International Journal of Research and Applications
Jan-Mar © 2014 Transactions 1(1): 1-5

eISSN : 2349 - 0020
pISSN : 2394 - 4544

COMPUTER SCIENCE ORIGINAL ARTICLE

Paper Title

Name [1] , Name [2] and Name [3]

[1] Designation , Department Name, College Name, Affiliated to University Name, Address-Pin code.
[2] Designation , Department Name, College Name, Affiliated to University Name, Address-Pin code.
[3] Designation , Department Name, College Name, Affiliated to University Name, Address-Pin code.
[1] aaaaa@gmail.com, [2] bbbbb @gmail.com, [3] cccccc @gmail.com

ABSTRACT

A key in cryptography is defined as a piece of information that determines the functional output of an algorithm or cipher. In the process of Encryption, a key specifies the conversion of a plaintext into cipher text and cipher text into a plaintext during decryption [1]. A *key* is a piece of variable data that is fed as input into a cryptographic algorithm to perform one such operation. Keys are widely used in other cryptographic algorithms, such as digital signature schemes and message authentication codes. Without the usage of keys, a specific algorithm would produce no valid result.

Keywords: Key, cipher text, Encryption, Decryption.

1. INTRODUCTION

Keys are used to control the operation of a cipher. Many ciphers are based on publicly known algorithms which are source. Claude Shannon and Auguste Kerckhoffs contributed towards the concepts of cryptography with the statements known as Kerckhoffs' principle and Shannon's Maxim respectively that the security of the system should depend on the key alone and this has been explicitly formulated.

2. SIGNIFICANCE OF KEYS

Cryptographic keys work as important elements w.r.to to the cryptographic operations. Most of the cryptographic schemes consist of a pair of operations such as encryption and decryption or signing and verification. In a well-designed cryptographic scheme, the security of the scheme depends only on the security of the keys used [1].

In general 80 bit key length is generally considered to be the minimum for strong security with symmetric encryption algorithms. A key should therefore be large enough that a brute force attack takes too long time to execute. Shannon's work on information theory showed that, to achieve the so called *perfect secrecy*, it is necessary for the key length to be at least as large as the message to be transmitted and only used once. This algorithm is called as One-time pad. Due to the practical difficulty of managing such long keys, modern cryptographic practices have discarded the notion of perfect secrecy as a requirement for encryption, and instead focus on *computational security*, under which the computational requirements of breaking an encrypted text must be infeasible for an attacker. On the other hand 128-bit keys are commonly used and considered to be very strong.

The concept of Encryption has been divided into two main types.

1. Symmetric systems and
2. Asymmetric systems.

The above two types are categorized according to the central algorithm used depending on the specified operation. As each of the above two are of different levels of cryptographic complexity, it is usual to have

different key sizes for the same level of security, depending upon the algorithm used.

3. SYMMETRIC KEY ALGORITHMS

In Symmetric key algorithms same key is used in the process of Encryption and the Decryption. This was proposed by Auguste Kerckhoffs. He was a Dutch cryptographer a professor of languages at the École des Hautes Études Commercials in Paris in the late 19th century. Kerckhoffs's principles are also called as Kerckhoffs's desiderata, Kerckhoffs's assumption, axiom, or law. A cryptosystem should be secure even if everything about the system, except the key, is made public [4]. The history of cryptography provides evidence that it can be difficult to keep the details of a widely used algorithm secret. This is known as Kerckhoffs' principle. Further the law specifies as "*only secrecy of the key provides security*", or defined as Shannon's maxim, "*the enemy knows the system*".

He is best known today for a series of two essays he published in 1883 in *le Journal des Sciences Militaries Journal of Military Science* entitled *La Cryptographie Militaire Military Cryptography*. These articles surveyed the then state-of-the-art in military cryptography, and made a plea for considerable improvements in French practice.

They also included many pieces of practical advice and rules of thumb, including six principles of practical cipher design:

1. The system should be, if not theoretically unbreakable, unbreakable in practice.
2. The design of a system should not require secrecy and compromise of the system should not inconvenience the correspondents (Kerckhoffs' principle).
3. The key should be memorable without notes and should be easily changeable
4. The cryptograms should be transmittable by telegraph
5. The apparatus or documents should be portable and operable by a single person
6. The system should be easy, neither requiring knowledge of a long list of rules nor involving mental strain

In the designing of security systems, it is wise to assume that the details of the cryptographic algorithm are already available to the attacker. A key is often easier to protect than an encryption algorithm, and easier to change if compromised. An attacker who obtains the key can recover the original message from the encrypted data and trying to keep keys secret is one of the most difficult problems in practical cryptography.

4. ASYMMETRIC CRYPTOGRAPHY

Asymmetric cryptography refers to a cryptographic algorithm which requires two separate keys, one of which is *secret* (or *private*) and one of which is *public*. Although different, the two parts of this key pair are mathematically linked. The public key is used to encrypt plaintext or to verify a digital signature; whereas the private key is used to decrypt cipher text or to create a digital signature [2]. The term "asymmetric" stems from the use of different keys to perform these opposite functions, each the inverse of the other as contrasted with conventional ("symmetric") cryptography which relies on the same key to perform both.

A newer class of "public key" cryptographic algorithms was invented in the 1970s which uses a pair of keys, one to encrypt and one to decrypt. These asymmetric key algorithms allow one key to be made public while retaining the private key in only one location. They are designed so that finding out the private key is extremely difficult, even if the corresponding public key is known. A user of public key technology can publish their public key, while keeping their private key secret, allowing anyone to send them an encrypted message [2]. Public-key algorithms are based on mathematical problems which currently admit no efficient solution that are inherent in certain integer factorization, discrete logarithm, and elliptic curve relationships.

It is computationally easy for a user to generate their own public and private key-pair and to use them for encryption and decryption. The strength lies in the fact that it is "impossible" (computationally unfeasible) for a properly generated private key to be determined from its corresponding public key. Thus the public key may be published without compromising security, whereas the private key must not be revealed to anyone not authorized to

read messages or perform digital signatures. Public key algorithms, unlike symmetric key algorithms, do *not* require a secure initial exchange of one (or more) secret keys between the parties.

5. ADVANTAGE OF SECRET KEYS

Using secure cryptography is supposed to replace the difficult problem of keeping messages secure with a much more manageable one, keeping relatively small keys secure. A system that requires long-term secrecy for something as large and complex as the whole design of a cryptographic system obviously cannot achieve that goal. It only replaces one hard problem with another. However, if a system is secure even when the enemy knows everything except the key, then all that is needed is to manage keeping the keys secret.

There are a large number of ways the internal details of a widely used system could be discovered. The most obvious is that someone could bribe, blackmail, or otherwise threaten staff or customers into explaining the system. In war, for example, one side will probably capture some equipment and people from the other side. Each side will also use spies to gather information.

If a method involves software, someone could do memory dumps or run the software under the control of a debugger in order to understand the method. If hardware is being used, someone could buy or steal some of the hardware and build whatever programs or gadgets needed to test it. Hardware can also be dismantled so that the chip details can be seen with microscopes.

6. MAINTAINING SECURITY

A generalization some make from Kerckhoffs's principle is: "The fewer and simpler the secrets that one must keep to ensure system security, the easier it is to maintain system security." Bruce Schneier ties it in with a belief that all security [3] systems must be designed to fail as gracefully as possible: Any security system depends crucially on keeping some things secret. However, Kerckhoffs's principle points out that the things kept secret ought to be those least costly to change if inadvertently disclosed.
For example, a cryptographic algorithm may be implemented by hardware and software that is widely distributed among users. If security depends on keeping that secret, then disclosure leads to major logistic difficulties in developing, testing, and distributing implementations of a new algorithm – it is "brittle". On the other hand, if keeping the algorithm secret is not important, but only the *keys* used with the algorithm must be secret, then disclosure of the keys simply requires the simpler, less costly process of generating and distributing new keys. Kerckhoffs's principle was reformulated (or perhaps independently formulated) by Claude Shannon as "the enemy knows the system", *i.e.*, "one ought to design systems under the assumption that the enemy will immediately gain full familiarity with them". In that form, it is called Shannon's maxim [5]. In contrast to "security through obscurity", it is widely embraced by cryptographers.

7. CRYPTOGRAPHY IN EVERYDAY LIFE

7.1 AUTHENTICATION/DIGITAL SIGNATURES

Authentication and digital signatures are a very important application of public-key cryptography. The only requirement is that public keys are associated with their users by a trusted manner, for example a trusted directory. To address this weakness, the standards community has invented an object called a certificate. A certificate contains, the certificate issuer's name, the name of the subject for whom the certificate is being issued, the public key of the subject, and some time stamps. You know the public key is good, because the certificate issuer has a certificate too.

Pretty Good Privacy (PGP) is a software package originally developed by Phil Zimmerman that provides encryption and authentication for e-mail and file storage applications. Zimmerman developed his freeware program using existing encryption techniques, and made it available on multiple platforms. It provides message encryption, digital signatures, data compression, and e-mail compatibility [3]. PGP uses RSA for key transport and IDEA for bulk encryption of messages. Zimmerman ran into legal problems with RSA over his use of the RSA algorithm in his program. PGP is now available in a couple of legal forms: MIT PGP versions 2.6 and later are legal freeware for non-commercial use, and Via crypt PGP versions 2.7 and later are legal commercial versions of the same software.

7.2 TIME STAMPING

Time stamping is a technique that can certify that a certain electronic document or communication existed or was delivered at a certain time. Time stamping uses an encryption model called a blind signature scheme. Blind signature schemes allow the sender to get a message receipted by another party without revealing any information about the message to the other party.

Time stamping is very similar to sending a registered letter through the U.S. mail, but provides an additional level of proof. It can prove that a recipient received a specific document. Possible applications include patent applications, copyright archives, and contracts. Time stamping is a critical application that will help make the transition to electronic legal documents possible.

7.3 ELECTRONIC MONEY

The definition of electronic money (also called electronic cash or digital cash) is a term that is still evolving. It includes transactions carried out electronically with a net transfer of funds from one party to another, which may be either debit or credit and can be either anonymous or identified. There are both hardware and software implementations.

Anonymous applications do not reveal the identity of the customer and are based on blind signature schemes. (Digicash's Ecash) Identified spending schemes reveal the identity of the customer and are based on more general forms of signature schemes. Anonymous schemes are the electronic analog of cash, while identified schemes are the electronic analog of a debit or credit card [6]. There are also some hybrid approaches where payments can be anonymous with respect to the merchant but not the bank (Cyber Cash credit card transactions); or anonymous to everyone, but traceable (a sequence of purchases can be related, but not linked directly to the spender's identity).

Encryption is used in electronic money schemes to protect conventional transaction data like account numbers and transaction amounts, digital signatures can replace handwritten signatures or a credit-card authorizations, and public-key encryption can provide confidentiality. There are several systems that cover this range of applications, from transactions mimicking conventional paper transactions with values of several dollars and up, to various micropayment schemes that batch extremely low cost transactions into amounts that will bear the overhead of encryption and clearing the bank.

7.4 SECURENETWORK COMMUNICATIONS

Secure Socket Layer (SSL) Netscape has developed a public-key protocol called Secure Socket Layer (SSL) for providing data security layered between TCP/IP (the foundation of Internet-based communications) and application protocols (such as HTTP, Telnet, NNTP, or FTP). SSL supports data encryption, server authentication, message integrity, and client authentication for TCP/IP connections.

The SSL Handshake Protocol authenticates each end of the connection (server and client), with the second or client authentication being optional. In phase 1, the client requests the server's certificate and its cipher preferences. When the client receives this information, it generates a master key and encrypts it with the server's public key, then sends the encrypted master key to the server. The server decrypts the master key with its private key, then authenticates itself to the client by returning a message encrypted with the master key. Following data is encrypted with keys derived from the master key. Phase 2, client authentication, is optional. The server challenges the client, and the client responds by returning the client's digital signature on the challenge with its public-key certificate.

