0) \\ E_s(0) \end{bmatrix} & \begin{bmatrix} E_c(0) \\ E_s(0) \end{bmatrix} & \cdots & \begin{bmatrix} E_c(0) \\ E_s(0) \end{bmatrix} \\ \begin{bmatrix} E_c(0) \\ E_s(0) \end{bmatrix} & \begin{bmatrix} E_c(0) \\ E_s(0) \end{bmatrix} & \begin{bmatrix} E_c(0) \\ E_s(0) \end{bmatrix} & \cdots & \begin{bmatrix} E_c(0) \\ E_s(0) \end{bmatrix} \\ \begin{bmatrix} E_c(0) \\ E_s(0) \end{bmatrix} & \begin{bmatrix} E_c(5) \\ E_s(275) \end{bmatrix} & \begin{bmatrix} E_c(4) \\ E_s(226) \end{bmatrix} & \cdots & \begin{bmatrix} E_c(0) \\ E_s(0) \end{bmatrix} \\ \begin{bmatrix} E_c(0) \\ E_s(0) \end{bmatrix} & \begin{bmatrix} E_c(1) \\ E_s(68) \end{bmatrix} & \begin{bmatrix} E_c(8) \\ E_s(577) \end{bmatrix} & \cdots & \begin{bmatrix} E_c(2) \\ E_s(105) \end{bmatrix} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \begin{bmatrix} E_c(0) \\ E_s(0) \end{bmatrix} & \begin{bmatrix} E_c(0) \\ E_s(0) \end{bmatrix} & \begin{bmatrix} E_c(0) \\ E_s(0) \end{bmatrix} & \cdots & \begin{bmatrix} E_c(0) \\ E_s(0) \end{bmatrix} \end{bmatrix} \\ & \text{Latitudes} \end{array}$$

Figure 6.3: An example of an aggregate noise map.

campaign is encrypted at the client side in the form of a personal aggregate map using the campaign's designated public key. The structure of each personal aggregate map is shown in Figure 6.3. It covers the whole geographical area of the campaign, not only the sub-area the user traversed. Each grid element corresponds to an area for which two values are computed: the number of measurements in the particular area (E_c), and the sum of measurements (E_s), in our case sound levels in dB(A). By the announced deadline, each NoiseTubePrime agent has received the encrypted personal aggregate map of its user (as in Figure 6.3) in case connectivity was possible with the mobile device.

When the computation deadline has been met, the distributed computation between the participating personal agents can start. Initially, the NoiseTube service selects one of the participating agents as the root-node and sends it a request to commence the map computation. Then, the root-node agent begins the computation. The computation is performed across the agent topology which provides a virtual distributed computation platform. Each agent receives the aggregate map from its predecessor and multiplies each value pair (E_s,E_c) with its own corresponding value pair. Then the result is forwarded to the successor agent in the topology, which repeats the same steps. This computation exploits the additive homomorphic property of the Paillier cryptosystem [141], which is an asymmetric cryptographic algorithm for public key cryptography (see Section 2.6 for details). Figure 6.4 presents a simple ring topology, and illustrates how the computation responsibility is passed from each agent to its successor.

At the end of the computation, the aggregate encrypted map is returned to the root-node which then forwards it to the NoiseTube service for decryption. The NoiseTube service receives the aggregate map, decrypts it with the private key, and calculates the measurement average for each grid element by dividing $D(E_s)/D(E_c)$. This produces the decrypted aggregate noise map, where for each element of the grid we have calculated the average noise value and the number of measurements that support it.

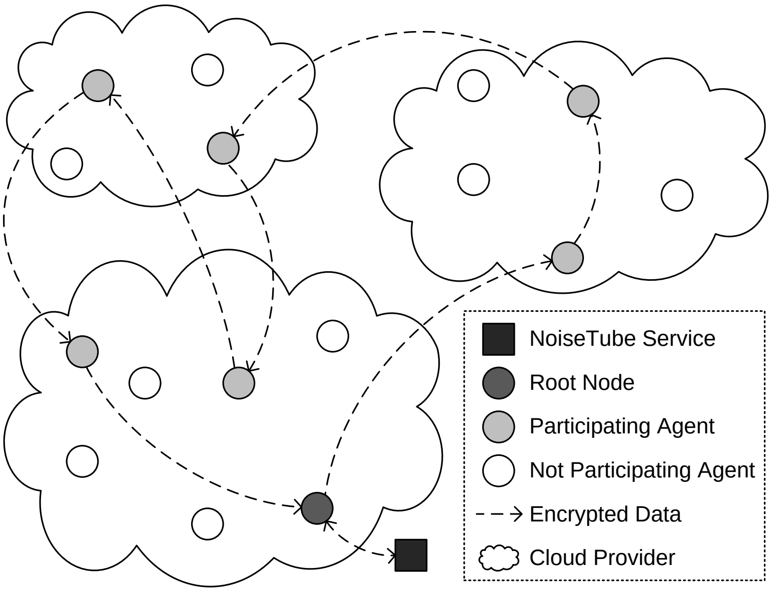


Figure 6.4: The NoiseTubePrime virtual distributed computation platform, where agents are deployed in different cloud services and form a simple ring network topology.

To avoid *side-channel privacy leaks*[1], a user can participate even without having data for a particular computation, by submitting a private encrypted map of zero values. In this way, not even his own agent is aware of the fact that the user does not have data for the particular computation. Similarly, when the mobile device cannot establish contact with the NoiseTubePrime agent, the agent may participate in the computation with a private encrypted map of zero values. In this way, the agent does not need to opt out from the ring, while the final result is the same and at the same time the privacy of its owner is protected.

The appropriate network topology depends on several factors like the number of participating agents, the requirements for tolerance on network failures and the

[1] Side-channel information leaks are information leaks that an adversary can obtain from the attributes of encrypted communications. Such side-channel leaks have been studied for example in [35].

limitations on the execution time. However, in this work execution time was not critical since maps need not be computed in real-time, and our experiments with several cloud services turned out to be fast enough. Consequently, we adopted a simple ring topology [121], and did not investigate this issue any further.

Our protocol ensures k-anonymity (see Section 2.1.3.1), where $k = N$ and N is the number of all participants that took part in the computation. Furthermore, the system could also support more statistical functions, such as covariance or frequency distribution for each grid element. Such capabilities are presented for example in Chapter 5.

With respect to fault tolerance, during the experimental evaluation the NoiseTubePrime system was remarkably stable and reliable, even when the distributed computation comprised cloud agents from three or four different cloud providers. The only variations noticed were that some agents of a specific provider occasionally needed a longer time to wake-up from their idle state; this issue was easily addressed by executing a wake-up round before the main computation. Overall, the behavior of the cloud services used by the agents was very reliable. This is probably not a surprise, due to the high availability of the cloud platforms provided by major players of the information technology field, like Amazon and Google. Nevertheless, a production-ready version of NoiseTubePrime should have some extra fault-tolerance features. For example, the directory server could simply skip a node of the logical ring topology if it does not respond within a predetermined time interval to its predecessor. We do not further elaborate on such issues related to implementation improvements.

6.4.3 Security

In this subsection, we demonstrate that the proposed protocol preserves the privacy of participants. Its security holds for the *Honest-But-Curious* (HBC) model (see Section 2.9.1) both for the users and for the cloud providers. In the NoiseTubePrime protocol, the information exchanged by agents is both aggregated and encrypted; thus, honest-but-curious party cannot infer any private information. The security of the Paillier cryptosystem (see Section 2.6) and its homomorphic property ensures that the personal data is not disclosed and cannot be associated with any particular user. To prove the privacy attribute of the protocol, we show that it satisfies the criterion of k-anonymity (see Section 2.1.3.1). The NoiseTubePrime protocol offers N-anonymity in the sense that the result computed at the end of the protocol cannot be attributed to any of the N participating agents, even if the list of participating users is known.

To summarize, the key security features of NoiseTubePrime protocol are:

- Each NoiseTubePrime agent receives an encrypted grid from the previous node. It cannot obtain information about the contents of the map, because the ciphertexts are encrypted with Paillier encryption.

- None of the cloud providers can obtain any information about the private content stored or computed by the agents, because all data and computations are in encrypted form.

- Each node alters the ciphertexts of the computation. Even the nodes that do not have data to participate multiply the ciphertexts with an encrypted number '0', which is the neutral element of the additive homomorphic property of Paillier. Again it is impossible to detect that an agent contributed with a grid consisting only of zeros.

- At the end of the protocol, only the aggregate noise map is revealed. As a result, no individual can be associated with his own measurements contributed in the computation. Consequently, the proposed protocol preserves k-anonymity for $k = N$, where N is the number of all participants that took part in the computation.

Our protocol can be extended to tolerate (at least some types of) malicious behavior. For example, a malicious NoiseTube service could collude with potential malicious cloud providers or user agents to obtain and decrypt intermediate results of the computation. This could possibly lead to the disclosure of the personal maps submitted by specific users. Such a threat can be effectively handled by deploying threshold decryption [46] for the decryption of the encrypted maps. Threshold decryption requires that the number of coordinating parties exceeds an appropriate threshold for decryption to be possible. We leave the comprehensive treatment of malicious user behavior within our application for future work.

6.5 Experimental Evaluation

To evaluate our approach, we developed a NoiseTubePrime prototype that implements the proposed privacy-preserving protocol for calculating participatory noise maps in the cloud. We used the implementation to set up an online demo of a NoiseTubePrime use case and to execute two sets of experiments for privacy-preserving noise map generation, showing that our protocol is able to reproduce noise maps correctly. We also analyse the performance of our protocol, in the context of realistic as well as artificial setups.

6.5.1 The NoiseTubePrime Prototype

The prototype consists of two parts: the mobile application, which runs on users' devices, and the NoiseTubePrime agent community, which runs on (a family of) cloud providers.

At the mobile device side, we implemented our solution on the Android platform[1], using Java. We have chosen Android because the existing NoiseTube system already supports it, and because it is currently the most popular smartphone platform – in terms of devices being sold [71]. However, there is no reason why our solution could not be ported to other mobile application platforms (e.g. Java ME/CLDC or Apple iOS). For convenience, our implementation uses the `BigInteger` class provided by Android (and by Java SE, but not Java ME/-CLDC), but on platforms that do not provide a similar type or class this could be implemented at the level of the application itself. NoiseTubePrime agents were also implemented in Java, as Java Web Servlets (WAR). They were deployed on several cloud infrastructure providers, namely Google App Engine, CloudBees, and Amazon EC2, without important differences in the implementation[2].

In the current stage of their development, NoiseTubePrime agents and the Android client application do not have all functionalities that were presented in the previous sections – in particular those parts pertaining to campaign definitions are currently lacking. However, we did fully implement and test the core of the protocol, i.e., the distributed homomorphic computations in a realistic setting. Our prototype supports both `http` and `https` for the communication among NoiseTubePrime agents and between the Android client and the cloud agent. The `https` protocol, which makes use of encrypted communication over secure sockets (SSL/TLS), is necessary to fully satisfy the security goals of NoiseTubePrime. In our experiments, however, for simplicity[3] we used `http`. Both the mobile and the cloud application implement the Paillier cryptosystem primitives for encrypting/decrypting data and performing secure calculations.

6.5.2 On-line Demonstration

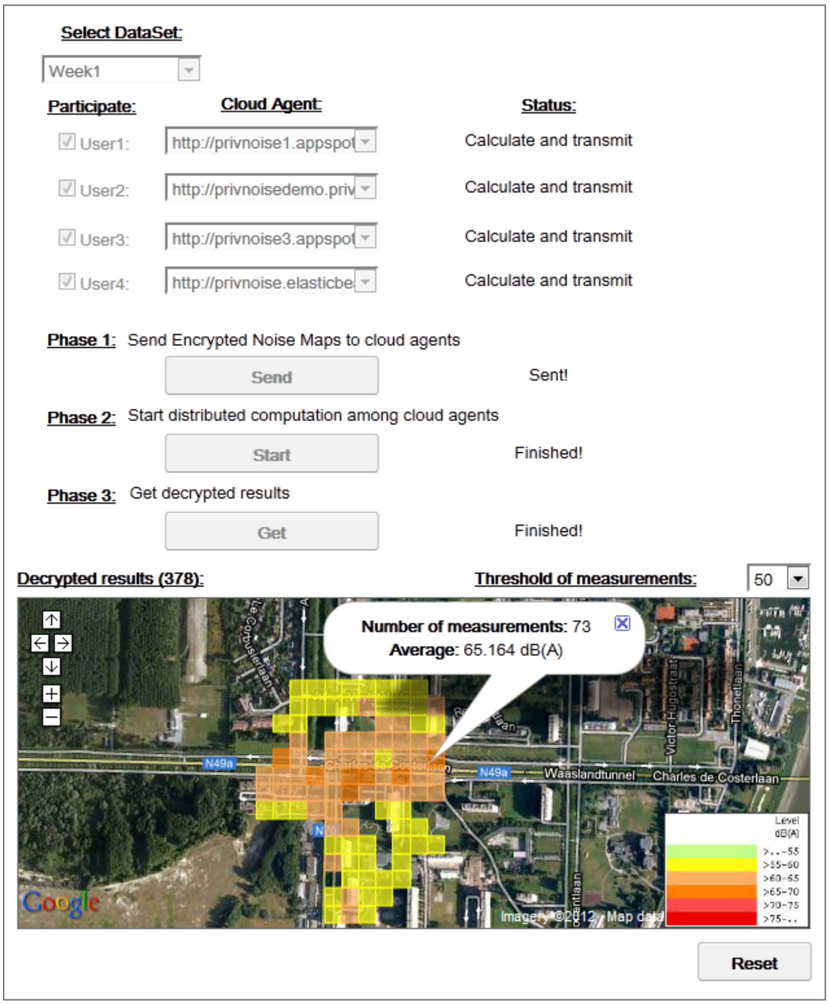
To demonstrate NoiseTubePrime functionality, we implemented an online demo (`http://polis.ee.duth.gr/NoiseTubePrime`), for a small scale experiment. As a proof of concept and at the same time a validation of correctness, our demo reproduces, in a privacy-preserving way, the results of a concrete noise

[1] In fact, we are targeting Android v2.2 "Froyo", or newer versions.

[2] *Google App Engine:* `http://appengine.google.com`, *CloudBees:* `http://www.cloudbees.com`, *Amazon EC2:* `http://aws.amazon.com/ec2`

[3] The configuration of https was a provider-specific task, which was complicated for some of the providers.

measuring campaign. For this purpose, we used real noise measurements collected in July 2010 by volunteering citizens in a 0.4 km×0.4 km area in the city of Antwerp in Belgium, as part of the "Ademloos experiment" set up by the BrusSense Team [49, 175].



Select DataSet:

Week1

Participate:	Cloud Agent:	Status:
☑ User1:	http://privnoise1.appspot	Calculate and transmit
☑ User2:	http://privnoisedemo.priv	Calculate and transmit
☑ User3:	http://privnoise3.appspot	Calculate and transmit
☑ User4:	http://privnoise.elasticbe	Calculate and transmit

Phase 1: Send Encrypted Noise Maps to cloud agents

Send — Sent!

Phase 2: Start distributed computation among cloud agents

Start — Finished!

Phase 3: Get decrypted results

Get — Finished!

Decrypted results (378): Threshold of measurements: 50

Number of measurements: 73
Average: 65.164 dB(A)

Reset

Figure 6.5: A screenshot of our demo.

Concretely, the campaign's goal was to map the chosen area during a peak-hour (7:30–8:30 am) and an off-peak hour (9:00–10:00 pm). To do this, four volunteers from the Antwerp-based Ademloos citizen action group followed a predefined measurement track twice daily for a week for each of the chosen hours. On the basis of these measurements (over 30,000 for each week) noise maps of the target area were produced. The standard NoiseTube approach is to analyse a collection of measurement tracks statistically to produce one single noise map. To do this the measured area is divided into smaller areas, the total set of measurements is divided over those areas, and a statistical analysis is carried out per unit area. In a final step color coded averages are mapped on each pertaining area. The resulting noise maps for this and other experiments can be found online[1].

In our online demo, we deploy four NoiseTubePrime agents which represent the four volunteers of the Ademloos experiment, and re-compute the same maps in a privacy-preserving manner using the NoiseTubePrime protocol. The demo is implemented with the Google Web Toolkit (GWT v2.4.0)[2] and consists both of a Web client and a server side application (servlet). The Web client is used to control the four mobile clients of the demo and visualizes the final results. The servlet initiates the computation, so that the four agents compute the aggregate map from the encrypted data of their users. Moreover, for the particular demo the servlet is used to simulate the four mobile devices. For this purpose, the original Java classes implementing the computational task of the Android application have been packaged within the server side servlet. The four agents have been tested on three different cloud providers (Google App Engine, CloudBees, and Amazon EC2). The key size of Paillier cryptosystem was chosen to be 512 bits. A screenshot of the demo during the execution of an experiment is shown in Figure 6.5. Each grid element corresponds to an area of 40 m×40 m.

The time needed by the NoiseTubePrime cloud agents for the multiplication of the encrypted maps, fluctuates between 875 to 1614 ms, which is acceptable for a grid of 21×18 elements. Note that this includes the time for receiving and transmitting the aggregate encrypted map, because we had no simple way to separate these two quantities from the cloud providers logs.

6.5.3 Computational Performance Evaluation

To evaluate the computational requirements of the NoiseTubePrime system for a wide range of realistic problem sizes and security parameters we conducted a large set of experiments using noise data which was generated artificially rather than actually measured. These comprised performance evaluation both of the mobile device-based computation and the distributed cloud-based computations. Naturally, the location trace of each simulated user and the number and the values of the corresponding noise measurements have only negligible impact on the computational requirement of the NoiseTubePrime application. Instead, the running time of the application is dominated by the size of the noise map, which is determined by the number of grid elements, and the size of the encryption keys. Thus, the computational requirements for processing the artificial data closely resembles the corresponding task on real data.

[1] http://www.brussense.be/experiments/

[2] http://code.google.com/webtoolkit/

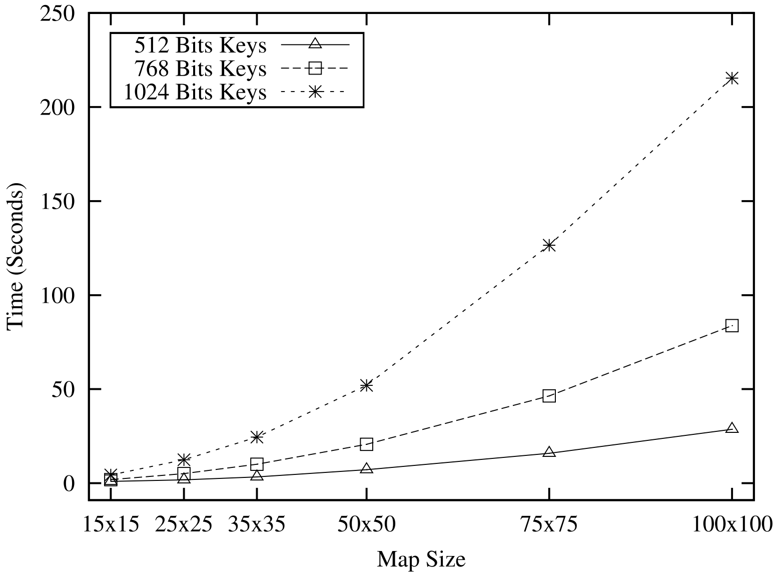


Figure 6.6: Execution times of encryption by the mobile client running on a Samsung Galaxy Note II.

The NoiseTubePrime solution comprises two main computational tasks (given that the noise measurements are collected through the existing NoiseTube application): The preprocessing step (encryption of the local map, step (c) in Section 6.3) which is executed locally on the mobile devices, and the distributed computation step (merging of the encrypted maps, step (f) in Section 6.3) which is outsourced to the personal agents located in the Cloud.

The first set of experiments concerns the computation task of the mobile devices, which have to prepare the encrypted map for each user. This task is highly parallelizable, and thus we can fully exploit the multi-core CPU architectures of modern mobile devices. In Figure 6.6 we show the execution times of the data encryption step that is performed by a mobile device for different map and public key sizes. In this experiment, we used the Android smartphone Samsung Galaxy Note II, which comes with a quad-core CPU ARM Cortex-A9 at 1.6 GHz and 2 GB of RAM. While execution times may run up to a few minutes, in particular for large maps, we note that the running time of this task is not critical for the NoiseTubePrime application, since it can be executed in batch mode as a background task on the mobile device at any time before the deadline of the specific campaign. It would also be possible to completely hide the running time of this preprocessing task, for example by incrementally building the local encrypted map during the noise sampling phase; a background process of the mobile device could immediately encrypt every new measurement and merge it with the local encrypted map.

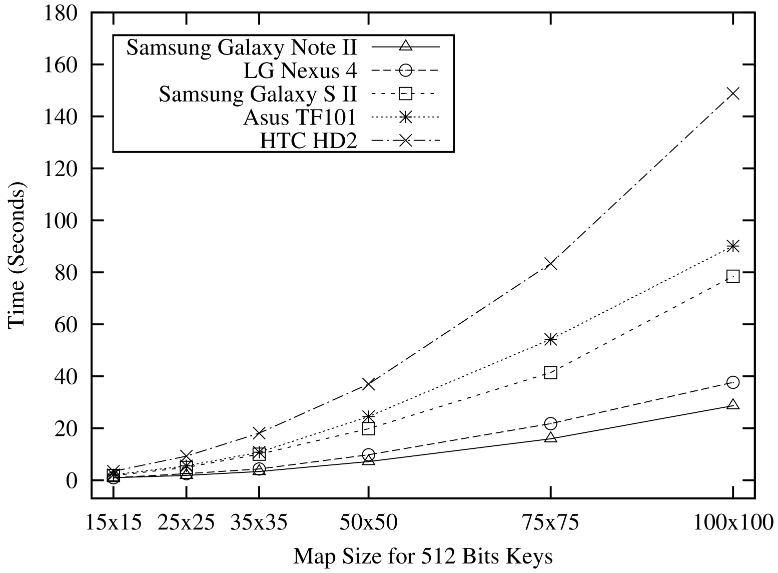


Figure 6.7: Execution times of encryption by the mobile client running on different mobile devices.

Next we examine how the execution time of the mobile application may vary between different mobile devices. In Figure 6.7 we show the execution time of the data encryption step that is performed on five different modern mobile devices, a Samsung Galaxy Note II (4-core CPU), a LG Nexus 4 (4-core CPU), a Samsung Galaxy S II (2-core CPU), an Asus TF101 (2-core CPU) and an HTC HD2 (single-core CPU). In these experiments the key size of the Paillier cryptosystem was fixed at 512 bits. Execution times do not vary substantially, at least not for the chosen set of devices. However, we do clearly see the effect of the parallel nature of the problem by the difference in single, dual and quad-core curves.

With respect to the Web-based component, Figure 6.8 shows how the execution time of the distributed computation in the cloud varies with respect to the number of NoiseTubePrime agents. In this simulation we deployed agents on a single cloud provider (Google App Engine), in a simple ring topology and with 512 bits Paillier keys. We use a single cloud provider in this set of experiments so as to minimize delays due to network transmission and as a result, we mainly capture the processing time of the cloud agents. Figure 6.8 shows that the total time for the distributed computation increases almost linearly with the number of agents, while the size (number of cells) of the map has only a small impact on the running time. In particular, for map sizes 15×15, 25×25 and 35×35, the delay per node is approximately 1.22, 1.26 and 1.31 seconds respectively.

We consider the execution times in all the above experiments to be entirely acceptable for noise mapping campaigns. There are of course feasible ways to

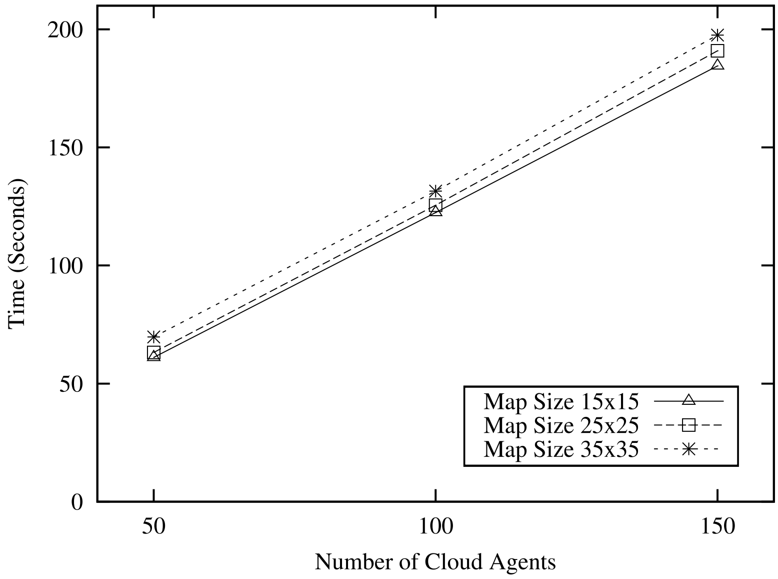


Figure 6.8: Execution times of computation in the cloud with artificially generated noise data.

improve the computational performance further, if necessary. The mobile device application can be further optimized by using more advanced programming techniques such as Renderscript [152] for the encryption process. Renderscript offers a high performance computation API for Android devices that gives the ability to run operations with automatic parallelization across all available processor cores of a device such as the CPU, GPU and DSP. On the other hand, the execution time of the cloud agent computation can be reduced by using a more efficient virtual topology, which would increase the concurrency of the distributed computation, by requesting more powerful resources from the cloud providers and by compressing the data that is transmitted during the distributed computation. However, we repeat that because both transmission of data as well as producing the agglomerated map are not required to proceed in real-time any delays that we find do not pose a concern.

