"CRYPTOGRAPHY AND DATA SECURITY"

Part I

Prof. Dr. V.N.Yarmolik

Lecture course

Contents

1. Introduction: Data security. Fundamental concepts of cryptography.

2. Information theory: Shannon's theory of the systems secrecy. The quantity of Information. Entropy.

3. