جامعة آل البيت

Al al-Bayt University

An Improved Secure Anonymity-Preserving Remote User Authentication and Session Key Agreement Scheme Based on ECC

تحسين مخطط حماية مصادقة المستخدم عن بعد بالحفاظ على هويته وتوافق مفتاح الجلسة اعتماداً على خوارزميه ECC

BY

Eman Mohammad Bil'as Khamis

Supervisor

Dr. Khaled Mohammad Batiha

This Thesis was submitted in Partial Fulfillment of the Requirements for the Master's Degree in Computer Science

Deanship of Graduate Studies

Al-Bayt

University

جامعة آل البيت

عمادة الدراسات العليا

اقرار والتزام بقوانين جامعة آل البيت وانظمتها.

انا ايمان محمد بلعاس خميس

الرقم الجامعي: ١٤٢٠٩٠١٠٠٧ تخصص: علم الحاسوب

أعلنُ بأني قد التزمت بقوانين جامعة آل البيت وانظمتها وتعليماتها وقراراتها السارية المفعول المتعلقة بإعداد رسائل الماجستير والدكتوراه عندما قمت شخصياً بإعداد رسالتي بعنوان:

An Improved Secure Anonymity-Preserving Remote User Authentication and Session Key Agreement Scheme Based on ECC

"تحسين مخطط حماية مصادقة المستخدم عن بعد بالحفاظ على هويته وتوافق مفتاح الجلسة اعتماداً على خوارزمية CCE"

وذلك بما ينسجم مع الأمانة العلمية المتعارف عليها في كتابة الرسائل والأطاريح العلمية. كما أنني أُعلن بأن رسالتي هذه غير منقولة أو مستلة من رسائل أو أطاريح أو كتب أو أبحاث أو أي منشورات علمية تم نشرها أو تخزينها في أي وسيلة اعلامية، وتأسيساً على ما تقدم فأنني اتحمل المسؤولية بأنواعها كافة فيما لو تبين غير ذلك بما فيه حق مجلس العمداء في جامعة آل البيت بإلغاء قرار منحي الدرجة العلمية التي حصلت عليها وسحب شهادة التخرّج مني بعد صدورها دون أن يكون لي الحق في التظلم أو الأعتراض أو الطعن بأي صورة كانت في القرار الصادر عن مجلس العمداء بهذا الصدد.

التوقيع ....

التاريخ :

Committee Decision

This Thesis (An Improved Secure Anonymity-Preserving Remote User Authentication and Session Key Agreement Scheme Based on ECC) was Successfully Defended and Approved on 8ndMay. 2018.

Examination Committee Signature

Dr. Khaled Batiha, (Supervisor)

Dep. of Computer Science, Al al-Bayt University …….……………………….

batihakhalid@aabu.edu.jo

Dr. Akram Aref Hamarsheh, (Member)

Dep. of Computer Science, Al al-Bayt University ....……………………………

hamarshi@aabu.edu.jo

Dr. Mofleh AlDiabat, (Member)

Dep. of Computer Science, Al al-Bayt University ………………………………

moflehd @aabu.edu.jo

Dr. Ahamad Mousa Al-aodat, (External Member)

Dep. of Computer Information Systems, Irbid National University ………..………………

DrAhmadOdat@inu.edu.jo

Dedication

To my dear father Mohammad Bil'as Khamis, to my dear mother, to my all family with love.

Acknowledgments

I would like to thank my father and mother for their support and their patience during my study. Thanks to my supervisor Dr. Khaled Batiha for his guidance provided throughout my research and thesis preparation.

Thanks to myself for her patience, diligence and insistence, and for her endure the pressure and stress until reaching to the end. Thanks to my uncle Faisal Bil'as for his support and encouragement. Thanks to my brothers Ayman, Kareem and my sisters Alaa', Ayah for their support and endure all my psychological fluctuations. And Thanks to my best friends for their support and encouragement during my study. Finally, the first and last thanks to God for all his graces in my life.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Symbol	Description
ECC	Elliptic Curve Cryptography
RSA	Rivest-Shamir-Adelman
S	Telecare medical information system server
E_p (a, b)	Elliptic curve
ΔT	Maximum transmission delay
h (.)	Collision-resistant one-way cryptographic hash function
^	XORed operation
||	Concatenation operation
GF	Galois Field
MIPS	Million Instructions Per Second

An improved secure anonymity-preserving remote user authentication and session key agreement scheme based on ECC

A Master Thesis By

Eman Mohammad Bil'as Khamis

Supervisor:

Dr. Khaled Mohammad Batiha

Department of Computer Science, Al al-Bayt University, 2018

Abstract

In field of remote user authentication protocols, there are many of researches that proposed to improve the security and the performance of these protocols, and providing a variety of services including users' privacy and authentication and to keep sensitive information secure and safe via communication domain in the client-server environment. In 2016, A.K. Sutrala, A. K. Das, V. Odelue, M. Wazid, S. Kumari proposed a secure anonymity-preserving password-based user authentication and session key agreement scheme for Telecare Medicine Information Systems (TMIS) which is an improvement over Amin-Biswas's scheme. (Sutrala et al., 2016) scheme vulnerable in adopting Rivest-Shamir-Adelman (RSA) algorithm making it slow scheme.

In this thesis, we overcome the weaknesses in (Sutrala et al., 2016) scheme that’s used RSA algorithm making it slow scheme by using Elliptic Curve Cryptography (ECC) instead of RSA algorithm through key generations and encryption/decryption processes. In addition, we modify some parts of the scheme such as an identity change phase to reach better performance.

Through the security analysis, we show that the proposed scheme increases the security against possible known attacks including replay attack, off-line password guessing attack and smart card lost/stolen verifier attack comparison with other schemes that use ECC such as Wang (2014) and Odelu (2015) schemes, where it insecure against privilege insider attack and replay attack. In addition, the simulation results show that the proposed scheme provided low computation and communication costs by using ECC where the total cost of computation and communication respectively, is 0.0028641ms, 2112 bit comparison with (Sutrala et al., 2016) scheme that use RSA, that's the computation and communication cost respectively, is 21.6250267ms, 5632 bit. In addition, the proposed scheme better than Wang (2014) scheme that use ECC in computation and communication cost. The proposed scheme is practically suitable for mobile devices in the client-server environment.

Key words: security, RSA, ECC, TMIS.

تحسين مخطط حماية مصادقة المستخدم عن بعد بالحفاظ على هويته وتوافق مفتاح الجلسة اعتماداً على خوارزميةECC

رسالة ماجستير قُدمت من قبل

ايمان محمد بلعاس خميس

المشرف:

د .خالد محمد بطيحه

قسم علم الحاسوب، جامعة آل البيت، ٢٠١٨م

الملخص باللغة العربية

في مجال بروتوكولات مصادقة المستخدم عن بعد، هناك العديد من الأبحاث التي أُقترحت لتحسين أمن وآداء هذه البروتوكولات، وتوفير خدمات متنوعة بما في ذلك خصوصية المستخدمين والمصادقة والحفاظ على أمن المعلومات الحساسة عبر مجال الاتصال في بيئة الخادم-العميل .في عام 2016 اقترح كل من A.K. Sutrala, A. K. Das, V. Odelue, M. Wazid, S. Kumariبرتوكول للحفاظ على عدم كشف الهوية وكلمة السر من أجل مصادقة المستخدم وتوافق مفتاح الجلسة لأنظمة المعلومات الطبية عن بعد، وهو تحسين لمخطط Amin-Biswas's scheme. برتوكول (Sutrala et al., 2016)يعتمد على خوارزمية RSAفي توليد المفاتيح وفي عمليات التشفير وفك التشفيرمما جعله بروتوكول بطئ في التنفيذ.

في هذه الرسالة، قمنا بتحسين بروتوكول (Sutrala et al., 2016)من خلال استخدام منحنى التشفير الإهليجي (ECC) في توليد المفاتيح وفي عملية التشفير/فك التشفير بدلا من خوارزمية ..RSA.

من خلال التحليل الأمني، تبين لنا أن البروتوكول المقترح يزيد من الأمن ضد الاختراقات المعروفة المحتملة بما في ذلك إختراق إعادة الإرسال، اختراق تخمين كلمة المرور في حالة عدم الاتصال بالشبكة، واختراق تحليل بيانات البطاقة الذكية المفقودة/المسروقة مقارنة مع البروتوكولات التي تستخدم خوارزمية ECC مثل بروتوكول Wang (2014)، وبروتوكول Odelu (2015) حيث أنها غير آمنه ضد الإختراق المتميز من الداخل واختراق إعادة الإرسال. بالإضافة الى ذلك، أظهرت نتائج المحاكاة التي قمنا بها أن التكلفة الحسابية وتكلفة الاتصال للبروتوكول المقترح باستخدام خوارزمية ECCمنخفضة وأقل بالمقارنة مع بروتوكول (Sutrala et al., 2016) الذي يستخدم خوارزمية RSA، حيث أن مجموع التكلفة الحسابية تساوي 0.0028641ms وتكلفة الإتصال تساوي2112 bit مقارنة مع البروتوكول باستخدام RSA الذي تكلفته الحسابية وتكلفة الإتصال تساوي 21.6250267ms، 5632 bit على التوالي. بالإضافة إلى أن البروتوكول المقترح أفضل من بروتوكول Wang (2014) الذي يستخدم خوارزمية ECC من ناحية التكلفة الحسلبية وتكلفة الإتصال. كما أن البروتوكول المقترح يعتبر مناسبا عمليا للأجهزة النقالة في بيئة الخادم-العميل.

الكلمات المفتاحية : أمن المعلومات، مصادقة المستخدم، ECC، RSA.

Chapter : Introduction

Overview of Remote User Authentication :

With the tremendous development of information technology and transfer data and information over Internet, security has become very important to overcome the threats to the network by attackers. In client-server environments, authentication protocols are one of the security mechanisms that have an important role in the protection of sensitive and important information against attackers and intruders by providing a variety of services. Remote user authentication is the mechanism by which the remote server identifies the legitimacy of the user on the internet (Kalra and Sood, 2013). Smart cards are become widely used in the electronic e-commerce applications for remote user authentication due to their efficiency, low cost, and portability (Kalra and Sood, 2013; Odelu et al., 2015). Remote user authentication based on either symmetric or asymmetric (public key) cryptographic, symmetric cryptographic use the same key for both encryption and decryption, so that the parameters are inexpensive in term of computation cost. Asymmetric (public key) cryptographic use two different keys; one private key and other public key for encryption and decryption, and it involves the calculation of exponential operations which need a lot of processing time, so that the authentication protocols that based on public key cryptography are very expensive (Kalra and Sood, 2013; Stallings, 2006; Sutrala et al., 2016)

RSA and ECC algorithms :

A number of algorithms have been proposed for public key cryptography, some of them are Rivest-Shamir-Adleman (RSA) algorithm that is used for secure data transmission. The RSA algorithm uses two different keys; public key for encryption and private key for decryption which is kept secret (Stallings, 2006).The key size in RSA is 1024 bits and it does involve calculation of exponential operations making it a slow algorithm. Therefore, RSA is less commonly used to encrypt user data.