6.5.4 A Realistic Use-Case of NoiseTubePrime

For our final experiment we set up what we consider to be a realistic use case of noise mapping and the NoiseTubePrime application. In this experiment we rely on a data set of real noise measurements gathered by 93 users in a 4 km^2 area in the city of Brussels, Belgium. The data set comprises 409.768 measurements at an average of 4.406 measurements per user, gathered in an uncoordinated way over a long period of time and including calibrated as well as uncalibrated devices. The largest user contribution consists of 76.337 and the smallest of 12 measurements[1].

With respect to privacy, we assume a mixed environment specified by user preference. In this way we consider two user types: “conventional” NoiseTube users, who contribute their data in plain format, and privacy-sensitive NoiseTubePrime users, who wish to share data only in a privacy-preserving way. Based on this assumption, we conducted a realistic set of experiments with a varying number of privacy-sensitive users.

With respect to the cloud platforms, we assumed heterogeneity too. Agents of the privacy-sensitive users run on three possible platforms: two commercial cloud providers, *CloudBees* and *Google App Engine*, and a server running in our lab. We performed a series of experiments with a gradually increasing number (up to 40 out of a total of 93) of privacy-aware users that deploy NoiseTubePrime agents, while the remaining users participate in the campaign as conventional NoiseTube users. In all experiments, the numbers of the agents assigned to each provider satisfy the ratio 1:1:2 for Google App Engine, CloudBees and the own server, respectively. For example, in the case of 40 privacy-aware users, we deployed 10 agents on the *Google App Engine*, 10 agents on *CloudBees*, and 20 agents on our own server.

We used a simple ring topology where the sequence of agents alternated between those residing on a cloud provider and on our own server. Note that this sequence of agents corresponds to a worst-case scenario with respect to the network load, since it generates the maximum possible network traffic for the particular mixture of agents.

Figure 6.9 shows how the execution time of the distributed computation varies with respect to the number of NoiseTubePrime agents and public key sizes. The map size is 100×100 elements and each grid element corresponds to an area of 40 m×40 m. In each experiment we also verified that the final aggregated noise map of the privacy-preserving and the conventionally computed results were identical.

We again find computation times which evolve linearly with the number of cloud agents. Moreover computation times are of a duration that is perfectly acceptable for a map of this size. Indeed, even without a privacy-preserving computation producing a noise map for this amount of measurements typically takes up a couple of minutes, and moreover we stress once more that producing noise maps is not something that is required to happen in real-time.

[1] Because of the heterogeneous nature of this dataset we have no guarantee about the quality of the resulting noise map and therefore choose not to include it here. However, that does not affect the usefulness of this dataset for the purpose of evaluating the NoiseTubePrime privacy-preserving map computation system.

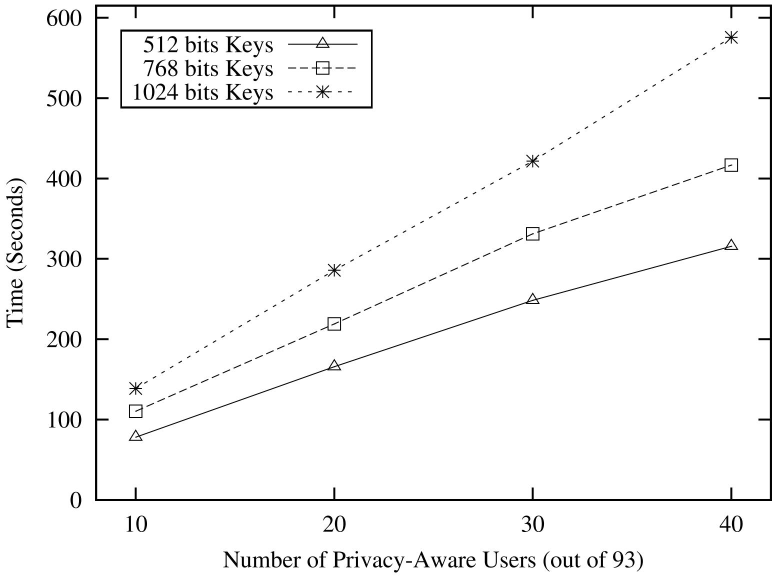


Figure 6.9: Execution times of computation in the cloud with real noise measurements.

6.6 Related Work/Discussion

6.6.1 On Privacy-Preserving Participatory Sensing

Privacy protection in mobile sensing systems [113] has recently attracted the interest of the scientific community. Because users of a participatory sensing system play an active role in the data collection process, it has been argued that they should also be actively engaged in privacy-related decisions [169], e.g. where and when to measure and what to share which whom. It has also been argued that, in order to protect user privacy and increase their negotiating power, data collection and data sharing should be decoupled by introducing a *personal data vault* that stores a user's data in a secure manner (i.e. encrypted), from which he can then selectively share subsets with various services or campaigns [66]. This idea is one of the ingredients of the NoiseTubePrime system presented above.

A comprehensive approach for opportunistic sensing is presented in [101]. The area under consideration is divided into appropriate regions (tessellation procedure), which have to be sufficiently large to preserve user anonymity. A similar approach is used in [37], where area cloaking is used to offer k-anonymity. The NoiseTubePrime approach is simpler and does not require any specific area division to preserve user privacy.

A very interesting related work is the PriSense system [168] which is based on a data slicing technique [89], and can offer functionality comparable to NoiseTube-

Prime for additive aggregation functions. However, the homomorphic encryption-based approach of NoiseTubePrime is simpler – no data scattering has to take place – and seems to be more general since homomorphic encryption is not limited to additive functions. Moreover, due to its simplicity, the NoiseTubePrime approach should be less error-prone.

In [18] the authors use advanced algorithmic techniques like sketches and approximate set cover to compute approximate statistic results in a privacy-preserving way. While theoretically interesting, in our opinion this approach is too complicated to be applied in practical, real-world settings, as opposed to NoiseTubePrime.

With respect to existing work, NoiseTubePrime is a simple but at the same time powerful approach for privacy-preserving participatory sensing, and to the best of our knowledge, the first operational privacy-preserving solution for participatory sensing. Moreover, NoiseTubePrime minimizes the requirements for the user by outsourcing the distributed computation to freely available cloud services[1]. NoiseTubePrime is based on plausible assumptions, is efficient for our purposes (as is shown in Section 6.5), has no requirements for special infrastructure and does not make any compromises in the quality of the computed results like cloaking or tessellation do. Similarly to PriSense, NoiseTubePrime is user-centred but, unlike PriSense, it can also support multiplicative functions by using an appropriate homomorphic cryptosystem.

6.6.2 On Using Cloud Agents for Preserving Privacy

The NoiseTubePrime software agents that we introduce in this work are based on the related idea of the Polis platform (Chapter 3) where each user is represented by a Polis agent. In a nutshell, Polis is a personal data management framework that abides by the following principle: every individual has absolute control over his personal data that reside only at his own side. The Polis agents constitute the backbone of the Polis architecture and run on the users' side; they are used to manage the personal data of a user, and provide controlled access at the entity's data. The service providers request personal data items of users from their personal agents. The agents provide the requested data if there is a corresponding license agreement (policies).

NoiseTubePrime agents are deployed in the cloud as Web services and are located on public servers, in contrast with Polis agents that are on the users' side. To avoid the obvious disclosure of personal data to cloud providers, only

[1] The requirements for computation and networking per user are very low and are served for free by several independent cloud providers. In addition, there is no reason why these needs would not continue to be served for free or, in the worst case, at a very low cost in the foreseeable future.

data in encrypted form is uploaded to the agents. The encryption of data solves both security and privacy issues that we have in clouds. Furthermore, NoiseTubePrime agents host only data that is destined for particular computations, and not all personal data, such as Personal Identifiable Information (PII), of users. For common security requirements like authentication of users who have the right to add personal encrypted data and to participate in a specific distributed computation, we can use standard security measures. Finally, the main reasons why we deploy our agents in the cloud are:

- Agents have to be online continuously during the distributed computation;
- Several cloud computing providers offer free services for low computational and bandwidth requirements, which are sufficient for our goals;
- The network connectivity offered by cloud infrastructures is fast and reliable, unlike mobile data connections;
- The cloud offers scalable computational resources.

6.7 Conclusion

This work presents a novel, privacy-preserving architecture for the creation of participatory noise maps, called NoiseTubePrime and built on top of the NoiseTube system [123, 175]. NoiseTubePrime allows aggregate noise maps to be generated from data collected by multiple users without disclosing their location traces. The resulting maps are exactly the same as those generated with conventional grid-based aggregation methods, as applied in [49]. However, our system allows users to preserve their privacy, and thus contributes to the realization of trustworthy computing systems. Our approach implements the '*fair information principle*' as privacy is respected when information is collected [132]. The protection of privacy is achieved by using cryptographic techniques and performing a distributed computation within the network of agents. The distributed computation is performed on encrypted data and no personal data items are disclosed to anyone, including the cloud service providers, at any time. Finally, we developed a prototype implementation and presented experimental results using a heterogeneous set of commercial cloud services, confirming the viability and the efficiency of the proposed solution.

Key features of the NoiseTubePrime system include:

- Accurate aggregate statistics are computed using the private measurement data of each user, while at the same time the privacy of the participating users is preserved: No location/time data is disclosed.

- Outsourcing the NoiseTubePrime agent to the cloud relieves the user from the trouble to run and manage his own software agent and to maintain permanent Internet access. The computational and networking requirements of each software agent are low and are (currently) provided without any cost by the various cloud service providers we used in our experiments.

- The main task of the NoiseTube service is to decrypt the final encrypted noise map that is the result of the distributed cloud computation. Hence the computational work of the NoiseTube service is independent of the number of participating users, making NoiseTubePrime a decentralized system that theoretically can be scaled to handle very large numbers of users.

The NoiseTubePrime architecture strikes a sound balance between providing secure, yet straightforward, privacy protection for those contributors that want or require it, while maintaining transparency for those that do not. We believe that the privacy-preserving solution presented in this work can make participatory sensing platforms like NoiseTube more suitable for large-scale (e.g. city-wide) deployments, in which the privacy concerns of individual contributors are expected to be significantly higher than in previous small-scale noise mapping campaigns [175, 49] – due to higher numbers of participants, weaker (or absent) acquaintance and trust relationships, and possibly the involvement of authorities.

Our future plans are to develop a stable and more complete version of NoiseTubePrime and demonstrate its use for real-world campaigns, also extending the platform towards more statistical parameters. To accomplish this we have to extend our prototype implementation. Roughly, the user-side Android application has to be to enriched with features supporting user policies and campaign participation and the resulting code has to be integrated into the existing NoiseTube Mobile for Android application[1], and then released to the public. The prototype NoiseTubePrime servlet, which implements the server side of our application, has to be extended with auxiliary functionalities for public key management, campaign management and a DS for supporting the distributed computations[2]. Current and future NoiseTube users should be oblivious to these privacy extensions insofar as possible. In the future a user study could be set up to evaluate the overall usability of the solution in different contexts.

Last but not least we should stress that the proposed architecture for privacy-preserving sharing, transmission, processing and management of sensitive (spatial) data is independent of the noise domain, and can thus potentially be applied

[1] https://play.google.com/store/apps/details?id=net.noisetube.

[2] We note that the NoiseTube system as it stands is currently undergoing a transition to support campaigns. However, the privacy extensions proposed in this article are not yet included.

in other participatory sensing systems. The only constraint is that the parameters of interest can be computed with efficient homomorphic cryptosystems.

Chapter 7

Privacy-Preserving Television Audience Measurement using Smart TVs

7.1 Introduction

Television is nowadays one of the dominant mediums for information and entertainment. Information about television audiences provide valuable insights to broadcasters and the advertising industry on recent trends. Television Audience Measurement (TAM) systems aim at calculating qualitative and quantitative TV audience measurements. For example, Nielsen[1], one of the leading companies in the field of media audience measurement, uses measurements from approximately 18,000 households (February 2010) in the U.S.A. to create the estimates the TV networks use. The viewer data is collected by the special metering equipment installed on the TV sets of the participating households; this data is transferred directly to the company's servers. Apparently, the above measurement process raises important privacy issues for the participants. A person's viewing record can reveal sensitive information about the person's preferences and habits. A privacy-preserving method for creating accountable TAMs is needed, in order to utilize television ratings information, while protecting the participants' privacy. Additionally, since TAM data bring important financial benefits to the industry (broadcasters, advertising companies, commercial products and more), some kind of fair financial compensation should be offered to the users that provide their viewing records.

Advances in communication and entertainment technologies have recently led to the introduction of Smart TVs, which are expected to be the next logical step in

[1] http://www.agbnielsen.net

television technology. The term *Smart TV*[1] is used to describe the current trend of integration of the Internet and Web 2.0 features into modern television sets and set-top boxes, as well as the technological convergence between computers and these television sets/set-top boxes. These new devices most often also have a much higher focus on online interactive media, Internet TV, over-the-top content, as well as on-demand streaming media, and less focus on traditional broadcast media like previous generations of television sets and set-top boxes always have had. For example, they allow viewers to check YouTube, Facebook, and other popular websites while watching TV. In essence, Smart TVs bring the Internet into the living room. As the technology improves, many of these sets are becoming as capable as standard computers when it comes to web browsing and even Internet video. This combination of traditional TV functionality with computational and networking capabilities, makes Smart TV technology capable of a whole new set of applications.

Modern Smart TVs from companies like Samsung and Sony have two major advantages over DVRs and set-top boxes. Many newer sets actually have full browsers, which means that content isn't restricted. Additionally, more recent initiatives, like the Google TV[2] platforms, also come with app stores, similar in design to the app store that Apple introduced for the iPhone. Apps add functionality to a Smart TV with video games, sports updates, specialized channels and much more. These apps are cheap or free and have already been used in exciting ways to make television viewing more interactive.

In this work, we present PrivTAM, a system for privacy-preserving TAM using Smart TV technology. The core of PrivTAM is a privacy-preserving cryptographic protocol, which accepts as input the viewing records from users' Smart TVs and performs secure multi-party computations [214] to calculate the TAMs. PrivTAM satisfies the following requirements for a reliable, privacy-preserving, TAM:

- Privacy - all records must be secret.
- Completeness - all valid records must be counted correctly.
- Soundness - dishonest records cannot disrupt the measurement process.
- Unreusability - no user can submit their record more than once.
- Eligibility - only those who are allowed to participate can submit their records.
- Verifiability - nobody can falsify the result of the TAM process.

The above requirements are a subset of the typical requirements of e-voting systems [148] and thus, our system borrows techniques from this field [17, 105, 111]. In addition, functionalities for the financial compensation of the participants

[1] http://en.wikipedia.org/wiki/Smart_TV

[2] http://www.google.com/tv/

are supported. The computations of PrivTAM are performed between software agents, which are located at the participants' Smart TVs, and a Trusted Authority (TA). Each Smart TV has an agent which continuously collects the viewing records of its owners.

The Trusted Authority coordinates the computation, verifies the validity of the records, collects the encrypted results and provides the compensation to the participants. This process is performed using encrypted viewing records, hence the record contents are never revealed to the Trusted Authority. Finally, we develop a prototype implementation and perform experiments that confirm the feasibility of the approach.

Some of the advantages of our approach in comparison to traditional TAM systems are:

- Preserving the privacy of participants' viewing records.
- More reliable measurements can be achieved, since a practically unrestricted number of participants can produce the PrivTAM results.
- Supports fine grained measurements which can be automatically calculated in small time intervals as well as specific one-time queries.
- Reducing the cost for conducting a TAM. No specialized equipment is required and only the participants in a calculation need to be compensated.
- Supporting measurements using records from any Internet-enabled broadcasting medium (e.g., Broadcast TV, Cable TV, IPTV and Satellite TV).

Our solution requires Smart TV's to have permanent Internet access, a requirement which is satisfied by default. Moreover, the computational and networking requirements of PrivTAM can be easily fulfilled by modern embedded Android-based platforms.

7.2 Related Work

To our knowledge this is the first attempt at creating a privacy-preserving TAM system, particularly one that supports an arbitrarily large amount of participants. In general, TAMs are products of aggregation operations and therefore our work is related to common privacy-preserving aggregation systems. For example, in [168], privacy-preserving data aggregation in people-centric urban sensing systems is discussed. A market for personal data, supporting anonymous data aggregation operations is presented in [6]. The economic aspects of personal privacy are discussed in [188, 3]. The fact that individuals need to be in control and be compensated when their personal information is used for commercial purposes, is discussed in [109, 179]. The sensitivity of the viewing records is stressed by both the Video Privacy Protection Act [185] and the Cable TV Privacy Act [184].

Overall, we consider that PrivTAM lies between privacy-preserving aggregation systems and e-voting systems, offering verifiable, privacy-preserving, aggregation functionalities. Additionally, PrivTAM takes into account the economic aspects of privacy and supports compensation functionalities for the measurement subjects.

7.3 The PrivTAM System

An overview of the PrivTAM system architecture, built on top of Smart TV technology, is shown in Figure 7.1. The main parts of the architecture are the participating Smart TVs, the Television Audience Measurement Service (TAM Service) and the Trusted Authority (TA). Every Smart TV contains a software agent that collects and stores its viewing records and maintains a set of demographic elements, such as gender, age and educational level of the viewers. The agent manages the viewers' personal data, provides controlled access to the data, and has the ability to participate in distributed protocols and computations.

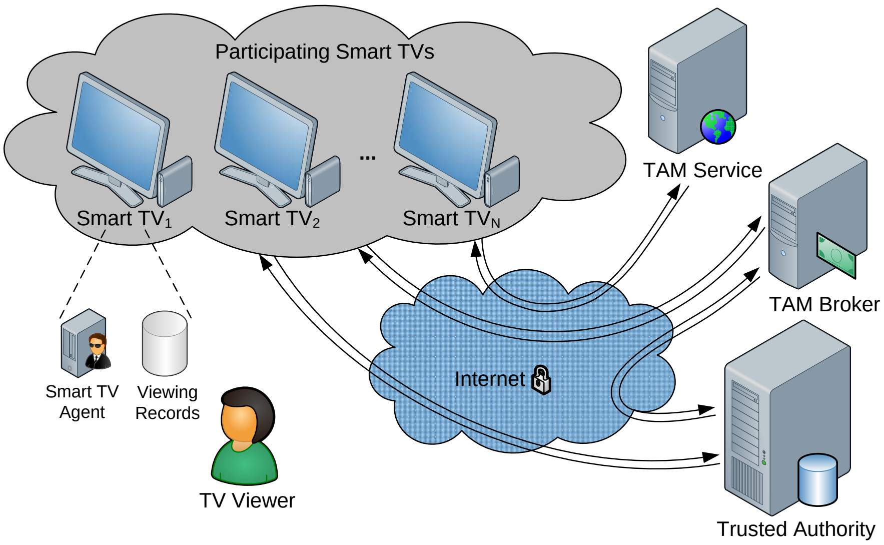


Figure 7.1: The general architecture of our system.

The TAM Service collects the measurements and is responsible for coordinating the distributed key generation [138] for the public-key cryptosystem between itself and a group of L TV agents. These L agents are chosen with a verifiable random selection [64] and participate in both the public-key creation phase and the decryption of the results phase. The selection of the set of L agents can be repeated on a regular basis, for example every one or few days. With each execution a new set of agents will have the responsibility of the procedure of distributed key generation. The verifiability of the agents' random selection can be found in Section 7.4.2.

The TA is responsible for coordinating the PrivTAM computation process. Time is divided into consecutive intervals, and for each interval, an aggregate result is periodically calculated using input from the participating Smart TV's. Many conventional TAM systems, for example, report audience results for intervals of 15 minutes. PrivTAM can obtain equivalent results by using the same interval value. The TAM Broker is used to facilitate the (optional) payment functionality described in Section 7.4.2. Each Smart TV agent encrypts its viewership vector with the public-key of the measurement and sends it to the TA for verification. The verification of the encrypted vector requires a cryptographic protocol that is described in Phase 2 of Section 7.4.2. Following a successful verification, the TA adds this vector to the current encrypted result of the measurement. The final encrypted TAM is transmitted to the participants in the distributed key generation for decryption and the result is revealed.

7.4 The PrivTAM Protocol

In this section, we present the cryptographic protocol used in PrivTAM. The communication between the entities in our protocol is performed over secure sockets (SSL/TLS) with both server and client authentication enabled. Our protocol is secure in the Malicious Model, assuming that the TA and the TAM Service are Honest-But-Curious (HBC) (see Section 2.9.1). During the calculation the actual users' personal data are not disclosed in any stage of the process, but only the final results are revealed at the end. Regarding privacy, the question we address is how to achieve privacy of type (b) (see details in Section 2.1.4), that is, how to compute the TAMs without pooling the viewing records, and in a way that reveals nothing but the final TAM results of the computation.

7.4.1 Problem Definition

We define the PrivTAM problem for verifiable, privacy-preserving TAM. A PrivTAM problem instance consists of:

- **N Smart TVs** - $TV_1, TV_2, \ldots, TV_N$ and the viewing records of their owners.

- **Input:** The viewership vector of each owner.
- **Output:** The TAM for the participating viewership vectors.

We assume that one viewership vector is submitted per Smart TV. We do not consider user identification issues within family members, as this is an existing issue in TAM systems and is out of the scope of this work. In current TAM systems the recording equipments is able to identify the members of the family who watch TV by using dedicated remote controls per family member. This could also be achieved on the Smart TV application, adding a simple interface on the device where the user can select who they are. Alternatively advanced techniques for user identification can be used, like face recognition.

7.4.2 Outline of the Computation

The computation consists of three main phases. In Figure 7.2, the participating entities of each phase are illustrated. The full descriptions of the three phases are given in the following paragraphs.

- In Phase 1 a distributed key generation for a Threshold Paillier Cryptosystem is performed.
- In Phase 2 the privacy-preserving TAM calculation takes place.
- In Phase 3 the final encrypted TAM is forwarded for decryption and the result is announced.

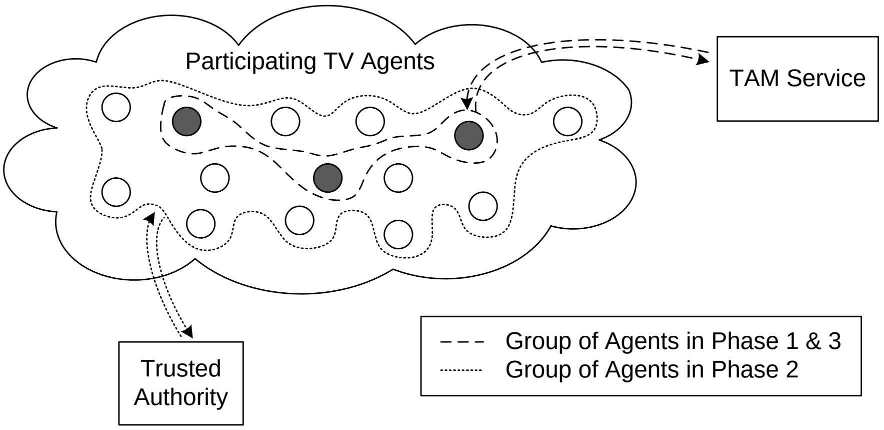


Figure 7.2: Illustration of protocol participants.

Phase 1. During Phase 1 the TAM Service selects an L-sized subset of the N-sized set of all the participating TV agents with a verifiably random procedure.

An example of a publicly verifiable random selection process is described in [64]. With the last phrase, we mean that the L agents are really randomly selected and the TAMS can prove that they are really random agents. This technique prevents the TAM Service from making a biased or impeachable group selection. Then, the TAM Service and the L selected TV agents execute a cryptographic protocol for the distributed key generation of the Threshold Paillier Cryptosystem [46]. We use the following Threshold Decryption Model, which is an adaptation of the corresponding definition in [17] to our needs, so that the distributed key generation can be performed without a trusted dealer [138]. Avoiding a trusted dealer makes the PrivTAM system more relevant for the TAM application context. This approach makes the Threshold Paillier Cryptosystem feasible in practice.

Definition 6 (Threshold Decryption Model) *In a threshold cryptosystem, instead of merely decrypting the encrypted message, we use n parties P_i with their secret keys, so that at least t parties, where $t \leq n$, are required to decrypt the message. The decryption process includes the following players: a combiner (can be one of the n parties), a set of n parties P_i, and users. We consider the following scenario:*

- *In an initialization phase, the parties use a distributed key generation algorithm to create the public key PK of their private keys SK_i. Next the parties publish their verification keys VK_i.*
- *To encrypt a message, any user can run the encryption algorithm using the public key PK.*
- *To decrypt a ciphertext c, we forward c to the combiner and n parties. Using their secret keys SK_i and their verification keys VK_i, each party runs the decryption algorithm and outputs a partial decryption c_i with a proof of validity of the partial decryption $proof_i$. Finally, the combiner uses the combining algorithm to recover the cleartext, provided that at least t partial decryptions are valid.*

In PrivTAM, we use the Paillier public key generated in Phase 1 for the encryption of the viewership vectors and utilize the Pailler Cryptosystem's homomorphic property in Phase 2. In addition, we specify that t is equal to n in our Threshold Decryption Model, meaning that all the parties are required to decrypt a message. Setting $t = n$ is important to ensure that the final result cannot be decrypted without the active participation of the TAM Service. Phase 1 should be repeated occasionally, to renew the keys and the set of L agents.

Phase 2. During this phase, the TA coordinates the voting process, and collects and verifies the encrypted viewership vectors of the participants. Upon successful verification, the TA adds the submitted viewership vector to the current TAM result, and sends the compensation to the participant.

In detail, Phase 2 begins with the TV agents that hold viewing records for the particular time period, creating their viewership vectors (Figure 7.3). Each such vector is submitted to the TA for the verification. The verification process is based on a zero-knowledge proof that an encrypted message lies in a given set of messages [17]. This way, when encrypting a message, it is possible to append a proof that the message lies in a public set $S = \{m_1, \cdots, m_p\}$ of p messages without revealing any further information. This proof is described in detail in Section 7.5.

