Encryption Technology in Client-Server Database Systems

By Doug Lhotka

1.0 Introduction

We live in an information age, a society where computing power doubles faster than every 10 years, and the collective coalescing of knowledge holds the promise to generate incredible advances. However there is dark side to this revolution: this same power can be used and abused (even with good intentions), to eliminate personal privacy. As more data flows into a wide variety of commercial and governmental databases, there is a tremendous danger to individual liberty and freedom. It is no longer possible to be anonymous – we are constantly inundated with junk mail, telephone solicitations, and targeted advertising. Two-way television, national medical databases, a proposed national ID card are just some examples of the continuing invasion into our lives by the government and business entities. The lack of privacy on the internet was demonstrated recently when the author of the Melissa virus was tracked down in a matter of days. This danger may lead our society to become that which George Orwell feared in his famous novel, 1984.

The area of Personal Privacy covers a multitude of topics, from public policy to business ethics, to technological solutions. One area of concern is the theft of personal data directly from the computer hardware, another the interception of data when transmitted across the network. Technology offers a solution to these issues in the form of encryption technology. In order to keep the scope of discussion to a reasonable level, some knowledge of encryption is assumed, as well as a familiarity with database and client-server technologies.

In this information age, each database developer has a responsibility to incorporate security features into the system, in order to protect unauthorized people from accessing the data. There are a variety of systems for ensuring this, from password rotation schemes to physical security of the server. However, one of the most important technologies is encryption – transforming the data from a human understandable format, through a mathematical process, into random gibberish – and then restoring it to it's original form when an authorized person requires access. This technology can be used in three different locations, the client machine, the network between the server and the client, and the database server itself.

2.0 Encryption

Encryption technology has a long history. Caesar used a simple substitution cyber to protect messages sent via courier from the battle front. In World War II, Germany used the Enigma cypher to transmit messages to it's U-boats. In no small part, the war of the north Atlantic ocean was won because the British managed to crack the cypher and read much of the submarine traffic.

> *“There are two kinds of cryptography in this world: cryptography that will stop your kid sister from reading your files, and cryptography that will stop major governments from reading your files.”*
>
> --Bruce Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C.

There is a considerable amount of what Phil Zimmerman refers to as ‘Snake Oil’ on the cryptography market today. One common example is the password protection on PKZIP files, which can be broken in a matter of minutes with commercially available cryptographic software. The technology that is discussed in this paper is the latter – strong encryption that will protect the information from both large governments as well as international corporations.

Certainly the foremost authority on cryptographic technology and knowledge is the National Security Agency of the United States Government. Charged with two separate tasks: First, to ensure that sensitive American data is transmitted safely, and Second, to gather intelligence by intercepting and analyzing the communications of other people and governments. Operating under a tight cloud of secrecy, The agency was not even publicly acknowledged to exist until the 1980’s. Again from *PGP’s Introduction to Cryptography:*

> In the 1980s, NSA had been pushing a conventional encryption algorithm that they designed (the COMSEC Endorsement Program), and they won’t tell anybody how it works because that’s classified. They wanted others to trust it and use it. But any cryptographer can tell you that a well-designed encryption algorithm does not have to be classified to remain secure. Only the keys should need protection. How does anyone else really know if NSA’s classified algorithm is secure? It’s not that hard for NSA to design an encryption algorithm that only they can crack, if no one else can review the algorithm.

1.1 Legal Issues

The legal issues surrounding encryption are complex and may vary tremendously from country to country. In some nations, such as Russia, Iraq, North Korea, and Iran, merely possessing cryptographic equipment or software will result in immediate imprisonment. In others, it is available, but the keys must be registered with the government, effectively eliminating one’s ability to keep data private. Currently in the United States, there are no domestic restrictions on cryptographic software (although it is illegal to encrypt a telephone conversation). However, there continue to be efforts in Congress to mandate either a key escrow system or to ban strong encryption outright.

For the purposes of this paper, the legal implications of the use of encryption will be largely ignored. For more information on these aspects, both domestically and abroad, please review the references listed below.

1.2 Technology

There are a variety of technologies available to a database developer in order to encrypt their data. These boil down into two basic categories – symmetric (or conventional) and public/private key systems. Symmetric key technology has the advantage of being very fast, but requires a separate secure transmission medium to exchange the single key that is used for both encryption and decryption. ‘One time’ cypher pads are an example of symmetric encryption. Public/Private key encryption eliminates that needs, but is a complex algorithm that is slow when used with large keys, even on today’s powerful computers.