Another algorithm for public-key cryptography is the Elliptic Curve Cryptography algorithm (ECC). ECC is a public-key encryption technique based on elliptic curve theory that can be used to generate faster, smaller and more efficient cryptographic keys (the key size is 160-512 bits comparison to RSA 1024-15360 bits). An elliptic curve is the set of points that satisfy the specific mathematical equation: $y^2 = x^3 + ax + b$ (Odelu et al., 2015; Stallings, 2006). ECC generates keys through the properties of the elliptic curve equation instead of the traditional method of generation as the product of very large prime numbers (RSA).

Problem definition :

A.K. Sutrala, A.K. Das, V. Odelu, M. Wazid, S. Kumari, (2016) have been proposed a Secure anonymity-preserving password-based user authentication and session key agreement scheme for Telecare Medicine Information Systems (TMIS) to overcome the weaknesses in Amin and Biswas scheme (2015) such as privileged insider attack, user impersonation attack, replay attack and offline password guessing attack, and to be strong against various known attacks.

Although the strength of (Sutrala et al., 2016) scheme, but this scheme vulnerable in term of the dependency of RSA algorithm which its use two large prime number to encryption and decryption processes that take a lot of processing time (high computation and communication cost) making it slow scheme.

Research objective :

In many of researches, security mechanisms such as authentication protocols have an important role in protection sensitive and important information against attackers and intruders, some of these researches focus on security side to increase strength the schemes against known attacks, and some others focus on cost side to decrease computation and communication cost. In this study, the main goal is to reach better security at low cost. So that, we focused on increasing security degree against known attacks and decreasing computation and communication cost.

Thesis Contribution :

In this study, we improved (Sutrala et al., 2016) scheme by implementing ECC algorithm in generation keys, and implementing ECC in encryption and decryption processes instead of RSA algorithm. The main principle of using ECC depending on the length of key, that's the length of key in ECC is much smaller than in RSA, and due to the elliptic curve discrete logarithm problem (ECDLP) in ECC that's for given a point P □ E_p (a, b) and scalar k □ Z_p, computing $Q = kP$ is relatively easy and simple. But, given point P □ E_p (a, b) and Q □ E_p (a, b), it's computationally hard to derive the scalar k □ Z_p.

Thesis Outline :

Chapter 2: Provides background about the security in general and explained security services and security attacks, then discuss remote user authentication and public cryptography algorithms RSA and ECC, and comparison between them.

Chapter 3: Shows the recent related works that discuss remote user authentication protocols.

Chapter 4: Explained and discuss the improvement on (Sutrala et al., 2016) scheme with details, and review the (Sutrala et al., 2016) scheme.

Chapter 5: Presents the security analysis of the improved proposed scheme and it shows the results of the simulation, then compared it with (Sutrala et al., 2016) scheme and other schemes that use ECC algorithm.

Chapter 6: Conclusion the thesis and presents the future work.

Chapter 2 : Security Algorithms Background

Introduction :

In the recent years, with the increased of the dependency on the technology in all fields of life, especially in communication systems where personal and business information being shared on computer network every day, security becomes one of the most important aspects of networking. Network security involves policies adopted to prevent and monitor unauthorized access, modification, denial of a computer network, misuse, and network accessible-recourses. A security process in networks divides to authentication, authorization, confidentiality, and integrity processes, its start with the authentication process to verify the users to allow them access to information and programs within their authority. This chapter discusses remote user authentication and the public key cryptography in details.

Remote User Authentication :

In most computer security contexts, user authentication is a process that allows a device to verify the identity of the user who wants to connect to a network resource (Gurung, 2016; Yu, 2012).

Remote users can be authenticated via various ways:

Something the individual knows: the user submits password, personal identification number or answer to a prearranged set of questions to be authenticated to a remote system and to access and use system resources (Garrett, 2016).

Something the individual has: this include the electronic key card, smart card, and physical keys. All authentication information stored in the smart card and then it's entered into reader machine to verify this information.

Something the individual is: it's called static biometric where the remote user is authenticated by fingerprint, retina, and face.

Public Key Cryptography :

Asymmetric or public key cryptography is a technique that uses two key; public and private key to encrypt and decrypt data. One key can be shared with everyone; it is known as a public key and other key kept secret; it is known as a private key (Stallings, 2006; Yu, 2012). There are a number of algorithms proposed for public key cryptography; Rivest-Shamir-Adleman (RSA), and Elliptic Curve Cryptography (ECC) algorithms.

-Rivest-Shamir-Adelman (RSA) :

RSA algorithm is one of the popular public key cryptosystems that used for secure data transmission. The algorithm uses two different keys; one public key for encryption and the other private key for decryption which is kept secret (Stallings, 2006; Yu, 2012). RSA algorithm divided into three phases; key generation, encryption, and decryption.

Generation key :

Before encryption and decryption of a message between users, public and private key pair must be generated. The generation key process for each user done as follows:

Select two prime number p and q such that p!= q.

Compute the number $n = p \times q$ *and* $ø\ (n) = (p – 1) \times (q – 1)$. $ø\ (n)$ is the phi function defined by $ø(n) = |\ \{a|0 < a < n,\ gcd\ (a,\ n) = 1\}\ |$ where *gcd* is the greatest common divisor, this phi function defined as the number of positive integers less than *n* and relatively prime to *n*.

Select *e*, with $gcd\ (e,\ ø\ (n)) = 1$, such that $1 < e < ø\ (n)$.

Compute *d* such that *d* □ *e* □^{1} *(mod (ø (n)),* and determine $d \times e = 1\ (mode\ ø\ (n))$.

Encryption :

An encryption process in RSA algorithm done as follow: suppose that we have two users *E* and *B*. *E* wants to send a message *M* to user *B*. first user *E* convert *M* into *k* number of plaintext blocks $Mi\ (1 < i < k\ )$. User *E* then computes the corresponding ciphertext blocks $Ci\ (1 < i < k\ )$, where $Ci = M\ (mod\ n)$ using the public key $(e,\ n\ )$ of the user *B*. finally user *E* sends the encrypted message blocks *Ci* to user *B* via a public channel (Stallings, 2006).

Decryption :

In decryption process, user *B* decrypts the ciphertext message $Ci = [C1,\ C2,\ …,\ Ci]$ using its own private key $(d,\ n)$ to get the original corresponding plaintext blocks M_i as $M_i = C_i\ (mod\ n)$. After that, user *B* concatenates all decrypted ciphertext blocks *Mi* to get the message *M*.

-Elliptic Curve Cryptography (ECC) :

ECC is a public key encryption technique based on elliptic curve theory which is the set of the point that satisfies the specific mathematical equation: $y^2 = x^3 + ax + b$ (Stallings, 2006; Yu, 2012).

Before talking about the ECC encryption/decryption, we will discuss in a brief elliptic curve equation.

-Elliptic Curve with Finite Field :

A finite field or Galois Field (GF) is a field that contains a finite number of elements which called its order, such as the fields of prime order GF(p) (also denoted Z_p or F_p), which is a field of prime number of order p (size of field). The finite field with elliptic curve defined as follow:

Let *a* and $b \in Zp$, where $Zp = \{0,\ 1, …\ p - 1\ \}$ and $p > 3$ be a prime, such that $4a^3 + 27b^2\ != 0\ (mod\ p)$. A non-singular elliptic curve $y^2 = x^3 + ax + b$ over the finite field $GF\ (p)$ is the set $E_p\ (a,\ b)$ of solutions $(x,y) \in Zp \times Zp$ to the congruence $y^2 = x^3 + ax + b\ (mod\ p)$, where *a* and $b \in Zp$ are constants such that $4a^3 + 27b^2$ □ $0\ (mod\ p)$. Together with a special point @ called the point at infinity or zero points (Odelu et al., 2015; Stallings, 2006; Yu, 2012).

Let $P\ (X_p,\ Y_p)$ and $Q\ (X_Q,\ Y_Q)$ be two point on an elliptic curve $y^2 = x^3 + ax + b\ (mod\ p)$, and let G be the base point on $E_p\ (a, b)$ whose order be n, that is $nG = G + G + \ldots + G\ (n\ times) =$ Ꝋ, $R\ (X_R,\ Y_R) = P + Q$ is computed as follows:

$X_R = (\square^2 - X_p - X_Q)\ (mod\ p),$

$Y_R = (\square(X_p - X_R) - Y_p)\ (mod\ p),$

Where $\square = (Y_Q - Y_p)/(X_Q - X_p)\ (mod\ p)$, if $P\ \square\ Q$ and $\square = (2X_p^2 + a)/(2Y_P)\ (mod\ p)$, if $P = Q$. Figure 2-1 explain the elliptic curve.

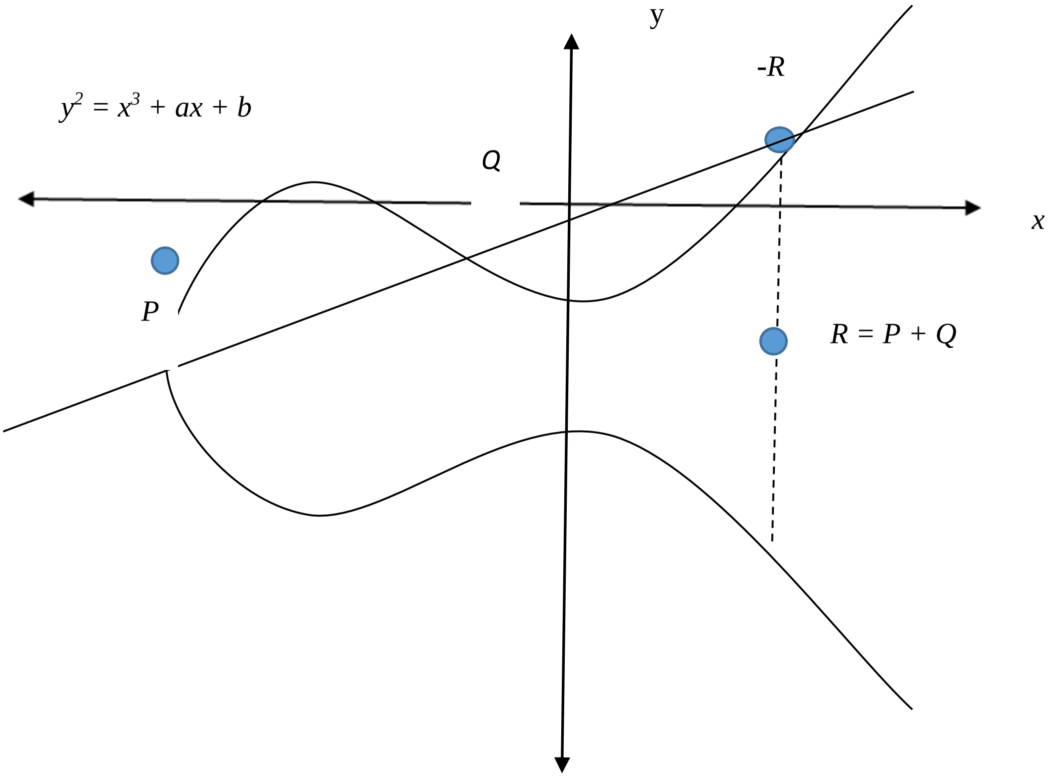


Figure 2-1. Elliptic Curve.