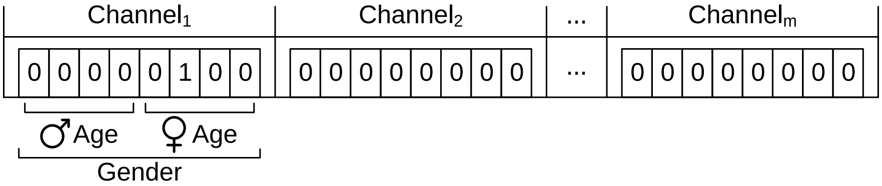


Figure 7.3: Example of a viewership vector.

In Figure 7.3, the viewership vector for m TV channels is illustrated. The vector section for each channel consists of a number of ciphertexts, which result from the number of demographic elements used in the vector. Every demographic element is respectively represented by some ciphertexts that its number depends on the number of parts that is separated the demographic element. Additionally, in our case the two demographic elements are dependent elements because we want to know the ages of each gender and they are represented as one element. Of course, it is possible to have and independent elements in our measurements. In our example, these elements are the age group and the gender of the viewer. The gender categories are male and female and the age groups are "$Age_1 \leq 24$", "$25 \leq Age_2 \leq 40$", "$41 \leq Age_3 \leq 55$" and "$Age_4 > 55$". Consequently, a combination of 8 ciphertexts is created to represent these elements. In order to indicate the channel the viewer was watching, the representation of their demographic elements are added to the viewership vector section for the corresponding channel. For example, in our case the gender is female, the age is between 25 and 40 years old and she was watching $Channel_1$. All the values in the viewership vector lie in the public set $S = \{0, 1\}$ and they are encrypted using the public key that is generated in Phase 1. Every participant should prove that their vector is valid so that the TA can avoid any malicious behavior from them. More specifically, the participants should prove that:

1. Every ciphertext in the viewership vector should lie in the set $S = \{0, 1\}$.

2. The multiplication of the ciphertexts in every channel should lie in the set $S = \{0, 1\}$.
3. Finally, the multiplication of all ciphertexts in the viewership vector should equal to "1". This means that the participant was watching TV.

The multiplication of the ciphertexts in the above proofs utilizes the additive homomorphic property of the Paillier Cryptosystem [141] (see details in Section 2.6). In Table 7.1, you can see in details the various levels of the above proofs.

	$Channel_1$	$Channel_2$	$\dots$	$Channel_m$
Ciphertexts of viewership vector	$E_{1,1}$ $E_{1,2}$ $\dots$ $E_{1,\delta}$	$E_{2,1}$ $E_{2,2}$ $\dots$ $E_{2,\delta}$	$\dots$	$E_{m,1}$ $E_{m,2}$ $\dots$ $E_{m,\delta}$
1^{st} Level Proofs	$P_{1,1}$ $P_{1,2}$ $\dots$ $P_{1,\delta}$	$P_{2,1}$ $P_{2,2}$ $\dots$ $P_{2,\delta}$		$P_{m,1}$ $P_{m,2}$ $\dots$ $P_{m,\delta}$
Multiplication of ciphertexts per channel	$E_1 = \prod_{d=1}^{\delta} E_{1,d}$	$E_2 = \prod_{d=1}^{\delta} E_{2,d}$	$\dots$	$E_m = \prod_{d=1}^{\delta} E_{m,d}$
2^{nd} Level Proofs	P_1	P_2		P_m
Multiplication of all ciphertexts		$E = \prod_{ch=1}^{m} \prod_{d=1}^{\delta} E_{ch,d}$		
3^{rd} Level Proof		P		

1^{st} and 2^{nd} Level of ZKPs: Ciphertexts should lie in the set $S = \{0, 1\}$.
3^{rd} Level of ZKP: Ciphertext should equal to "1".

Table 7.1: The levels of proofs for a viewership vector.

Once the viewership vector is confirmed by the TA, the vector is multiplied, using the homomorphic property, with the current TAM result. More specifically, every ciphertext of the viewership vector is multiplied with the corresponding ciphertext of the current TAM by taking advantage of the homomorphic property. We assume that the TA only logs the participants in a measurement in order to ensure unreusability of the vectors. However, even if the vectors were stored, the TA would not be able to reveal their contents, unless all the participants of the threshold decryption are malicious and collude towards this purpose. The final result of Phase 2 is the encrypted TAM of the particular query, which ensures k-anonymity (see Section 2.1.3.1), where $k = N$ and N is the number of all participants who take part in the TAM.

Payments in PrivTAM. The PrivTAM system can support functionalities for the compensation of participants, either in the form of financial payments or in the form of vouchers or points. The requirement for a participant to be compensated is that they provide a valid viewership vector to the computation. After the successful verification of a participant's viewership vector, the TA sends the compensation to the participant.

In case of financial compensation, the payment scheme within PrivTAM needs to be efficient enough to facilitate large numbers of small amount payments, without entailing substantial transaction costs. Therefore, we draw techniques along the lines of micropayments, as proposed in [161]. The main actors in micropayment schemes are Brokers, Vendors and Users. A User becomes authorized to make micropayments by the Broker. A Vendor receives micropayments from authorized users and redeems them through the Broker. Relationships of Users and Vendors with the Broker are long term. In PrivTAM, the Smart TV owners can act as Vendors and the TA can act as a user making micropayments. The TAM Broker is introduced in the architecture to facilitate the payments (Figure 7.1). A micropayment scheme suitable for PrivTAM is Payword, presented in [161]. Payword is a credit-based scheme, based on chains of hash values (called Paywords) and the Broker does not need to be online in order for a transaction between a User and a Vendor to take place.

Alternatively, non-monetary compensation, including points that can be redeemed with participating companies, can be offered to participants. The amount of compensation for each PrivTAM calculation is fixed for simplicity, but methods for providing different pricing could be introduced into the system. It is important to stress that the collected points of each participant are not recorded in a profile by a centralized service, but are kept at the participant's side.

Phase 3. In Phase 3, the final encrypted result of Phase 2 is forwarded to the L selected TV agents of Phase 1 and the TAM Service. The L agents perform partial decryptions and send the results to the TAM Service which acts as the final participant and combiner of the threshold decryption. This way, only the TAM Service can see the final result of the calculation, which is acceptable if the TAM Service is considered honest and reports accurately the decrypted result. In order for the PrivTAM calculation to be protected from inaccurate reporting of the results from the TAM Service, a verification mechanism can be introduced to validate the announced results. This verification could be accomplished by using multiple combiners in the threshold decryption, to confirm the announced results from the TAM Service.

7.5 The Protocol's Security

In this section, we show that the PrivTAM protocol achieves the requirements described in the introduction, i.e., privacy, completeness, soundness, unreusability, eligibility, and verifiability. The security model holds for Malicious viewers, with the assumption that the TA is Honest-But-Curious (HBC) and the TAM Service is the final receiver of TAM results. Malicious users can submit any value as input to the computation or even abandon the protocol at any step. See the description of the Malicious Model given in Section 2.9.1 or the more detailed description in [108, 78]. An Honest-But-Curious party (adversary) follows the prescribed protocol properly, but may keep intermediate computation results, e.g. messages exchanged, and try to deduce additional information from them other than the protocol result.

In the PrivTAM protocol described above, the information exchanged by TV agents is encrypted with the Paillier cryptosystem [141], which is known to offer Semantic Security [82], that is, it is infeasible for a computationally bounded adversary to derive significant information about a message (plaintext) when given only its ciphertext and the corresponding public encryption key. More details about the semantic security of Paillier scheme (known and as Decisional Composite Residuosity Assumption (DCRA)) you can find in Section 2.6. Furthermore, the security analysis of the Threshold Version of Paillier Cryptosystem is described in [138].

The security of the Threshold Paillier cryptosystem and its homomorphic property ensures that the viewing records are never disclosed and cannot be associated with any particular participant. To prove the privacy attribute of the protocol, we show that it satisfies the criterion of k-anonymity (see Section 2.1.3.1). In the context of this work, k-anonymity means that no less than k individual users can be associated with a particular personal viewing record.

The following zero-knowledge proof illustrates the steps of the verification process in Phase 2. The security of this zero-knowledge proof is shown in [17].

Proof that an encrypted message lies in a given set of messages [17]. Let N be a k-bit RSA modulus, $\mathcal{S} = \{m_1, \cdots, m_p\}$ a public set of p messages, and $c = g^{m_i} r^N \mod N^2$ an encryption of m_i where i is secret. In the protocol, the prover P convinces the verifier V that c encrypts a message in $\mathcal{S}$.

1. P picks at random ρ in $\mathbb{Z}_N^*$. He randomly picks $p-1$ values $\{e_j\}_{j \neq i}$ in $\mathbb{Z}_N$ and $p-1$ values $\{\upsilon_j\}_{j \neq i}$ in $\mathbb{Z}_N^*$. Then, he computes $u_i = \rho^N \mod N^2$ and $\{u_j = \upsilon^N (g^{m_j}/c)^{e_j} \mod N^2\}_{j \neq i}$. Finally, he sends $\{u_j\}_{j \in \{1, \cdots, p\}}$ to V.

2. V chooses a random challenge e in $[0, A[$ and sends it to P.

3. P computes $e_i = e - \sum_{j \neq i} e_j \mod N$ and $\upsilon_i = \rho r^{e_i} g^{(e - \sum_{j \neq i} e_j) \div N} \mod N$ and sends $\{\upsilon_j, e_j\}_{j \in \{1, \cdots, p\}}$ to V.
4. V checks that $e = \sum_j e_j \mod N$ and that $\upsilon_j^N = u_j (c/g^{m_j})^{e_j} \mod N^2$ for each $j \in \{1, \cdots, p\}$.

We note that r is the random number which was used for the encryption of message m_i and $a \div b$ is the quotient in the division of a by b. According to Theorem 2 of [17], it holds that t iterations of the above protocol is a perfect zero-knowledge proof (against an honest verifier) that the decryption of c is a member of $\mathcal{S}$, for any non-zero parameters A and t such that $1/A^t$ is negligible.

The main security features of the protocol are discussed below. Note that these features can also expressed in a more way, for example by using the tools of [14]. In Table 7.2, we show the summary of the data items (encrypted or not) that are generated with our approach combined with the participating entities. The main security features are:

- The TA cannot obtain information about the contents of the viewership vector ($\checkmark_{(1)}$ in Table 7.2) and the encrypted TAM result ($\checkmark_{(4)}$), since the ciphertexts are encrypted with the Paillier encryption.
- In case the TA stores the viewership vector ($\checkmark_{(1)}$), the contents cannot be revealed unless all the participants in the threshold decryption are malicious and collude towards this purpose.
- The participants cannot submit invalid viewership vectors ($\checkmark_{(1)}$) and disrupt the calculation, due to the verification process.
- None of L selected TV agents can obtain information about the content of the final encrypted TAM result ($\checkmark_{(5)}$), since the collaboration all of them and the TAM Service is required to reveal the TAM result in plain form.
- The TAM Service can learn the content of the encrypted TAM result ($\checkmark_{(3)}$) if and only if the L agents (all of them) send the partial decrypted results to it.
- At the end of the protocol, only the aggregate TAM result ($\checkmark_{(2)}$) is revealed. As a result, no individual can be associated with the viewership vector that they submitted. Consequently, the proposed protocol preserves k-anonymity for $k = N$, where N is the number of all the participants who take part in the measurement.
- In order to be protected from inaccurate result ($\checkmark_{(2)}$) reporting from the TAM Service, multiple combiners can be introduced in Phase 3, to confirm the announced results.

Data items	Participants			
	TAM Service	TA	TV agents	
			All	L-selected
TV agents' viewership vector				
Plain form	✗	✗	✗	✗
Encrypted form	✗	✓$_{(1)}$	✗	✗
TAM result				
Plain form	✓$_{(2)}$	✗	✗	✗
Encrypted form	✓$_{(3)}$	✓$_{(4)}$	✗	✓$_{(5)}$

Table 7.2: The scope (columns) of the data items (rows).

7.6 Experimental Results

To evaluate our solution, we developed a prototype that implements the PrivTAM calculation. The prototype can be separated into two main parts, the first being the application on the Smart TVs and the second the application on the TA. The Smart TV application is implemented using the Google TV platform[1] and the Java for Android 3.1 SDK. Of course, it is possible and other platforms, like Samsung Smart TV[2], to support our cryptographic protocol because the only requirement is to support operations with big integers. The application on the TA is also implemented in Java. Both applications use the cryptographic primitives of the Paillier Threshold Encryption Toolbox [186]. In this library, a centralized mechanism (with a trusted dealer) for threshold key generation [46] is implemented, instead of a distributed Paillier key generation [138]. In our view, this is enough for this prototype implementation.

The TV agents use production-ready cryptographic libraries and employ 1024 bits RSA X.509 certificates. The communication between agents is performed over secure sockets (SSL/TLS) with both client and server authentication. At this stage, the full functionalities of the TV agents described in our proposed system are not implemented, rather, we only implement the privacy-preserving cryptographic TAM computation.

We performed an experiment of the PrivTAM calculation, where 6 TV agents, the TA and the TAM Service participated and four channels exist. Each agent

[1] http://www.google.com/tv/

[2] http://www.samsung-smarttv.com

generated random values for the submitted viewing record, as well as for the gender and the age of the viewer. Initially, the TAM Service randomly chooses two of the participating TV agents ($L = 2$), $TVAgent_2$ and $TVAgent_5$, for the first phase of the protocol. Therefore, the final encrypted measurement will be decrypted from $TVAgent_2$, $TVAgent_5$ and the TAM Service (n, $t = 3$ parties).

Next, each TV agent encrypts the viewership vector and transmits it to the TA for verification. This process in our experiments takes less than 8 seconds. Once the viewership vector is verified, the TA multiplies it with the current encrypted TAM result. In Table 7.3 the values used to create the viewership vector of each agent are shown, along with the resulting current encrypted measurement after the submitted viewership vector is calculated by the TA.

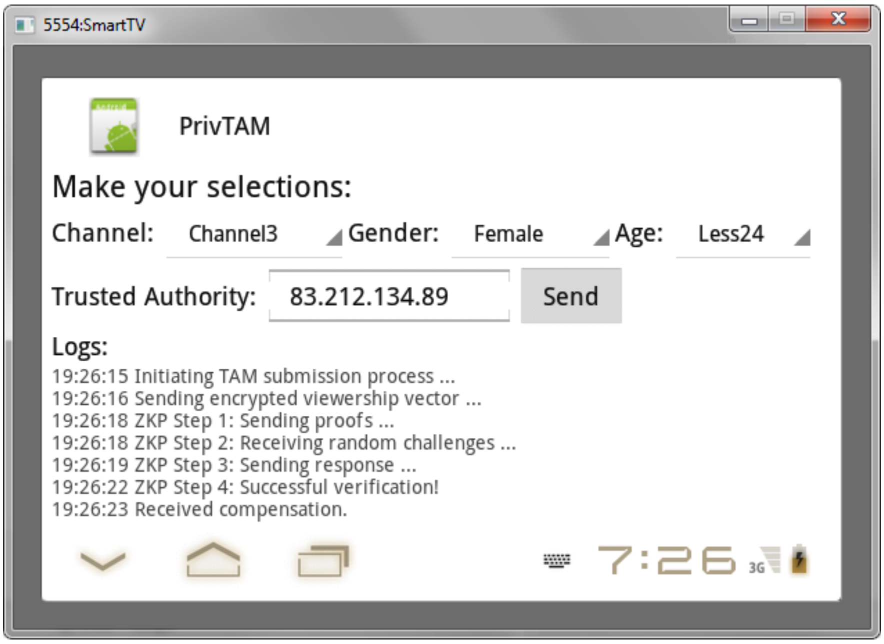
TV Agents Values				**Current Encrypted TAM**			
Agent	Channel	Gender	Age	$Channel_1$	$Channel_2$	$Channel_3$	$Channel_4$
$TVAgent_1$	$Channel_3$	Male	23	0000 0000	0000 0000	1000 0000	0000 0000
$TVAgent_6$	$Channel_1$	Female	45	0000 0010	0000 0000	1000 0000	0000 0000
$TVAgent_2$	$Channel_1$	Male	32	0100 0010	0000 0000	1000 0000	0000 0000
$TVAgent_4$	$Channel_4$	Female	29	0100 0010	0000 0000	1000 0000	0000 0100
$TVAgent_3$	$Channel_3$	Female	53	0100 0010	0000 0000	1000 0010	0000 0100
$TVAgent_5$	$Channel_3$	Female	22	0100 0010	0000 0000	1000 1010	0000 0100

Table 7.3: Example of a PrivTAM.

At the end of the computation, the TA sends the encrypted results to $TVAgent_2$, $TVAgent_5$ and the TAM Service. The TAM Service collects the partial decryption results from $TVAgent_2$ and $TVAgent_5$, and combines the partial decryption results. The decrypted TAM result, is shown in the last row of Table 7.3, where $Channel_3$ has the highest audience (50%) and the 66.66% of viewers were women. A snapshot of the application during the execution of the experiment is shown in Figure 7.4.

7.7 Conclusions

The introduction of Internet connectivity and computation capabilities to contemporary television systems, opens the possibility of conducting TAMs using larger samples of viewers. In this work we design an efficient protocol for privacy-preserving TAMS and test the applicability of the proposed solution. The accuracy and trustworthiness of the produced results act as strong incentives for TAM Services to adopt the PrivTAM system. From the viewers' perspective, PrivTAM offers the privacy assurance necessary for them to participate in a TAM



Make your selections:

Channel: Channel3 Gender: Female Age: Less24

Trusted Authority: 83.212.134.89

Logs:

```
19:26:15 Initiating TAM submission process ...
19:26:16 Sending encrypted viewership vector ...
19:26:18 ZKP Step 1: Sending proofs ...
19:26:18 ZKP Step 2: Receiving random challenges ...
19:26:19 ZKP Step 3: Sending response ...
19:26:22 ZKP Step 4: Successful verification!
19:26:23 Received compensation.
```

Figure 7.4: A snapshot of the $TV Agent_5$.

system, while fair compensation can be offered for their participation, returning some of the economic benefits of TAMs back to the viewer. Additionally, PrivTAM can support alternative kinds of measurements, providing interesting information about audiences to the TV industry. These results are achieved without using any specialized equipment and can take into account data from multiple broadcast sources. Overall, we believe this is an interesting approach, which, using contemporary cryptographic tools, achieves reliable TAMs while protecting the viewers' privacy.

A future direction for improving our solution could be to investigate if it is possible to create a decentralized architecture, like a peer-to-peer topology, where the TV agents would be self-organized and they can independently calculate and prove the correctness of the TAM results. In this case, a TA is not required. Finally, an interesting extension of the PrivTAM would be to support polls where the viewers could express their opinions about current social or political issues.

Chapter 8

Semantic Query Scrambling for Search Privacy on the Internet

8.1 Introduction

The Internet has gradually become the primary source of information for many people. More often than not, users submit queries to search engines in order to locate content. Considering the Internet as a huge library, web-search corresponds to a search within this library. While conventional library records are private under law, at least in the U.S., Internet users might be exposed by their searches.

Every time a user submits a query to a web search engine, some private information about the user and his interests might be leaked with the query. The query representing the interest will be saved in the engine's session-logs, or it may be intercepted by the Internet provider or any other node in the network path. Table 8.1 presents some queries, which—depending on culture, country laws, or corporation rules—may have privacy issues. Some of those queries may correspond to malicious intentions, but we will not distinguish.

There is ongoing research on web-log anonymization, which has turned out to be a non-trivial problem. The use of fairly advanced techniques like token-based hashing [112] and query-log bundling [97] shows that web-log anonymization is by far not solved. Another server-based approach for anonymizing query logs is based on micro-aggregation [65]. The above approaches require the user to trust the good intentions of the search engine (with respect to the user's privacy) and additionally to tolerate the inevitable possibility of personal data leakage of the server-based methods. Thus, it currently makes sense to investigate the issue also from the other side: *how users can protect themselves.*

In September 2006, AOL released a collection with search query-log data

welfare fraud	post traumatic stress
rehabs in harrisburg pa	herpes
how to make bombs	lawyers for victims of child rape
hazardous materials	acute hepatitis
gun racks	police scanner

Table 8.1: Queries which may have privacy issues.

containing about 21 million web queries collected from about 650 thousand users over three months [145]. To protect user privacy, each real IP address had been replaced with a random ID number. Soon after the release, the first 'anonymous' user had been identified from the log data. In particular, the user given the ID 4417749 in AOL's query-log was identified as the 62-old Thelma [16]. Interestingly, this identification was based solely on the queries attributed to her ID. Even though AOL withdrew the data a few days after the privacy breach, copies of the collection still circulate freely online. The incident only substantiated what was already known: web search can pose serious threats on the privacy of Internet users.

There are some countermeasures a common user can take to protect his privacy. One is to submit the query anonymously by employing standard tools, like the Tor network[1] or some anonymization proxy. This might seem as a step in right direction, but it does not solve the privacy problem. In the AOL incident, the origin of each query was hidden, since each IP address was replaced with a random ID. However, all queries originating from the same IP were assigned the same ID. This linkability between queries submitted by the same user, resolutely increased the leakage of personal data from his query set and led to the exposition of Thelma and possibly other users. Consequently, a further step would be to make the queries of a user unlinkable. To accomplish this, a user has to continuously change his IP address and to cancel out several other information leak issues that may originate elsewhere, e.g. cookies and embedded javascript.

Alternatively or in parallel, a user can try to obfuscate his 'profile' by submitting some additional random queries. In this way, the real queries are hidden in a larger set, and the task of identifying the actual interests of the user is hindered to some extent. The TrackMeNot add-on [92] for the Firefox browser implements such a feature. Another interesting add-on is OptimizeGoogle which, among other features, trims information leaking data from the interaction of a user with Google. An interesting combination of anonymization tools is employed in the Private Web Search tool [162], which is also available as an (outdated[2])

[1] http://www.torproject.org

[2] The Private Web Search (PWS) tool is a Firefox Add-on. It is available on-line but seems

Firefox add-on. An interesting recent Firefox Add-on is Google Privacy, which removes the redirected links from the web search results. While this does not protect the user query, it helps to prevent the monitoring of which of the search results the user will actually retrieve. A community-based approach to support user privacy in information retrieval is presented in [53]; a user gets his query submitted by other users of a peer-to-peer community.

An interesting approach for improving search privacy was presented in [54], where a single-term query of a user is mixed with a set of $k-1$ random search terms. This approach achieves at most k-*anonymity*, which means that each keyword can be assumed to be the actual keyword with probability of $1/k$. In our view, the concept of k-anonymity provides a handy tool to quantify privacy. However, as it is applied in [54] it raises practical issues; the number of terms in a search query is often bounded; for example, Google's API allows a maximum of 32 keywords. The problem further escalates for multi-term queries, where the mixed query consists of k multi-term expressions. Another related work is the plausibly deniable search technique of [134] where a query is transformed into a canonical form and then submitted along with $k-1$ appropriately selected cover queries. A survey on issues and techniques for preserving privacy in web-search personalization is given in [167].

There is an important reason why the above tools and methods alone might be inadequate: in all cases, the query is revealed in its clear form. Thus, privacy-enhancing approaches employing proxies, anonymous connections, or k-anonymity, would not hide the *existence of the interest* at the search engine's end or from any sites in the network path. In addition, using anonymization tools or encryption, the plausible deniability against the *existence of a private search task* at the user's end is weakened. *Plausible deniability* is a legal concept which refers to the lack of evidence proving an allegation.[1] If a query is never disclosed to the network (never leaves the user's device/computer), then the user can deny the information need it represents. Such a denial may be deemed credible, believable, or else, plausible, due to the lack of sufficient evidence of the contrary. One way to achieve plausible deniability is to submit other related—but less exposing—queries instead, such that each of the latter queries is pointing to many plausible information needs. A related application of the notion of plausible deniability can be found in the aforementioned work of [134].

Finally, there is also the related field of Private Information Retrieval (PIR). In PIR, the main problem addressed is to retrieve data from a database without revealing the query but only some encrypted or obfuscated form of it, e.g. see [215,

not to be further developed. Its latest version is v0.4.2, which supports Firefox up to version 2. The PWS as well as the TrackMeNot tool have been developed in the context of the Portia project (http://crypto.stanford.edu/portia/).

[1] http://en.wikipedia.org/wiki/Plausible_deniability

139, 36]. An interesting approach for private information retrieval that combines homomorphic encryption with the embellishment of user queries with decoy terms is presented in [144]. Another work in this line of research is the secure anonymous database search system presented in [155]. However, all the above PIR methods have an important limitation: they assume collaborative engines.

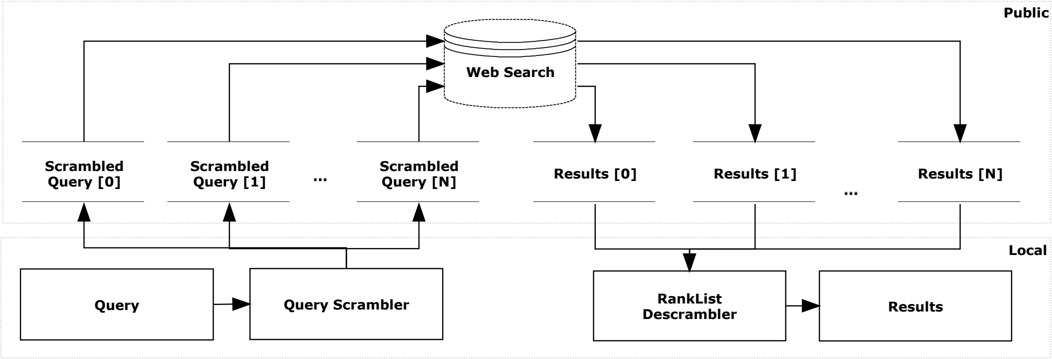


Figure 8.1: Architecture of a privacy-enhancing search system.