Symmetric key or traditional encryption is very fast, and works well for data that is not being transmitted or distributed. As Phil Zimmerman puts it in his *Introduction to Cryptography*,

> “For a sender and recipient to communicate securely using conventional encryption, they must agree upon a key and keep it secret between themselves. If they are in different physical locations, they must trust a courier, the Bat Phone, or some other secure communication medium to prevent the disclosure of the secret key during transmission. Anyone who overhears or intercepts the key in transit can later read, modify, and forge all information encrypted or authenticated with that key. From DES to Captain Midnight’s Secret Decoder Ring, the persistent problem with conventional encryption is key distribution: how do you get the key to the recipient without someone intercepting it?”

Public/private key encryption was invented by Whitfield Diffie and Martin Hellman in 1975. It involves a two-part key and a one-way hash algorithm. Each person has a ‘public’ key that is freely and widely distributed. This key can be used to encrypt information for the recipient, even if they two people have never met. Because it is computationally infeasible to compute the private key from the public one, the algorithm is secure. The downside to public key encryption is that it is slow, so many commercial encryption products (such as PGP) are hybrid systems that use a public key to encrypt a randomly generated session specific symmetric key. The data is encrypted with the symmetric key, then encrypted session key is appended to the cypertext. These types of hybrid products encompass the best of both worlds.

1.3 Points of Encryption

There are three points of failure for data security on the client machine. First is the old-fashioned password. Second is the local disk storage, and third is the risk of a Tempest attack.

Probably the greatest risk to sensitive data are client applications accessible from public networks, or on large private networks where the machines are not physically secured. In most cases these are simply protected by a user password. End users are notorious for choosing easy to guess passwords, and for never changing them. In one security audit of a company that this author worked for, the auditor was able to penetrate 47% of our user accounts within the first 8 hours! Without adequate application password security, and other data security approaches are nearly worthless. Password rotation and auditing schemes are well documented in system administration literature.

The second area of risk is occurs when user download information to a local machine and store it in unencrypted format. Unless the application program prohibits the user from extracting the data and storing it locally, this will be a constant risk – and most likely will occur. For very sensitive data, the client machine may be physically secured, but for large commercial databases with hundreds of users this is not really an option. However, there are some new products on the market that mitigate these risks. PGPDisk is a product that creates an encrypted partition on the local machine. All files transferred to it are encrypted, and it incorporates special technology to ensure that sensitive data is not stored in the swap file. Symantec Corporation also offers a similar product. This is probably best addressed by Corporate or Governmental Policy – backed up by consequences for violations.

A tempest attack is generally something that most consumers and businesses need not be concerned with. Using sophisticated equipment it is possible to intercept and decode the emissions from monitors and printers – essentially watching everything a user does from a remote location. There are commercially available computers, monitors, and printers that provide shielding to prevent such an attack from succeeding. In addition, there are screen fonts available, that purport to be resistant to tempest attacks. Since the interception technology is highly restricted, the validity of these claims is unable to be validated.

Sensitive data is at the greatest risk of interception when it is being transmitted across the network (especially if it is on the Internet). Packet sniffing is relatively simple to accomplish, and is the source of a significant amount of data interception (including credit card fraud). In order to protect information, some sort of streaming technology must be incorporated into the networking protocols. There are several proprietary and public standards that are currently competing in order to be chosen by the IEEE standards committee as the algorithm of choice.

These protocols run at the lowest level, just above TCP/IP itself. However, a better option may be to encrypt the actual database information contained within the packets. Then the control is maintained by the database application, and can be coordinated with application passwords. Most database vendors offer some form of encrypted network transmission as an option for their products. Oracle Corporation offers a version of their SQL*Net (recently renamed Net8) that incorporates RSA encryption. By using this protocol, it becomes essentially impossible for someone to read an intercepted package.

Clearly the place where a database developer or administrator can have the greatest impact on data security is in the database engine itself. Regardless of the type of engine used, data is normally stored in unencrypted format and protected by either application or database level passwords.

For very high-security databases the information stored in the system may itself be encrypted. This can be done at either the application level, or at the database engine level before the information is actually written to disk. Either scenario can place tremendous strain on the CPU of the database server, and some form of specialized hardware may be required to offload this work.