-ECC Encryption/Decryption :

Before encryption and decryption process in elliptic curve algorithm, the plaintext message *m* will be encoded to be sent as an elliptic curve point $P_m \in Ep\ (a, b)$. This point P_m then will be encrypted as a cipher text and then subsequently decrypted.

Key generation :

In the key generation process, user *B* has the elliptic curve $Ep\ (a, b)$ defined over a finite field $GF\ (p)$, and a base point $G \in E_p\ (a, b)$ whose order is n, that $nG = \mathcal{O}$. User *B* selects a private key nB randomly on the interval $[\ 1,\ n - 1\ ]$ and computes his/her public key $P_B = nB\ G$.

Encryption :

To encrypt the message P_m, user *E* chooses a random integer k in the interval $[1, n - 1]$ and produce the cipher text C_m consisting of the pair of points C_1 and C_2 where $C_m = (C_1, C_2)$, with $C_1 = kG$, and $C_2 = P_m + kP_B$, where P_B is the public key of user *B*. then user *E* sends the cipher text C_m to user *B* via public channel.

Decryption :

To decrypt the cipher text C_m, user *B* multiplies the first point C_1 with his/her private key nB and obtain $C_1nB = kGnB = k\ P_B$. After that, user *B* recovers the plaintext message P_m as $C_2 – nB\ C_1 = (P_m + k\ P_B) – nB\ (\ kG) = P_m + k\ P_B – k\ P_B = P_m$.

-Elliptic curve discrete logarithm problem (ECDLP) :

In elliptic curve algorithm, there is a problem known as Elliptic Curve Discrete Logarithm Problem (ECDLP) makes the schemes that use ECC secure and strong against known attacks. The problem defined as follow: given a point P □ $E_p\ (a, b)$ and scalar k □ Z_p, computing $Q = kP$ is relatively easy and simple. But, given point P □ $E_p\ (a, b)$ and Q □ $E_p\ (a, b)$, it's computationally hard to derive the scalar k □ Z_p, where $Q = kP$ (Hankerson et al., 2006; Stallings, 2006).

ECC versus to RSA :

ECC and RSA algorithms have several clear different that explained in many researches, and proved that ECC better than RSA in remote user authentication protocols. For example, the key size 160-bit in ECC is provided equivalent security compared with the key size 1024-bit in RSA (Lee and Lee, 2016). In addition, ECC has the elliptic curve discrete logarithm problem (ECDLP) that is to determine k given P and Q where $Q = kP$. It's computationally easy to calculate Q given k and P, but it is computationally infeasible to determine k given Q and P, this problem increase the security of ECC and make the schemes that use ECC stronger than the schemes that use RSA.

In (Gura et al., 2004), Gura et al. have a result through implementing assembly language for ECC and RSA. The result shows that a 160-bit requires 0.8s for point multiplication in ECC, whereas 1024-bit requires 0.43s for public key operation and 10.99s for private key operation in RSA. In addition, (Lee and Lee, 2016) shown that the key size of 160 bit in ECC and the key size of 1024 bit in RSA provide equivalent security level that equals 80, where 80 its mean that the attacker needs to 80 operation to do the attack process. Table 2-1 shows the comparison of key length, security level and MIPS years to attack for ECC and RSA, where MIPS is Million Instruction Per Second that means the speed of a device that's used in attack process.

As result, ECC better than RSA and provides best security and low computation and communication cost.

Table 2-1: Comparison of key size and security level for ECC and RSA (Lee and Lee, 2016)

Security level	MIPS years to attack	ECC key size	RSA key size
80	10^{12}	160	1024
112	10^{24}	224	2048
128	10^{28}	256	3072
192	10^{47}	384	7680
256	10^{71}	512	15360

Chapter 3 : Related Works

Introduction :

Many of researches have been presented in term of remote user authentication protocols.

The authentication process is done in various ways, some of these researches used one-way hash function and bitwise XOR operations, and other used two-way that use the smart card and three-way that use biometrics authentication. This chapter reviews the researches that apply these various ways and have a relation to our proposed scheme.

Works used one-way hash function and bitwise XOR operations :

L. Lamport (1981) proposed the first password-based authentication scheme, in which was used in authentication of legitimate users (Lamport, 1981). But the scheme has several drawbacks making it insecure such that passive attacks and an adversary can perform on-line or off-line password guessing attacks and stolen verifier attacks (Wang, 2014). The first multi-server password-based authentication scheme was proposed by Ford and Kaliski (2000). The scheme splits the password information among multiple servers and therefore a malicious user cannot compromise the password by launching various attacks (Ford and Kaliski, 2000). The scheme uses public-key systems to achieve authentication and therefore is computationally expensive.

Das, Odelu, Goswami,(2014) have been proposed a Robust and Effective Smart-Card-Based Remote User Authentication Mechanism using Hash Function (Das et al., 2014). This scheme used only one-way hash function and

bitwise XOR operations. The results show that the scheme is secure against possible known attacks and perform better than other existing schemes in term of communication and computation overhead. But in earliest years many of researches proved that the remote user authentication schemes using biometrics are better than one-way hash function and bitwise XOR operations, such as (Odelu et al., 2015; Sutrala et al., 2016). In addition, the scheme insecure against impersonation attack and brute force attack.

Works that use smart card and ECC algorithm :

Kalra and Sood (2013) proposed an efficient multi-server authentication scheme uses smart cards based on elliptic curve cryptography (ECC) (Kalra and Sood, 2013). The scheme improved Sood et al., (2011) schema by increasing the security and reducing the computation cost due to the use of ECC algorithm. Although the improvement of this scheme, it's still insecure against some attacks such as impersonation attack.

Zhang, et, al., (2014) have been proposed a novel remote user mutual authentication scheme for multi-server environments using elliptic curve cryptography to provide secure communication in a mobile environment (Zhang et al., 2014). The proposed scheme used ECC and secure hash function to protect the communication between entities; user, server and registration center (RC).

The security analysis shows that the proposed scheme solves a various type of the security problems in other schemes such as (Yang and Chang, 2009) scheme, and it's suitable for a multi-server environment with low-power mobile devices.

(Odelu, Das, and Goswami; 2015) proposed an efficient ECC-based privacy-preserving client authentication protocol with key agreement using a smart card. The protocol proposed to overcomes the weaknesses and security pitfalls in Wang's scheme such as user anonymity, off-line password guessing attack, credential leaking and smart card lost/stolen verifier attack (Wang, 2014).The proposed scheme based on ECC, so it provides low computational and communication costs and establishes a secret session key between a client and server for their future secure communication between them.

Through the formal and informal security analysis, they have shown that the proposed scheme is secure against possible known attack including the attacks found in Wang's scheme and the simulation results indicate that the scheme is secure against passive and active attacks. Although the results were proved the strength of the protocol, it's based on single server environment that makes it less security compared to the protocols that based on multi-server.

D. Giri, T. Maritra, R. Amin, et al., (2015) proposed a new authentication scheme for Telecare Medical Information System (TMIS) (Giri et al., 2015) to overcome the weaknesses in (Khan and Kumari, 2013) scheme, where they have shown that Khan and Kumari's scheme is valuable against off-line password guessing attack. Although the scheme improved the security and efficiency, it does not provide or support most of the functionality features such as off-line password guessing attack, strong reply attack, user impersonation attack, and stolen smart card attack (Sutrala et al., 2016).

Troung, et, al., (2017) have been overcome the limitation in (Yeh, 2014) scheme which it cannot achieves mutual authentication and session key agreement. The improvement in (Truong et al., 2017) scheme done by using elliptic curve cryptosystem as security foundation in the multi-server environment.

Works that use smart card and RSA algorithm.:

Amin and Biswas (2015) proposed a scheme for improving (Giri et al., 2015) scheme, but this scheme vulnerable to some attacks such as user impersonation attack and its failure to protect strong reply attack (Sutrala et al., 2016). In addition its slow scheme due to depending on RSA algorithm.

Sutrala, Das, Odelu, et al., (2016) proposed a new RSA-based user authentication scheme for Telecare Medical Information System (TMIS) which overcomes the security drawbacks found in (Amin and Biswas, 2015) scheme such as privileged insider attack, user impersonation attack, reply attack and also offline password guessing attack. In addition, the scheme preserves the user anonymity property and the security analysis shows the robustness of the scheme against the various known attacks.

Although the strength of the scheme, it is very slow due to using RSA public-key cryptography that involves calculation of exponential operations which needs a lot of processing time.

Chapter 4 : The Improved Proposed Scheme

Introduction :

The goal of the improvement that we proposed is to increase the security and decrease the communication and computation cost of the (Sutrala et al., 2016) scheme. We apply the ECC in the scheme instead of RSA and we updated some phases to achieve better performance. This chapter shows the proposed improvement on (Sutrala et al., 2016) scheme and explains it in details for each phase.

The improved proposed scheme for (Sutrala et al., 2016) scheme :

(Sutrala et al., 2016) scheme is a scheme to authenticate client and server in network communication domain and establishes the communication between them by session key agreement. The scheme consists of six phases; initialization, registration, login, authentication, password change, and identity change phases. In this section we review each phase in (Sutrala et al., 2016) scheme and proposed the improvement for these phases and how it work in details.

-Initialization phase :

In the initialization phase for (Sutrala et al., 2016) scheme, the server S selects two large distinct prime numbers p and q, and compute $n = p \times q$. then, S chooses a prime number e and an integer d such that $e \times d = 1$ (mod $(p - 1)$ $(q - 1)$), that is $d = e^{-1}$ (mod $(p - 1)$ $(q - 1)$). Server S marks d as its private key and keeps it secret, where it publishes the corresponding e as the public key. A one-way hash function h(.) is chosen by S.

In the proposed scheme, we applied elliptic curve cryptography in generate keys as follow:

In the beginning of communication, the server S chooses the elliptic curve E_p (a,b) and the generator point P, then S choose secret key d_s and computes public key $P_{pub} = d_s \times P$.

Finally, server S publishes the public parameter < P_{pub}, P> to the public domain.

-Registration phase :

In registration phase for (Sutrala et al., 2016) scheme, a new user U_i, perform registration at server S as follow:

Step R1. U_i selects an identity ID_i, password pwi and a random number bi of his/her choice. U_i then compute the pseudo-identity idbi = h (ID_i || b_i) and send the registration request message <idb_i> to the telecare server S securely.