In view of the limitations of the aforementioned approaches, we define the Query Scrambling Problem (QSP) for privacy-preserving web search as: Given a query for a web search, it is requested to obtain related web documents. To achieve this, it is allowed to interact with search engines, but without revealing the query; the query and the actual interest of the user must be protected. The engines cannot be assumed to be collaborative with respect to user privacy. Moreover, the amount of information disclosed in the process about the query should be kept as low as possible.

To address QSP, we propose the QueryScrambler; in a nutshell, it works as follows. Given a query corresponding to the intended interest, we generate a set of *scrambled queries* corresponding loosely to the interest, thus blurring the true intentions of the searcher. The set of scrambled queries is then submitted to an engine in order to obtain a set of top-n result-lists which we call *scrambled rankings*. Given the scrambled rankings, we attempt to reconstruct, at the searcher's end, a ranking similar to the one that the query would have produced, which we call *target ranking*. The process of reconstruction we call *descrambling*. Figure 8.1 depicts the architecture of such a system.

The novelty of the QueryScrambler is that it does not reveal the important terms of the exposing query, but it employs semantically related and less exposing terms. The amount of privacy gained can be controlled by users via a parameter which determines the minimum semantic distance between the intended query and each of the scrambled queries issued. In this respect, the QueryScrambler only protects the query against query-logs or sites in the network path. Thus, an adversary with knowledge of the method and access to all (or many) of the scrambled queries of a particular scrambled search could potentially reverse the procedure getting to the actual interest, nevertheless, this is easy to fix. In practice, the QueryScrambler can—and should—be combined with other orthogonal methods, such as those mentioned earlier. Especially, adding random queries and/or querying via multiple proxies/agents can make adversarial descrambling nearly impossible.

Inevitably, the QueryScrambler introduces an overhead over traditional web-search. We are currently not interested in its efficiency, as long as its requirements are within the reaches of current commodity desktop systems and retail Internet speeds. What we are interested in is its feasibility, focusing on the trade-off between privacy and quality of retrieved results: the method may be lossy, in the sense that the quality of results may degrade with enhanced privacy.

8.2 A Query Scrambler

The proposed QueryScrambler is based on a semantic framework (Section 8.2.2). First we discuss feasibility issues.

8.2.1 Theoretical & Practical Feasibility

There is no question of the theoretical feasibility of a near lossless QueryScrambler. Suppose we submit to the engine scrambled queries consisting of very frequent words, e.g. *near* stop-words. A few such scrambled queries could cover almost all the collection, which then could be downloaded to the user's site, indexed, and searched with the query. Accounting for the difference between the retrieval models, that of the engine's (usually proprietary) and that of the user's, a near-target or satisfactory ranking could be produced locally without revealing the user's target interest. In reality, such a procedure would be highly impractical or impossible for large web search engines.

Having established the theoretical feasibility of near lossless solution to QSP with the procedure described above, what we are interested in is the trade-off between the *descrambled ranking quality* and the following three quantities:

1. *scrambling intensity*, i.e., the minimum semantic distance between the query and the set of scrambled queries,
2. *query volume*, in terms of the cardinality of the scrambled query set, and
3. *ranking depth*, i.e., the number of results returned by the engine for a scrambled query.

The scrambling intensity represents the degree of hiding the true intentions; it should be given the highest priority and be kept high, affecting negatively the ranking quality. Query volume and ranking depth have the largest impact on the practical feasibility of the task; they should be kept low, affecting again negatively the ranking quality.

In practice, web search engines usually do not return the full set of results, but truncate at some rank n. For example, the Google API returns a maximum of top-1000 results per query. In this respect, we could eliminate the depth from the parameters by setting it to top-1000, a rather sufficient and practical value.

8.2.2 A Semantic Framework

Simplifying the analysis, let us assume that a query represents a single concept. Concepts more general to the query, i.e., *hyper-concepts*, would completely cover the query's concept, as well as other concepts. In this respect, some other query representing one of the hyper-concepts of the query would target more results than the query but include all results targeted by the query. Privacy for the query can be enhanced by searching for any of the hyper-concepts instead and then filtering the results for the query concept. Thus, queries representing hyper-concepts of the query can be used as scrambled queries (SQ).

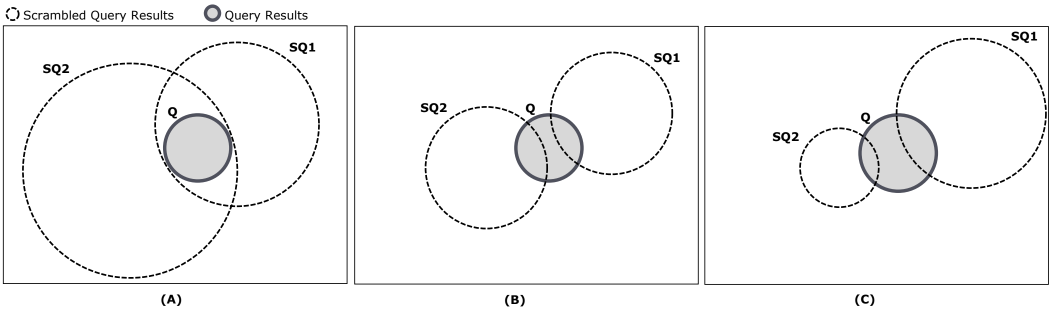


Figure 8.2: Results for two scrambled queries in relation to a query Q: (A) all results in a concept space of uniform density, (B) top-n results in a uniform document space, (C) top-n results in a non-uniform document space. Q represents all relevant results.

Figure 8.2A depicts an idealized concept space. As an example consider a query Q representing the concept 'herpes' (the disease), but searching for the concept of 'infectious disease'. SQ1 could represent 'infectious disease'. SQ2 could represent 'health problem', a more general concept than this of SQ1 denoted by covering a larger area in the space. We assume that the space has a uniform concept density. Both SQ1 and SQ2 cover Q completely.

Trying to transform Figure 8.2A to a document space, some important issues come at play:

1. Concept retrieval via the bag-of-words paradigm is inherently noisy. Semantic relations between keywords or phrases are seldom used. Thus, using concept names as keywords, e.g. using 'infectious disease' directly as SQ1, would count on 100% co-occurrence of this phrase on all documents containing the word 'herpes' in order to fulfil Figure 8.2A.

2. Web search engines usually do not return the full set of results but truncate at some rank n.

3. Document spaces are non-uniform, a direct result of the collection at hand not covering all concepts equally.

Let us first consider an idealized uniform document space. The first issue would result to SQ1 and SQ2 circles not covering Q completely, with their centers positioned at slightly different areas (assuming keyword retrieval approximates well concept retrieval). The second issue would enforce equal circle areas for SQ1 and SQ2, denoting n results (assuming that both scrambled queries have $\geq n$ results). These are depicted in Figure 8.2B.

Factoring in non-uniformity of the document space, i.e., the third issue, the picture changes to Figure 8.2C; the SQ2 area is denser than the area of SQ1, denoted by the reduced area covered by n results. The size of the Q area may also change, depending on the number of relevant results in the collection. Obviously, a single SQ would not cover all results corresponding to the query, so for a full coverage multiple SQs would have to be used.

Next, we investigate theoretically the trade-off between the number of SQs used and the expected coverage. Then, we describe the current implementation of the QueryScrambler.

8.2.3 Scrambled Query Volume

One important parameter of the QueryScrambler is the number of scrambled queries that should be executed. Naturally, a larger number of scrambled queries will increase the recall. We provide a simple probabilistic argument for how the number of scrambled queries trades off with recall.

Assume that we are interested in $\ell \leq n$ target items, where n is the search engine's truncation rank. Also, assume that we manage to generate a set of scrambled queries, such that each scrambled query catches r of the target items. If for example $r = 5$, $\ell = 50$ and $n = 1000$, then each scrambled query will retrieve (on average) $r = 5$ target items in 1000 retrieved items. The precision of the scrambled ranking will only be 0.5%, a value which should be considered sufficiently low for protecting the user's privacy.

How many scrambled queries should we submit in order to catch, with a high probability, all target items? If we assume that the target items in the results of each scrambled query are independent random items of the set of ℓ target items (of course, in reality the items will not be independent, but we will make this simplifying assumption here to obtain an indication about the number of scrambled queries that are needed), then this problem can be modelled as a *balls and bins* problem; each target item corresponds to a bin and we throw balls randomly into the bins until all bins have at least one ball. In particular, this specific problem corresponds to *the coupon collector's problem* in which there are n types of coupons and independent random coupons are chosen until a coupon of each type has been found. The following result which gives the average number of coupons that have to be drawn in order to find all ℓ coupons is well-known; see for example [130, 129]. For completeness, we provide a short proof of it in the context of QSP.

Lemma 1 *The average number of random target items that have to be drawn in order to find all ℓ target items is ℓH_ℓ, where H_ℓ is the harmonic number. The harmonic number satisfies $\ln \ell \le H_\ell \le \ln \ell + 1$, which implies that $H_\ell = \ln \ell + \Theta(1)$.*

Proof 1 *For $0 \le i \le \ell - 1$, assume that i distinct target items have been found. Let X_i be the number of random target items that have to be drawn until the next distinct target item is found. Then, the sum $Y_\ell = \sum_{i=0}^{\ell-1} X_i$ is a random variable that corresponds to the total number of random target items that are drawn until all ℓ distinct target items are found. Each random variable X_i is geometrically distributed with parameter $p_i = \frac{\ell - i}{\ell}$. Thus, the expected value of X_i is $E[X_i] = 1/p_i$ and the expected value of the sum Y_ℓ is*

$$E[Y_\ell] = E[X_1 + \cdots + X_\ell] = E[X_1] + \cdots + E[X_\ell] = \ell \sum_{i=1}^{\ell} \frac{1}{i} = \ell H_\ell \, . \qquad (8.1)$$

For the harmonic number H_ℓ, it holds

$$H_\ell = \sum_{i=1}^{\ell} \frac{1}{i} \le \sum_{i=0}^{\lfloor \log \ell \rfloor} \sum_{j=0}^{2^i - 1} \frac{1}{2^i + j} \le \sum_{i=0}^{\lfloor \log \ell \rfloor} \sum_{j=0}^{2^i - 1} \frac{1}{2^i} \le \sum_{i=0}^{\lfloor \log \ell \rfloor} 1 \le \log \ell + 1 \, . \qquad (8.2)$$

We now apply the above arguments to the QueryScrambler. Let Y_ℓ be the number of random target items that have to be retrieved to obtain the ℓ target items. As noted earlier, Y_ℓ is a random variable and Figure 8.3 shows how its expected value $E[Y_\ell]$ is related to ℓ. For $\ell = 50$, the expected number of random items that have to be drawn to retrieve all target items is $s = 50\,H_{50} \simeq 225$. If every scrambled ranking includes $r = 5$ target items, then this implies that on average $v = 45$ scrambled queries have to be executed. This number is reduced to $v = 15$, if the scrambled queries return on average $r = 15$ target items.

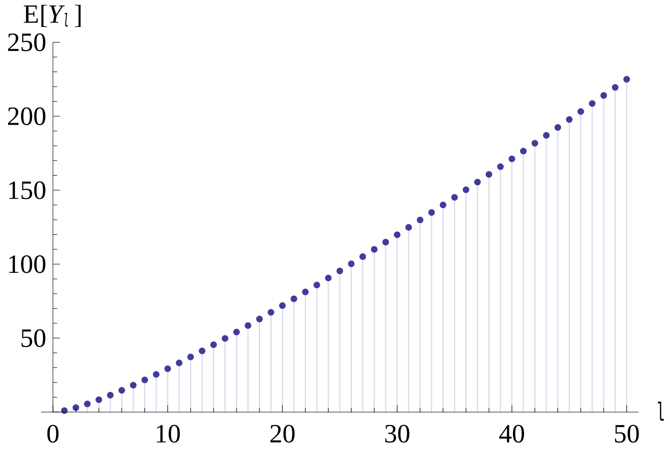


Figure 8.3: The expected number $E[Y_\ell]$, where Y_ℓ is the number of random items that have to be retrieved until all ℓ target items have been found.

To account for deviations, we may set a more conservative goal where we focus on obtaining not necessarily all distinct target items but only a (large) fraction of them. Let Z be the number of distinct random items after m random target items have been retrieved and let μ be its expected value $\mu = E[Z]$. Then, it is not hard to show the following Lemma.

Lemma 2 *The average number μ of distinct random items after drawing m random target items is*

$$\mu = E[Z] = \ell(1 - (1 - 1/\ell)^m)\ . \tag{8.3}$$

Proof 2 *For each distinct target item i, let Z_i be an indicator random variable such that*

$$Z_i = \begin{cases} 0, & \textit{if item } i \textit{ has not been selected after } m \textit{ random items,} \\ 1, & \textit{if item } i \textit{ has been selected after } m \textit{ random items.} \end{cases} \tag{8.4}$$

Then

$$E[Z_i] = Pr[Z_i = 1] = 1 - Pr[Z_i = 0] = 1 - (1 - 1/\ell)^m\ . \tag{8.5}$$

For $Z = Z_1 + \cdots + Z_\ell$ the average number of distinct target items after m random items is $E[Z] = E[Z_1 + \cdots + Z_\ell] = \ell(1 - (1 - 1/\ell)^m)$.

In Figure 8.4, the upper line shows for $\ell = 50$ how the expected number of target items found increases with the number of random items retrieved. Again, dividing the number of random items by $r = 5$ gives the average number of scrambled queries.

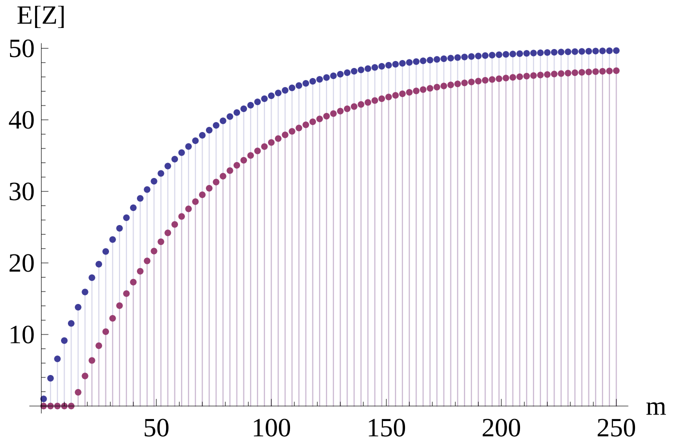


Figure 8.4: The random variable Z is the number of distinct target items (out of a total of $\ell = 50$ distinct target items) that have been found after m random target items have been retrieved. For $\ell = 50$, the upper line shows the expected value of $E[Z]$ with respect to m. The lower line shows a lower (tail) bound on Z (with probability at least 0.9) with respect to m; with probability at least 0.9 the value of Z is not below the lower line.

The lower line in Figure 8.4 presents a lower (tail) bound on the number of target items found. More precisely, the line shows that with probability at least 0.9, at least so many items have been found. The corresponding tail inequality is obtained from [130, Theorem 4.18] by focusing on the number Z of occupied bins instead of the number of empty bins. Then, as in the original theorem of [130], a corresponding martingale sequence[1] is defined and then Azuma's inequality is applied. The outcome is that for $\lambda > 0$,

$$Pr[|Z - \mu| \geq \lambda] \leq 2 \exp\left(-\frac{\lambda^2(\ell - 1/2)}{\ell^2 - \mu^2}\right) , \tag{8.6}$$

where Z is the number of distinct target items found after m random target items and $\mu = E[Z]$. Setting the right hand side in the above equation to be $\leq \rho = 0.1$ and solving for λ gives that $\lambda \geq \sqrt{\frac{(\ell^2 - \mu^2)\ln(2/\rho)}{\ell - 1/2}}$. The minimum possible value of λ given by the above inequality is used to draw the lower line in Figure 8.4.

[1] A simple definition of a martingale sequence from [130]: A sequence of random variables $X_0, X_1, \ldots,$ is said to be a martingale sequence if for all $i > 0$, $E[X_i | X_0, \ldots, X_{i-1}] = X_{i-1}$.

8.2.4 Current Implementation

In order to generate scrambled queries representing hyper concepts of the query, we currently employ an ontology for natural language terms. The approach taken is a brute force one which does not involve deep semantic analysis of the query.

First, we perform a massive indiscriminate generalization taking all possible combinations of generalized query terms up to a certain higher conceptual level. Then, we apply a similarity measure to determine the distance between the query and scrambled queries; the further the distance, the better the privacy enhancement. In this respect, the similarity measure is 'loaded' with the task of classifying the scrambled queries into privacy levels, getting rid at the same time of generalized queries unsuitable to the task.

8.2.4.1 Query Generalization

As an ontology, we employ WordNet, version 3.0 (2006), a freely available lexical database used extensively for supporting automatic text analysis and artificial intelligence applications [127]. WordNet attempts to model the lexical knowledge of a native speaker of English. Its database contains about 150000 words or collocations[1], organized in over 115000 synonym sets (synsets) across four types of part of speech (PoS): noun, verb, adjective, and adverb.

A synset is the smallest unit, which represents a specific meaning of a word or collocation. Synsets are connected to each other through explicit semantic relations. The hypernymy/hyponymy relations for nouns and the hypernymy/troponymy for verbs constitute an 'is-a-(kind-of)' hierarchy. The holonymy/meronymy relations for nouns constitute 'is-a-part/member/substance-of' hierarchies. Such taxonomic properties for adverbs and adjectives do not exist in the ontology. The synsets are also organized into senses.

Initially, WordNet's lemmatization process is applied to each keyword of the query, followed by stopword removal using the traditional SMART system's English stoplist. Then, possible collocations are recognized by checking consequent query words against WordNet. All resulting *terms* (i.e., single keywords or collocations) go through part-of-speech (PoS) and sense disambiguation.

PoS and sense disambiguation cannot be performed well without enough contextual information, e.g. a complete sentence. Thus, we used a manual approach which gives the user more control over the whole procedure; the extra user effort is deemed insignificant in the big picture of privacy enhancement, considering also the fact that web queries consist of only 2 to 3 terms on average. The system

[1] A collocation is two or more words that often go together to form a specific meaning, e.g., 'hot dog'.

finds all possible PoS for each term using Wordnet and prompts the user to select the proper one. Similarly, the user selects the proper sense.

Hyper-concepts for query's terms are approximated via hypernyms and holonyms for nouns, and hypernyms for verbs. For each query term, a bag of related terms is generated following the hypernymy and holonymy relations in the ontology up to a minimum level of 2 or up to 3 if level 2 results to less than 300 scrambled queries. The set of scrambled queries is the Cartesian product of those bags of words. Thus, accounting for collocations, scrambled queries have length equal to the query.

We do not generalize adverbs or adjectives since WordNet does not have similar relations, but keep them in scrambled queries. This does not seem to be a problem; adverbs and adjectives are unlikely to have privacy issues, since they are usually modifiers to verbs and nouns, respectively.

8.2.4.2 Measuring Privacy Enhancement

Several methods for determining semantic similarity between terms have been proposed in the literature. We apply the approach of [209] to estimate the semantic similarity between two terms. The method has been found to be among the best edge counting methods applied on WordNet [187], and it has been used widely in the literature, e.g. [177, 212]. It measures the depth of the two concepts in the WordNet taxonomy as well as the depth of the least common subsumer (LCS)[1], and combines these figures into a similarity score

$$\mathrm{sim}_{i,j} = \frac{2\,\mathrm{depth}(\mathrm{LCS})}{\mathrm{depth}(i) + \mathrm{depth}(j)} \tag{8.7}$$

where, for the task at hand, we will denote a query term with i and a scrambled query term with j.

The similarity between pairs of terms is used to calculate the similarity between each scrambled query and the query. Let SQ be a scrambled query. If q is the length of the query, then any SQ has also length q. Thus, there are q^2 term(SQ)-to-term(query) similarities. For each scrambled query term j, what determines the privacy level is its max similarity with any of the query terms, i.e., $\max_i \mathrm{sim}_{i,j}$; the larger the max, the lesser the privacy. Similarly, for a multi-term query what determines the privacy level is the least private term, justifying again the use of max. Thus, the similarity $\mathrm{sim}_{\mathrm{SQ}}$ between the scrambled query and the query is

$$\mathrm{sim}_{\mathrm{SQ}} = \max_j \max_i \mathrm{sim}_{i,j} \tag{8.8}$$

[1] The LCS is defined as the ancestor node common to both input synsets whose shortest path to the root node is the longest.

where $\max_j$ selects the most exposing scrambled query term with respect to the query terms.

The last measure is a very strict criterion for privacy. In the current implementation, considering that adverbs and adjectives appear in scrambled queries unchanged, the measure would return 1 denoting no privacy. In this respect, we relax the criterion by taking the average instead:

$$\mathrm{sim}_{\mathrm{SQ}} = \frac{1}{q} \sum_j \max_i \mathrm{sim}_{i,j} \tag{8.9}$$

On the one hand, this implies that adverbs and adjectives reduce privacy, but not destroying it altogether. This reduction makes the measure safer from a privacy perspective. On the other hand, a too general term would not contribute too much to increasing the privacy of a multi-term scrambled query: too general terms are filtered out by limiting the paths on the ontology to 2 or 3 edges, as described in Section 8.2.4.1.

Table 8.2 shows all scrambled queries generated with the current query generalization method for the query 'gun racks', together with their similarities to the query as these are calculated by Equation 8.9.

8.2.5 Descrambling Ranked-lists

Each scrambled query run on a search engine produces a scrambled ranking. We investigate two ways of reconstructing the target ranking from many scrambled rankings.

8.2.5.1 Fusion

A natural and efficient approach to reconstructing the target ranking would be to fuse the scrambled rankings. However, standard fusion methods from metasearch, such as CompSUM, Borda Count, etc., may not be suitable: the scrambled rankings are results of queries targeting different, more general than the query, information needs.

Figure 8.2C depicts a document space, with the areas targeted by a query and two scrambled queries. The further from a query's center, the deeper in the ranking. The results we are interested in appear deeper in scrambled rankings than their top ranks. To complicate things further, web search engines usually do not return scores. Thus, a fusion approach should be based solely on ranks and have a positive bias at deep or possibly middle ranks of scrambled rankings.

A simple method that may indirectly achieve the desired result is to fuse by the number of scrambled rankings an item appears in. Assuming that sets of top results of scrambled rankings, as well as sets of noisy results, would be

sim_{SQ}	SQ
0.9442725	weapon system support
0.9442725	weapon support
0.9442725	arm support
0.9150327	instrument support
0.9111842	weapon system device
0.9111842	weapon device
0.9111842	arm device
0.8952206	device support
0.8819444	instrument device
0.8736842	weapon system instrumentation
0.8736842	weapon system instrumentality
0.8736842	weapon instrumentation
0.8736842	weapon instrumentality
0.8736842	arm instrumentation
0.8736842	arm instrumentality
0.8621324	device device
0.8503268	instrument instrumentation
0.8503268	instrument instrumentality
0.8433824	device instrumentation
0.8433824	device instrumentality

Table 8.2: All scrambled queries for the query ‘gun racks’.

more disjoint than sets of deep to middle results, such a fusion method would over-weigh and rank at the top the common good results. We will call this *fusion by occurrence count* (FOC) descrambling. The method results to a rough fused ranking since it classifies items into v ranks, where v is the number of scrambled queries or rankings.

In order to determine whether Figure 8.2 corresponds well to the reality of the proposed scrambler, we will also fuse with Borda Count (BC). BC is a consensus-based electoral system which in its simplest form assigns votes to ranks as $N -$ rank $+ 1$, where N is the total number of items. Since N is unknown for web search engines, we set it to 1000, i.e., the depth of the scrambled lists. Then, votes per item are added for all rankings, and items are sorted in a decreasing number of total votes. Note that BC results in a smoother ranking than FOC.

8.2.5.2 Local Re-indexing

Another approach to re-constructing a target ranking, which does not suffer from low correspondence of ranks to relevance and produces smoother rankings than FOC or BC, would be to recover item scores. This can be achieved by re-indexing the union of scrambled results at the user's end, and running the query against a local engine. We will call this method *local re-indexing* (LR) descrambling.

Re-indexing such non-random subsets of a web collection would locally create different frequency statistics than these at the remote end. This may result in a ranking quality inferior to the target ranking, even if all target results are found by the scrambled queries and re-indexed. Furthermore, it is inefficient compared to the fusion approaches: retrieving and indexing the union of results may introduce a significant network load, increased disk usage, and CPU load.

8.3 Evaluation

In order to evaluate the effectiveness of the QueryScrambler and how its quality trades off with scrambling intensity and scrambled query volume, we set up an offline experiment. We are currently not interested in the efficiency of the approach, as long as the time and space needed is within the reaches of current commodity desktop systems and retail Internet speeds.

First, we describe the datasets, the software and parameters, and the effectiveness measures used. Then, we present the experimental results.

8.3.1 Datasets, Tools, & Methods

The query dataset consisted of 95 queries selected independently by four human subjects from various query-logs. The selection was based on the rather subjective criterion of: queries which may have required some degree of privacy. Table 8.1 presents a sample of the test queries; the full set of the test queries is available online.[1]

The ClueWeb09 dataset consists of about 1 billion web pages, in 10 languages, crawled in January and February, 2009.[2] It was created by the Language Technologies Institute at CMU. As a document collection, we used the ClueWeb09_B dataset consisting of the first 50 million English pages of the ClueWeb09 dataset. The dataset was indexed with the Lemur Toolkit V4.11 and Indri V2.11, using the default settings of these versions, except that we enabled the Krovetz stemmer.[3] We used the baseline language model for retrieval, also with the default smoothing rules and parameters. This index and retrieval model simulate the remote web search engine.

