Symmetric vs asymmetric algorithms computer science essay

Technology, Computer

Data integrity, confidentiality and availability over the web, applications and storage means have become the major concerns in computer world. If data are exposed to an attacker, it will have a significant impact to business. Cryptography has a major role to play to prevent attacks to sensitive data employing encryption and decryption mechanisms. There are two main approaches to encryption: symmetric and asymmetric, and each of them contains its own variety of encryption algorithms. Both types have their advantages and disadvantages as none of them excels at both efficiency and high security. As illustrated in this paper it is not sufficient enough to use a single type of encryption methods in applications. Rapid development of network technology and expansion of information around the world, information security has to be balanced with processing efficiency requiring hybrid approaches. This paper discusses the advantages and disadvantages of each type of cryptography and proposes in reference to literature integration approaches.

Index Terms -encryption, decryption, symmetric algorithm, asymmetric algorithm, public keys, ciphertext, PGP, GnuPG, hybrid encryption.

Introduction

Cryptography is the study of the mathematical techniques related to aspects of information security such as confidentiality, data integrity, message authentication, and entity authentication (Piper, 2002). Cryptography has many applications including those in key management for digital communication (communication equipment, electronic mail and data

interchange, access control and audit trails, e-banking), commercial software (software verification and virus detection).

Encryption of electronic messages has been considcred for Electronic Data Interchange (EDI), where contracts and purchase orders are signed and delivered electronically. Similar system has been used by the British banks for Electronic Funds Transfer (EFT) for Point Of Sales (POS). For Local Area Networks (L. 4Ns) the IEEE 802. 10 LAN Security Working Group is currently drafting security standard using public-key techniques for key management.

Access control to buildings or computers relies on use of passwords or Personal Identification Number (PIN). Passwords are either stored to the computer or are dynamically generated using battery-powered devices (“ tokens”). In some cases (e. g. in banking) the token is activated by entering a PIN. State-of-the art smart card devices embed personal or payment data which can be decrypted as long as the user enters a password or provides biometric data (e. g. voice, fingerprint, handwritten signature, or scanned picture).

In theory of cryptography, the information to be encrypted is called the message and the operation of disguising is known as encryption (or enciphering). The enciphered message is called the ciphertext or cryptogram. The algorithm used for this operation also has a second input known as the enciphering key. The process of obtaining the message from the ciphertext is known as decryption, and, in addition to the ciphertext, the

deciphering algorithm needs a deciphering key. The receiver will obtain the correct message, if they use the right deciphering key.

An encryption system is said to be symmetric if, for each corresponding pair of enciphering and deciphering keys, it is easy to determine the deciphering key from the enciphering key. If, on the other hand, it is computationally impossible to determine the deciphering key from the enciphering key, then we have a public key system or, else, an asymmetric system.

Symmetric and asymmetric cryptography are almost two different subjects, therefore the algorithms are different, and the key management problems are different. In the sections below we present the main algorithms of each system and a comparison between the algorithms of each system is performed.

SYMMETRIC ALGORITHMS

In symmetric algorithms, the sender and receiver of messages and files share the same key for encryption and decryption. Symmetric algorithms have the advantage of not consuming too much computing power. The most typical examples are: DES, Triple-DES (3DES), IDEA, CAST5, BLOWFISH, TWOFISH.

In symmetric algorithms an important aspect of their effectiveness is the strength of the key encryption or else the size of the key used. Since the same key is used for encryption and decryption, the longer the keys are, the

harder to unlock them. Strong versus weak key is one of the typical classification of symmetric algorithms.

DES (Data Encryption Standard) was the first symmetric algorithm to be introduced by NIST (National Institute of Standards and Technology) in 1974. DES uses one 64-bits key and many attacks have been recorded in literature against it and that caused the need to propose 3DES (Triple DES). Triple DES (3DES) uses three 64-bits keys. Therefore, 3DES applies 3 times the core encryption method of DES however this makes it slower than other symmetric algorithms (Nadeem and Javed, 2005).

In addition, NIST introduced AES (Advanced Encryption Standard) in 1997 as replacement of DES. AES uses various (128, 192, 256) bits keys. Yet Brute Force attack was the only effective unlocking mechanism against it trying all character combinations. Bruce Schneier’s Blowfish encryption method is even stronger with no effective attack to have been recorded since its release in 1993. Its effectiveness relies on using variable length key (32-448) and 128-bit is its default and a 64-bit block size although taking the risk of allowing the definition of weak keys (Nadeem and Javed, 2005).

Running simulation tests, Nadeem and Javed (2005) showed that Blowfish has better performance than other symmetric algorithms and no any security weak points to its record.

AES and 3DES showed poor performance since it requires more processing power. Similar results have been produced in (Elminaam et al, 2009).

ASSYMETRIC ALGORITHMS

Nowadays confidential messages around the world are encrypted and decrypted relying on asymmetric techniques. This is because the key used for encryption and decryption is not the same but rather it relies on a key distribution mechanism which is called public-private key distribution. Confidential messages are encrypted using the public key and can only be decrypted using the private key. RSA, DSA, ELGAMAL, TLS, PGP are some of the examples of asymmetric algorithms.

RSA is one of the well known public key (asymmetric type of) algorithms used for generating digital signatures over messages (Das and Madhavan, 2009). In addition, NIST published the Digital Signature Standard (DSS) in 1991 for generating digital signatures. DSS uses the SHA-1 algorithm for calculating the message digest of the plain message and then applies the DSA (Digital Signature Algorithm) for creating the digital signature of the message based on the message digest. DSA is only used for performing digital signatures. It cannot be used for encryption and this is the main difference with RSA (Das and Madhavan, 2009).

COMPARING SYMMETRIC AND ASSYMETRIC CRYPTOGRAPHY TECHNIQUES

Advantages and disadvantages of symmetric algorithms are illustrated in the table below (Panda and Kumar, 2011):

Advantages

Disadvantages

Simple method of encryption

Agree a priori a secret key before any message exchange

Encryption of personal user files and messages

Maintain multiple keys one for each pair of message exchange or collaboration

Faster than asymmetric techniques

Sharing the secret key does not prove authenticity of sender or receiver of messages

Requires less computer resources

Key management is a task requiring significant effort

Key compromise to a communication pair does not affect communication with other pairs

Key exchange should also be a secure process requiring the implementation of its own secure channel

Can use the same publicly known algorithm for encryption

strength of security depends on size of key

Advantages and disadvantages of symmetric algorithms are illustrated in the table below (Panda and Kumar, 2011):

Advantages

Disadvantages

Makes convenient the key distribution as only the public keys are shared

Public key generation and distribution is required for activating the encryption mechanism requiring a verification process in its own

Digital signatures are generated through public key encryption hence verifying the authenticity of the sender

It is rather slow in comparison to symmetric algorithms and is not preferred for short messages

Messages accompanied by digital signature cannot be modified during transfer

It requires a lot more processing power

Digitally signing a message is equivalent to a physical signature thus the recipient of the message cannot deny the authenticity of the sender

Losing a private key can cause can cause widespread security compromise as all messages can be read

HYBRID approaches

The main disadvantage of asymmetric algorithms is their slowness is comparison to the symmetric algorithms. One way to address this issue in

many applications, is to apply a combination of both. A typical integration approach is the following:

use asymmetric keys for authentication

then one or more symmetric keys can be generated and exchanged using the asymmetric encryption.

Typical examples of this procedure are the RSA/IDEA combination of PGP2 or the DSA/BLOWFISH used by GnuPG.

For instance, Mantoro and Zakariya (2010) proposed a secure method of e-mail communication for Android-based mobile devices using a hybrid cryptosystem which combines AES 128 bit (symmetric) encryption, RSA 1024 bit (asymmetric) encryption and SHA-1 (hash) function. This approach had been tested with plain text but not with email attachments or communication between Android devices and other mobile platforms.

PGP Configuration

One of the main applications of public key encryption techniques is in the PGP (Pretty Good Privacy) program for data communication. It is primarily used for signing, encrypting emails, files, text and anything else that is involved in email communication (Zimmermann, 1995).

RSA key of 1024 bits are still considered secure (given the available options of 512, 1024, 2048 bits). On the other hand, Kaliski (2003) considers that an 1024-bit RSA key can be broken in one year and that an 2048-bit RSA public

key is secure enough for a PGP configuration which includes also the AES and the SHA-1 algorithm. RSA/IDEA/MD5 or any other other similar configuration is less secure according to Lenstra and Verheul (2003).

PGP works as follows (Benz, 2001):

compresses the plain message in order to reduce the pattern of the typical plaintext

creates a session key which is a random number, usually generated given mouse movements or keystrokes

a symmetric encryption algorithm is applied on the random number (e. g. Triple DES, Twofish, CAST, or AES) to generate a one-time-only secret key (session key)

additional input might be required by the user if the collected information is not sufficient enough (e. g. additional mouse movements, keystrokes)

the session key is used along a symmetric algorithm to encrypt the message to a ciphertext.

The session key is also encrypted using an asymmetric technique such as RSA

The recipient receives the ciphertext along with the public key-encrypted session key.

Indeed the combination of the two encryption methods exploits the convenience of public-key encryption with the speed of symmetric encryption (Benz, 2001). Symmetric encryption is about 100 to 1, 000 times faster than public-key encryption, solving the problem of slow encryption which asymmetric algorithms suffer from. Public-key encryption provides a solution to key distribution and data transmission issues. Hence, performance and key distribution are improved by combining both approaches without any sacrifice in security.

On the othe hand, the PGP process described above can be a complex process for users requiring though some training. Apart of awareness it is important both parties to have installed PGP-compatible programs to be able to exchange PGP messages (Benz, 2001).

PGP2 Configuration

PGP2 implements the public key encryption using the RSA and IDEA algorithms to provide secure electronic mail communication either between individuals or known sources (e. g. EDI). This approach of authentication through cryptography ensures that the encrypted message does not provide any information that it is carrying even if it is intercepted by attackers. The recipient of the mail can decrypt the mail using the associated private or public key (Simmons, 1993).

This approach verifies that the recipient is certain that the message is from the original sender and the contents has not been changed or lost during data transmission from sender to recipient. Furthermore, a digital signature

can be attached together with the mail to further confirm the identity of the sender. This approach complements password authentication and is therefore utilized in interpersonal communication between known parties.

Similar combination of asymmetric and symmetric encryption techniques can be generalized to other application domains involving collaboration between large groups. Common prerequisite is to obtain the secret key information before the recipient is able to decrypt the mail.

HARDWARE-BASED IMPLEMENTATION OF RSA/IDEA ENCRYPTION

A VLSI implementation of an encryption process, which combines RSA for key exchange and IDEA for block encryption, are proposed in (Buldas and Poldre, 1997). The encryption process consists of 8 rounds. Each around contains 16-bit modular additions and multiplications which simulate the integer calculations used in RSA. Also the key inversion algorithms for both ciphers are similar. When the circuit is in block encryption mode it starts the IDEA cipher process. One The IDEA cipher applies 128-bit key for block encryption.

GnuPG Configuration

The aim of GnuPG was to create a “ digital signature” mechanism which would be compatible with OpenPGP but at the same time it should avoid the use of patented algorithms like RSA. GnuPG provides therefore encryption and decryption services based on a range of both symmetric and asymmetric algorithms (Garloff and Jaeger, 2000).

Keyring: is the key management solution of GnuPG maintaining a database of private keys and a range of corresponding public keys

Hashing modules verify the authenticity of the public keys

Web of trust: a collection of signatures which have been declared as trusted by other users form a web of trusted keys

In comparison to PGP2, GnuPG rejects emai; s signed with RSA and IDEA keys, potentially been produced by PGP2. And vice versa, PGP2 rejects emails signed with DSA/ELGAMAL keys from GnuPG. GnuPG is compatible with PGP5 (Garloff and Jaeger, 2000).

Conclusion

In this paper we illustrated that symmetric and asymmetric techniques in cryptography have their own advantages and disadvantages. Symmetric algorithms are better in performance than asymmetric methods, and require all parties to share a key before they collaborate. The asymmetric ciphers allow public key infrastructures and key exchange systems, but they consume a lot of processing resources. A hybrid cryptosystem is an approach forward which combines multiple algorithms of different types, in order to exploit the best advantage of each type.

One typical approach is to apply a symmetric module to generate a random secret key, and then apply an asymmetric module to encrypt this key using as input the receiver’s public key. After that, the message is also encrypted

using the symmetric algorithm and the secret key. Both the encrypted secret key and the ciphertext of the message are then sent to the receiver.

Since it is insufficient to use a single kind of cryptographic algorithm in applications, the hybrid approach in cryptography fills the gap of efficiency or performance of each of the types of cryptography (symmetric, asymmetric) along with the objectives of confidentiality, data integrity, authentication, and trust between two communication parties. This need is generalized to any type of collaboration between individuals (email exchange, group collaboration, file sharing, etc).