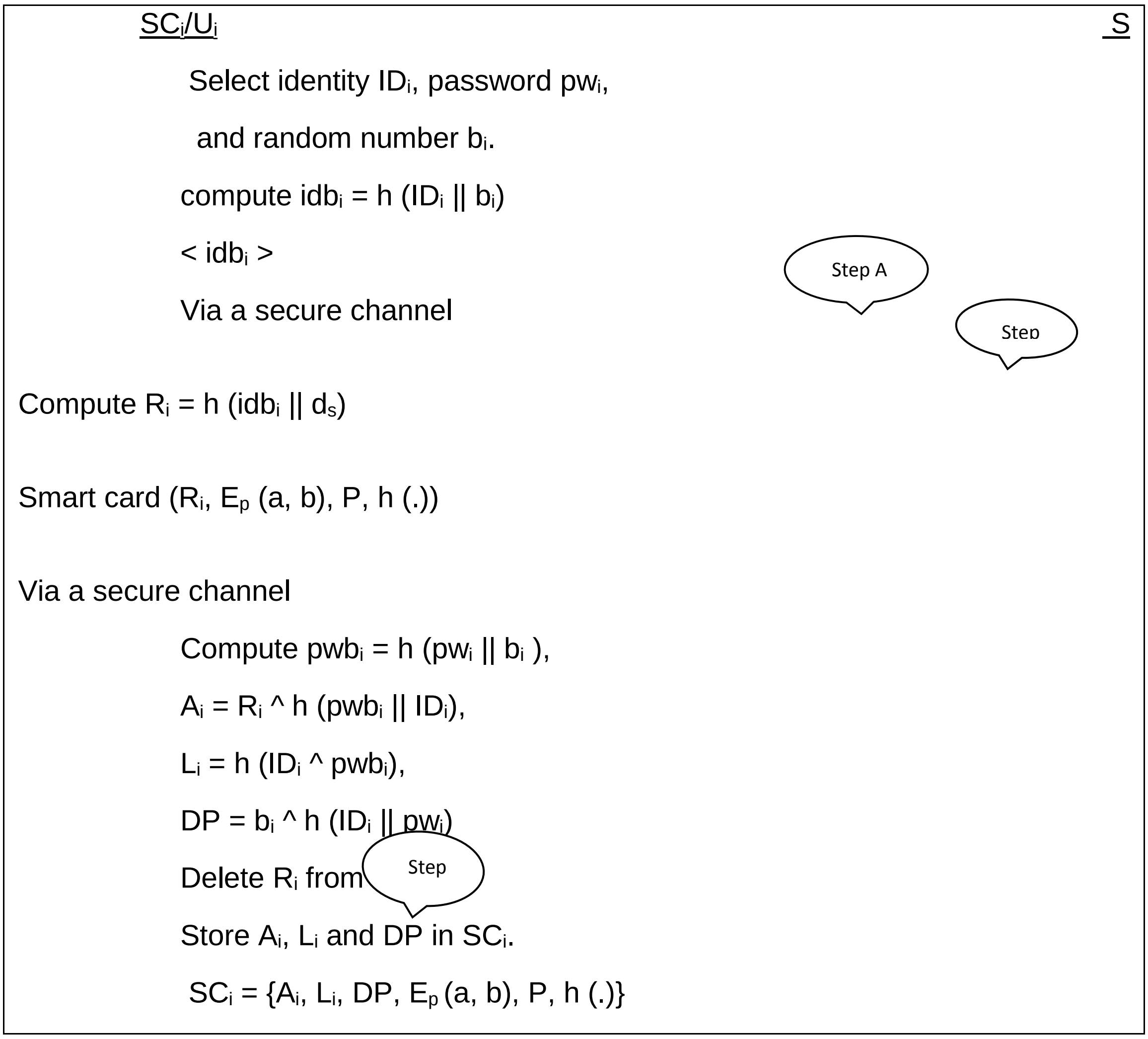
Step R2. after receiving the registration request message <idb_i> from user U_i, server S computes R_i = h (idb_i || d) where d is the private key of server S. a smart card SC_i for user U_i containing the information {R_i, e, n, h(.)} is generated by S and further sends it to U_i via a secure channel.

Step R3. After receiving the smart card CS_i, U_i computes the pseudo-password $pwb_i = h(pw_i \| b_i)$ and $A_i = R_i \wedge h(ID_i \| pwb_i)$, $L_i = h(ID_i \| pwb_i)$, $DP = b_i \wedge h(ID_i \| pw_i)$. Finally, U_i stores A_i, L_i, and DP into memory of smart card CS_i. U_i then remove R_i from the memory of smart card U_i. Thus, the smart card CS_i contains the information $\{A_i, L_i, Dp, e, n, h(.)\}$. See figure 4-1.

In the proposed scheme, we changed some parts of this phase as follow:

Step A: We used the generated key by ECC in server side, that's the server S compute $R_i = h(idb_i \| d_s)$, where d_s is the private key of server S. in addition, a smart card SC_i for user U_i in our improved scheme, will contain the information $\{R_i, E_p(a, b), P, h(.)\}$ is generated by server S, and then sends it to user U_i via secure channel.

Step B: Finally, after U_i/SC_i computed all the operation in Step R3 for (Sutrala et al., 2016) scheme, the smart card SC_i will contains the information $\{A_i, L_i, DP, E_p(a, b), P, h(.)\}$. Figure. 4-1 explains the registration phase for the scheme after improvement.



SC_i/U_i S

Select identity ID_i, password pw_i,

and random number b_i.

compute $idb_i = h(ID_i \| b_i)$

$< idb_i >$

Via a secure channel

Compute $R_i = h(idb_i \| d_s)$

Smart card $(R_i, E_p(a, b), P, h(.))$

Via a secure channel

Compute $pwb_i = h(pw_i \| b_i)$,

$A_i = R_i \wedge h(pwb_i \| ID_i)$,

$L_i = h(ID_i \wedge pwb_i)$,

$DP = b_i \wedge h(ID_i \| pw_i)$

Delete R_i from

Store A_i, L_i and DP in SC_i.

$SC_i = \{A_i, L_i, DP, E_p(a, b), P, h(.)\}$

Figure.4-1. Registration phase of the proposed scheme.

-Login phase :

In login phase for (Sutrala et al., 2016) scheme, the user U_i first insert his/her smart card SC_i into a card reader and then provides his/her identity ID_i and password pw_i. Login phase contains the following steps:

Step L1. SC_i computes bi' = DP ^ $h(ID_i \| pw_i)$, $idb_i' = h(ID_i \| b_i')$, $pwb_i' = h(pw_i \| b_i')$, $L_i' = h(ID_i \| pwb_i')$, $R_i' = A_i$ ^ $h(pwb_i' \| ID_i)$ and check whether L_i stored in smart card SC_i matches with computed L_i'. If it does not match, SC_i rejects the user U_i and terminates this phase. Otherwise, it proceeds further to the next step.

Step L2. SC_i generates a random nonce N_1, computes $C_i = h(pwb_i' \| N_1 \| R_i' \| T_1)$, $D_i = h(idb_i' \| pwb_i' \| R_i')$ ^ N_1, $B_i = h(idb_i' \| pwb_i' \| N_1 \| R_i' \| T_1)^e$ (mod n), where T_1 is the current timestamp generated by SC_i, and then send the login request message $<C_i, B_i, D_i >$ to S via public channel.

In the proposed scheme, the login phase works as (Sutrala et al., 2016) scheme expect the step L2 where we encrypted the information idb_i', pwb_i', N_1, R_i' and T_1 by elliptic curve cryptography and generate public and secret keys for the user U_i/SC_i. The user U_i use the secret key for encryption process and public key to send it to the server S. the changes that done in our improved scheme are as follow:

Step A: SC_i chooses private key x_i, generates a random nonce N_1, and calculates secret key $K_1 = x_i \times P_{pub}$ where K_1 used for the encryption then computes $F_i = E_{K1}$ $(idb_i' \| pwb_i' \| N_1 \| R_i' \| T_1)$, $C_i = h\ (pwb_i' \| N_1 \| R_i' \| T_1)$, $D_i =\ (idb_i' \| pwb_i' \| R_i')$ ^ N_1, $X_i = x_i \times P$ where T_1 is the current timestamp generated by SC and X_i is the public key for user U_i, and then sends the login request message $<C_i, F_i, D_i, X_i >$ to server S via public channel. See figure 4-2.

-Authentication and key agreement phase:

The authentication and key agreement phase in (Sutrala et al., 2016) scheme very important to prove the legitimacy of the user. So that, after receiving the login request message $< C_i, B_i, D_i >$ by the server S, it performs the following steps to achieve mutual authentication and session key agreement with the user U_i:

Step AK1. S obtain the information $(idb_i, pwb_i, N_1, R_i, T_1)$ by decrypting B_i using its own private key d as B_i^e (mod n) = $(idb_i, pwb_i, N_1, R_i, T_1)$. S then generates the current timestamp T_1^* and check the condition $| T_1^* - T_1 | \leq \Delta T$. if this condition does not hold, the server S abort this phase; otherwise, it proceeds to the next step.

Step AK2. S computes $R_i^* = h\ (idb_i \| d)$ and checks if R_i^* matches with R_i which is computed from Bi. If it does not match, S aborts the login request and stops. Otherwise S computes $N_1^* = D_i$ ^ $h\ (idb_i \| pwb_i \| R_i^*)$ and check if N_1^* matches with N_1. If it does not match, S abort this phase. Otherwise, S proceeds to the next step.

Step AK3. S compute $C_i^* = h\ (idb_i \| N_1^* \| R_i^* \| T_1)$ and checks the condition $C_i^* = C_i$. If it does not hold, S abort this phase and stops. Otherwise, S believes the authenticity of the user U_i and proceeds to the next step.

Step AK4. S generates a random nonce N_2 and computes $SK = h(idb_i \| pwb_i \| N_1^* \| N_1 \| T_1 \| T_2)$, $N_3 = N_1^*$ ^ N_2, $K_i = h(SK \| R_i^* \| N_2 \| T_2 \| N_1 \| T_1)$, where T_2 is the current timestamp generated by the server S. S then send the authentication request message $< N_3, K_i, T_2 >$ to the user U_i via a public channel.

Step AK5. whenever the smart card SC_i receives the authentication request message < N_3, K_i, T_2 > for U_i, it checks the condition $| T_2^* - T_2| \leq \Delta T$, where T_2^* is the current timestamp generated by SC_i. if the condition does not hold, SC_i aborts the session. Otherwise, it derives $N_2^* = N_1$ ^ N_3, $SK = h$ (idb_i' || pwb_i' || N_1 || N_2^* || T_1 || T_2), $K_i^* = h$ (SK || R_i' || N_2^* || T_2 || N_1 || T_1), and checks if $K_i^* = K_i$. If this condition does not hold, SC_i/U_i thinks that the authentication request message has been tampered and aborts the session. Otherwise, the mutual authentication property of the protocol is satisfied and go to the next step.

Step AK6. smart cart SC_i computes SKV = h(SK || R_i' || idb_i || T_3 || T_2), where T_3 is the current timestamp generated by SC_i, and then sends the authentication reply message < SKV, T_3 > to S using a public channel.

Step AK7. After receiving the authentication reply message < SKV, T_3 >, the server S checks the condition $| T_3^* - T_3 | \leq \Delta T$, where T_3^* is the current timestamp generated by the server S. if it does not match, S immediately discard the message and abort the connection. Otherwise, it computes $SKV^* = h$ (SK^* || R_i^* || idb_i || T_3 || T_2) and checks if $SKV^* = SKV$. If this condition holds, the session key verification of the protocol is achieved, and both parties U_i and S can now communicate securely using the computed session key SK. See figure 4-2.

In the proposed scheme, the authentication and key agreement phase works as (Sutrala et al., 2016) scheme expect that in step AK1 we decrypted the information idb_i, pwb_i, N_1, R_i' and T_1 by elliptic curve cryptography instead of the decryption by RSA algorithm as follow:

Step B. After receiving the login request message <C_i, F_i, D_i, X_i >, the server S decrypts F_i using its own secret key $K_2 = k_s \times X_i$ as $F_i = D_{K2}$ (idb_i, pwb_i, N_1, R_i', T_1) and obtains the information (idb_i, pwb_i, N_1, R_i', T_1). S generates the current timestamp T_1^* and checks the condition $|T_1^* - T_1| \leq \Delta T$. if the condition does not hold, the server S aborts this phase; otherwise, it goes to the next step (Step AK2 in (Sutrala et al., 2016) scheme). Figure 4-2 explain the login and authentication and key agreement phase of the scheme after improvement.

-Password change phase :

The password change phase in the proposed scheme is the same as in (Sutrala et al., 2016) scheme. If user U_i wants to change his/her password, first U_i insert the smart card SC_i in a card reader, and then provides the inputs such as his/her identity ID_i and password pw_i^{old}. SC_i then performs the following operations.

Step P1. SC_i derives $b_i' = DP$ ^ h(ID_i || pw_i^{old}), $idb_i' = h(ID_i$ || $b_i')$, $pwb_i^{old} = h(pw_i^{old}$ || $b_i')$, $L_i' = h(ID_i$ ^ $pwb_i^{old})$, $R_i' = A_i$ ^ $h(pwb_i^{old}$ || $ID_i)$,and checks whether L_i stored in the smart card matches with the computed L_i'. If it does not match, the SC_i reject the user U_i and stops. Otherwise, SC_i prompts U_i to input the new password pw_i^{new} and proceeds to the next step.

Step P2. On inputting the new password pw_i^{new}, SC_i computes $pwb_i^{new} = h$ (pw_i^{new} || b_i'), $A_i^{new} = R_i'$ ^ h(pwb_i^{new} || IDi), $L_i^{new} = h(ID_i$ ^$pwb_i^{new})$, $DP_i^{new} = b_i'$ ^ h(ID_i || pw_i^{new}). SC_i then replaces the parameters A_i, L_i, and DP_i with A_i^{new}, L_i^{new}, and DP_i^{new} into its memory.

-Identity change phase :

In (Sutrala et al., 2016) scheme, if a user U_i wants to changes his/her identity, first he/she inserts the smart card SC_i in a card reader and then provides inputs such as his/ her identity ID_i, password pw_i. SC_i then performs the following steps:

Step I1. SC_i computes $b_i' = DP \wedge h(ID_i || pw_i)$, $idb_i' = h(ID_i || b_i')$, $pwb_i' = h(pw_i || bi')$, $L_i' = h(ID_i \wedge pwb_i')$, $R_i' = A_i \wedge h(pwb_i' || ID_i)$, and checks whether L_i stored in the smart card matches with the computed Li'. If it does not match, the SC_i reject the user U_i and stops. Otherwise, SC_i prompts to input the new identity ID_i^{new} and proceeds to the next step.

Step I2. after reading the new identity ID_i^{new}, SC_i generates a random nonce N_1 and computes $idb_i^{new} = h(ID_i^{new} || bi')$, $C_i = h(pwb_i || N_1 || R_i' || T_1)$, $D_i = h(idb_i' || pwb_i') \wedge N_1$, $Bi = (idb_i' || idb_i^{new} || pwb_i' || N_1 || R_i' || T_1)$^e (mod n), where T_1 is the current timestamp, and send the message < C_i, D_i, B_i > to the server S using a public channel.

Step I3. After receiving above message, S decrypt B_i using its own private key d to obtain (idb_i', idb_i^{new}, pwb_i', N_1, R_i', T_1). S then generates the current timestamp T_1^* and checks the condition. If this condition does not hold, S immediately abort this phase. Otherwise, it computes $R_i^* = h(idb_i || d)$ and check if $R_i^* = R_i$. If it does not match, S abort this phase and stops. Otherwise, it compute $N_1^* = D_i \wedge h (idb_i || pwb_i)$ and checks if $N_1^* = N_1$. If there is a mismatch, S discards the message and terminates this phase. Otherwise, S proceeds to the next step.

Step I4. S computes $C_i^* = h(pwb_i || N_1^* || R_i^* || T_1)$ and checks if $C_i^* = C_i$. If there is a mismatch, S abort this phase and stops. Otherwise, S trusts that the request received from the legitimate user U_i, and proceeds further.

Step I5. S computes $R_i^{new} = h(idb_i^{new} || d)$, $N_2 = N_1 \wedge R_i^{new}$, $K_i = h(R_i* || N_1 || T_2)$ where T_2 is the current timestamp generated by the server S and then sends the message < K_i, N_2, T_2 > to the user U_i via public channel.

Step I6. Upon receiving the message < K_i, N_2, T_2 > from S, SC_i generates the current timestamp T_2^* and checks whether $| T_2^* - T_2 | \leq \Delta T$ or not. If it does not hold, SC_i discards the message and terminates this phase. Otherwise, SC_i computes $K_i^* = h (R_i' || N_1 || T_1)$ and checks whether $K_i^* = K_i$ or not. If the condition turns false, SC_i assumes that the message has tampered or it is not from the legitimate server and terminates this phase.

Step I7. Finally, SC_i computes $R_i^{new} = N_2 \wedge N_1$, $A_i^{new} = R_i^{new} \wedge h (pwb' || ID_i^{new})$, $DP_i^{new} = b_i' \wedge h(ID_i^{new} || pw_i)$, $L_i^{new} = h(ID_i^{new} \wedge pwbi')$, and stores A_i^{new}, L_i^{new}, and DP_i^{new} into its memory by replacing A_i, L_i, and DP respectively.

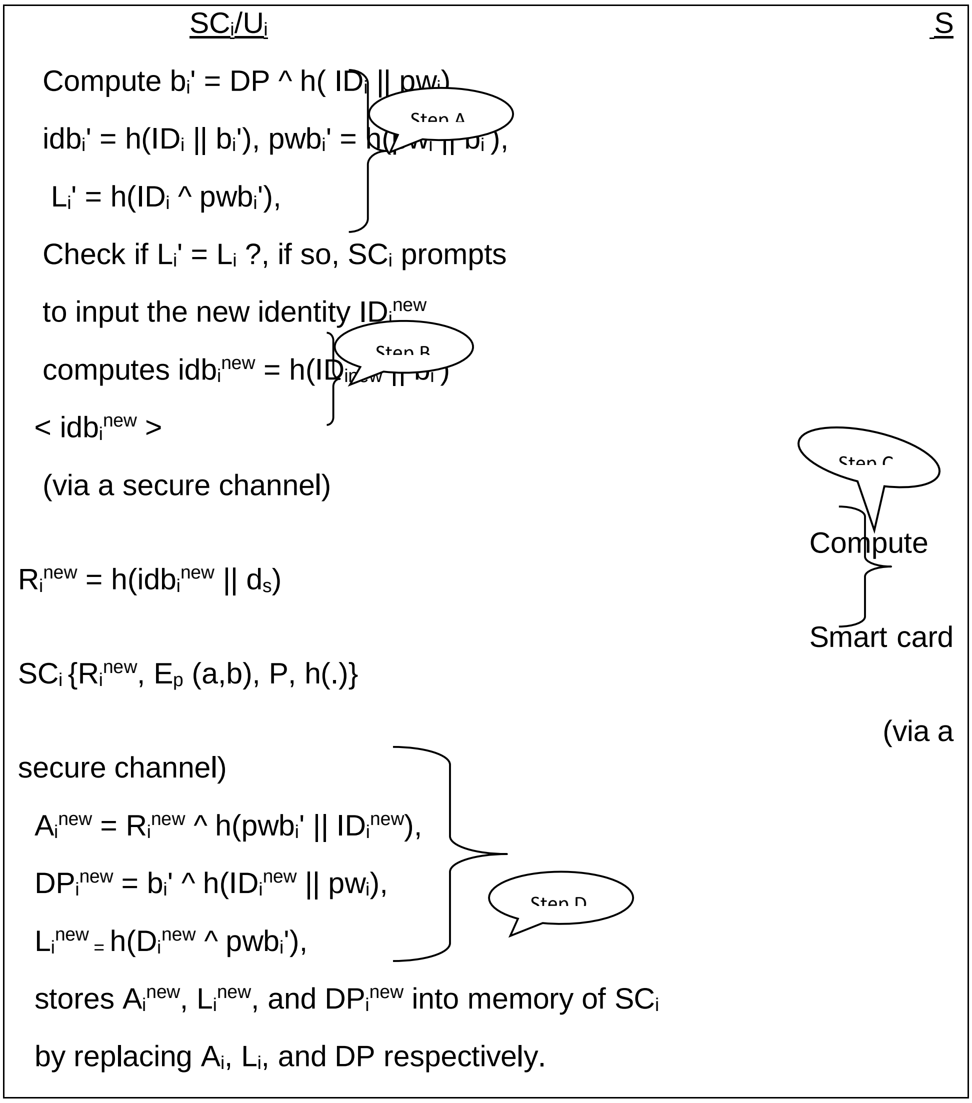
The identity change phase of (Sutrala et al., 2016) scheme contain repeated and needless operations to change the identity ID_i of a user U_i. So that, in the proposed scheme we made this phase more efficient than the identity change phase in (Sutrala et al., 2016) scheme. The improvement of this phase done as follow:

Step A. SC_i computes $b_i' = DP \wedge h (ID_i || pw_i)$, $idb_i' = h(ID_i || b_i')$, $pwb_i' = h(pw_i || b_i')$, $L_i' = h(ID_i \wedge pwb_i')$, $R_i' = A_i \wedge h(pwb_i' || ID_i)$, and checks whether L_i stored in the smart card matches with the computed L_i'. If it does not match, the SC_i reject the user U_i and stops. Otherwise, SC_i prompts to input the new identity ID_i^{new} and proceeds to the next step.

Step B. After reading the new identity ID_i^{new}, SC_i computes $idb_i^{new} = h\ (ID_{inew} \| b_i')$, and send the message < idb_i^{new} > to the server S via a secure channel.

Step C. After receiving above message, S computes $R_i^{new} = h\ (idb_i^{new} \| d_s)$ and generates a smart card SC for user U containing the information $\{R_i, E_p\ (a, b), P, h\ (.)\}$. And then sends it to U_i via a secure channel.

Step D. Finally, after receiving the smart card SC_i, U_i computes $A_i^{new} = R_i^{new} \wedge h\ (pwb_i' \| ID_i^{new})$, $DP_i^{new} = b_i' \wedge h(ID_i^{new} \| pw_i)$, $L_i^{new} = h(D_i^{new} \wedge pwb_i')$, and stores A_i^{new}, L_i^{new}, and DP_i^{new} into its memory by replacing A_i, L_i, and DP respectively. Figure.4-3, show the identity change phase of the proposed scheme.



SC_i/U_i S

Compute $b_i' = DP \wedge h(\ ID_i \| pw_i)$

$idb_i' = h(ID_i \| b_i')$, $pwb_i' = h(pw_i \| b_i')$,

$L_i' = h(ID_i \wedge pwb_i')$,

Check if $L_i' = L_i$?, if so, SC_i prompts

to input the new identity ID_i^{new}

computes $idb_i^{new} = h(ID_{inew} \| b_i')$

< idb_i^{new} >

(via a secure channel)

Compute

$R_i^{new} = h(idb_i^{new} \| d_s)$

Smart card

$SC_i\ \{R_i^{new}, E_p\ (a,b), P, h(.)\}$

(via a secure channel)

$A_i^{new} = R_i^{new} \wedge h(pwb_i' \| ID_i^{new})$,

$DP_i^{new} = b_i' \wedge h(ID_i^{new} \| pw_i)$,

$L_i^{new} = h(D_i^{new} \wedge pwb_i')$,

stores A_i^{new}, L_i^{new}, and DP_i^{new} into memory of SC_i

by replacing A_i, L_i, and DP respectively.

Figure.4-3. Identity change phase.

Chapter 5 : Simulation results analysis

Introduction :

This chapter analyses the security of the proposed scheme and shows that the scheme is secure against different known attacks through the informal security analysis.

Elliptic Curve Discrete Logarithm Problem (ECDLP) :

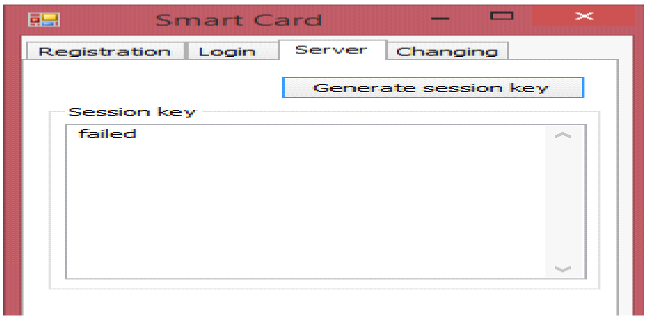
As mention in chapter two, the elliptic curve contains the discrete logarithm problem; for given scalar k € Z_p and a point P € E_p (a, b), to compute Q = kP is relatively easy. But given P, Q € E_p (a, b), it's computationally hard to derive the scalar k € Z_p (Hankerson et al., 2006; Stallings, 2006).

Security analysis:

The security analysis shows some known attacks and attempts by attackers to breach the scheme to obtain sensitive information. This analysis shows that the proposed scheme is strong against these attacks and show the failure of the attackers to obtain the sensitive information.

- Replay attack :

The replay attack is one of the known attacks that the attacker attempt to interrupts the message and resends it in later time. Suppose an adversary V intercepts the authentication message <D_i, C_i, F_i, X_i> from the legitimate user U_i, and tries to replay the request message at a later point of the time. The proposed scheme make of use the freshly generated timestamp T for login and authentication. In each communication message the fresh timestamp is send in plain text like <D_i, C_i, F_i, X_i> as well as embedded in some secret message such as F_i = (idb_i || pwb_i || N_1 || R_i || T_1) it contains timestamps T_1. Therefore, if adversary V replays the old message it will not pass the freshness test, and if V may sends new timestamp along with the old message, it will not pass the next verification test where the proposed scheme checks the embedded timestamp by $|T_1^* - T_1| \leq \Delta T$. Figure 5 shows this operation, where if $|T_1^* - T_1| > \Delta T$, it will be fails to generate shared session key between client and server. So that, the proposed scheme prevents the replay attack. See figure 5-1.



Session key

failed

Figure. 5-1. Failure of generate session key.

-Impersonation attack :

In impersonation attack, an adversary V try to impersonate the server S or user U_i. In server impersonation attack, the adversary V tries to generate <K_i, N_3, T_2>. However, it's impossible to generate $K_i = h\ (SK \,||\, R_i \,||\, N_2 \,||\, T_2 \,||\, N_1 \,||\, T_1)$ without the knowledge of pw_i, b_i, and d_s, where d_s is a private key for server S that is hard to derive due to ECDLP problem in elliptic curve algorithm. In user impersonation attack the adversary V cannot impersonate user U_i during login phase and cannot modify the login request message <C_i, F_i, D_i> due to the unknown parameters pw_i, b_i, and R_i that's used in the computation of this login request message. Also, the authentication request message cannot be modified by adversary V, that's the computation of SKV embedded SK that requires R_i, N_1, and T_1. In addition to, it's hard to know the private key d_s of the server S (ECDLP problem).

-Lost/Stolen smart-card attack :

If the adversary V stole the smart-card SC and get the information in the SC (A_i, DP, L_i, E_P (a, b), h (.)), there is no way for V to guess the identity ID and password pw. For example, $A_i = R_i \wedge h\ (pwb_i \,||\, ID)$, $R_i = h\ (idb_i \,||\, d_s)$, the adversary V cant guess the ID and pw due to the hash function, and cannot know private key d_s for server S due to the difficulty of solving ECDLP in elliptic curve algorithm. So that the proposed scheme prevents stolen smart attack.

-Session key security :

During the login phase and authentication and session key agreement phase, if the adversary V intercepts the login request message <C_i, F_i, D_i>, the authentication request message <N_3, K_i, T_2> and the authentication reply message <SKV,T_3> that are sent via a public channels during this phases. The adversary V cannot derive session key SK that's embedded in SKV due to use one-way hash function h (.). Furthermore, it is impossible to compute session key SK, where $SK = h\ (idb_i \,||\, pwb_i \,||\, N_1 \,||\, N_2 \,||\, T_1 \,||\, T_2)$, because of unknown parameters ID_i, pw_i, b_i, N_1 and N_2. Hence, the proposed scheme provides the session key agreement.

-Mutual authentication :

The proposed scheme achieved the mutual authentication as follow: after receiving the login request message <C_i, F_i, D_i> from user U_i, the server S decrypts the parameters from F_i and computes C_i^* with the decrypted parameters, then checks whether the computed C_i^* matches with C_i or not. If those match, S authenticates U_i. Otherwise, the server S aborts the authentication phase. The same thing done after receiving the authentication reply message <N_3, K_i, T_2> from server S, U_i computes K_i^* and compares it with the received K_i, if it matches, the U_i authenticates S. it's noted that before authentication process it should check the matches of parameters.

-Privileged-insider attack :

During the registration phase of the proposed scheme, the user U_i does not sends his/her identity ID_i in plaintext. Instead, the user U_i sends the hashed identity $idb_i = h$ (IDi || b_i) to the server S. An adversary V cannot compute either of ID_i and b_i due to use of a one-way property of the hash function h (.). Therefore, the adversary V cannot compute any of the user parameters in the smart card SC that using idb_i.

-User unlink ability :

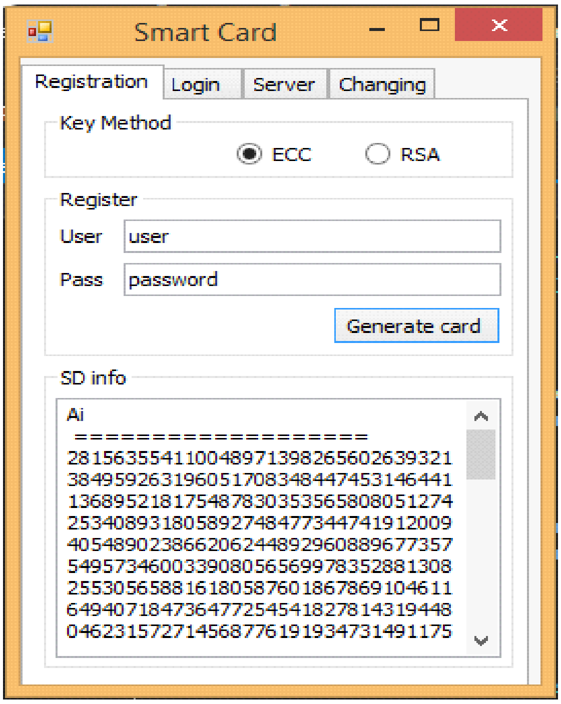
In this property, the adversary V wants to check if the two login request <C_i, F_i, D_i> and < C_i^*, F_i^*, D_i^*> are of the same user U_i or not. These parameters using N_1; C_i = h (pwb_i || N_1 || R_i || T_1), D_i = h (idb_i || pwb_i) ^ N_1, F_i = (pwb_i || idb_i || N_1 || R_i || T_1)^e (mod n), where N_1 is a random nonce generated freshly for every login request message, so that, <C_i, F_i, D_i> must differ for every login. Therefore, the adversary V cannot be finding whether two login requests are from the same user or not. For that, the proposed scheme preserve this property.

Performance analysis

In this study, we evaluate the performance of (Sutrala et al., 2016) scheme and the proposed scheme by programming both schemes in C# language using visual studio 2017, and we compared computation and communication cost between both schemes during login, authentication and key agreement, and identity change phases.

Figure 5-2 shows the registration phase of the proposed scheme. The user U_i insert identity ID_i in user field, and password pw in pass field then generate card that contain the secure information A_i, L_i, DP, P, E_p (a,b) and one-way hash function h (.).

The login and authentication phases of the proposed scheme shown in figure 5-3, where the user U_i insert his/her identity ID_i and password pw in the fields and then submit it to the server S by submit button. The authentication process then done by the server S and authenticate the user U_i to share the session key agreement between server and user.