[1] http://lethe.nonrelevant.net/datasets/95-seed-queries-v1.0.txt

[2] http://boston.lti.cs.cmu.edu/Data/clueweb09/

[3] http://www.lemurproject.org

A local engine re-indexes, per query, the union of sets of results returned by the remote engine for all scrambled queries. For the local engine, we again used the Lemur Toolkit and Indri, but in order to simulate that a remote engine's model is usually proprietary, we switched the local retrieval model to tf.idf. The items for re-indexing were extracted as term vectors directly from the remote engine's index; this implies a common pre-processing (e.g. tokenization, stemming, etc.) across the remote and local engines.

8.3.2 Effectiveness Measures

There are several ways for measuring the top-n quality of an IR system, e.g. precision and recall at various values of n, mean average precision (MAP), etc. These compare two top-n lists by comparing them both to the ground truth, but this presents two limitations in the current setup. First, such measures typically give absolute ratings of top-n lists, rather than a relative measure of distance. Second, in the context of the web, there is often no clear notion of what ground truth is, so they are harder to use.

We are interested in the quality of the re-constructed ranking in terms of how well it approximates the target ranking, not in the degree of relevance of the re-constructed result-list. Although, this could still be measured indirectly as a percentage loss of a traditional IR measure (assuming ground-truth exists), e.g. MAP, we find more suitable to opt for direct measures of result set intersection and rank distance. In this way we will still measure the effectiveness even for queries poorly formulated for the information need, or information needs with near zero relevance in a collection.

A simple approach to measure the distance between two top-n lists τ_1,τ_2, is to regard them as sets and capture the extent of overlap between them. We measure the overlap with the following disjointness metric (DM), which is based on the symmetric difference of the two lists:

$$\mathrm{DM}(\tau_1, \tau_2) = \frac{|(\tau_1 - \tau_2) \cup (\tau_2 - \tau_1)|}{|\tau_1| + |\tau_2|} \, . \tag{8.10}$$

It lies in $[0, 1]$, with 1 denoting disjoint lists. For lists of the same size, DM equals 1 minus the fraction of overlap.

Traditional measures of rank distance (i.e., distance between two permutations), such as Kendall's tau distance [107] or Spearman's rho, are not very suitable because our lists are truncated so they may rank different results. Thus, we use *Kendall's distance with penalty parameter* p, denoted $K^{(p)}$, which is a generalization of Kendall's tau distance to the case of truncated lists. $K^{(p)}$ was introduced in [69], where it was shown that it is not a metric in the strict mathematical sense, but still a near metric in the sense of satisfying a ‘relaxed’ triangle inequality. On the other hand, DM is a metric.

The original Kendall distance between two permutations is essentially equal to the number of exchanges in a bubblesort to convert one permutation to the other. The generalized $\bar{K}^{(p)}_{i,j}(\tau_1, \tau_2)$ measure is also related to the permutation distance between the two truncated lists, albeit with some plausible assumptions about items that do not belong to both lists. The detailed description of $K^{(p)}$ is out of the scope of this work and the interested reader is referred to [69]. In short, we first define a penalty $\bar{K}^{(p)}_{i,j}(\tau_1, \tau_2)$ for each pair of items in $P(\tau_1, \tau_2)$, where $P(\tau_1, \tau_2)$ is the union of the sets of items of the two lists. Then,

$$K^{(p)}(\tau_1, \tau_2) = \sum^{\{i,j\} \in P(\tau_1,\tau_2)} \bar{K}^{(p)}_{i,j}(\tau_1, \tau_2) \ . \tag{8.11}$$

From the definition of $K^{(p)}(\tau_1, \tau_2)$ and assuming that the penalty parameter is $p \in [0, 1]$, the maximum distance between two top-k lists occurs when the lists are disjoint. In this case the value of the distance measure is $k((p + 1)k + 2 - p)$. We use the above maximum value of the $K^{(p)}$ measure to normalize it; the normalized distance takes values in the interval $[0, 1]$. We report results with $p = 0.5$; this corresponds to the ‘neutral’ approach, and moreover, $K^{(0.5)}$ is equivalent to other rank distance measures (K_{avg}, $K_{\text{Hausdorff}}$), as it is shown in [69].

A very important feature of the Kendall’s distance with penalty parameter p is that it is a measure that can be applied even if the lists are obtained from a very large universe whose exact size might be unknown, thus it is suitable in the web retrieval context.

We evaluate with the averages of both measures on the test query dataset at top-ℓ for $\ell = 50$ instead of $n = 1000$. We find top-50 to be more realistic for web retrieval than the top-1000 of traditional IR evaluations. In addition, this allows us to put our results somewhat in perspective with the $K^{(0)}$ results for top-50 reported in [69] where rankings returned from different web search engines for the same query are compared to each other. In initial experiments, we found that $K^{(0)}$ and $K^{(0.5)}$ give values not too far away from each other. The authors in the last-mentioned study regard values of around 0.3 as ‘very similar’ rankings, while comparing a ranking fused from several engines to the individual rankings generated $K^{(0)}$ distances between 0.3 and 0.8.

8.3.3 Experiments & Results

We run experiments for 3 levels of scrambling intensity and 3 levels of query volume. By looking into the sets of scrambled queries generated via the method described in Section 8.2.4, it seemed that a test query to scrambled query similarity of less than 0.70 results in extremely weak semantic relationship between the two. Consequently, we took the similarity intervals of $(1, 0.7]$, $(0.9, 0.7]$, and $(0.8, 0.7]$, for low, medium, and high scrambling respectively. For scrambled query volume, we arbitrarily selected volumes in $\{1, 10\}$, $\{11, 25\}$, and $\{26, 50\}$, for low, medium, and high volume respectively.

When a combination of intensity and volume levels had 0 scrambled queries for a test query, we did not take that test query into account in averaging results. In such cases, search privacy for the query at the requested scrambling intensity and volume is not possible with the proposed method and other methods must be applied. Table 8.3 presents the number of test queries averaged per combination. In the parentheses, we further give the minimum, median, and maximum numbers (in this order) of scrambled queries that the test queries had for the combination at hand. The combinations with the fewest test queries are the ones where a high volume was requested, especially at high scrambling; the proposed method can generate a limited number of scrambled queries. This can be a limitation of all ontology-based methods which statistical methods may not have.

		scrambling		
		low	med	high
volume	high	55 (27,50,50)	33 (29,50,50)	19 (26,50,50)
	med	72 (11,25,25)	62 (13,25,25)	30 (11,25,25)
	low	94 (3,10,10)	88 (1,10,10)	58 (1,10,10)

Table 8.3: Numbers of test queries and (min, median, max) numbers of scrambled queries per scrambling-volume combination.

Tables 8.4 and 8.5 present the mean $K^{(0.5)}$ and DM (Section 8.3.2) for FOC and BC descrambling (Section 8.2.5.1) respectively. The best results are expected at the top-left corners of the tables for both measures, i.e., high-volume/low-scrambling, and are expected to decline with decreasing volume and/or increasing scrambling. The best experimental results are in boldface. In all experiments, the two measures appear correlated, in the sense that a better DM also implies a better ranking or $K^{(0.5)}$.

		mean $K^{(0.5)}$			mean DM		
		scrambling			scrambling		
		low	med	high	low	med	high
volume	high	.980	.989	.998	.985	.992	.999
	med	**.961**	.978	.998	**.968**	.983	.999
	low	.962	.969	.993	.971	.977	.996

Table 8.4: Mean $K^{(0.5)}$ and DM for FOC descrambling.

		mean $K^{(0.5)}$			mean DM		
		scrambling			scrambling		
		low	med	high	low	med	high
volume	high	.970	.981	.994	.978	.987	.996
	med	.944	.971	.994	.956	.978	.996
	low	**.927**	.958	.983	**.944**	.969	.988

Table 8.5: Mean $K^{(0.5)}$ and DM for BC descrambling.

The best DM results correspond to an average intersection of only 2 or 3 results between fused and target top-50 rankings, for both fusion methods. In any case or measure, BC works better than FOC. This seems to be a result of the rougher ranking that FOC provides, since the results of the two methods become closer as volume increases. Results degrade with increasing scrambling, as expected, but also degrade with increasing volume. The later is due to the fact that larger volumes of scrambled queries presuppose larger degrees of scrambling even within the same scrambling interval.

Table 8.6 presents results for LR descrambling (Section 8.2.5.2); they are much better than the fusion descrambling results. The unexpected degradation with increasing volume appears again, but only at low or med scrambling. However, it is now more difficult to explain, and we can only speculate that it is a result of having biased global statistics in the local collection. Here, the best DM result corresponds to an average intersection of 7 to 8 results between descrambled and target top-50 rankings.

		mean $K^{(0.5)}$			mean DM		
		scrambling			scrambling		
		low	med	high	low	med	high
volume	high	.848	.898	.864	.891	.926	.906
	med	.832	.883	.901	.876	.915	.932
	low	**.812**	.870	.914	**.856**	.903	.940

Table 8.6: Mean $K^{(0.5)}$ and DM for LR descrambling.

8.4 Retrieval Failure Analysis and Improvements

The task we set out to perform is daunting. Nevertheless, on average, we get to the same 7 or 8 results of the top-50 of the plain query, without submitting its important keywords; we consider this a decent result. In this section, however, we examine further the results in order to identify what may have a negative impact on retrieval effectiveness and suggest future improvements.

Firstly, it is easy to see that we have a problem in fusing scrambled ranked lists, since the Local Re-indexing and re-ranking approach (LR descrambling) yields the triple effectiveness of the two fusion methods we have tried. Nevertheless, we do not believe that LR descrambling represents the ceiling of achievable performance. In order to measure the quality of scrambled queries without the influence of descrambling, (i.e., neither fusion nor local re-indexing is used), we can look at the number of the target top-50 results found by all scrambled queries combined. Table 8.7 presents these numbers, averaged over all test queries. The previously best result of 7 or 8 is now raised to almost 13. We see improvements of at least 40% and up to 100% all over the table. In other words, although the scrambled queries retrieve quite a few of the target top-50 results, local re-indexing can rank roughly half or two-thirds of those in the descrambled top-50. This is clearly due to having biased term frequency statistics in the local collection, and results could be improved by using a generic source of frequencies instead.

		scrambling		
		low	med	high
volume	high	11.1	9.7	7.5
	med	12.1	7.8	5.1
	low	**12.7**	8.0	4.3

Table 8.7: Mean number of the target top-50 results found by all scrambled queries combined.

Secondly, there seems to be amble room for improving the method of generating scrambled queries. Let us consider a user who wants maximum privacy (i.e., high scrambling) irrespective of cost (i.e., he is willing to trade off time and use a high query volume). Assuming ideal descrambling (i.e. yielding the results of Table 8.7), the ceiling of performance would be 7.5 items out of 50, or a 15% intersection between targeted and obtained results in the top-50. The 'missing' 85% represents the price that such a user needs to pay for privacy, under the currently proposed method for generating scrambled queries. Milder privacy requirements (i.e., low scrambling) can raise the intersection to 22–25% (or lower the missing items to 75–78%). In any case, many target items are missed, thus we examined the recall of the scrambled queries with respect to the overall recall that they achieve for the original user query. To this end, we used the results of Section 8.2.3 to predict the overall recall from the recall of the scrambled queries and compared this number with the experimental values. Indicative results are presented in Figure 8.5.

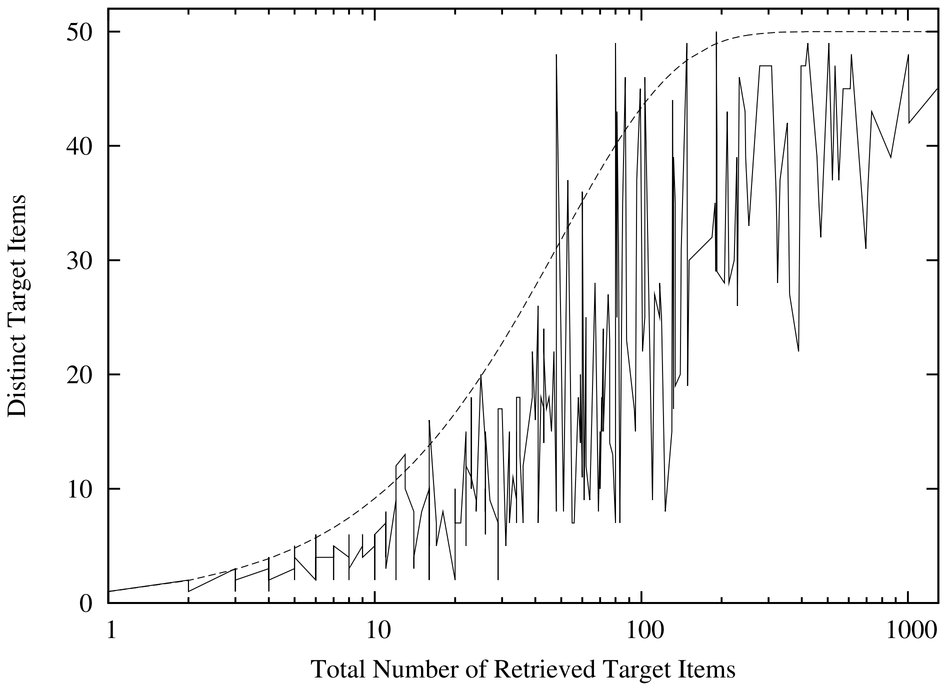


Figure 8.5: The total number of distinct target items with respect to the total number of random target items for each query-experiment. The dashed line shows the expected number of distinct target items, if the target items are independently randomly selected. The continuous line presents the experimental results. The line simply connects a large number of points, where each point corresponds to a single experiment.

In general, the measured overall recall is lower then its predicted value. The distance becomes larger for larger volumes of random target items (which in most cases implies a larger number of scrambled queries or scrambled queries of lower scrambling degree). A plausible explanation for the divergence of the measured recall is that the target items captured by the scrambled queries are correlated and not independent samples of the set of target items as it should be in the ideal case. This in turn provides a strong indication that the scrambled queries do not catch independent random subsets of the set of target items. Instead many scrambled queries return practically the same target items in their results.

The above observation defines an important issue related to the retrieval procedure with scrambled queries. The challenge is how to select scrambled queries such that they cover more effectively the whole range of target items of the original query. The Wordnet-based approach used in this work is a first step in this direction but the results show that it can be improved. Specifically, the problem seems to be that the nearness of two terms in Wordnet's graph does not imply a high co-occurrence of them in documents. In this context, we are considering to enhance our scrambled query generation procedure with statistical methods, e.g. incorporate term co-occurrence statistics. Wordnet also presents a couple of other limitations in this context: there is no obvious way to deal with phrases (except collocations), and it is a rather generic, domain-independent, thesaurus. Domain-specific knowledge could be beneficial.

Thirdly, using a search engine based on semantics might improve results. The approach taken is based on the premise that a generalization of a concept X appears in some documents that treat X. However, most of the currently big and popular commercial search engines—as well as the research engine we used in the experiments—use very little of the semantic structure behind the concepts for ranking items. Consequently, we are falling back to simply targeting co-occurrence between user and scrambled terms, missing relevant results. To this end, we submitted a dozen of our test queries to the standard non-semantic engine of Google as well as two semantic engines: Cognition[1] and Hakia[2]. Instead of the user query “acute hepatitis” (which returned 10 good results in all three engines), we submitted the scrambled query “acute liver disease”. By visually examining the snippets of the top-10 results, Google returned 0 results on hepatitis, while Medline.Cognition returned 4, and Hakia (which groups results in categories) returned 1 in Credible, 3 in Pubmed, 0 in News, and 7 in Blogs. This example suggests that using semantic engines (as well as domain-knowledge) would improve results. Nevertheless, we had a difficult time finding another so good example, thus it is unclear how big the benefits may currently be. Although there are risks in exchanging a popular big and proven non-semantic search engine with an unproven semantic engine of possibly less coverage, this matter certainly deserves a further investigation in the current context.

Lastly, our results may be more promising when measured in absolute retrieval effectiveness than goodness of approximating the target ranking. We set our ultimate goal to reconstructing the target ranking, i.e., the one the original query would have returned. However, by submitting multiple scrambled queries, we may retrieve new relevant documents not appearing in a target top-50 ranking. However, we cannot measure this in our current testbed, since it consists of custom queries with privacy issues for which no relevance judgements exist. This may even be ‘tricky’ in other standard testbeds: Given that it is customary to approximate ground-truths through pooling processes (i.e., humans judging only the union of top results of many different systems assuming all the rest non-relevant), and that most systems participating in a pool are based on the bag-of-words paradigm, the ground-truth provided with standard Web test col-

[1] http://www.congition.com
[2] http://www.hakia.com

lections may not be sufficient for our purpose. Again, further experimentation with semantic engines seems important.

8.5 Privacy Analysis

In the previous section we identified parts of our approach which may have a negative impact on retrieval effectiveness. Here, after a brief discussion on IR privacy, we investigate whether our approach achieves its the privacy goals.

8.5.1 Discussion on IR Privacy

An interesting concept that fits into the context of privacy enhanced web search is *plausible deniability*, a legal concept which refers to the lack of evidence proving an allegation. The scrambler may enhance the plausible deniability, since the original query is never disclosed and the real interest of the user is hidden within a broader concept space. This is the main privacy-enhancing feature of the QueryScrambler. Instead of the user query, a set of scrambled queries representing more general concepts is used.

Also of great importance is the fact that the QueryScrambler can perform searches while protecting not only the privacy of the user but also the query itself. A query may pose serious privacy threats even if it is submitted anonymously. In competitive fields like business or academic research, a query may contain some interesting new idea which should not be disclosed at least while the user is still making background searches on it.

In addition to the original query, the privacy of a web search might also be endangered by the results of the query. If the results have very high precision, then important information about the original user interest might be inferred from them. Regarding this issue, an inherent privacy-enhancing principle of the QueryScrambler is that each scrambled query usually retrieves only a small number of items that are in the real interest of the user. The results of an ideal scrambled query should contain some of the target items but only mixed with a large volume of unrelated or loosely related items and preferably not in the first ranks. More specifically, each scrambled query should have small recall and small precision with respect to the ℓ most related target items. In this way, the privacy of the user is not seriously endangered by simply monitoring the results. A decisive parameter of the QueryScrambler that can be used to achieve this is the degree of scrambling. If the scrambling degree is too high, then the recall in the scrambled results will be too low; privacy will be preserved, but the query will remain unanswered. If the scrambling degree is too low, then the final recall will be high but the user privacy will be endangered.

A formal criterion that can be used for privacy is *k-anonymity* [178] (see Section 2.1.3.1), which demands that every piece of information about items in a set be indistinguishably related to no fewer than k items. There are more than one ways this concept can be used with the QSP. One is the approach used in [54] to hide a term of the plain query within a group of k terms. As noted earlier, the weakness of this approach is that in this case k is bounded by a very small constant number. A more robust approach would be to achieve k-anonymity or k-indistinguishability in the concept space of a search query. The higher-level scrambled query should indistinguishably correspond to a large number k of conceptually lower-level terms. This way, the real interest of the user is hidden within a large field. This is our aim, and as we show in Section 8.5.2, on average the QueryScrambler can accomplish this goal. However, there is still work to be done. For example, there are particular cases where the semantic generalization applied by the QueryScrambler may achieve k-anonymity only for small values of k. This happens when a higher-level term corresponds only to a small number of lower-level terms. We will further investigate this issue and examine ways to overcome it, for example, by employing additional sources of semantic information or extending our approach with statistical techniques.

8.5.2 Privacy of the QueryScrambler

The privacy goal of the QueryScrambler is to protect the actual interest of a user who wants to submit a query to a search engine. We assume that the user's interest is expressed with the original query, thus, this query should not be disclosed. We also assume that the ground-truth with respect to the retrieval task is given with the results (in this work the top-50 items) of the search engine for the original query. Consequently, these results should also be protected. In summary, we define the following privacy requirements:

(A) The original user query should not be disclosed.

(B) The results of the original query must be protected.

A common way to examine the effectiveness of security or privacy measures is to evaluate the system against a so-called *adversary*, who (in the QSP context) represents the malicious entity whose aim is to violate the privacy of the user by identifying his true search interest. What are the features of the adversary? If an adversary can monitor all scrambled queries, then we expect that the privacy of the plain query can be violated. However, as we discussed earlier, we can encounter the possibility of such an attack; for example, the query scrambling can be combined with other approaches like submitting individual queries via different agents or through the Tor network etc. If the adversary captures only individual scrambled queries, then the privacy of the user depends on the minimum distance between the scrambled queries and the plain query. A different attack is to use the search results of a scrambled query to extract information about the interest. As noted earlier, the QueryScrambler has the potential to generate scrambled queries of low-enough precision and high-enough recall to unbrace such attacks. We assume that an adversary

- knows that a particular scrambled query is actually scrambled,
- can capture any scrambled query but cannot link independent scrambled queries to the same original query,
- can also capture the results that are returned by the search engine for any particular scrambled query, and
- has no background knowledge about the original query.

Next, we consider the criterion of k-anonymity, or more appropriately in this context, k-indistinguishability for the user's interest as a plausible criterion for the privacy goals of the QueryScrambler. An adversary should not be able to come closer to the real interest of the user than a set of k possible interests.

Requirement A, is addressed with the query scrambling procedure described in Section 8.2.4. In the experiments, each scrambled term can correspond to any of each descendant nodes in the 2 or 3 lower levels of the Wordnet hierarchy. In our query test-set there were on average about 319 distinct words in the two lower levels. Any of these words could be the original query term that gave the corresponding scrambled query term that was intercepted by the adversary. For example, the original term 'cortisone'[1], gives the scrambled term 'hormone'[2] (in this case a scrambled term three levels higher than the original term), and the indistinguishability for cortisone is 1 out of 138, i.e., $k = 138$. For multi term queries, indistinguishability increases multiplicatively with each additional query term. Thus, on average, every original query is indistinguishably hidden within a large number of possible terms.

An important issue is that, given a scrambled query, the corresponding candidate seed queries may not all be equally plausible. However, this non-uniformity of the candidate seed query set does not mean that an adversary who is attacking the system can with certainty exclude some of them. Let us consider the following example: An adversary who is aware of the QueryScrambling approach intercepts the scrambled query "manufacturer portable device", and then calculates a corresponding (large) set of candidate seed queries. Assume that the queries "Nokia

[1] More precisely, "cortisone#n#1" in Wordnet, i.e., its 1st, most-frequent, sense as a noun.
[2] More precisely, "hormone#n#1" in Wordnet, i.e. its 1st, most-frequent, sense as a noun.

tablet”[1] and “Apple Tablet” both belong to this set. Can the adversary exclude one of them, for example “Nokia tablet”? In our view, the fact that the query about Apple is more likely does not necessarily mean that the user did not submit the other one. In fact, the less common candidate seed query can even be more interesting, because it might reveal some interest about a less expected topic. Concluding, despite the fact that some of the candidate seed queries may be more likely than others, in most cases it should not be possible to significantly reduce the set of candidate queries using this information. Combining this with the large average number of candidate seed queries per scrambled query we believe that indistinguishability holds for the QueryScrambler approach.

Requirement B, is addressed implicitly by keeping the recall and the precision of each scrambled query low as discussed above. Even though the low precision of the scrambled queries seems to be an unavoidable consequence of the query scrambling procedure, it suits very well this requirement. In our experiments we measured an average precision of 0.0053 (or 0.018 if we exclude scrambled queries that completely failed, i.e., zero precision) at the top-1000 results. Even if an adversary knew this average precision, the low precision makes the relevant result item indistinguishable within the 1000 returned items. For example, if there are 6 relevant items, then each item of the scrambled results is a real item with probability 0.006 and even worse the correct set of the 6 relevant item is indistinguishable from a total of $C(1000, 6) \simeq 1.37 \cdot 10^{15}$ combinations (here subsets) of 1000 items taken 6 at a time. Of course, the remaining 994 items may also convey some information about the user query but they do not belong to the assumed ground-truth for the query.

In our view, the above arguments indicate that a good scrambled query can force an adversary to examine a prohibitive large set of possible interests of the user. Thus, on the one hand, we achieve the privacy goals. On the other hand, the high level of privacy for several queries in our test-bed may also justify the low retrieval effectiveness achieved. Concluding, further investigation is needed to strike a better, more usable, balance between privacy and retrieval effectiveness.

8.6 Conclusions

We introduced a method for search privacy on the Internet, which is orthogonal to standard methods such as using anonymized connections, agents, obfuscating by random additional queries or added keywords, and other techniques preventing private information leakage. The method enhances plausible deniability against query-logs by employing semantically more general queries for the intended information need. The key assumption is: the more general a concept is,

[1] At the time of the writing of this work, there was no Nokia tablet in the market.

the less private information it conveys; an assumption deemed true by example. We theoretically modelled the problem, providing a framework on which similar approaches may be built in the future.

The current implementation is based on a semantic ontology without using sophisticated natural language processing techniques or deep semantic analysis. It is arguably a brute force approach focusing on investigating the practical feasibility of the proposed method and the trade-off between quality of retrieved results and privacy enhancement. The proposed scrambling method gets up to 25% of the top-50 target results, in the ceiling of its performance. Obviously, there is a price to pay for privacy, i.e., a retrieval effectiveness loss. We investigated this trade-off in a system study; it should also be investigated in a user study in order to determine the levels of trade-off users find acceptable. Overall, the exercise demonstrated promising aspects and revealed important issues that future research should tackle.

There seems to be room for improving the method of generating scrambled queries. A thorough study of query transitions, from which one might be able to take ideas for improving the scrambled queries, is in [23]. Also, knowledge of user behavior [172] could help to improve such privacy protocols. Most importantly, the failure analysis suggested that using semantic engines, as well as domain-knowledge, would improve results. Another direction to pursue is the fusion of loosely-related data such as results corresponding to queries targeting different but related topics. This may have further extensions for meta-search, or ad hoc retrieval via multiple queries. Also, it seems interesting to investigate the retrieval effectiveness on non-uniform collection samples such as samples obtained via related queries. We have merely scratched the surface of a series of interesting aspects which beyond enhancing privacy may also prove useful for improving retrieval.

