

“Overview of Modern Cryptography”

M.Kundalakesi MS(IT&M),M.Phil.,

Assistant Professor,

*Department of computer applications and software system,
Sri Krishna Arts and Science College,
Coimbatore-641008.*

Sharmathi.R, Akshaya.R

(Bachelor of computer applications)

*Department of computer applications and software system,
Sri Krishna Arts and Science College,
Coimbatore-641008.*

Abstract-Cryptography is the practice and study of techniques for secure communication in the presence of third parties. More generally, it is about constructing and analyzing protocols that overcome the influence of adversaries and which are related to various aspects in information security such as data confidentiality, data integrity, authentication, and non-repudiation. Modern cryptography intersects the disciplines of mathematics, computer science, and electrical engineering. Applications of cryptography include ATM cards, computer passwords, and electronic commerce.

Keywords- Plain Text, Cipher Text, Encryption.

I.INTRODUCTION

Cryptography is the science of writing in secret code and is an ancient art. It is no surprise, then, that new forms of cryptography came soon after the widespread development of computer communications. In data and telecommunications, cryptography is necessary when communicating over any untrusted medium, which includes just about any network, particularly the Internet. Within the context of any application-to-application communication, there are some specific security requirements, including:

- **Authentication:** The process of proving one's identity. (The primary forms of host-to-host authentication on the Internet today are name-based or address-based, both of which are notoriously weak.)
- **Privacy/confidentiality:** Ensuring that no one can read the message except the intended receiver.
- **Integrity:** Assuring the receiver that the received message has not been altered in any way from the original.
- **Non-repudiation:** A mechanism to prove that the sender really sent this message.

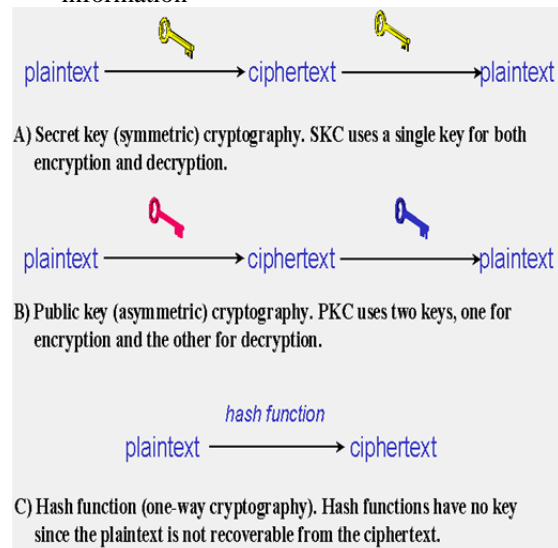
Cryptography, then, not only protects data from theft or alteration, but can also be used for user authentication. There are, in general, three types of cryptographic schemes typically used to accomplish these goals: secret key (or symmetric) cryptography, public-key (or asymmetric) cryptography, and hash functions, each of which is described below. In all cases, the initial unencrypted data is referred to as *plaintext*. It is encrypted into *cipher text*, which will in turn (usually) be decrypted into usable plaintext.

II. TYPES OF CRYPTOGRAPHIC ALGORITHMS

There are several ways of classifying cryptographic algorithms. For purposes of this paper, they will be categorized based on the number of keys that are employed

for encryption and decryption, and further defined by their application and use.

- **Secret Key Cryptography (SKC):** Uses a single key for both encryption and decryption
- **Public Key Cryptography (PKC):** Uses one key for encryption and another for decryption
- **Hash Functions:** Uses a mathematical transformation to irreversibly "encrypt" information



A. Secret Key Cryptography

With *secret key cryptography*, a single key is used for both encryption and decryption. The sender uses the key (or some set of rules) to encrypt the plaintext and sends the ciphertext to the receiver. The receiver applies the same key (or ruleset) to decrypt the message and recover the plaintext. Because a single key is used for both functions, secret key cryptography is also called *symmetric encryption*.

With this form of cryptography, it is obvious that the key must be known to both the sender and the receiver; that, in fact, is the secret. The biggest difficulty with this approach, of course, is the distribution of the key.

Secret key cryptography algorithms that are in use today include:

- **DATA ENCRYPTION STANDARD (DES):** The most common SKC scheme used today, DES was designed by IBM in the 1970s and adopted by the National Bureau of Standards (NBS) [now the National Institute for Standards and Technology (NIST)] in 1977 for commercial and unclassified

government applications. DES is a block-cipher employing a 56-bit key that operates on 64-bit blocks. DES has a complex set of rules and transformations that were designed specifically to yield fast hardware implementations and slow software implementations, although this latter point is becoming less significant today since the speed of computer processors is several orders of magnitude faster today than twenty years ago. IBM also proposed a 112-bit key for DES, which was rejected at the time by the government; the use of 112-bit keys was considered in the 1990s, however, conversion was never seriously considered.

- **ADVANCED ENCRYPTION STANDARD (AES):** In 1997, NIST initiated a very public, 4-1/2 year process to develop a new secure cryptosystem for U.S. government applications. The result, the Advanced Encryption Standard became the official successor to DES in December 2001. AES uses an SKC scheme called Rijndael, a block cipher designed by Belgian cryptographers Joan Daemen and Vincent Rijmen. The algorithm can use a variable block length and key length; the latest specification allowed any combination of keys lengths of 128, 192, or 256 bits and blocks of length 128, 192, or 256 bits. NIST initially selected Rijndael in October 2000 and formal adoption as the AES standard came in December 2001. FIPS PUB 197 describes a 128-bit block cipher employing a 128-, 192-, or 256-bit key.
- **CAST-128/256:** CAST-128, described in Request for Comments (RFC) 2144, is a DES-like substitution-permutation crypto algorithm, employing a 128-bit key operating on a 64-bit block. CAST-256 (RFC 2612) is an extension of CAST-128, using a 128-bit block size and a variable length (128, 160, 192, 224, or 256 bit) key. CAST is named for its developers, Carlisle Adams and Stafford Tavares and is available internationally. CAST-256 was one of the Round 1 algorithms in the AES process.
- **INTERNATIONAL DATA ENCRYPTION ALGORITHM (IDEA):** Secret-key cryptosystem written by Xuejia Lai and James Massey, in 1992 and patented by Ascom; a 64-bit SKC block cipher using a 128-bit key. Also available internationally.
- **RIVEST CIPHERS (aka Ron's Code):** Named for Ron Rivest, a series of SKC algorithms.
 - ✓ **RC1:** Designed on paper but never implemented.
 - ✓ **RC2:** A 64-bit block cipher using variable-sized keys designed to replace DES. Its code has not been made public although many companies have licensed RC2 for use in their products. Described in RFC 2268.
 - ✓ **RC3:** Found to be breakable during development.

- ✓ **RC4:** A stream cipher using variable-sized keys; it is widely used in commercial cryptography products. An update to RC4, called Spritz, was designed by Rivest and Jacob Schuldt.
- ✓ **RC5:** A block-cipher supporting a variety of block sizes (32, 64, or 128 bits), key sizes, and number of encryption passes over the dRFC_2040RC6

- **BLOWFISH:** A symmetric 64-bit block cipher invented by Bruce Schneier; optimized for 32-bit processors with large data caches, it is significantly faster than DES on a Pentium/PowerPC-class machine. Key lengths can vary from 32 to 448 bits in length. Blowfish, available freely and intended as a substitute for DES or IDEA, is in use in a large number of products.
- **TWOFISH:** A 128-bit block cipher using 128-, 192-, or 256-bit keys. Designed to be highly secure and highly flexible, well-suited for large microprocessors, 8-bit smart card microprocessors, and dedicated hardware. Designed by a team led by Bruce Schneier and was one of the Round 2 algorithms in the AES process.

B. Public-Key Cryptography

Public-key cryptography has been said to be the most significant new development in cryptography in the last 300-400 years. Modern PKC was first described publicly by Stanford University professor Martin Hellman and graduate student Whitfield Diffie in 1976. Their paper described a two-key crypto system in which two parties could engage in a secure communication over a non-secure communications channel without having to share a secret key.

PKC depends upon the existence of so-called *one-way functions*, or mathematical functions that are easy to compute whereas their inverse function is relatively difficult to compute. Let me give you two simple examples:

1. *Multiplication vs. factorization*
2. *Exponentiation vs. logarithms*

Public-key cryptography algorithms that are in use today for key exchange or digital signatures include:

- **RSA:** The first, and still most common, PKC implementation, named for the three MIT mathematicians who developed it — Ronald Rivest, Adi Shamir, and Leonard Adleman. RSA today is used in hundreds of software products and can be used for key exchange, digital signatures, or encryption of small blocks of data. RSA uses a variable size encryption block and a variable size key. The key-pair is derived from a very large number, n , that is the product of two prime numbers chosen according to special rules; these primes may be 100 or more digits in length each, yielding an n with roughly twice as many digits as the prime factors. The public key information includes n and a derivative of one of the factors