SSL uses the RSA public-key cryptosystem for the authentication steps. After the exchange of keys, a number of different cryptosystems are used, including RC2, RC4, IDEA, DES and triple-DES.

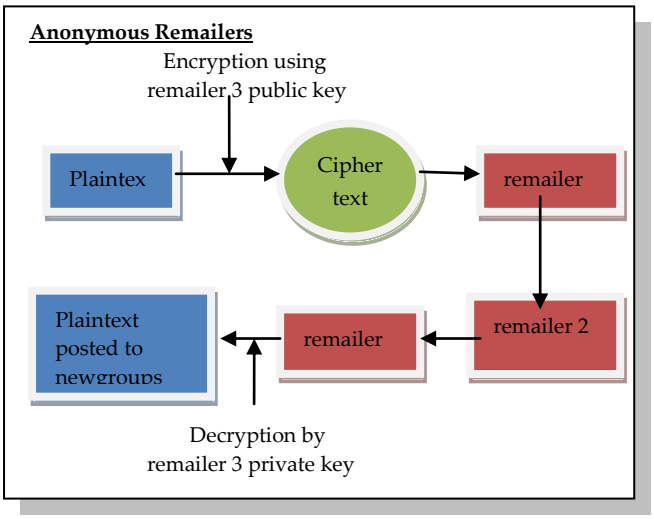
7.5 KERBEROS

Kerberos is an authentication service developed by MIT which uses secret-key ciphers for encryption and authentication. Kerberos was designed to authenticate requests for network resources and does not authenticate authorship of documents.
In a Kerberos system, there is a site on the network, called the Kerberos server, to perform centralized key management and administrative functions. The server maintains a key database with the secret keys of all users, authenticates the identities of users, and

distributes session keys to users and servers who need to authenticate one another [7]. Kerberos depends on a trusted third party, the Kerberos server, and if the server were compromised, the integrity of the whole system would be lost. Kerberos is generally used within an administrative domain (for example across a company's closed network); across domains (e.g., the Internet), the more robust functions and properties of public-key systems are often preferred.

7.6 ANONYMOUS REMAILERS

A remailer is a free service that strips off the header information from an electronic message and passes along only the content. It's important to note that the remailer may retain your identity, and rather than trusting the operator, many users may relay their message through several anonymous remailers before sending it to its intended recipient. That way only the first remailer has your identity, and from the end point, it's nearly impossible to retrace.



Here's a typical scenario - the sender intends to post a message to a news group via three remailers (remailer 1, remailer 2, and remailer 3). He encrypts the message with the last remailer's (remailer 3's) public key. He sends the encrypted message to remailer 1, which strips away his identity, then forwards it to remailer 2, which forwards it to remailer 3. Remailer 3 decrypts the message and then posts it to the intended newsgroup.

REFERENCES

1. Delfs, Hans & Knebl, Helmut (2007). "Symmetric-key encryption". *Introduction to cryptography: principles and applications*. Springer. ISBN 9783540492436.
2. Mullen, Gary & Mummert, Carl (2007). *Finite fields and applications*. American Mathematical Society. p. 112. ISBN 9780821844182.
3. Pelzl & Paar (2010). *Understanding Cryptography*. Berlin: Springer-Verlag. p. 30.
4. Frederick J. Hirsch. "SSL/TLS Strong Encryption: An Introduction". *Apache HTTP Server*. Retrieved 2013-04-17.
5. N. Ferguson; B. Schneier (2003). *Practical Cryptography*. Wiley. ISBN 0-471-22357-3.
6. J. Katz; Y. Lindell (2007). *Introduction to Modern Cryptography*. CRC Press. ISBN 1-58488-551-3.
7. A. J. Menezes; P. C. van Oorschot; S. A. Vanstone (1997). *Handbook of Applied Cryptography*. ISBN 0-8493-8523-7.
8. IEEE 1363: Standard Specifications for Public-KeyCryptography.