A complete scrambler-based system for privacy-preserving Internet search could be as follows. The steps to obtain a set of scrambled queries for an original user query can be executed locally at the user's side. The scrambled queries can then be submitted to search engines or any appropriate information providers. This step should not reveal the IP of the user. Furthermore, the scrambled queries should not be linkable with each other, thus, the interaction with search engines should not leak any information that might link the scrambled queries. Existing tools like Tor and OptimizedGoogle Search show how this can be done. Results are de-scrambled locally. An important feature of the proposed method is that it can be deployed in the current Internet; there are no requirements or assumptions on current search engines and, moreover, there is no need for external trusted parties or other external parties at all.

Chapter 9

Statistical Query Scrambling for Privacy-Enhanced Web Search

9.1 Introduction

In 2006, AOL released query-log data containing about 21 million web queries collected from about 650,000 users over three months [145]. To protect user privacy, each real IP address had been replaced with a random ID. Soon after the release, the first 'anonymous' user had been identified from the data [16]. Interestingly, this identification was made solely on the queries attributed to an anonymous ID. Even though AOL withdrew the data a few days after the privacy breach, copies of the collection still circulate freely online. The incident only substantiated what was already known: web search can pose serious threats on the privacy of Internet users.

The incident has motivated lots of research in web-log anonymization and solutions using anonymized or encrypted connections, agents, obfuscating by random additional queries, and other techniques; for a extensive review on the literature, we refer the reader to Chapter 8. There is an important reason why all the aforementioned methods alone might be inadequate: in all cases, the query is revealed in its clear form. Thus, such approaches would not hide the *existence of the interest* at the search engine's end or from any sites in the network path. In addition, using anonymization tools or encryption, the plausible deniability towards the *existence of a private search task* at the user's end is weakened. In other words, when a user employs the above technologies, the engine still knows that *someone* is looking for "lawyers for victims of child rape", and the user cannot deny that he has a private search task which may be the aforementioned one.

A way to achieve plausible deniability was presented in Chapter 8, called *query scrambler*, and works as follows. Given a private query, generate a set of *scrambled* *queries* corresponding loosely to the interest, thus blurring the true intentions of the searcher. The set of scrambled queries is then submitted to an engine in order to obtain a set of result-lists called *scrambled rankings*. Given the scrambled rankings, it is attempted to reconstruct, at the searcher's end, a ranking similar to the one that the private query would have produced, called *target ranking*. The process of reconstruction is called *descrambling*. The scrambler employed semantically more general queries for the private query, by using WordNet's ontology. The key assumption was: the more general a concept is, the less private information it conveys. Addressing privacy issues has the inherent difficulty to define what privacy really means. Privacy is an elusive concept, encompassing different things in different contexts and for different people [171].

The main contributions of this work are the following. In contrast to the semantic framework used in previous work, we employ a purely statistical framework. Within this statistical framework, we define three comprehensive privacy objectives—including the equivalent of the privacy objective introduced in Chapter 8. These objectives are used to define and quantify the privacy guarantees for a given web search task. All statistics needed for generating scrambled queries are estimated on a query-based document sample of the remote engine [29]; consequently, the tools presented in this work are corpus-specific. Compared to the semantic approach, the new proposed methods are found to be significantly better in retrieval effectiveness, better defined, more versatile, predictably behaved, applicable to a wider range of information needs, and the privacy they provide is more comprehensible to the end-user.

9.2 A Statistical Approach to Query Scrambling

We assume an Internet user with an information need expressed as a query for a public web search engine like Google, Bing or Baidu. The retrieval task we focus on is document discovery, i.e. finding documents that fulfill the information need of the user.

The Query Scrambling Problem (QSP) (defined in Chapter 8) for privacy-preserving web search is defined as: Given a private query q for a web search, it is requested to obtain the related web documents as if q had been submitted to a search engine. To achieve this, it is allowed to interact with search engines, but without revealing q; the query and the actual interest of the user must be protected. The engines cannot be assumed to be collaborative with respect to user privacy. Moreover, the amount of information disclosed in the process about q should be kept as low as possible.

Given a private query q, we identify two types of privacy-sensitive resources:

- The q itself and the corresponding information need of the user. In this work, we use q and information need interchangeably.

- The document set matching q, given by a public search engine. An adversary monitoring these results can extract significant information about the information need.

We will define two privacy primitives for web-search. Let N be the size of the document collection, H_q the set of documents matching q, and $\mathrm{df}_q = |H_q|$ the document frequency of q. Finally, let $\mathrm{df}_{w,q} = |H_w \cap H_q|$, for any query w and q. Let us, for now, imagine that w and q are single-term queries, so H_w and H_q are determined simply by the document sets their terms occur in; in Sections 9.3.1 and 9.3.2 we will see how we deal with multi-term queries.

A popular privacy primitive is *k-anonymity* [178] (see Section 2.1.3.1 for details), or *k-indistinghuishability*, which in the context of our work means that an adversary should not be able to come closer than a set of k possible alternatives to the private resource. Given q, for a candidate scrambled query w the first primitive k_w is

$$k_w = \frac{\mathrm{df}_w}{\mathrm{df}_{w,q}} \ , \tag{9.1}$$

a privacy measure between the two queries (each query can be single or multi-term) based on the concept of k-indistinguishability of the results. Note, that k_w is the inverse precision of the retrieval results of w with respect to the results of q. From a privacy perspective, submitting w instead of q, each of q's target documents is 'hidden' within at least $k_w - 1$ other documents.

The second primitive g_w is

$$g_w = \frac{\mathrm{df}_w}{N} \ , \tag{9.2}$$

a measure of the generality of w. The rationale behind g_w is that a general query can be assumed to be less exposing. As an indication of how general a query is, we use a pure statistical measure: *The more documents of the collection a query hits, the more general the query is.*

Based on the above primitives we define the following privacy objectives and present a use-case for each of them:

- Anything-But-This privacy or ABT_k: Assume a researcher in academia or industry who is working on some new application or product. The researcher might be interested in searching about his new idea, but might hesitate to submit a query in a clear form to a public search engine. Additionally, he doesn't care about what else will be revealed as long as it isn't his true interest. With ABT_k the researcher can conduct a scrambled search where each scrambled query satisfies $k_w > k$.

- Relative-Generalization privacy or RG_r: A citizen might be looking for information about some disease, but would not like to disclose the exact disease. A scrambled search based on scrambled queries more general than q by a factor of r might serve his need, while significantly reducing his privacy risks. Formally, RG_r means that every w must satisfy $g_w > r \cdot g_q$.

- Absolute-Generalization privacy or AG_g: Consider a citizen in some totalitarian regime. The user might decide to scramble one or more sensitive queries, for example about specific human rights, into queries with generality above a given user-specified threshold. In this case, every scrambled query must satisfy $g_w > g$.

These three privacy types may be combined, if such a privacy request arises. We will not investigate such scenarios in this work. Note that the minimum RG privacy (RG_1) also assures the minimum ABT privacy (ABT_1) but not the other way around.

Clearly, in realistic settings, it is not be feasible to calculate the exact values of the privacy measures defined above, since no one but the engine itself has access to its full collection. However, we can resort to estimating the needed quantities from a query-based document sample of the engine. We show in Section 9.3 how to estimate their values from statistical information of the local document sample.

We can now model our query scrambling approach as a set covering problem [33]. More precisely, we define Scrambled Set Covering $SSC(v, k, g)$, a multi-objective extension of set covering. Given a finite universe U of all documents of a collection, a partition of U into sets H_q and $U - H_q$, and a collection S of subsets of U, the requirement is to find a subset C of S to satisfy the following objectives and/or constraints:

- maximize $(\bigcup_{H_w \in C} H_w) \bigcap H_q$, i.e., to maximize the coverage of H_q,

- $|C| \leq v$, where v is the maximum number of scrambled queries,

- for each $H_w \in C$, the corresponding scrambled query w must satisfy $k_w > k$,

- for each $H_w \in C$, the corresponding scrambled query w must satisfy $g_w > g$.

For example, the SSC instance $SSC(10, 2, 0.01)$ refers to query with 10 scrambled queries, ABT_2 and $\mathrm{AG}_{0.01}$. The same example with RG_2 would be $SSC(10, 2, 2\, g_q)$.

Let us give an overview of our approach for query scrambling. First, we obtain a collection sample of size N with a query-based document sampling tool; this is done offline, however, the sample should be updated often enough to correspond to significant collection updates at the remote engine. In the online phase:

1. A private query q is decomposed into a set of scrambled queries. The scrambled queries are chosen to satisfy the user-specified privacy objectives of Section 9.2. To this end, we employ statistical information from the collection sample.

2. The scrambled queries are submitted as independent searches and all results are collected. To avoid a reverse engineering attack, the scrambled queries should not be linkable to each other. The user should use Tor[1] or other anonymization tools for the submissions, taking care to assure unlinkability between the scrambled queries.

3. The query q may be locally executed on the scrambled results (local re-indexing), or the scrambled ranked-lists may be fused with some combination method.

The tool we propose is intended to be used in the following way: A user can install it locally and then use it to scramble privacy-sensitive queries. It does not rely on some trusted third party for the scrambling process.

9.3 Generating Scrambled Queries

For generating scrambled queries, we follow a statistical approach using a local document sample of the remote search engine. So far, for simplicity, we have assumed single-term private and scrambled queries. In Sections 9.3.1 and 9.3.2, we will see how we can generalize the methods to work with multi-term queries. As soon as we generate a set of candidate scrambled queries, these are filtered for privacy according to the measures defined in Section 9.2. The remaining candidates are ranked according to their expected retrieval effectiveness, described in Section 9.3.3, before they are submitted.

9.3.1 Dealing with Multi-term Private Queries

If q is a single-term query, then its document frequency df_q can be determined directly from the document sample. The question is how to treat a multi-term q, or else, what the df_q of such a query is and which subset of df_q documents will be assumed as matching q so we can harvest from it related terms to be used as scrambled queries.

Given df_q, the question of which subset of documents is matching q can be settled as: we rank the sample documents with respect to q using some best-match

[1] http://www.torproject.org

retrieval model and ORed q, and take the top-df_q documents. We determine the threshold df_q by submitting the ANDed q to the collection sample and count the number of results, enforcing a minimum of 1 for practical reasons. We will refer to this estimate of df_q as aDF. The maximum number of results an ANDed query can retrieve is $\min_i \mathrm{df}_i$ when i is a term of the query; we will refer to such an estimate of df_q as mDF. This happens when the query term with the least df is 100% positively correlated with all other query terms. The term with the least df is also the most informative: if we were to reduce a multi-term q to a single term, this is the term we would keep. In this respect, df_q cannot be larger than mDF in any case.

While aDF may be too restrictive especially for a long q, mDF may be too 'loose' especially if q contains high frequency common terms. So, we will employ both aDF and mDF for estimating df_q. From a retrieval perspective, it is easier to create scrambled queries to retrieve smaller sets of documents, thus, using aDF makes the task easier than using mDF. From a privacy perspective, mDF is the largest df possible so it is safer. For example let us consider the information need represented by the query "big bad wolf". Using aDF will point to documents about the "Little Red Riding Hood" fairy tale, while using mDF will point to all documents referring to wolves including the fairy tale. Since aDF's target set is smaller, it can be easier retrieved by scrambled queries. But using mDF instead corresponds to trying to hide all wolves.

9.3.2 Generating Multi-term Scrambled Queries

For single-term scrambled queries, df_w can be determined directly from the document sample. However, we can also generate multi-word scrambled queries. The question is how to treat these, or else, what the df_w of such a scrambled query is and which subset of df_w sample documents will be assumed as occurring in.

From the documents matching q, we enrich the set of candidate scrambled single-term queries by using a sliding window of length W and generating all unique unordered combinations of 2 and 3 terms. We use a window instead of whole documents so as to limit the number of combinations; currently, we set $W = 16$ which was shown in past literature to perform best in ensuring some relatedness between terms [180] (see also Section 9.3.3). We limit the scrambled query length to 3, which also helps to keep the number of combinations practically manageable. In this procedure, we exclude all stopwords except those occurring in q.

The document set hit by such a scrambled query is estimated similarly to the method of aDF described in Section 9.3.1: The ORed scrambled query is submitted to the sample and the top-df_w documents are considered matching, where df_w is the number of documents matching the ANDed scrambled query. The choice of aDF over mDF is made purely on targeting the best privacy. aDF produces lower df_w estimates than mDF, so these queries will be removed earlier as g increases. Also, using aDF implies that queries are more targeted, achieving higher precision, so they will be removed earlier as k increases.

9.3.3 Ranking Scrambled Queries

After dropping candidate scrambled queries that violate any privacy criteria on k_w and g_w, the remaining queries should be ranked according to their expected retrieval quality with respect to the document set matching the query, i.e. the target set. For example, we can measure this quality in terms of precision and recall, and combine those in one number such as the F_β-measure [124]. Although F_β is suitable for our purpose, it has not been commonly used before for detecting the best related terms.

Topically-related terms can be ranked via several methods; a common one is by computing pointwise mutual information (PMI) using large co-occurrence windows [27]. For the task at hand, it is appropriate to consider whole documents as windows, and score each w co-occurring with q as

$$\text{PMI}_w = \log \frac{P(q,w)}{P(q)P(w)} = \log N \frac{\text{df}_{q,w}}{\text{df}_q \ \text{df}_w} \tag{9.3}$$

where $P(q,w)$ is the probability of q and w co-occurring in a document, and $P(q)$, $P(w)$, the probabilities of occurrence of q, w, in a document, respectively. Using a large corpus and human-oriented tests, [180] did a comprehensive study of a dozen word similarity measures and co-occurrence estimates. From all combinations of estimates and measures, document retrieval with a maximum window of 16 words and PMI (run tagged DR-PMI16) performed best on average.

Although PMI has been widely used in computational linguistics literature, classification, and elsewhere, it has a major drawback in our task. Removing constant factors from Eq. 9.3, which do not affect the relative ranking of terms for a given q and collection, PMI ranks terms identically to the ratio: $\text{df}_{q,w}/\text{df}_w$. Considering this ratio, an 1/1 term is ranked higher than a 9/10 term although the latter is clearly a better term from a retrieval perspective leading to a better recall; moreover, the former may be some accidental/spurious match. Or else, the PMI of perfectly correlated terms is higher when the combination is less frequent. This low-frequency bias may not undesirable for some tasks (e.g. collocation extraction), but it is in our case due to our high precision *and* recall preference. A workaround is instead to use a normalized version of PMI such as NPMI [24], which divides PMI by $-\log P(q,w)$, reducing some of the low frequency bias but not all. In any case, our task—while related—is not exactly a linguistic similarity one, where PMI works well in finding synonyms for TOEFL synonym tests [180], or collocation identification, where NPMI works well [24].

Our task seems more related to scoring features for feature selection in classification. [213] review feature selection methods and their impact on classification effectiveness. They find that PMI (which confusingly they refer to as just MI) is not competitive with other methods, and that the best methods are the χ^2-statistic and the expected mutual information (MI) [124, Ch. 13.5.1, Eq. 13.17] (which they refer to as information gain) with similar effectiveness. Still, our task is different than a straightforward term selection for classification. In classification, all selected terms are intended to be used simultaneously in order to classify a new object. Here, we use selected terms as queries *one by one* in order to cover the target set of documents. Beyond query volume, other parameters such as the number of documents retrieved per related query and the cardinality of the target document set may impact the effectiveness of the procedure.

All in all, since our task is different than determining linguistic similarity or feature selection, it makes sense to evaluate again some common term similarity measures and feature selection methods, as well as some uncommon ones, in this context.

9.4 Evaluation

In order to evaluate the effectiveness of the scrambler and how its retrieval quality trades off with scrambled query volume (v) and scrambling intensity (k or g) over the different privacy types (ABT/RG/AG) and methods (aDF/mDF), we set up an offline experiment. For comparison purposes, we re-constructed the set-up that was presented in Chapter 8 as close as possible.

9.4.1 Datasets, Tools and Methods

The private query dataset is available online[1] and consists of 95 queries selected independently by four human subjects from various query-logs (the same with the Chapter 8). As a document collection, we used the ClueWeb09_B dataset consisting of the first 50 million English pages of the ClueWeb09 dataset[2]. The dataset was indexed with the Lemur Toolkit, Indri V5.2, using the default settings, except that we enabled the Krovetz stemmer[3]. We used the baseline language model for retrieval, also with the default smoothing rules and parameters. This index and retrieval model simulate the remote web search engine.

[1] http://lethe.nonrelevant.net/datasets/95-seed-queries-v1.0.txt

[2] http://boston.lti.cs.cmu.edu/Data/clueweb09/

[3] http://www.lemurproject.org

We took a document sample of the remote collection using random queries similarly to [29]. We bootstrapped the procedure with the initial query "www". At each step, the procedure retrieves the first K results of the random query and adds them to the sample; we set $K = 1$. Previous research has shown that the choice of the initial query is not important and that $K = 1$ is best suited for heterogeneous collections such as the web. Then, a term is uniformly selected from the unique terms of the current sample and used as the next random query until the desired sample size is reached. Candidate terms are at least 3 characters long and cannot be numbers. After initial experiments we decided to use a sample of 5,000 documents which provides a good compromise between effectiveness and practical feasibility (as well as speed). We used the same types of indexing and retrieval model for the sample as for the remote engine.

In initial experiments we compared PMI, NPMI, MI, F_1, F_2 and centroid weight, and found that MI and centroid weight work best for the task of ranking scrambled queries. F_β with $\beta = 2$, i.e. weighing recall twice to precision, is slightly behind but competitive; the F-measure however requires an extra parameter (β). NPMI works better than PMI, but both are left quite behind. We will not present these results for space reasons, and will stick with MI.

We targeted the top-50 documents of the remote engine. Our local sample (5,000 documents) was so small in relation to the engine's collection that all target documents corresponded to less than 1 document in the sample. In this respect, in order to improve the focus of the scrambled queries, it makes sense to harvest those from a set of sample documents of a smaller cardinality than df_q. In initial experiments we found that a good compromise between focus and reasonably good statistics of document frequencies is to take the top-df'_q sample documents returned by q, where $\mathrm{df}'_q = \min(10, \mathrm{df}_q)$, i.e. we harvested scrambled queries from the *at most* top-10 sample documents. Also, we adjusted df'_w and $\mathrm{df}'_{q,w}$ to the new set and calculated MI using these numbers instead; this was found to improve retrieval effectiveness. Of course, the privacy constraints were applied to the unmodified frequencies as described in Section 9.2.

Concerning the evaluation measures, we simplified the matters in relation to Chapter 8 where scrambled rankings were fused via several combination methods and the fused ranking was evaluated against the target one via Kendall's τ and a set intersection metric. The fusion methods tried in the previous study were deemed weak in comparison to a local re-indexing approach, i.e. index locally the union of top-1000 documents retrieved by all scrambled queries and run the private query against the local index in order to re-construct the target ranking. Nevertheless, even with local re-indexing the ceiling of achievable performance was not reached: there were quite a few target documents retrieved by scrambled queries that could not be locally ranked in the top-50. This was attributed to having biased DF statistics in the local index. The experimental

	unfiltered		$k = 1$		$k = 2$		$k = 4$		$k = 8$		$k = 16$	
v	aDF	mDF	aDF	mDF	aDF	mDF	aDF	mDF	aDF	mDF	aDF	mDF
2	30.1	34.2	28.6	30.3	19.9	12.1	11.8	5.05	7.49	2.02	3.57	1.13
10	36.2	39.6	35.3	37.5	30.9	23.6	22.8	11.2	15.6	5.13	8.33	2.41
50	40.8	44.2	40.2	42.5	37.3	33.1	31.8	19.3	23.7	10.8	14.2	5.25

Table 9.1: ABT privacy, top-50 target documents found by the top-v scrambled queries.

effort in the aforementioned study concluded with a bare experiment evaluating only the number of target top-50 documents found by the union of the top-1000 documents retrieved by all scrambled queries. This allowed to remove the effect of de-scrambling and evaluate only the quality of scrambling; this is what we will also do.

9.4.2 Results

The two left-most columns of Table 9.1, marked as 'unfiltered', show results with no privacy; these can be considered as the ceiling of achievable performance when de-composing a user query q with the current methods. Even with no privacy, we do not get 50 out of 50 target documents because there are cases where we cannot exactly reproduce q from the sample for the following reasons. First, a term of q may not occur in the sample, e.g. 'chamblee' from "definition of chamblee cancer". However, such a term may occur in the remote collection. Second, the terms of a multi-term q, e.g. 'definition', 'chamblee', and 'cancer', may not occur within a window of 16 terms in sample documents. Third, we generate scrambled queries only up to 3 terms. All these already suggest future improvements: use larger samples, use larger or no windows at all but whole documents, and generate longer scrambled queries.

Table 9.1 also shows results for ABT privacy. The minimum privacy ($k = 1$) removes only scrambled queries which occur in all documents of the sample target set. This has a larger impact to a single-term q which may loose its 50 out 50 effectiveness. The table also shows that for light or no privacy requirements mDF works better than aDF; this happens because the sample target set of mDF is larger than this of aDF, so more scrambled queries are harvested/generated leading to better results. However, the effectiveness of mDF degrades faster than aDF as k increases, so aDF works better, as expected and explained in Section 9.3.1. For large k (e.g. for $k \geq 2$), the effectiveness of mDF roughly halves for every doubling of k, suggesting a linear relation in log-log space or a power-law.

Tables 9.2 and 9.3 show results for RG and AG privacy respectively. Using mDF, RG effectiveness roughly halves for every doubling of generalization, sug-

	$g = g_q$		$g = 2\,g_q$		$g = 4\,g_q$		$g = 8\,g_q$	
v	aDF	mDF	aDF	mDF	aDF	mDF	aDF	mDF
2	22.1	13.5	19.9	8.17	12.6	4.33	7.35	1.65
10	31.1	21.2	31.4	12.6	22.1	6.83	13.3	3.42
50	38.3	28.6	36.1	19.2	28.9	10.3	20.1	6.28

Table 9.2: Top-50 target documents with RG privacy.

	$g = .0064$		$g = .0128$		$g = .0256$		$g = .0512$	
v	aDF	mDF	aDF	mDF	aDF	mDF	aDF	mDF
2	13.7	5.22	13.5	6.82	9.29	5.40	7.80	4.83
10	21.2	11.4	21.8	11.8	15.9	12.3	11.7	7.79
50	28.0	17.1	26.5	19.0	23.1	18.0	16.0	11.4
$\#q$	69	27	82	44	87	63	94	81

Table 9.3: Top-50 target documents with AG privacy.

gesting again a power-law. Concerning AG privacy, the g values shown correspond to document frequency cut-offs of 64, 128, 256 and 512 in the current sample size. If a private query is already general enough for a g value, it is not scrambled since it has no privacy issues. Such queries are excluded from the average results of the right table. The numbers of private queries scrambled per g value and choice of aDF/mDF are shown in the last row ($\#q$). The effectiveness of mDF is similar for the first three small g cut-offs but then falls off. In other words, we can generalize private queries relatively well by using scrambled queries hitting up to 2.5% sample documents. At such an AG level, 66% (63 out of 95) of the private query dataset is deemed as not general enough so it is scrambled. Again, the aDF method is much better than mDF in all cases, providing a less steep decrease in effectiveness as generalization increases.

The fact that aDF is more effective than mDF in all privacy types when more than light privacy is required, does not mean that it should be the preferred method. As we noted in Section 9.3.1, mDF represents stricter privacy than aDF which is experimentally proved to trade off with retrieval effectiveness. The final choice between aDF/mDF should be left to the end-user or determined via a user-study.

Concerning scrambled query volume, in all privacy types and methods effectiveness increases with higher volumes. However, due to the nature of the experimental setup, we see diminishing returns as effectiveness gets closer to 50 documents. At high privacy levels where effectiveness suffers, we can see roughly a doubling of effectiveness for every fivefold increase in volume, i.e. another power-law albeit a very steep one suggesting that a few dozens of scrambled queries are enough.

9.4.3 A Comparison to Semantic Query Scrambling

The previous work (Chapter 8) dealt only with RG privacy, so we will compare our RG method and results to it. The best effectiveness reported in Chapter 8 is 12.7, obtained at low volume (i.e. as many scrambled queries as can be produced up to 10) and low scrambling by averaging the results for 94 of the 95 user queries. One query did not produce any scrambled queries at low scrambling. At higher volume, ironically, effectiveness slightly decreased, an effect we attribute to averaging only the 55 user queries having numbers of low-scrambled queries in the 26–50 range. Effectiveness decreased fast—below 10 and even 5 documents—at medium or high scrambling.

The most obvious problems of the semantic approach are the following. First, not all user queries can be scrambled at a requested scrambling intensity, due to WordNet's ontology being generic thus not 'dense' enough. The problem seems severe: at high scrambling, only 58 out of the 95 user queries had at least 1 scrambled query. Second, the levels of low/medium/high scrambling were defined by taking arbitrary ranges of values of some semantic similarity measure between each scrambled query and q. Thus, scrambling intensity is difficult to be explained to the end-user: how much exposing is a scrambled query with, say, 0.8 similarity to q?

The statistical approach does not have the problems of the semantic. First, we always seem to produce enough scrambled queries. This may not be the case for very small document samples, but it does hold for our—reasonably small—5,000 sample. Second, our approach to RG can easier be explained to the end-user: the information need expressed by a scrambled query is satisfied by at least X times more documents than his private query. This can give him a better idea on how much he is exposed, in contrast to giving him a raw similarity threshold as in the semantic approach.

Moreover, we seem to get much better effectiveness. Although the two approaches are not directly comparable due to the weak definitions of low/medium/high scrambling of the semantic approach, comparing the methods at minimum scrambling (i.e. low scrambling vs. $g = g_q$) at volume 10 we see improvements of +145% or +67% (12.7 vs. 31.1 with aDF or 21.2 with mDF). Nevertheless, we should investigate which levels of privacy are roughly comparable across the two approaches.

