NETWORK LABORATORY
Electrical Engineering Department, 2nd floor
Universitas Indonesia
Depok. 16424

NETWORK SECURITY 2019/2020

BASIC THEORY

MODULE 2: SECURE COMMUNICATION (CRYPTOGRAPHY)

OBJECTIVES:

1. Define cryptography
2. Describe hash, symmetric, and asymmetric cryptographic algorithms
3. List the various ways in which cryptography is used

BASIC THEORY:

What is Cryptography?

"Scrambling" data so that it cannot be read is a process known as cryptography (from Greek words meaning hidden writing). Cryptography is the science of transforming information into a secure form so that unauthorized persons cannot access it. Whereas cryptography scrambles a message so that it cannot be understood, steganography hides the existence of the data. What appears to be a harmless image can contain hidden data, usually some type of message, embedded within the image. Steganography takes the data, divides it into smaller sections, and hides it in unused portions of the file.

Steganography may hide data in the file header fields that describe the file, between sections of the metadata (data that is used to describe the content or structure of the actual data), or in the areas of a file that contain the content itself. Steganography can use a wide variety of file types—image files, audio files, video files, etc.—to hide messages and data.

Data in an unencrypted form is called cleartext data. Cleartext data is "in the clear" and thus can be displayed as is without any decryption being necessary. Plaintext data is cleartext data that is to be encrypted and is the result of decryption as well. Plaintext may be considered as a special instance of cleartext.

Plaintext data is input into a cryptographic algorithm, which consists of procedures based on a mathematical formula used to encrypt and decrypt the data. A key is a mathematical value entered into the algorithm to produce ciphertext, or encrypted data. Just as a key is inserted into a door lock to lock the door, in cryptography a unique mathematical key is input into the encryption algorithm to "lock down" the data by creating the ciphertext. Once the ciphertext needs to be returned to plaintext, the reverse process occurs with a decryption algorithm and key. The cryptographic process is illustrated in Figure

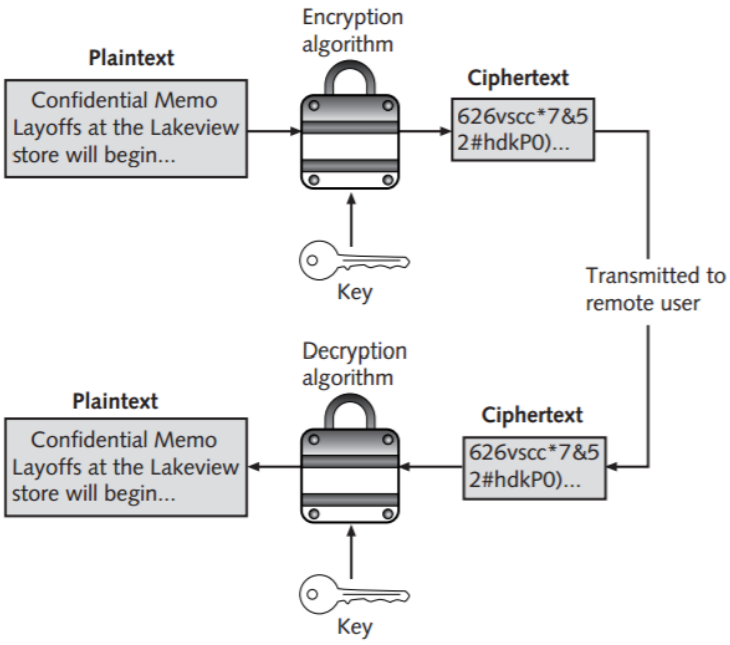

Figure 1. Cryptography

Cryptographic Algorithms

One of the fundamental differences in cryptographic algorithms is the amount of data that is processed at a time. Some algorithms use a stream cipher. A stream cipher takes one character and replaces it with one character, as shown in Figure

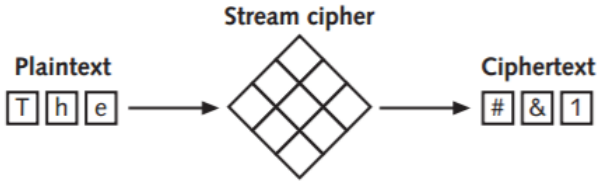

Figure 2. Stream Cipher

The simplest type of stream cipher is a substitution cipher. Substitution ciphers simply substitute one letter or character for another (a monoalphabetic substitution cipher), as shown in Figure 2. A more complex stream cipher that can be more difficult to break is a homoalphabetic substitution cipher that maps a single plaintext character to multiple ciphertext characters. For example, an F may map to ILS.

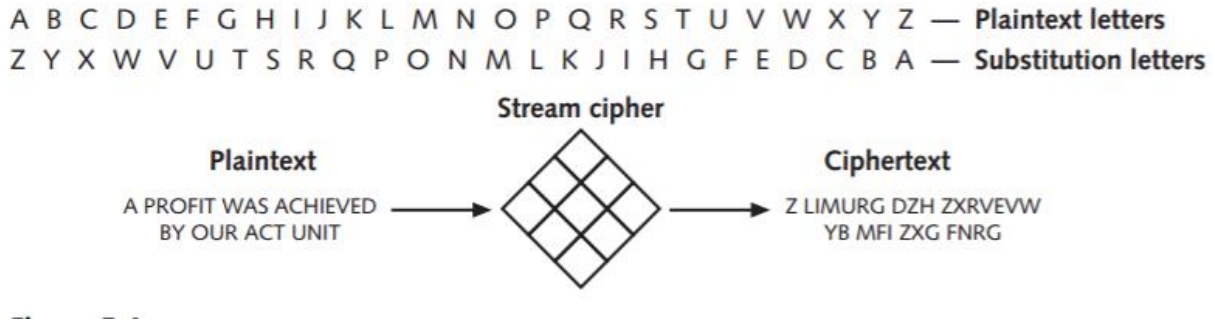

Fig 3. Subtitution Cipher

Other algorithms make use of a block cipher. Whereas a stream cipher works on one character at a time, a block cipher manipulates an entire block of plaintext at one time. The plaintext message is divided into separate blocks of 8 to 16 bytes, and then each block is encrypted independently. For additional security, the blocks can be randomized. Stream and block ciphers each have advantages and disadvantages. A stream cipher is fast when the plaintext is short, but can consume much more processing power if the plaintext is long. In addition, stream ciphers are more prone to attack because the engine that generates the stream does not vary; the only change is the plaintext itself. Because of this consistency, an attacker can examine streams and may be able to determine the key. Block ciphers are considered more secure because the output is more random. When using a block cipher, the cipher is reset to its original state after each block is processed. This results in the ciphertext being more difficult to break.

There are three broad categories of cryptographic algorithms. These are known as hash algorithms, symmetric cryptographic algorithms, and asymmetric cryptographic algorithms

Hash Algorithm

The most basic type of cryptographic algorithm is a one-way hash algorithm. A hash algorithm creates a unique “digital fingerprint” of a set of data and is commonly called hashing. This fingerprint, called a digest (sometimes called a message digest or hash), represents the contents. Although hashing is considered a cryptographic algorithm, its purpose is not to create ciphertext that can later be decrypted. Instead, hashing is “one-way” in that its contents cannot be used to reveal the original set of data. Hashing is used primarily for comparison purposes. A secure hash that is created from a set of data cannot be reversed. For example, if 12 is multiplied by 34 the result is 408. If a user was asked to determine the two numbers used to create the number 408, it would not be possible to “work backward” and derive the original numbers with absolute certainty because there are too many mathematical possibilities (204 +204, 407+1, 999–591, 361+47, etc.). Hashing is similar in that it is used to create a value, but it is not possible to determine the original set of data.

A hashing algorithm is considered secure if it has these characteristics:

- Fixed size.
  A digest of a short set of data should produce the same size as a digest of a long set of data. For example, a digest of the single letter a is 86be7afa339d0fc7cfc 785e72f578d33, while a digest of 1 million occurrences of the letter a is 4a7f5723f95 4eba1216c9d8f6320431f, the same length.
- Unique.
  Two different sets of data cannot produce the same digest, which is known as a collision. Changing a single letter in one data set should produce an entirely different digest. For example, a digest of Sunday is 0d716e73a2a7910bd4ae63407056d79b while a digest of sunday (lowercase s) is 3464eb71bd7a4377967a30da798a1b54.
- Original.
  It should be impossible to produce a data set that has a desired or predefined hash
- Secure.
  The resulting hash cannot be reversed in order to determine the original plaintext.