of n ; an attacker cannot determine the prime factors of n (and, therefore, the private key) from this information alone and that is what makes the RSA algorithm so secure. (Some descriptions of PKC erroneously state that RSA's safety is due to the difficulty *factoring* large prime numbers. In fact, large prime numbers, like small prime numbers, only have two factors!) The ability for computers to factor large numbers, and therefore attack schemes such as RSA, is rapidly improving and systems today can find the prime factors of numbers with more than 200 digits. Nevertheless, if a large number is created from two prime factors that are roughly the same size, there is no known factorization algorithm that will solve the problem in a reasonable amount of time; a 2005 test to factor a 200-digit number took 1.5 years and over 50 years of compute time. Regardless, one presumed protection of RSA is that users can easily increase the key size to always stay ahead of the computer processing curve. As an aside, the patent for RSA expired in September 2000 which does not appear to have affected RSA's popularity one way or the other.

- *DIFFIE-HELLMAN*: After the RSA algorithm was published, Diffie and Hellman came up with their own algorithm. D-H is used for secret-key key exchange only, and not for authentication or digital signatures.
- *DIGITAL SIGNATURE ALGORITHM (DSA)*: The algorithm specified in NIST's Digital Signature Standard (DSS), provides digital signature capability for the authentication of messages.
- *ElGamal*: Designed by Taher Elgamal, a PKC system similar to Diffie-Hellman and used for key exchange.
- *Elliptic Curve Cryptography (ECC)*: A PKC algorithm based upon elliptic curves. ECC can offer levels of security with small keys comparable to RSA and other PKC methods. It was designed for devices with limited compute power and/or memory, such as smartcards and PDAs. Other references include "The Importance of ECC" Web page and the "Online Elliptic Curve Cryptography Tutorial", both from Certicom. See also RFC 6090 for a review of fundamental ECC algorithms and *The Elliptic Curve Digital Signature Algorithm (ECDSA)* for details about the use of ECC for digital signatures.
- *Public-Key Cryptography Standards (PKCS)*: A set of interoperable standards and guidelines for public-key cryptography, designed by RSA Data Security Inc.
 - PKCS #1: RSA Cryptography Standard (Also RFC 3447)
 - PKCS #2: *Incorporated into PKCS #1.etc*
- *Cramer-Shoup*: A public-key cryptosystem proposed by R. Cramer and V. Shoup of IBM in 1998.
- *Key Exchange Algorithm (KEA)*: A variation on Diffie-Hellman; proposed as the key exchange method for Capstone.
- *LUC*: A public-key cryptosystem designed by P.J. Smith and based on Lucas sequences. Can be used for encryption and signatures, using integer factoring.

C. Hash Functions

Hash functions, also called *message digests* and *one-way encryption*, are algorithms that, in some sense, use no key. Instead, a fixed-length hash value is computed based upon the plaintext that makes it impossible for either the contents or length of the plaintext to be recovered. Hash algorithms are typically used to provide a *digital fingerprint* of a file's contents, often used to ensure that the file has not been altered by an intruder or virus. Hash functions are also commonly employed by many operating systems to encrypt passwords. Hash functions, then, provide a measure of the integrity of a file. Hash algorithms that are in common use today include:

- *Message Digest (MD) algorithms*: A series of byte-oriented algorithms that produce a 128-bit hash value from an arbitrary-length message.
 - *MD2 (RFC 1319)*: Designed for systems with limited memory, such as smart cards. (MD2 has been relegated to historical status, per [RFC 6149](#).)
 - *MD4 (RFC 1320)*: Developed by Rivest, similar to MD2 but designed specifically for fast processing in software. (MD4 has been relegated to historical status, per RFC 6150.)
 - *MD5 (RFC 1321)*: Also developed by Rivest after potential weaknesses were reported in MD4; this scheme is similar to MD4 but is slower because more manipulation is made to the original data. MD5 has been implemented in a large number of products although several weaknesses in the algorithm were demonstrated by German cryptographer Hans Dobbertin in 1996 ("Cryptanalysis of MD5 Compress").
- *Secure Hash Algorithm (SHA)*: Algorithm for NIST's Secure Hash Standard (SHS).
 - SHA-1 produces a 160-bit hash value and was originally published as FIPS PUB 180-1 and RFC 3174.
 - SHA-2, originally described in FIPS PUB 180-2 and eventually replaced by FIPS PUB 180-3 and FIPS PUB 180-4), comprises five algorithms in the SHS: SHA-1 plus SHA-224, SHA-256, SHA-384, and SHA-512 which can produce hash values that are 224, 256, 384, or 512 bits in length, respectively. SHA-2 recommends use of SHA-1, SHA-224, and SHA-256 for messages less than 2^{64} bits in length, and employs a 512 bit

block size; SHA-384 and SHA-512 are recommended for messages less than 2^{128} bits in length, and employs a 1,024 bit block size. FIPS PUB 180-4 also introduces the concept of a truncated hash in SHA-512/ t , a generic name referring to a hash value based upon the SHA-512 algorithm that has been truncated to t bits; SHA-512/224 and SHA-512/256 are specifically described. SHA-224, -256, -384, and -512 are also described in RFC 4634.

- SHA-3 is a proposed new SHS algorithm. Although there have not been any successful attacks on SHA-2, NIST decided that having an alternative to SHA-2 using a different algorithm would be prudent. In 2007, they launched a SHA-3 Competition to find that alternative. In 2012, NIST announced that after reviewing 64 submissions, the winner was Keccak (pronounced "catch-ack"). Based upon a sponge function — which is different from the algorithm used for SHA-1 and SHA-2 — SHA-3 will employ the same hash lengths as SHA-2. SHA-3 should be published by NIST by the middle of 2014.
- *RIPEMD*: A series of message digests that initially came from the RIPE (RACE Integrity Primitives Evaluation) project. RIPEMD-160 was designed by Hans Dobbertin, Antoon Bosselaers, and Bart Preneel, and optimized for 32-bit processors to replace the then-current 128-bit hash functions. Other versions include RIPEMD-256, RIPEMD-320, and RIPEMD-128.
- *HAVAL (HAsH of VArIable Length)*: Designed by Y. Zheng, J. Pieprzyk and J. Seberry, a hash algorithm with many levels of security. HAVAL can create hash values that are 128, 160, 192, 224, or 256 bits in length.
- *Whirlpool*: A relatively new hash function, designed by V. Rijmen and P.S.L.M. Barreto. Whirlpool operates on messages less than 2^{256} bits in length, and produces a message digest of 512 bits. The design of this has function is very different than that of MD5 and SHA-1, making it immune to the same attacks as on those hashes (see below).
- *Tiger*: Designed by Ross Anderson and Eli Biham, Tiger is designed to be secure, run efficiently on 64-bit processors, and easily replace MD4, MD5, SHA and SHA-1 in other applications. Tiger/192 produces a 192-bit output and is compatible with 64-bit architectures; Tiger/128 and Tiger/160 produce a hash of length 128 and 160 bits, respectively, to provide compatibility with the other hash functions mentioned above.

III. TRUST MODELS

Secure use of cryptography requires trust. While secret key cryptography can ensure message confidentiality and hash codes can ensure integrity, none of this works without trust. In SKC, Alice and Bob had to share a secret key. PKC solved the secret distribution problem, but how does Alice really know that Bob is who he says he is? Just because Bob has a public and private key, and purports to be "Bob," how does Alice know that a malicious person (Mallory) is not pretending to be Bob?

There are a number of *trust models* employed by various cryptographic schemes. This section will explore three of them:

- The web of trust employed by Pretty Good Privacy (PGP) users, who hold their own set of trusted public keys.
- Kerberos, a secret key distribution scheme using a trusted third party.
- Certificates, which allow a set of trusted third parties to authenticate each other and, by implication, each other's users.

Each of these trust models differs in complexity, general applicability, scope, and scalability.

CONCLUSION

Cryptography is used to achieve few goals like Confidentiality, Data integrity, Authentication etc. of the sent data. Now, in order to achieve these goals various cryptographic algorithms are developed by various people. For a very minimal amount of data those algorithms wouldn't be cost effective since those are not designed for small amount of data. A single algorithm is used for both encryption and decryption i.e. it is fallen under secret key cryptographic algorithm. But as public key cryptography is more secured then secret key algorithm. It is hard to say that any one is better than the others; it depend upon your application. One of the biggest and fastest growing applications of cryptography today, though, is electronic commerce (e-commerce), a term that itself begs for a formal definition. PGP's web of trust is easy to maintain and very much based on the reality of users as people. Kerberos overcomes many of the problems of PGP's web of trust, in that it is scalable and its scope can be very large.

REFERENCES

1. <http://en.wikipedia.org/wiki/Cryptography#>
2. <http://www.wisegeek.com/what-are-cryptographic-algorithms.htm#didyouknowout>
3. http://en.wikipedia.org/wiki/Asymmetric_key_algorithm
4. http://en.wikipedia.org/wiki/Digital_Signature_Algorithm
5. http://en.wikipedia.org/wiki/Elliptic_curve_cryptography
6. [http://en.wikipedia.org/wiki/Blowfish_\(cipher\)](http://en.wikipedia.org/wiki/Blowfish_(cipher))