Let us attempt a comparison of RG at the minimum level, as well as, at levels of the statistical approach which result to around 12.7 target documents on average for volume 10, according to Table 9.2. For the user query "gun racks", Table 9.4 compares the scrambled queries resulting from the semantic approach (the two left-most columns of Table 9.4 are taken from the Table 8.2 appearing in Chapter 8) against the scrambled queries of the statistical approach. The semantic

low scrambling	medium scrambling	mDF, $g = g_q$	mDF, $g = 2g_q$	aDF, $g = 8g_q$
weapon system support	device support	light replacement	air power	air power
weapon support	instrument device	gun light **39**	light power	light power
arm support	weapon system instrumentation	air book cover	weight	weight
instrument support	weapon system instrumentality	electric light machine	accessory	accessory
weapon system device	weapon instrumentation	pull	machine power	machine power
weapon device	weapon instrumentality	air kit	light supply	light model
arm device	arm instrumentation	air cover	22 light	fire light
—	arm instrumentality	air gun home **3**	cover picture	gun **40**
—	device device	light pump	light model	trailer
—	instrument instrumentation	brake	fire light	air picture
0	**0**	**39**	**0**	**40**

Table 9.4: Top-10 RG scrambled queries for private query 'gun racks' and # of target docs found.

approach is capable of generating only 7 scrambled queries at low scrambling but 10 at medium scrambling. None of the scrambled queries hit any of the target documents at any scrambling intensity. A bold number next to a query is the number of target results hit (if any), while the last row shows the number of distinct target results hit by all scrambled queries per column. The statistical approach achieves good results (above the 12.7 average) in two out of three cases. Nevertheless, it seems difficult to decide where the methods stand privacy-wise: is "weapon support" less exposing than "gun light" or just "gun"? In our opinion, the user should have the last word on this by reviewing the set of scrambled queries before submission.

All in all, using the strictest privacy provided by mDF, we roughly matched or improved the best retrieval result of the semantic approach, for k up to 4 and g up to $2g_q$ or 2.5% at volume 10, and for k up to 8 and g up to $4g_q$ or 5% at volume 50. At lighter privacy requirements, we outperformed the semantic approach by far. In all cases, our methods managed to scramble all private queries where this was needed, in contrast to the semantic approach. Moreover, we detected power-law relations between the privacy levels and retrieval effectiveness of ABT and AG, as well as, between volume and retrieval effectiveness. Thus, our methods are more well-defined and easier explained to the end-user, can be applied to a wider-range of private information needs, are more effective and behave predictably, retrieval-wise.

Last, there are two other advantages of the statistical approach over the semantic one. First, in the semantic approach the user had to manually select the part-of-speech and sense of every term in his query in order to select the right node in WordNet's ontology. The statistical approach does not require these time-consuming steps. Second, the semantic approach arrived at the conclusion that the best method to de-scramble ranked-lists is to locally re-index the union of documents hit by all scrambled queries and run q against this local index. Nevertheless, even with local re-indexing the ceiling of achievable performance was not reached: there were quite a few target documents retrieved by scrambled queries that could not be locally ranked in the top-50. This was attributed to having biased DF statistics in the local index due to the fact that the local documents represented a far from uniform collection sample: they were all retrieved by a set of semantically-related scrambled queries. The document sample used by the statistical approach is more representative of the remote collection, so its DF statistics can be used in the local re-indexing approach removing most of the bias.

9.5 Conclusion

We introduced a method for search privacy on the Internet, which is orthogonal to—and should be combined with—standard methods such as using anonymized connections, agents, obfuscating by random additional queries or added keywords, and other techniques reducing private information leakage. The method enhances plausible deniability towards query-logs by employing alternative less-exposing queries for a private query. We defined and modeled theoretically three types of privacy, providing a framework on which similar approaches may be built in the future.

In contrast to previous work (Chapter 8), we followed a statistical approach which does not use word/concept ontologies, semantic analysis or natural language processing. We investigated the practical feasibility of the proposed method and the trade-off between quality of retrieved results and privacy enhancement. In the semantic approach, the best result was 25% of the top-50 target documents found, and was achieved at the lightest possible privacy requirements; our new method can match this at higher-than-minimum privacy levels and for more and better-defined privacy types which can easier be explained to the end-user. At our lightest privacy level, our new method outperforms the semantic one by far; we retrieve up to 56–76% of the target results. Moreover, the proposed method can be applied to a wider range of information needs and performs more predictably retrieval-wise.

Privacy is an elusive concept. While it is easy to evaluate the retrieval effectiveness of our methods, it is difficult to evaluate the actual privacy perceived by the end users. We investigated our approach in a system-study; it should also be investigated in a user-study in order to determine the levels of privacy trade-offs users find acceptable.

Chapter 10

Conclusion

In this thesis, we proposed the usage of ubiquitous personal data from sensors, mobile devices or other resources in order to create beneficial services/applications for society. The proposed (four) applications utilize the personal data of users while ensuring their privacy. The protection of privacy is achieved using cryptographic techniques and protocols that perform privacy-preserving computations in communities of personal software agents. Additionally, the implementation of such applications confirmed that is feasible to make use of and, at the same time, to protect the personal data of individuals, and do so in an efficient way. More specifically, the conclusions for each one of these four applications are:

- In the Nearest Doctor Problem (NDP) (Chapter 4), we proposed the use of the current location of each doctor for supporting services (finding the nearest doctor in an emergency) for public well-being. In our view, the NDP solution for offering help in case of an emergency should be considered complementary to existing emergency handling services. The NDP solution would probably make a difference only in some cases of emergencies. However, even a small number of successful applications of NDP, justifies, at least in our view, the approach. An interesting extension of the NDP problem would be to require the location of the emergency to be private as well.

- In the privacy-preserving statistical analysis (Chapter 5), we presented an architecture for a privacy-enhanced Ubiquitous Health Monitoring System (UHMS) and proposed the use of the ubiquitous health data that are obtained by the wearable sensors in a UHMS for carrying out statistical research. To this end, we described how representative statistical functions can be executed in a distributed and privacy-preserving way. Moreover, the proposed solution offered insights into how we can calculate other more complex statistical functions, such as the polynomial regression and so on.

- In NoiseTubePrime (Chapter 6), we presented a novel, privacy-preserving architecture for the creation of participatory noise maps and on top of the NoiseTube system [123]. The proposed distributed computation is performed on encrypted data that is located in the cloud and is kept by personal software agents. Our future plans are to develop a stable and more complete version of NoiseTubePrime and demonstrate its use for real-world campaigns, also extending the platform towards more statistical parameters. Also, a user study could be set up to evaluate the overall usability of the solution in different contexts.

- In PrivTAM (Chapter 7), we designed an efficient protocol for privacy-preserving television audience measurements and tested the applicability of the proposed solution. The produced results are achieved without using any specialized equipment (only Smart TVs) and can take into account data from multiple broadcast sources. A future direction for the improvement of our solution could be to investigate if it is possible to have a decentralized architecture, like a peer-to-peer topology, where the TV agents would be self-organized and they can independently calculate and prove the correctness of the TAM results.

Additionally, we presented two methods (Chapter 8 and 9) for the protection of web searches on the Internet against search engine query-logs. The first method enhances plausible deniability against query-logs by employing semantically more general queries for the intended information need and the second one by employing alternative, statistically less-exposing queries. Compared to the semantic approach, the statistical approach is found to be significantly better in retrieval effectiveness, better defined, more versatile, predictably behaved, applicable to a wider range of information needs, and the privacy provided is more comprehensible to the end-user. While it is easy to evaluate the retrieval effectiveness of our methods, it is difficult to evaluate the actual privacy perceived by the end users. We investigated our approaches in a system-study; they could also be investigated in a user-study in order to determine the levels of privacy trade-offs users find acceptable.

Overall, this work is mostly based on a theoretical approach and confirmed with experimental results of the prototype implementations. For this reason, a more thorough evaluation of our approach would be useful, such as case studies in small or large scale. Furthermore, it would be appealing to investigate other innovative applications, that can be used in every day life, with emphasis on users' privacy, where personal data is kept on user's side. Today's data management technologies give the opportunity for users to control and protect their personal data in platforms such as *OwnCloud*[1] and *TonidoPlug*[2]. Finally, this thesis gave the general directions for future research in the area of personal data privacy which will continue to be a challenge in new information technologies.

[1] www.owncloud.org

[2] www.tonidoplug.com

References

[1] 104th U.S. Congress. Health Insurance Portability and Accountability Act. In *Public Law 104-191*. Aug. 21 1996. 12, 81

[2] M. Ackerman. The intellectual challenge of cscw: The gap between social requirements and technical feasibility, 2000. 53

[3] A. Acquisti. Privacy and security of personal information: Technological solutions and economic incentives. In J. Camp and R. Lewis, editors, *The Economics of Information Security*, pages 165–178. Kluwer, 2004. 38, 43, 47, 130

[4] A. Acquisti, S. Gritzalis, C. Lambrinoudakis, and S. De Capitani di Vimercati. *Digital privacy*. Auerbach Publications, Taylor & Francis Group, 2008. 32

[5] N. R. Adam and J. C. Worthmann. Security-control methods for statistical databases: a comparative study. *ACM Comput. Surv.*, **21**:515–556, December 1989. 96

[6] E. Adar and B. Huberman. A market for secrets. *First Monday*, **6**[8], 2001. 130

[7] C. C. Aggarwal. On k-anonymity and the curse of dimensionality. In *Proceedings of the 31st international conference on Very large data bases*, VLDB '05, pages 901–909. VLDB Endowment, 2005. 83

[8] G. Aggarwal, M. Bawa, P. Ganesan, H. Garcia-Molina, K. Kenthapadi, R. Motwani, U. Srivastava, D. Thomas, and Y. X. 0002. Two can keep a secret: A distributed architecture for secure database services. In *CIDR*, pages 186–199, 2005. 45

[9] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Hippocratic databases. In *VLDB '2002: Proceedings of the 28th international conference on Very Large Data Bases*, pages 143–154. VLDB Endowment, 2002. 41

[10] R. Anderson. U.k. government loses personal data on 25 million citizens. *EDRI-gram*, **Number 5.22**, 21 November 2007. 4, 43

[11] L. Atallah, B. Lo, G.-Z. Yang, and F. Siegemund. Wirelessly accessible sensor populations (wasp) for elderly care monitoring. In *Proceedings of the 2nd International Conference on Pervasive Computing Technologies for Healthcare*, PervasiveHealth '08, pages 2 –7, 2008. 80

[12] Australian Government. Privacy Act 1988. In *Office of the Australian Information Commissioner*. 1988. http://www.privacy.gov.au/law/act. 12

[13] C. Badica, Z. Budimac, H.-D. Burkhard, and M. Ivanovic. Software agents: Languages, tools, platforms. *Comput. Sci. Inf. Syst.*, **8**[2]:255–298, 2011. 83

[14] T. Balopoulos, S. Gritzalis, and S. Katsikas. Specifying and implementing privacy-preserving cryptographic protocols. *International Journal of Information Security*, **7**[6]:395–420, 2008. 139

[15] E. Bangerter, J. Camenisch, and A. Lysyanskaya. A cryptographic framework for the controlled release of certified data. In B. Christianson, B. Crispo, J. Malcolm, and M. Roe, editors, *Security Protocols Workshop*, **3957** of *LNCS*, pages 20–42. Springer, 2004. 37

[16] M. Barbaro and T. Zeller. *A Face Is Exposed for AOL Searcher No. 4417749*, 2006 (accessed June 3, 2010). http://www.nytimes.com/2006/08/09/technology/09aol.html. 4, 145, 172

[17] O. Baudron, P.-A. Fouque, D. Pointcheval, J. Stern, and G. Poupard. Practical multi-candidate election system. In *Proceedings of the twentieth annual ACM symposium on Principles of distributed computing*, PODC '01, pages 274–283, New York, NY, USA, 2001. ACM. 30, 129, 134, 135, 138, 139

[18] L. Becchetti, L. Filipponi, and A. Vitaletti. Opportunistic privacy preserving monitoring. In *PhoneSense '10: International Workshop on Sensing for App Phones, held at ACM SenSys '10*, pages 51–55, Nov. 2010. 103, 124

[19] D. Bickson, D. Dolev, G. Bezman, and B. Pinkas. Peer-to-peer secure multi-party numerical computation. *IEEE International Conference on Peer-to-Peer Computing*, pages 257–266, 2008. 57

[20] I. Bilogrevic, M. Jadliwala, P. Kumar, S. S. Walia, J.-P. Hubaux, I. Aad, and V. Niemi. Meetings through the cloud: Privacy-preserving scheduling on mobile devices. *Journal of Systems and Software*, **84**[11]:1910 – 1927, 2011. 103

[21] P. Bogetoft, D. L. Christensen, I. Damgård, M. Geisler, T. Jakobsen, M. Krøigaard, J. D. Nielsen, J. B. Nielsen, K. Nielsen, J. Pagter, M. Schwartzbach, and T. Toft. Secure multiparty computation goes live. In R. Dingledine and P. Golle, editors, *Financial Cryptography and Data Security*, pages 325–343, Berlin, Heidelberg, 2009. Springer-Verlag. 57

[22] K. Bohrer and B. Holland, editors. *Customer Profile Exchange (CPExchange) Specification*. IDEAlliance, 2000. http://www.idealliance.org/cpexchange. 38

[23] P. Boldi, F. Bonchi, C. Castillo, and S. Vigna. From "Dango" to "Japanese Cakes": Query reformulation models and patterns. In *WI-IAT '09: Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, pages 183–190, Washington, DC, USA, 2009. IEEE Computer Society. 170

[24] G. Bouma. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, pages 31–40, 2009. 178, 179

[25] Bouncycastle. Legion of the bouncy castle, January 2011. http://www.bouncycastle.org/. 96

[26] F. Brandt. Efficient cryptographic protocol design based on distributed el gamal encryption. In *Proceedings of the 8th International Conference on Information Security and Cryptology (ICISC 2005)*, **3935** of *LNCS*, pages 32–47. Springer, 2005. 70

[27] P. F. Brown, V. J. D. Pietra, P. V. de Souza, J. C. Lai, and R. L. Mercer. Class-based n-gram models of natural language. *Computational Linguistics*, **18**[4]:467–479, 1992. 178

[28] J. A. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava. Participatory sensing. In *WSW '06: Workshop on World-Sensor-Web, held at ACM SenSys '06*, Oct. 2006. 4, 102

[29] J. P. Callan and M. E. Connell. Query-based sampling of text databases. *ACM Trans. Inf. Syst.*, **19**[2]:97–130, 2001. 173, 180

[30] F. Camous, D. McCann, and M. Roantree. Capturing personal health data from wearable sensors. In *Proceedings of the 2008 International Symposium on Applications and the Internet*, pages 153–156, Washington, DC, USA, 2008. IEEE Computer Society. 81

[31] A. Campan, T. M. Truta, and N. Cooper. P-sensitive k-anonymity with generalization constraints. *Transactions on Data Privacy*, **3**[2]:65–89, 2010. 13

[32] Canadian Parliament. Personal Information Protection and Electronic Documents Act. In *Consolidated Acts, S.C. 2000, c. 5.* 13 April 2000. `http://laws-lois.justice.gc.ca/eng/acts/P-8.6`. 12

[33] A. Caprara, M. Fischetti, and P. Toth. Algorithms for the set covering problem. *Annals of Operations Research*, **98**:2000, 1998. 175

[34] S.-C. Cha and Y.-J. Joung. From p3p to data licenses. In *Privacy Enhancing Technologies*, pages 205–222, 2003. 40, 45

[35] S. Chen, R. Wang, X. Wang, and K. Zhang. Side-channel leaks in web applications: A reality today, a challenge tomorrow. In *Proceedings of the 2010 IEEE Symposium on Security and Privacy (SP '10)*, pages 191–206, Washington, DC, USA, 2010. IEEE Computer Society. 113

[36] B. Chor, N. Gilboa, and M. Naor. Private information retrieval by keywords. Technical Report Technical Report TR CS0917, Department of Computer Science, Technion, Israel Institute of Technology, Haifa, 1997. 147

[37] C.-Y. Chow, M. F. Mokbel, and T. He. A privacy-preserving location monitoring system for wireless sensor networks. *IEEE Transactions on Mobile Computing*, **10**[1]:94–107, January 2011. 123

[38] D. Christin, A. Reinhardt, S. S. Kanhere, and M. Hollick. A survey on privacy in mobile participatory sensing applications. *Journal of Systems and Software*, **84**[11]:1928 – 1946, 2011. 103

[39] V. Ciriani, S. Capitani di Vimercati, S. Foresti, and P. Samarati. κ-anonymity. In T. Yu and S. Jajodia, editors, *Secure Data Management in Decentralized Systems*, **33** of *Advances in Information Security*, pages 323–353. Springer US, 2007. 13

[40] M. Clarkson, S. Chong, and A. Myers. Civitas: Toward a secure voting system. In *IEEE Symposium on Security and Privacy (SP 2008)*, pages 354 –368, may 2008. 69

[41] ConsumerReports. C.r. investigates: Your privacy for sale. *Consumer Reports*, **71**[10]:41, October 2006. http://www.accessmylibrary.com/coms2/summary_0286-29062087_ITM. 4, 43

[42] R. Cramer, I. Damgård, and J. B. Nielsen. Multiparty computation from threshold homomorphic encryption. In *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques: Advances in Cryptology*, EUROCRYPT '01, pages 280–299, London, UK, 2001. Springer-Verlag. 68

[43] R. Cramer, R. Gennaro, and B. Schoenmakers. A secure and optimally efficient multi-authority election scheme. In *Proceedings of the 16th annual international conference on Theory and application of cryptographic techniques*, EUROCRYPT'97, pages 103–118, Berlin, Heidelberg, 1997. Springer-Verlag. 70

[44] J. Crosby. Challenges and opportunities in identity assurance. Technical report, HM Treasury, United Kingdom, March 2008. http://www.hm-treasury.gov.uk/d/identity_assurance060308.pdf. 41

[45] T. Dalenius. Finding a needle in a haystack or identifying anonymous census records. *Journal of Official Statistics*, **2**[3]:329–336, 1986. 13

[46] I. Damgård and M. Jurik. A generalisation, a simplification and some applications of paillier's probabilistic public-key system. In *Proceedings of the 4th International Workshop on Practice and Theory in Public Key Cryptography: Public Key Cryptography*, PKC '01, pages 119–136, London, UK, 2001. Springer-Verlag. 30, 96, 110, 115, 134, 140

[47] I. Damgård and J. Nielsen. Universally composable efficient multiparty computation from threshold homomorphic encryption. In *Advances in Cryptology - CRYPTO '03*, **2729** of *Lecture Notes in Computer Science*, pages 247–264. Springer Berlin / Heidelberg, 2003. 68

[48] E. D'Hondt and M. Stevens. Participatory noise mapping. In R. T. Ballagas and D. K. Rosner, editors, *Demo Proceedings of the 9th International Conference on Pervasive Computing (Pervasive '11)*, pages 33–36, June 2011. 104, 105

[49] E. D'Hondt, M. Stevens, and A. Jacobs. Participatory noise mapping works! An evaluation of participatory sensing as an alternative to standard techniques for environmental monitoring. *Pervasive and Mobile Computing: Special Issue on Pervasive Urban Applications*, in press 2012. 104, 105, 107, 117, 125, 126

[50] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, **22**[6]:644–654, November 1976. 25

[51] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, pages 303–320, August 2004. 48, 49, 64, 67

[52] DISCREET. Discreet service provision in smart environments. FP6-2004-IST-4 contract no. 27679, 2008. `http://www.ist-discreet.org/`. 38

[53] J. Domingo-Ferrer, M. Bras-Amorós, Q. Wu, and J. A. Manjón. User-private information retrieval based on a peer-to-peer community. *Data Knowl. Eng.*, **68**[11]:1237–1252, 2009. 146

[54] J. Domingo-Ferrer, A. Solanas, and J. Castella-Roca. h(k)-private information retrieval from privacy-uncooperative queryable databases. *Online Information Review*, **33**[4]:720–744, 2009. 146, 167

[55] S. Dritsas, J. Mallios, D. Gritzalis, and C. Lambrinoudakis. Applicability of privacy enhancing technologies in ubiquitous computing environments. In *Proceedings of the IEEE Workshop on Security, Privacy and Trust in Ubiquitous Computing (SecPerU '05)*, pages 61–70. IEEE, 2005. 54

[56] S. Dritsas, D. Gritzalis, and C. Lambrinoudakis. Protecting privacy and anonymity in pervasive computing: trends and perspectives. *Telemat. Inf.*, **23**[3]:196–210, August 2006. 54

[57] W. Du and M. Atallah. Privacy-preserving cooperative statistical analysis. In *Proceedings of the 17th Annual Computer Security Applications Conference*, pages 102–112, Washington, DC, USA, 2001. IEEE Computer Society. 83

[58] W. Du, S. Chen, and Y. S. Han. Privacy-preserving multivariate statistical analysis: Linear regression and classification. In *Proceedings of the 4th SIAM International Conference on Data Mining*, pages 222–233, 2004. 83

[59] Y. Duan, N. Youdao, J. Canny, and J. Z. Zhan. P4P: practical large-scale privacy-preserving distributed computation robust against malicious users. In *USENIX Security Symposium*, pages 207–222, 2010. 83

[60] A. Durresi, A. Merkoci, M. Durresi, and L. Barolli. Integrated biomedical system for ubiquitous health monitoring. In *Proceedings of*

the 1st international conference on Network-based information systems, NBiS'07, pages 397–405, Berlin, Heidelberg, 2007. Springer-Verlag. 80, 81

[61] C. Dwork. Differential privacy: a survey of results. In *Proceedings of the 5th international conference on Theory and applications of models of computation*, TAMC'08, pages 1–19, Berlin, Heidelberg, 2008. Springer-Verlag. 14, 95

[62] C. Dwork. A firm foundation for private data analysis. *Commun. ACM*, **54**:86–95, January 2011. 14, 95

[63] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284, 2006. 95

[64] D. Eastlake 3rd. Publicly Verifiable Nominations Committee (NomCom) Random Selection. RFC 3797 (Informational), June 2004. 132, 134

[65] A. Erola, J. Castellà-Roca, G. Navarro-Arribas, and V. c Torra. Semantic microaggregation for the anonymization of query logs using the open directory project. *SORT - Statistics and Operations Research Transactions*, pages 41–58, 2011. 144

[66] D. Estrin. Participatory sensing: Applications and architecture. *IEEE Internet Computing*, **14**[1]:12–14, Jan./Feb. 2010. 123

[67] European Parliament. Directive 95/46/EC. In *Official Journal L 281*, pages 0031–0050. 24 October 1995. http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:EN:HTML. 11, 81

[68] Europe's Information Society. eSafety, November 2009. http://ec.europa.eu/esafety. 55

[69] R. Fagin, R. Kumar, and D. Sivakumar. Comparing top k lists. *SIAM J. Discrete Math.*, **17**[1]:134–160, 2003. 159, 160

[70] M. Fahrmair, W. Sitou, and B. Spanfelner. Security and privacy rights management for mobile and ubiquitous computing. In *Workshop on UbiComp Privacy*, 2005. 40

[71] Gartner, Inc. Gartner says sales of mobile devices grew 5.6 percent in third quarter of 2011; Smartphone sales increased 42 percent, 2011. http://www.gartner.com/it/page.jsp?id=1848514. 116

[72] M. R. Genesereth and S. P. Ketchpel. Software agents. *Commun. ACM*, **37**[7]:48–ff., July 1994. 83

[73] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Secure distributed key generation for discrete-log based cryptosystems. *Journal of Cryptology*, **20**:51–83, 2007. 69

[74] C. Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st annual ACM symposium on Theory of computing (STOC '09)*, pages 169–178, New York, NY, USA, 2009. ACM. 21

[75] C. Gentry. Computing arbitrary functions of encrypted data. *Commun. ACM*, **53**[3]:97–105, March 2010. 21

[76] I. Goldberg. *A Pseudonymous Communications Infrastructure for the Internet*. PhD thesis, Univ. of California at Berkeley, 2000. 37

[77] I. Goldberg. Privacy-enancing technologies for the internet iii: Ten years later. In A. Acquisti, S. Gritzalis, C. Lambrinoudakis, and S. di Vimercati, editors, *Chapter 1 of Digital Privacy: Theory, Technologies, and Practices*. December 2007. 37, 47, 83

[78] O. Goldreich. *The Foundations of Cryptography*, **2**. Cambridge University Press, 2004. 32, 138

[79] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, STOC '87, pages 218–229, New York, NY, USA, 1987. ACM. 68

[80] O. Goldreich, S. Micali, and A. Wigderson. How to prove all NP statements in zero-knowledge and a methodology of cryptographic protocol design (extended abstract). In A. Odlyzko, editor, *Advances in Cryptology - CRYPTO '86*, **263** of *Lecture Notes in Computer Science*, pages 171–185. Springer Berlin / Heidelberg, 1987. 68

[81] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems. In *Proceedings of the seventeenth annual ACM symposium on Theory of computing*, STOC '85, pages 291–304, New York, NY, USA, 1985. ACM. 34

[82] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, **28**[2]:270 – 299, 1984. 27, 30, 94, 138

[83] P. Golle, F. McSherry, and I. Mironov. Data collection with self-enforcing privacy. In *CCS '06: 13th ACM conference on Computer and*

communications security, pages 69–78, New York, NY, USA, 2006. ACM. 45

[84] D. Gritzalis and C. Lambrinoudakis. A data protection scheme for a remote vital signs monitoring service. *Medical Informatics Journal*, **25**[2]:207–224, 2000. 80

[85] D. Gritzalis. Embedding privacy in IT applications development. *Inf. Manag. Comput. Security*, **12**[1]:8–26, 2004. 37

[86] D. Gritzalis and K. Moulinos. Cryptographic libraries as a means to support privacy-enhanced information systems. In *Proceedings of the 7th ACM Workshop on Security and Privacy in e-Commerce (SPEC '00)*. ACM, 2001. 30

[87] D. Gritzalis, K. Moulinos, and K. Kostis. A privacy-enhancing e-business model based on infomediaries. In *Proceedings of the International Workshop on Information Assurance in Computer Networks: Methods, Models, and Architectures for Network Security*, MMM-ACNS '01, pages 72–83, London, UK, UK, 2001. Springer-Verlag. 37

[88] S. Gritzalis. Enhancing web privacy and anonymity in the digital era. *Information Management and Computer Security*, **12**[3]:255–287, 2004. 37

[89] W. He, X. Liu, H. Nguyen, K. Nahrstedt, and T. Abdelzaher. PDA: Privacy-preserving data aggregation in wireless sensor networks. In *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM '07)*, pages 2045 –2053. IEEE, May 2007. 123

[90] J. Hong. *An Architecture for Privacy-Sensitive Ubiquitous Computing*. PhD thesis, University of California at Berkeley, Computer Science Division, Berkeley, 2005. 37, 45, 83

[91] D. Hook. *Beginning Cryptography with Java*. Wiley Publishing, Inc., Indianapolis, USA, 2005. 72

[92] D. C. Howe and H. Nissenbaum. TrackMeNot: Resisting surveillance in web search. In *Lessons from the Identity Trail: Anonymity, Privacy, and Identity in a Networked Society*, chapter 23, pages 417–436. Oxford University Press, Oxford, UK, 2009. 145

[93] G. Huang. Monitoring mom: As population matures, so do assisted-living technologies. In *Technical Review 20*, July 2003. 80

[94] J. Iliadis, D. Spinellis, D. Gritzalis, B. Preneel, and S. Katsikas. Evaluating certificate status information mechanisms. In *Proceedings of the 7th ACM conference on Computer and communications security (CCS '00)*, pages 1–8, New York, NY, USA, 2000. ACM. 31

[95] P. Jäppinen. *ME - Mobile Electronic Personality*. PhD thesis, Lappeenranta University of Technology, Finland, 2004. 37

[96] N. Jentzsch. *Theory of Information and Privacy*, pages 7–59. Springer, 2007. 53

[97] R. Jones, R. Kumar, B. Pang, and A. Tomkins. Vanity fair: privacy in querylog bundles. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 853–862, New York, NY, USA, 2008. ACM. 144

[98] M. Kandias, L. Mitrou, V. Stavrou, and D. Gritzalis. Which side are you on? A new panopticon vs. privacy. In *Proceedings of the 10th International Conference on Security and Cryptography (SECRYPT '13)*, e-Business and Telecommunications. Springer, July 2013. 105

[99] M. Kandias, K. Galbogini, L. Mitrou, and D. Gritzalis. Insiders trapped in the mirror reveal themselves in social media. In *Proceedings of the 7th International Conference on Network and System Security (NSS '13)*, **7873** of *LNCS*, pages 220–235. Springer Berlin Heidelberg, 2013. 105

[100] M. Kantarcioglu and O. Kardes. Privacy-preserving data mining in the malicious model. *International Journal of Information and Computer Security*, **2**:353–375, January 2008. 83

[101] A. Kapadia, N. Triandopoulos, C. Cornelius, D. Peebles, and D. Kotz. AnonySense: Opportunistic and privacy-preserving context collection. In J. Indulska, D. Patterson, T. Rodden, and M. Ott, editors, *Proceedings of the 6th International Conference on Pervasive Computing (Pervasive '08)*, **5013** of *LNCS*, pages 280–297. Springer Berlin / Heidelberg, May 2008. 103, 123

[102] D. R. Karger and M. Ruhl. Diminished chord: a protocol for heterogeneous subgroup formation in peer-to-peer networks. In *Proceedings of the Third international conference on Peer-to-Peer Systems*, IPTPS'04, pages 288–297, Berlin, Heidelberg, 2004. Springer-Verlag. 65

[103] G. Karjoth and M. Schunter. A privacy policy model for enterprises. In *Proceedings of the 15th IEEE workshop on Computer Security Foundations*, CSFW '02, pages 271–282, Washington, DC, USA, 2002. IEEE Computer Society. 38

[104] G. Karjoth, M. Schunter, and M. Waidner. Platform for enterprise privacy practices: privacy-enabled management of customer data. In *Proceedings of the 2nd international conference on Privacy enhancing technologies*, PET'02, pages 69–84, Berlin, Heidelberg, 2003. Springer-Verlag. 37, 38

[105] C. Karlof, N. Sastry, and D. Wagner. Cryptographic voting protocols: a systems perspective. In *Proceedings of the 14th conference on USENIX Security Symposium - Volume 14*, pages 33–50, Berkeley, CA, USA, 2005. USENIX Association. 129

[106] S. Katsikas, J. Lopez, and G. Pernul. Trust, privacy and security in e-business: Requirements and solutions. In *Panhellenic Conference on Informatics*, pages 548–558, 2005. 46

[107] M. G. Kendall. A new measure of rank correlation. *Biometrika*, **30**[1/2]:81–93, June 1938. 159

[108] L. Kissner and D. Song. Privacy-preserving set operations. In V. Shoup, editor, *Advances in Cryptology - CRYPTO '05*, **3621** of *Lecture Notes in Computer Science*, pages 241–257. Springer Berlin / Heidelberg, 2005. 138

[109] J. Kleinberg, C. Papadimitriou, and P. Raghavan. On the value of private information. In *Proceedings of the 8th conference on Theoretical aspects of rationality and knowledge*, pages 249–257. Morgan Kaufmann Publishers Inc., 2001. 47, 131

[110] L. Korba and S. Kenny. Towards meeting the privacy challenge: Adapting drm. In *Digital Rights Management (LNCS 2696/2003)*, pages 118–136. Springer Berlin / Heidelberg, 2003. 40

[111] S. Kremer, M. Ryan, and B. Smyth. Election verifiability in electronic voting protocols. In *ESORICS 2010*, pages 389–404, Heidelberg, 2010. Springer. 96, 129

[112] R. Kumar, J. Novak, B. Pang, and A. Tomkins. On anonymizing query logs via token-based hashing. In *WWW '07: Proceedings of the 16th*

international conference on World Wide Web, pages 629–638, New York, NY, USA, 2007. ACM. 144

[113] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell. A survey of mobile phone sensing. *IEEE Communications Magazine*, **48**[9]:140–150, Sept. 2010. 103, 123

[114] K. Laudon. Markets and privacy. *Commun. ACM*, **39**[9]:92–104, 1996. 36, 47

[115] X. H. Le, S. Lee, Y.-K. Lee, H. Lee, M. Khalid, and R. Sankar. Activity-oriented access control to ubiquitous hospital information and services. *Information Sciences*, **180**[16]:2979 – 2990, 2010. 80

[116] S. Lederer, J. Hong, A. Dey, and J. Landay. Personal privacy through understanding and action: Five pitfalls for designers. In *Designing Secure Systems That People Can Use*, pages 421–445. 2005. 42, 53

[117] H.-H. Lee and M. Stamp. An agent-based privacy-enhancing model. *Information Management & Computer Security*, **16**[3]:305–319, 2008. 37

[118] N. Li and T. Li. t-Closeness: Privacy Beyond k-Anonymity and -Diversity. In *In Proceedings of IEEE International Conference on Data Engineering*, 2007. 13

[119] Y. Lindell and B. Pinkas. Secure multiparty computation for privacy-preserving data mining. *Journal of Privacy and Confidentiality*, **1**:59–98, 2009. 14

[120] G. Lioudakis, E. Koutsoloukas, N. Dellas, N. Tselikas, S. Kapellaki, G. Prezerakos, D. Kaklamani, and I. Venieris. A middleware architecture for privacy protection. *Comput. Networks*, **51**[16]:4679–4696, 2007. 37, 51, 83

[121] N. A. Lynch. *Distributed Algorithms.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1996. 114

[122] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data*, **1**[1], March 2007. 13

[123] N. Maisonneuve, M. Stevens, and B. Ochab. Participatory noise pollution monitoring using mobile phones. *Information Polity*, **15**[1-2]:51–71, Aug. 2010. 4, 102, 104, 125, 187

[124] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008. 178, 179

[125] N. Matatov, L. Rokach, and O. Maimon. Privacy-preserving data mining: A feature set partitioning approach. *Information Sciences*, **180**[14]:2696 – 2720, 2010. 83

[126] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Inc., Boca Raton, FL, USA, 1997. 25, 62

[127] G. A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, **38**[1]:39–41, 1995. 154

[128] L. Millett, B. Friedman, and E. Felten. Cookies and web browser design: toward realizing informed consent online. In *SIGCHI Conference on Human factors in computing systems*, pages 46–52, New York, USA, 2001. ACM. 42

[129] M. Mitzenmacher and E. Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005. 151

[130] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995. 151, 153

[131] D. Mulligan and A. Schwartz. Your place or mine?: privacy concerns and solutions for server and client-side storage of personal information. In *Proceedings of the tenth conference on Computers, freedom and privacy: challenging the assumptions (CFP '00)*, pages 81–84, New York, NY, USA, 2000. ACM. 37, 44, 86

[132] C. Mundie, P. de Vries, P. Haynes, and M. Corwine. Trustworthy Computing. Microsoft white paper, Microsoft Corporation, 2002. 102, 125

[133] V. Muntés-Mulero and J. Nin. Privacy and anonymization for very large datasets. In *Proceeding of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 2117–2118, New York, NY, USA, 2009. ACM. 83

[134] M. Murugesan and C. Clifton. Providing privacy through plausibly deniable search. In *SDM*, pages 768–779. SIAM, 2009. 146

[135] A. Mylonas, A. Kastania, and D. Gritzalis. Delegate the smartphone user? Security awareness in smartphone platforms. *Computers & Security*, **34**:47–66, 2013. 104

[136] M. Naehrig, K. Lauter, and V. Vaikuntanathan. Can homomorphic encryption be practical? In *Proceedings of the 3rd ACM workshop on Cloud computing security workshop (CCSW '11)*, pages 113–124, New York, NY, USA, 2011. ACM. 21

[137] M. E. Nergiz, M. Atzori, and C. Clifton. Hiding the presence of individuals from shared databases. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, SIGMOD '07, pages 665–676, New York, NY, USA, 2007. ACM. 13

[138] T. Nishide and K. Sakurai. Distributed paillier cryptosystem without trusted dealer. In *Proceedings of the 11th international conference on Information security applications*, WISA'10, pages 44–60, Berlin, Heidelberg, 2011. Springer-Verlag. 131, 134, 138, 140

[139] R. Ostrovsky and W. I. Skeith. A survey of single-database PIR: techniques and applications. In *Public Key Cryptography (PKC 2007), Lecture Notes in Computer Science*, **4450**, pages 393–411, Berlin and Heidelberg, 2007. Springer. 147

[140] C. Otto, A. Milenkovic, C. Sanders, and E. Jovanov. System Architecture of a Wireless Body Area Sensor Network for Ubiquitous Health Monitoring. *Journal of Mobile Multimedia*, **1**:307–326, January 2006. 80, 81

[141] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques*, **1592** of *LNCS*, pages 223–238. Springer, 1999. 28, 62, 90, 94, 112, 136, 138

[142] P. Paillier and D. Pointcheval. Efficient public-key cryptosystems provably secure against active adversaries. In K.-Y. Lam, E. Okamoto, and C. Xing, editors, *Advances in Cryptology - ASIACRYPT'99*, **1716** of *LNCS*, pages 165–179. Springer Berlin Heidelberg, 1999. 30

[143] L. Palen and P. Dourish. Unpacking "privacy" for a networked world. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 129–136, New York, NY, USA, 2003. ACM. 53

[144] H. Pang, X. Ding, and X. Xiao. Embellishing text search queries to protect user privacy. *Proc. VLDB Endow.*, **3**[1]:598–607, September 2010. 147

[145] G. Pass, A. Chowdhury, and C. Torgeson. A picture of search. In *InfoScale '06: Proceedings of the 1st international conference on Scalable information systems*. ACM Press, New York NY, USA, 2006. 4, 145, 172

[146] E. Paulos. Citizen science: Enabling participatory urbanism. In M. Foth, editor, *Handbook of Research on Urban Informatics: The Practice and Promise of the Real-Time City*, chapter 28, pages 414–436. Information Science Reference, IGI Global, 2009. 4, 102

[147] K. Peng, C. Boyd, E. Dawson, and B. Lee. Ciphertext comparison, a new solution to the millionaire problem. In *Proceedings of the 7th International Conference on Information and Communications Security (ICICS 2005)*, **3783** of *LNCS*, pages 84–96. Springer, 2005. 72, 83

[148] J. Pieprzyk, T. Hardjono, and J. Seberry. *Fundamentals of computer security*, chapter 15. Monographs in theoretical computer science. Springer, 2003. 129

[149] Polis. The polis project, 2008. `http://polis.ee.duth.gr`. 47

[150] PRIME. Privacy and identity management for europe. EC Contract No. IST-2002-507591, 2008. https://www.prime-project.eu/. 46

[151] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket location-support system. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 32–43, New York, NY, USA, 2000. ACM. 56

[152] X. Qian, G. Zhu, and X.-F. Li. Comparison and analysis of the three programming models in google android. In *Proceedings of the 1st Asia-Pacific Programming Languages and Compilers Workshop (APPLC), in conjunction with PLDI 2012*, June 2012. 121

[153] J.-J. Quisquater, L. Guillou, M. Annick, and T. Berson. How to explain zero-knowledge protocols to your children. In *Proceedings on Advances in cryptology*, CRYPTO '89, pages 628–631, New York, NY, USA, 1989. Springer-Verlag New York, Inc. 68

[154] E. Rahm and H. H. Do. Data cleaning: Problems and current approaches. *IEEE Bulletin of the Technical Committee on Data Engineering*, **23**[4], December 2000. 43

[155] M. Raykova, B. Vo, S. M. Bellovin, and T. Malkin. Secure anonymous database search. In R. Sion and D. Song, editors, *CCSW*, pages 115–126. ACM, 2009. 147

[156] M. Reed, P. Syverson, and D. Goldschlag. Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications*, **16**[4]:482–494, 1998. 62, 64

[157] M. K. Reiter and A. D. Rubin. Crowds: anonymity for web transactions. *ACM Trans. Inf. Syst. Secur.*, **1**:66–92, November 1998. 72

[158] Reuters. Axel springer hit by new german data leak scandal, 18th October 2008. `http://www.reuters.com/article/internetNews/idUSTRE49H1GH20081018`. 4, 43

[159] R. Rivest, L. Adleman, and M. Dertouzos. On data banks and privacy homomorphisms. In *Foundations of Secure Computation*, pages 169–177. Academic Press, 1978. 21

[160] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, **21**[2]:120–126, February 1978. 23

[161] R. L. Rivest and A. Shamir. PayWord and MicroMint: two simple micropayment schemes. In *CryptoBytes*, **2**, pages 69–87, 1996. 137

[162] F. Saint-Jean, A. Johnson, D. Boneh, and J. Feigenbaum. Private web search. In *WPES '07: Proceedings of the 2007 ACM workshop on Privacy in electronic society*, pages 84–90, New York, NY, USA, 2007. ACM. 145

[163] F. Salim, N. Sheppard, and R. Safavi-Naini. Enforcing p3p policies using a digital rights management system. In N. Borisov and P. Golle, editors, *Privacy Enhancing Technologies*, **4776** of *LNCS*, pages 200–217. Springer, 2007. 37

[164] P. Samuelson. Privacy as intellectual property? *Stanford Law Review*, **52**:1125, 2000. 36

[165] R. Sarathy and K. Muralidhar. Evaluating laplace noise addition to satisfy differential privacy for numeric data. *Transactions on Data Privacy*, **4**[1]:1–17, April 2011. 95

[166] B. Schneier. *Applied Cryptography*. John Wiley & Sons, Inc., 2nd edition, 1996. 16, 18, 19, 20, 22

[167] X. Shen, B. Tan, and C. Zhai. Privacy protection in personalized search. *SIGIR Forum*, **41**[1]:4–17, 2007. 146

[168] J. Shi, R. Zhang, Y. Liu, and Y. Zhang. PriSense: Privacy-preserving data aggregation in people-centric urban sensing systems. In *Proceedings of the 29th IEEE International Conference on Computer Communications (INFOCOM '10)*, pages 1–9. IEEE, March 2010. 103, 123, 130

[169] K. Shilton. Four billion little brothers?: Privacy, mobile phones, and ubiquitous data collection. *Commun. ACM*, **52**:48–53, November 2009. 123

[170] L. Shundong, W. Daoshun, D. Yiqi, and L. Ping. Symmetric cryptographic solution to yao s millionaires problem and an evaluation of secure multiparty computations. *Information Sciences*, **178**[1]:244 – 255, 2008. 83

[171] D. J. Solove. *Understanding Privacy*. Harvard University Press, 2008. 173

[172] A. Spink, D. Wolfram, M. B. J. Jansen, and T. Saracevic. Searching the web: The public and their queries. *JASIST*, **52**[3]:226–234, 2001. 170

[173] G. Stamatelatos, G. Drosatos, and P. S. Efraimidis. Quantum: A peer-to-peer network for distributed computations with enhanced privacy. In *EYRHKA 2009 Conference Proceedings*, pages 201–210. 3rd Panhellenic Scientific Student Conference on Informatics, September 2009. Written in Modern Greek. 55, 65

[174] L. Steels. Community Memories for Sustainable Societies. Technical report, Sony Computer Science Laboratory – Paris, November 2007. 104

[175] M. Stevens. *Community memories for sustainable societies: The case of environmental noise*. PhD thesis, Vrije Universiteit Brussel, June 2012. 4, 102, 104, 107, 117, 125, 126

[176] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *ACM SIGCOMM '01*, pages 149–160. San Diego, CA, August 2001. 55, 65

[177] M. Strube and S. P. Ponzetto. Wikirelate! computing semantic relatedness using wikipedia. In *Proceedings of the 21st national conference on Artificial intelligence - Volume 2*, pages 1419–1424. AAAI Press, 2006. 155

[178] L. Sweeney. k-anonymity: a model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, **10**[5]:557–570, October 2002. 13, 167, 174

[179] A. Tasidou, P. S. Efraimidis, and V. Katos. Economics of personal data management: Fair personal information trades. In A. B. Sideridis and C. Z. Patrikakis, editors, *Next Generation Society. Technological and Legal Issues*, **26**, chapter 14, pages 151–160. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. 47, 131

[180] E. L. Terra and C. L. A. Clarke. Frequency estimates for statistical word similarity measures. In *HLT-NAACL*, 2003. 177, 178, 179

[181] To Vima Online. Missing doctor incident, August 2007. `http://www.tovima.gr/relatedarticles/article/?aid=209954`. 4, 54

[182] UK Parliament. Data Protection Act 1998. In *Office of Public Sector Information, Chapter 29*. 16 July 1998. `http://www.legislation.gov.uk/ukpga/1998/29`. 12

[183] US Government. Privacy Act of 1974. In *5 USC, Part I, Chapter 5, Subchapter II, Sec. 552a*. U.S. Gov. Printing Office, Washington, DC, 1974. 12

[184] US Government. The cable tv privacy act of 1984. In *47 USC, Chapter 5, Subchapter V-A, Part IV, Sec. 551*. U.S. Gov. Printing Office, Washington, DC, 1984. 12, 131

[185] US Government. Video privacy protection act. In *18 USC, Part I, Chapter 121, Sec. 2710, Pub.L. 100-618*. U.S. Gov. Printing Office, Washington, DC, 1988. 12, 131

[186] UTD Data Security and Privacy Lab. Paillier threshold encryption toolbox, November 2011. `http://www.utdallas.edu/~mxk093120/cgi-bin/paillier/`. 140

[187] G. Varelas, E. Voutsakis, P. Raftopoulou, E. G. Petrakis, and E. E. Milios. Semantic similarity methods in wordnet and their application to information retrieval on the web. In *Proceedings of the 7th annual ACM international workshop on Web information and data management*, WIDM '05, pages 10–16, New York, NY, USA, 2005. ACM. 155

[188] H. Varian. Economic aspects of personal privacy. In *Privacy and self-regulation in the information age*. U.S. Dept. of Commerce, National Telecommunications and Information Administration, 1996. 37, 47, 130

[189] P. Vassiliadis, A. Simitsis, and S. Skiadopoulos. Conceptual modeling for etl processes. In *5th ACM international workshop on Data Warehousing and OLAP*, pages 14–21, New York, USA, 2002. ACM. 43

[190] D. D. Vergados. Service personalization for assistive living in a mobile ambient healthcare-networked environment. *Personal Ubiquitous Comput.*, **14**:575–590, September 2010. 80, 85

[191] W3C. The platform for privacy preferences 1.0 (p3p1.0) specification, 2002. http://www.w3.org/TR/P3P. 38

[192] R. Want, A. Hopper, V. Falcão, and J. Gibbons. The active badge location system. *ACM Transactions on Information Systems (TOIS)*, **10**:91–102, January 1992. 56

[193] A. Ward, A. Jones, and A. Hopper. A new location technique for the active office. *IEEE Personal Communications*, **4**[5]:42–47, October 1997. 56

[194] S. D. Warren and L. D. Brandeis. The right to privacy. *Harvard Law Review, Vol. IV, No. 5*, December 15, 1890. 11

[195] A. Westin. Privacy and freedom. *New York, U.S.A.: Atheneum*, 1967. 11

[196] Wikipedia. Elgamal encryption, May 2013. http://en.wikipedia.org/wiki/ElGamal_encryption. 26

[197] Wikipedia. Homomorphic encryption, May 2013. http://en.wikipedia.org/wiki/Homomorphic_encryption. 20

[198] Wikipedia. Informational self-determination, May 2013. http://en.wikipedia.org/wiki/Informational_self-determination. 11

[199] Wikipedia. Paillier cryptosystem, May 2013. http://en.wikipedia.org/wiki/Paillier_cryptosystem. 28, 30

[200] Wikipedia. Personally identifiable information, May 2013. http://en.wikipedia.org/wiki/Personally_identifiable_information. 15

[201] Wikipedia. Privacy, May 2013. http://en.wikipedia.org/wiki/Privacy. 10, 11

[202] Wikipedia. Public key certificate, May 2013. http://en.wikipedia.org/wiki/Public_key_certificate. 30

[203] Wikipedia. Rsa, May 2013. http://en.wikipedia.org/wiki/RSA_(algorithm). 23

[204] Wikipedia. Secure multi-party computation, May 2013. http://en.wikipedia.org/wiki/Secure_multi-party_computation. 32

[205] Wikipedia. Sha hash functions, May 2013. http://en.wikipedia.org/wiki/SHA_hash_functions. 23

[206] Wikipedia. Transport layer security, May 2013. http://en.wikipedia.org/wiki/Transport_Layer_Security. 31

[207] Wikipedia. Zero-knowledge proof, May 2013. http://en.wikipedia.org/wiki/Zero-knowledge_proof. 34, 35

[208] A. Wood, G. Virone, T. Doan, Q. Cao, L. Selavo, Y. Wu, L. Fang, Z. He, S. Lin, and J. Stankovic. Alarm-net: Wireless sensor networks for assisted-living and residential monitoring. Technical report, Department of Computer Science, University of Virginia, 2006. 80

[209] Z. Wu and M. Palmer. Verb semantics and lexical selection. In *Proceedings of the 32nd annual meeting of the Association for Computational Linguistics*, pages 133–138. Las Cruces, New Mexico, 1994. 155

[210] X. Xiao and Y. Tao. M-invariance: towards privacy preserving republication of dynamic datasets. In *Proc. ACM SIGMOD Int'l Conference on Management of Data*, pages 689–700. ACM, 2007. 13

[211] A. Yamazaki, A. Koyama, J. Arai, and L. Barolli. Design and implementation of a ubiquitous health monitoring system. *Int. J. Web Grid Serv.*, **5**:339–355, December 2009. 80

[212] P. Yan, Y. Jiao, A. R. Hurson, and T. E. Potok. Semantic-based information retrieval of biomedical data. In *Proceedings of the 2006 ACM symposium on Applied computing*, SAC '06, pages 1700–1704, New York, NY, USA, 2006. ACM. 155

[213] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In D. H. Fisher, editor, *ICML*, pages 412–420. Morgan Kaufmann, 1997. 179

[214] A. C.-C. Yao. Protocols for secure computations (extended abstract). In *Proceedings of Twenty-third IEEE Symposium on Foundations of Computer Science*, pages 160–164. Chicago, Illinois, November 1982. 32, 57, 83, 107, 109, 129

[215] S. Yekhanin. Private information retrieval. *Commun. ACM*, **53**[4]:68–73, 2010. 147

[216] M. Yokoo and K. Suzuki. Secure multi-agent dynamic programming based on homomorphic encryption and its application to combinatorial auctions. In *Proceedings of the first international joint conference on Autonomous agents and multi-agent systems: part 1*, AAMAS '02, pages 112–119, New York, NY, USA, 2002. ACM. 57, 62

[217] A. Zaidi. Features and challenges of population ageing: The european perspective. In *This Policy Brief is derived from the presentation made at the Social and Economic Council of Spain (CONSEJO ECONOMICOY SOCIAL, CES, Madrid), as their keynote speaker in the conference "Ageing of Population"*. European Centre for Social Welfare Policy and Research, 2008. `http://www.euro.centre.org/data/1204800003_27721.pdf`. 80

[218] S. Zhong, Z. Yang, and T. Chen. k-anonymous data collection. *Information Sciences*, **179**[17]:2948 – 2963, 2009. 83