2 Conclusion

As with any project, a tradeoff analysis must be performed, balancing the sensitivity of the data, the downside risk of it's exposure, and the cost of securing it. For applications such as the 'joke of the day' database server, there is likely little need for encrypted transmission or security beyond the database administrator's password. However, for transmitting the launch codes for America's nuclear missiles, or for conducting Federal Reserve Bank transactions, some additional levels of security are required. As the speed of computers increases, the overhead to encrypt routine data traffic will become irrelevant, and it will become ubiquitous.

First, for any system where there is any data sensitivity, the administrator must institute some form of password auditing and rotation. Second, if the information is being transmitted over a public network, or if it is extremely sensitive, over a private network as well, it should incorporate encryption in the network protocol used. Only for those most critical applications should the data be stored in encrypted format.

Encryption is not an end-all to the problem of data security, but it can provide one answer to some of the vexing question. However, without reforms in public and private policies regarding the overall personal privacy issues, no technology solution can provide a complete solution.

3 References

Testimony of Phillip R. Zimmerman to the Subcommittee on Science, Technology, and Space of the US Senate Committee on Commerce, Science, and Transportation., June 26, 1996. http://www.pgp.com/phil/phil-quotes.cgi

Authors preface to the book: "PGP Source Code and Internals," by Phillip Zimmerman, http://www.pgp.com/phil/phil-src-intro.cgi

Crypto Law Survey http://cwis.kub.nl/~frw/people/koops/lawsurvy.htm

International PGP Site http://www.pgpi.net

PGP Corporate Web Site http://www.pgp.com

Electronic Freedom Foundation http://www.eff.org

References From the PGP User’s Manual:

Non-Technical and beginning technical books

• Whitfield Diffie and Susan Eva Landau, “Privacy on the Line,” *MIT Pres*s; ISBN: 0262041677
This book is a discussion of the history and policy surrounding cryptography and communications security. It is an excellent read, even for beginners and non-technical people, but with information that even a lot of experts don't know.

• David Kahn, “The Codebreakers” *Scribne*r; ISBN: 0684831309
This book is a history of codes and code breakers from the time of the Egyptians to the end of WWII. Kahn first wrote it in the sixties, and there is a revised edition published in 1996. This book won't teach you anything about how cryptography is done, but it has been the inspiration of the whole modern generation of cryptographers.

• Charlie Kaufman, Radia Perlman, and Mike Spencer, “Network Security: Private Communication in a Public World,” *Prentice Hall;* ISBN: 0-13-061466-1
This is a good description of network security systems and protocols, including descriptions of what works, what doesn't work, and why. Published in 1995, so it doesn't have many of the latest advances, but is still a good book. It also contains one of the most clear descriptions of how DES works of any book written.

Intermediate books

• Bruce Schneier, “Applied Cryptography: Protocols, Algorithms, and Source Code in C,” *John Wiley & Sons*; ISBN: 0-471-12845-7
This is a good beginning technical book on how a lot of cryptography works. If you want to become an expert, this is the place to start.

• Alfred J. Menezes, Paul C. van Oorschot, and Scott Vanstone, “Handbook of Applied Cryptography,” *CRC Press;* ISBN: 0-8493-8523-7
This is the technical book you should get after Schneier. There is a lot of heavy-duty math in this book, but it is nonetheless usable for those who do not understand the math.

• Richard E. Smith, “Internet Cryptography,” *Addison-Wesley Pub C*o; ISBN: 020192480
This book describes how many Internet security protocols. Most importantly, it describes how systems that are designed well nonetheless end up with flaws through careless operation. This book is light on math,

and heavy on practical information.

- William R. Cheswick and Steven M. Bellovin, “Firewalls and Internet Security: Repelling the Wily Hacker” *Addison-Wesley Pub C*o;
  ISBN: 0201633574
  This book is written by two senior researcher at AT&T Bell Labs, about their experiences maintaining and redesigning AT&T's Internet connection. Very readable.

Advanced books

- Neal Koblitz, “A Course in Number Theory and Cryptography”
  *Springer-Verla*g; ISBN: 0-387-94293-9
  An excellent graduate-level mathematics textbook on number theory and cryptography.

- Eli Biham and Adi Shamir, “Differential Cryptanalysis of the Data Encryption Standard,” *Springer-Verla*g; ISBN: 0-387-97930-1
  This book describes the technique of differential cryptanalysis as applied to DES. It is an excellent book for learning about this technique.