Hashing is used primarily to determine the integrity of a message or contents of a file. In this case, the digest serves as a check to verify that the original contents have not changed. For example, digest values are often posted on websites in order to verify the integrity of files that can be downloaded. A user can create a digest on a file after it has been downloaded and then compare that value with the original digest value posted on the website. A match indicates that the integrity of the file has been preserved.

The most common hash algorithms are Message Digest, Secure Hash Algorithm, Whirlpool, and RIPEMD.

Message Digest (MD)

One of the most common one-way hash algorithms is the Message Digest (MD), which has three different versions. Message Digest 2 (MD2) was one of the early hash algorithms. It takes plaintext of any length and creates a digest 128 bits in length. MD2 divides the plaintext into multiple 128-bit sections. If the message is less than 128 bits, however, extra padding is added. MD2 was developed in 1989 and was optimized to run on Intel-based microcomputers that processed 8 bits at a time. MD2 is no longer considered secure. Message Digest 4 (MD4) was developed in 1990 for computers that processed 32 bits at a time. Like MD2, MD4 creates a digest of 128 bits. The plaintext message itself is padded to a length of 512 bits instead of 128 bits as with MD2. Flaws in the MD4 hash algorithm have prevented this MD from being widely accepted.

Message Digest 5 (MD5), the current MD version and a revision of MD4, was created the following year and designed to address MD4's weaknesses. Like MD4, the length of a message is padded to 512 bits in length. The hash algorithm then uses four variables of 32 bits each in a round-robin fashion to create a value that is compressed to generate the digest. Weaknesses have been

revealed in the compression function that could lead to collisions, so some security experts recommend that a more secure hash algorithm be used instead.

Secure Hash Algorithm (SHA)

A more secure hash than MD is the Secure Hash Algorithm (SHA). Like MD, the SHA is a family of hashes. The first version was SHA-0, which due to a flaw was withdrawn shortly after it was first released. Its successor, SHA-1, was developed in 1993 by the U.S. National Security Agency (NSA) and the National Institute of Standards and Technology (NIST). It is patterned after MD4 and MD5, but creates a digest that is 160 bits instead of 128 bits in length. SHA pads messages of less than 512 bits with zeros and an integer that describes the original length of the message. The padded message is then run through the SHA algorithm to produce the digest.

Another family of SHA hashes are known as SHA-2. SHA-2 actually is comprised of six variations: SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, and SHA-512/256 (the last number indicates the length in bits of the digest that is generated). SHA-2 is currently considered to be a secure hash. In 2007, an open competition for a new SHA-3 hash algorithm was announced. Of the 51 entries that were accepted to Round 1 of the competition, only 14 were selected for Round 2 (one of the entries rejected was a new MD6). In late 2010, five finalists moved to Round 3. In late 2012 the final winner of the competition was announced. The winning algorithm, Keccak (pronounced catch-ack), was created by four security researchers from Italy and Belgium. Keccak will become NIST's SHA-3 hash algorithm.

One of the design goals of SHA-3 was for it to be dissimilar to previous hash algorithms like MD5 and SHA-0, SHA-1, and SHA-2. Because successful attacks have been launched against MD5 and SHA-0 as well as theoretical attacks on SHA-1, making SHA-3 different would prevent attackers from building upon any previous work to compromise hashing algorithms. SHA-3 uses a sponge function instead of stream or block ciphers.

Whirlpool

Whirlpool is a relatively recent cryptographic hash function that has received international recognition and adoption by standards organizations, including the International Organization for Standardization (ISO). Named after the first galaxy recognized to have a spiral structure, it creates a digest of 512 bits. Whirlpool is being implemented in several new commercial cryptography applications.

Symmetric Cryptographic Algorithms

The original cryptographic algorithms for encrypting and decrypting data are symmetric cryptographic algorithms. Symmetric cryptographic algorithms use the same single key to encrypt and decrypt a document. Unlike hashing in which the hash is not intended to be decrypted, symmetric algorithms are designed to encrypt and decrypt the ciphertext. It is therefore essential

that the key be kept private (confidential), because if an attacker obtained the key he could read all the encrypted documents. For this reason, symmetric encryption is also called private key cryptography. Symmetric encryption is illustrated in Figure where identical keys are used to encrypt and decrypt a document.

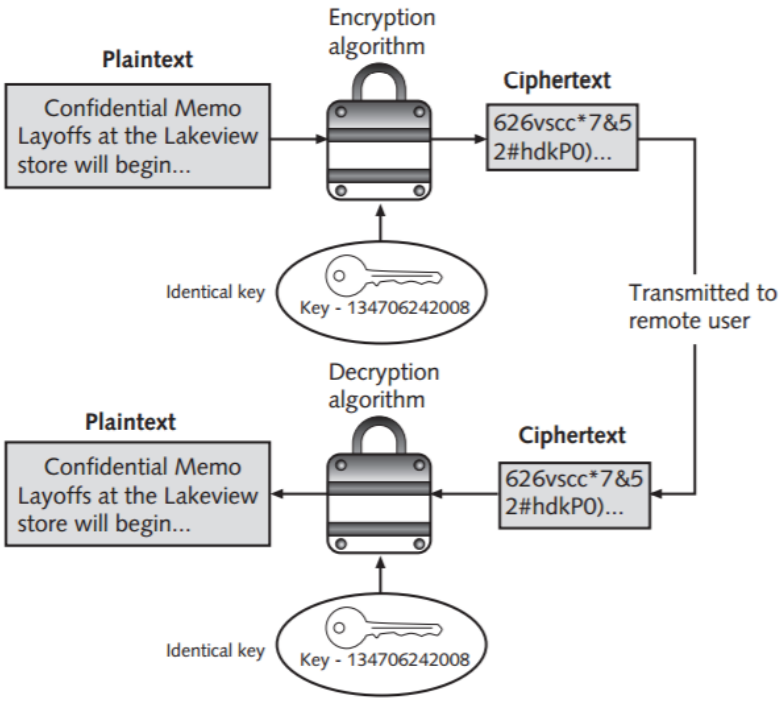

Figure 4. Symmetric cryptography

Common symmetric cryptographic algorithms include the Data Encryption Standard, Triple Data Encryption Standard, Advanced Encryption Standard, and several other algorithms.

Data Encryption Standard (DES)

DES is a block cipher. It divides plaintext into 64-bit blocks and then executes the algorithm 16 times. Four modes of DES encryption exist. Although DES was once widely implemented, its 56-bit key is no longer considered secure and has been broken several times. It is not recommended for use.

Triple Data Encryption Standard (3DES)

Triple Data Encryption Standard (3DES) is designed to replace DES. As its name implies, 3DES uses three rounds of encryption instead of just one. The ciphertext of one round becomes the entire input for the second iteration. 3DES employs a total of 48 iterations in its encryption (3 iterations times 16 rounds). The most secure versions of 3DES use different keys for each round, as shown in Figure 5. By design 3DES performs better in hardware than as software.