● ECC ○ RSA

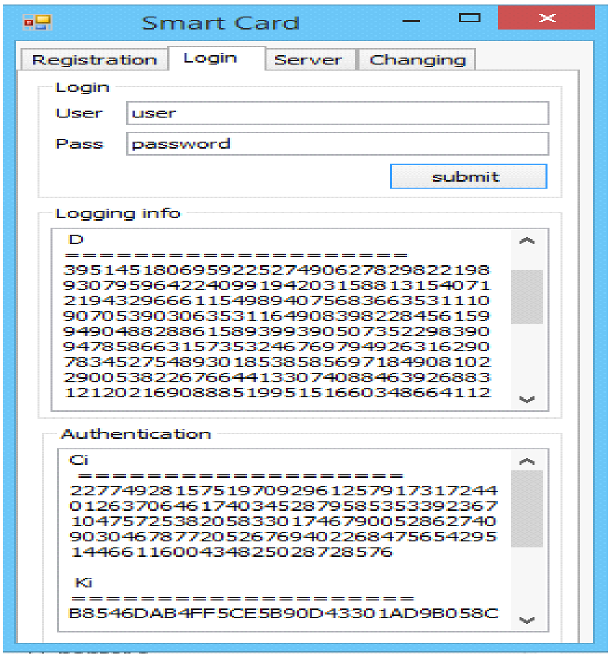
User user

Pass password

```
Ai
 ===================
281563554110048971398265602639321
384959263196051708348447453146441
136895218175487830353565808051274
253408931805892748477344741912009
405489023866206244892960889677357
549573460033908056569978352881308
255305658816180587601867869104611
649407184736477254541827814319448
046231572714568776191934731491175
```

Figure. 5-2. Registration phase of the proposed scheme.

Figure 5-2 explain the required time for key generation by using RSA is 0.00355372ms and the time by using ECC is 0.0355664ms. It's clear that the Time of key generation using ECC is more than RSA due to point multiplication operation to generate the table that's contain the generated keys, While in RSA, public and private keys are generated by using simple operations. Figure 5-6 explain that the required time for encryption and decryption processes when we use RSA algorithm is much more than the required time when we use ECC algorithm. The gap between ECC and RSA due to the key size, where the key size of public and private keys that used to encryption/decryption in RSA is 1024 bit. While in ECC, the key size of private key and public key is 256 bit.



Login

User user

Pass password

Logging info

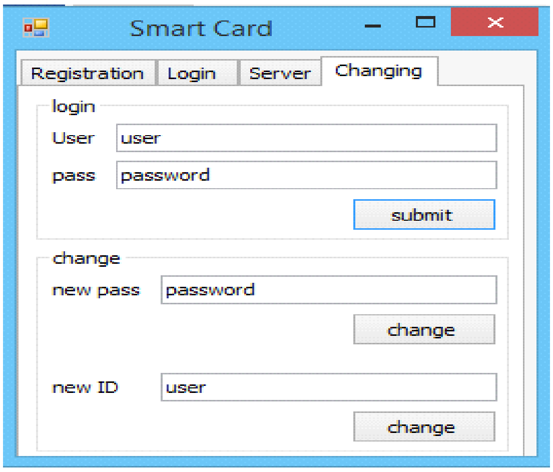
```
 D
====================
395145180695922527490627829822198
930795964224099194203158813154071
219432966611549894075683663531110
907053903063531164908398228456159
949048828861589399390507352298390
947858663157353246769794926316290
783452754893018538585697184908102
290053822676644133074088463926883
121202169088851995151660348664112
```

Authentication

```
 Ci
====================
227749281575197092961257917317244
012637064617403452879585353392367
104757253820583301746790052862740
903046787720526769402268475654295
1446611600434825028728576

 Ki
====================
B8546DAB4FF5CE5B90D43301AD9B058C
```

Figure. 5-3. Login and authentication phase of the proposed scheme.



login

User user

pass password

change

new pass password

new ID user

Figure. 5-4. Password and identity change phases.

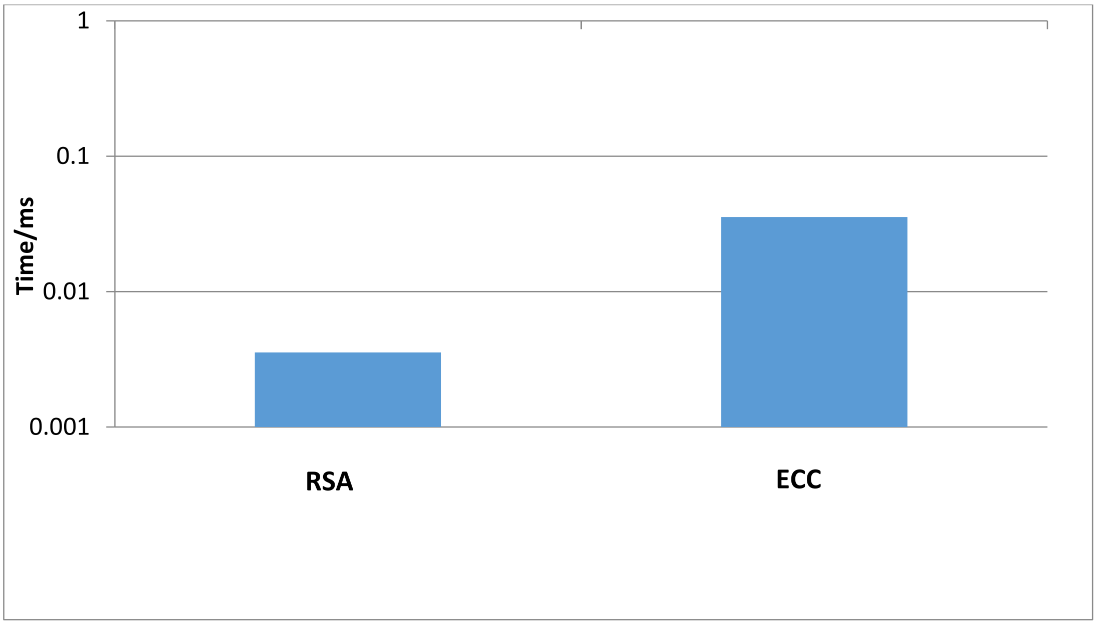


Figure.5-5. Time for key generation in ECC and RSA.

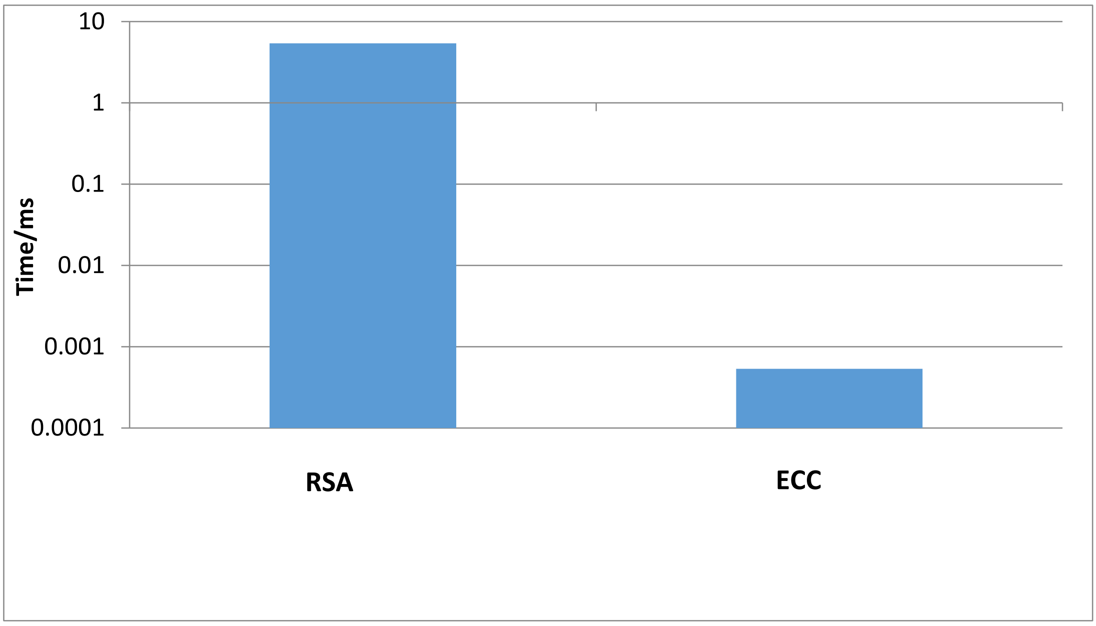


Figure. 5-6. Time for encryption and decryption in ECC and RSA algorithms.

The notation that we used to compare the results in Table 5-1 is: T_h denote to computational time for the one-way hash function h(.), T_e denoted to encryption/decryption using RSA public-key cryptosystem, T_c denoted for encryption/decryption using ECC public-key cryptosystem and T_{pm} denoted to perform an elliptic curve point multiplication.

The average of experimental values that we have with Intel Core i3-3110M CPU @ 2.40GHz 2.40 GHz processor and 4.00GB RAM is: T_h = 0.0000719ms, T_e = 5.40562754ms, T_c = 0.00053334ms and T_{pm} = 0.03555664ms.

Table 5-1 indicates the comparison of experimental computation time for the proposed scheme with (Sutrala et al., 2016) scheme and other schemes that use ECC such as Odelu (2015) and Wang (2014) schemes.

The total cost required in (Sutrala et al., 2016) scheme is 21.6250267ms and in the proposed scheme the total cost is 0.00286418ms, it's clear that the proposed scheme is better than (Sutrala et al., 2016) scheme, due to use ECC and the editing on identity change phase. In addition, we compare the proposed scheme with other existing scheme that used ECC such as Odelu (2015) and Wang (2014), it's clear that the proposed scheme better than Wang (2014) scheme, while Odelu (2015) scheme better than the proposed scheme that's because the number of hash function used in the proposed scheme more than in Odelu (2015) scheme, but in other hand, Odelu (2015) scheme has several security pitfalls as will show later in table 5-2, while proposed scheme strong against these security pitfalls. Figure 5-7 indicates the different between the proposed scheme with (Sutrala et al., 2016), Odelu (2015) and Wang (2014) schemes.

Table.5-1. Computation cost comparison.

scheme Phase	(Sutrala et al., 2016) scheme	Proposed scheme	Wang (2014)	Odelu (2015)
login	$7T_h + 1T_e$	$7T_h + 1T_c$	$6\ T_h + 1T_c$	$5T_h + 1T_c$
Authentication and key agreement	$9T_h + 1T_e$	$9T_h + 1T_c$	$6T_h + 3T_c$	$8T_h + 1T_c$
Identity change	$19T_h + 2T_e$	$9T_h$	_	_
Total cost	$35T_h + 4T_e$ = 21.6250267ms	$25T_h + 2T_c$ = 0.00286418ms	$12T_h + 4T_c$ = 0.002996ms	$13T_h + 2T_c$ = 0.0020015ms

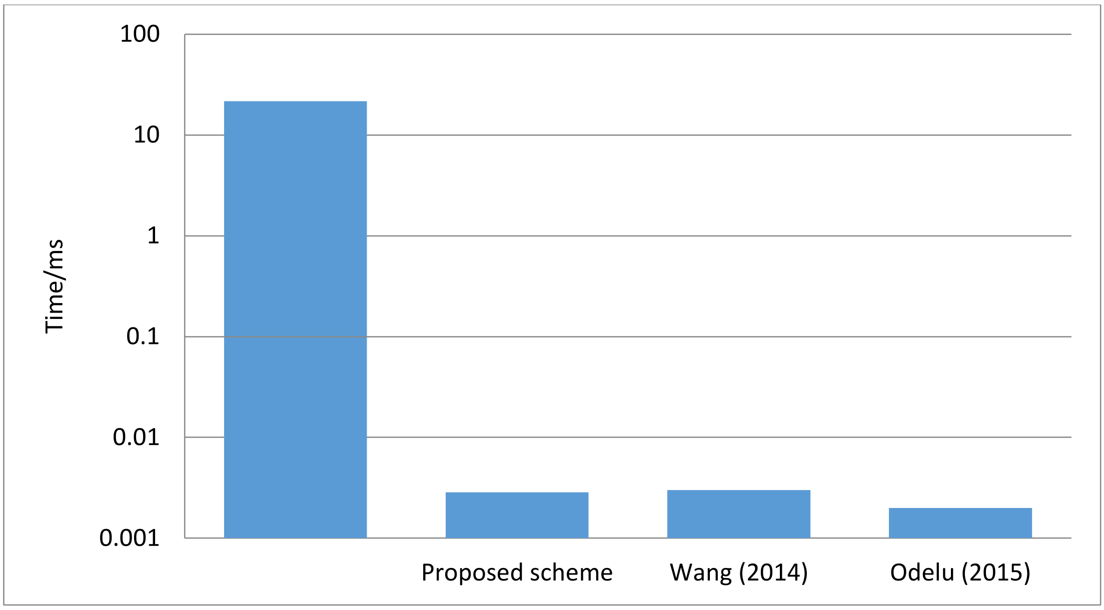


Figure. 5-7. Computation cost comparison.

In Table 5-2, we have compared the functionality features of the proposed scheme with Odelu (2015) and Wang (2014) schemes. It's clear that the proposed scheme provides better security than Odelu (2015) and Wang (2014) schemes, where both schemes Odelu (2015) and Wang (2014) are valuable against reply attack and privilege insider attack because of both schemes doesn't used timestamps during the communication and the user U_i sends the identity ID in plaintext, while the proposed scheme used timestamps during the communication and the user sends the ID in hashed value so that, its strong against replay attack and privilege insider attack. In addition, Wang (2014) scheme insecure against user unlink ability attack, that's the adversary V can finding whether two login requests are from the same user or not, that’s because the freshly random nonce not used in the parameters within request messages, while the proposed scheme used the freshly random nonce in the parameter within request messages, so that its secure against user unlink ability attack.

In addition, in this thesis, we used key length of 256 bit for ECC in encryption and decryption processes that provide better security level than key length of 1024 bit for RSA (Lee and Lee, 2016). See chapter 2. Where the security level of ECC when using key length of 256 bit is 128, while the security level of RSA when using key length of 1024 bit is 80. The security level of 128 or 80, its mean that the attacker needs to 128 or 80 operation to do the attack process.

Table 5-2: Functionality features comparison.

Feature / scheme	Replay attack	Impersonation attack	Session key security	Mutual authentication	Privilege insider attack	User unlink ability
Odelu (2015)	No	Yes	Yes	Yes	No	Yes
Wang (2014)	No	Yes	Yes	Yes	No	No
Proposed scheme	Yes	Yes	Yes	Yes	Yes	Yes

In the proposed scheme, we assume that the length of ID is 128 bits; timestamp T is of length 64 bits; public and private keys used in ECC are of length 256 bits; and the one-way cryptographic hash function h (.) is 256 bits (we use SHA-256 as one-way cryptographic hash function h (.)). While in (Sutrala et al., 2016) scheme, the length of ID is 160 bits; timestamp is of 32 bits; the primes p and q used in RSA are of length 1024 bits each; e, d and n are of length 2048 bits each; and one-way cryptographic hash function h (.) produces 160 bit output.

Table 5-3 and Table 5-4 indicate the comparison of communication costs of proposed scheme with the (Sutrala et al., 2016) scheme and other schemes that use ECC such as Odelu (2015) and Wang (2014) schemes. We compared the proposed scheme with others schemes by using 160 bit for hash function in Table 5-3, and then we compared the proposed scheme with other schemes by using 256 bit for hash function in Table 5-4.

It's clear that the total cost in Table 5-3 for proposed scheme less than the total cost for (Sutrala et al., 2016), Odelu (2015) and Wang (2014) schemes. The total cost for Odelu (2015) more than the total cost for proposed scheme due to the number of parameters in messages during the communication domain. And the number of messages in Wang (2014) and (Sutrala et al., 2016) schemes more than the number of messages in proposed scheme during communication domain.

In Table 5-4, the total cost for proposed scheme more than the total cost for Odelu (2015) due to 256 bit hash function that we used in proposed scheme, and the total cost for proposed scheme more than Wang (2014) scheme due to number of messages and to number of parameters within messages during communication domain, in addition to the hash function that used in both schemes.

Table. 5-3. Communication cost comparison for 160 bit hash function.

scheme Parameters	(Sutrala et al., 2016) scheme	Proposed scheme	Odelu (2015)	Wang (2014)
ID	160bit	128bit	160bit	160bit
Timestamp	32bit	64bit	_	_
Public and private keys	1024bit	160bit	160bit	160bit
Hash function h (.)	160bit	160bit	160bit	160bit
Total cost	5632bit	1440bit	1504bit	2016bit

Table. 5-4. Communication cost comparison with 256 bit for hash function.

scheme Parameters	(Sutrala et al., 2016) scheme	Proposed scheme	Odelu (2015)	Wang (2014)
ID	160bit	128bit	160bit	160bit
Timestamp	32bit	64bit	_	_
Public and private keys	1024bit	256bit	160bit	160bit
Hash function h (.)	160bit	256bit	160bit	160bit
Total cost	5632bit	2112bit	1504bit	2016bit

Chapter 6 : Conclusion and future works

Conclusion :

This thesis has outlined the recently proposed (Sutrala et al., 2016) scheme for TMIS and shows that the scheme vulnerable in depending on the RSA public key cryptosystem. The dependency of RSA makes the scheme requires more computation and communication cost. In the last years, researches became using elliptic curve cryptography ECC in remote user authentication protocols to be more efficient. ECC algorithm has several advantages, for example; the key size of 162-bit provides equivalent security comparison to the key size of 1024 bit in RSA and ECC has ECDLP problem which makes it more secure against known attacks and. So that we have improved (Sutrala et al., 2016) scheme by using elliptic curve cryptosystem ECC instead of RSA.

The proposed scheme overcomes the pitfalls in (Sutrala et al., 2016) scheme and increased the security degree, where the results show that the proposed scheme increases the security against possible known attacks such as replay attack, off-line password guessing attack and smart card lost/stolen verifier attack comparison with other schemes that use ECC such as Wang (2014) and Odelu (2015) schemes, where it insecure against privilege insider attack and replay attack. In addition, the simulation results show that the proposed scheme provided low computation and communication costs by using ECC algorithm, where the computation and communication cost respectively, is 0.0028641ms, 2112 bit comparison with (Sutrala et al., 2016) scheme that use RSA algorithm, where the computation and communication cost respectively, is 21.6250267ms, 5632 bit. In addition, the proposed scheme better than Wang (2014) scheme that use ECC in computation and communication cost.

Future works :

In the future, the presented contribution in this theses will be improved by:

Implementing the protocol in a multi-server environment, with focus on the performance, efficiency, and security of the protocol.

Develop the protocol by using the fingerprint to increase the degree of security.

References

Amin, R., Biswas, G.P., 2015. An improved RSA based user authentication and session key agreement protocol usable in TMIS. J. Med. Syst. 39, 79.

Das, A.K., Odelu, V., Goswami, A., 2014. A robust and effective smart-card-based remote user authentication mechanism using hash function. Sci. World J. 2014.

Ford, W., Kaliski, B.S., 2000. Server-assisted generation of a strong secret from a password, in: Enabling Technologies: Infrastructure for Collaborative Enterprises, 2000.(WET ICE 2000). Proceedings. IEEE 9th International Workshops On. IEEE, pp. 176–180.

Garrett, K., 2016. Vulnerability Analysis of Multi-Factor Authentication Protocols.

Giri, D., Maitra, T., Amin, R., Srivastava, P.D., 2015. An efficient and robust RSA-based remote user authentication for telecare medical information systems. J. Med. Syst. 39, 145.

Gura, N., Patel, A., Wander, A., Eberle, H., Shantz, S.C., 2004. Comparing elliptic curve cryptography and RSA on 8-bit CPUs, in: CHES. Springer, pp. 119–132.

Gurung, S., 2016. Data Authentication Principles for Online Transactions.

Hankerson, D., Menezes, A.J., Vanstone, S., 2006. Guide to elliptic curve cryptography. Springer Science & Business Media.

Kalra, S., Sood, S., 2013. Advanced remote user authentication protocol for multi-server architecture based on ECC. J. Inf. Secur. Appl. 18, 98–107.

Khan, M.K., Kumari, S., 2013. An authentication scheme for secure access to healthcare services. J. Med. Syst. 37, 9954.

Lamport, L., 1981. Password authentication with insecure communication. Commun. ACM 24, 770–772.

Lee, J.J., Lee, K.Y., 2016. An User Authentication Scheme Based on the ECC and OpenID Techniques in the Internet of Things. Int. J. Secur. Its Appl. 10, 79–88.

Odelu, V., Das, A.K., Goswami, A., 2015. An efficient ECC-based privacy-preserving client authentication protocol with key agreement using smart card. J. Inf. Secur. Appl. 21, 1–19.

Stallings, W., 2006. Cryptography and network security: principles and practices. Pearson Education India.

Sutrala, A.K., Das, A.K., Odelu, V., Wazid, M., Kumari, S., 2016. Secure anonymity-preserving password-based user authentication and session key agreement scheme for telecare medicine information systems. Comput. Methods Programs Biomed. 135, 167–185 .

Truong, T.-T., Tran, M.-T., Duong, A.-D., Echizen, I., 2017. Provable Identity Based User Authentication Scheme on ECC in Multi-server Environment. Wirel. Pers. Commun. 95, 2785–2801.

Wang, L., 2014. Analysis and enhancement of a password authentication and update scheme based on elliptic curve cryptography. J. Appl. Math. 2014.

Yang, J.-H., Chang, C.-C., 2009. An ID-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem. Comput. Secur. 28, 138–143.

Yeh, K.-H., 2014. A provably secure multi-server based authentication scheme. Wirel. Pers. Commun. 79, 1621–1634.

Yu, J., 2012. Remote user authentication in distributed networks and systems.

Zhang, J., Ma, J., Li, X., Wang, W., 2014. A Secure and Efficient Remote User Authentication Scheme for Multi-server Environments Using ECC. TIIS 8, 2930–2947.