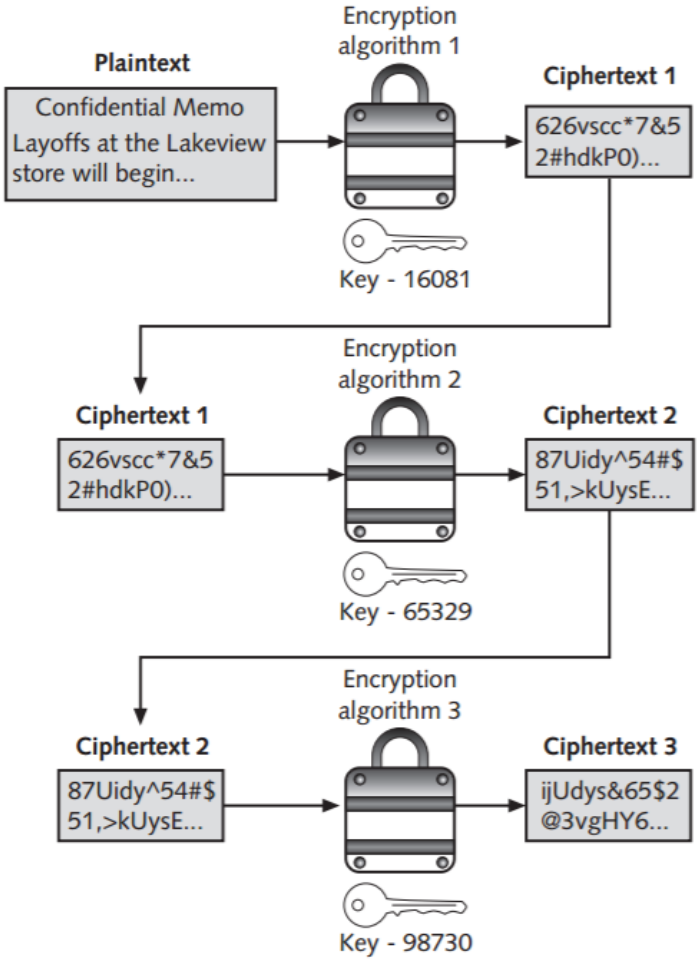

Figure 5. 3DES

Advanced Encryption Standard (AES)

AES performs three steps on every block (128 bits) of plaintext. Within step 2, multiple rounds are performed depending upon the key size: a 128-bit key performs 9 rounds, a 192-bit key performs 11 rounds, and a 256-bit key, known as AES-256, uses 13 rounds. Within each round, bytes are substituted and rearranged, and then special multiplication is performed based on the new arrangement. To date, no attacks have been successful against AES.

Asymmetric Cryptographic Algorithms

Asymmetric encryption uses two keys instead of only one. These keys are mathematically related and are known as the public key and the private key. The public key is known to everyone and can be freely distributed, while the private key is known only to the individual to whom it belongs. When Kiel wants to send a secure message to Nindya, he uses Nindya's public key to encrypt the message. Nindya then uses her private key to decrypt it

Several important principles regarding asymmetric cryptography are:

- Key pairs.
  Unlike symmetric cryptography that uses only one key, asymmetric cryptography requires a pair of keys.
- Public key.
  Public keys by their nature are designed to be "public" and do not need to be protected. They can be freely given to anyone or even posted on the Internet.
- Private key.
  The private key should be kept confidential and never shared.

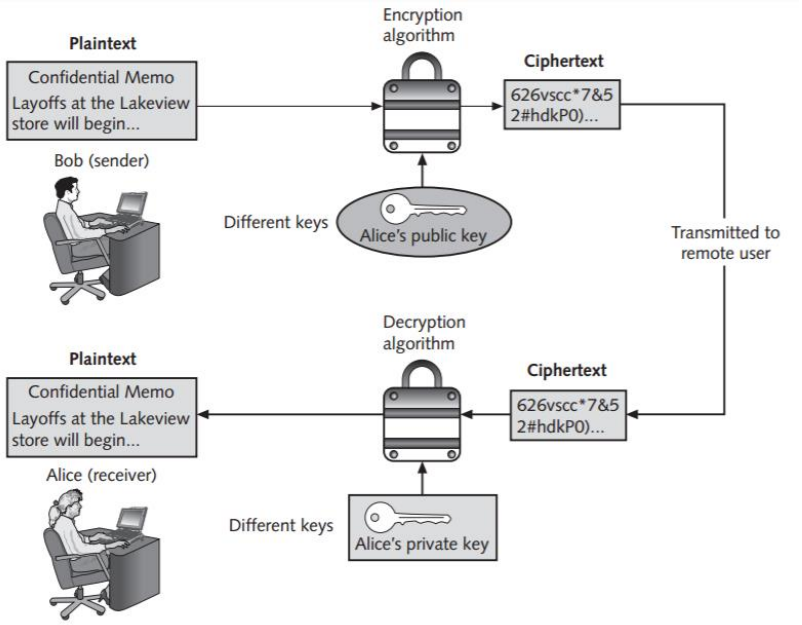

Figure 6. Asymmetric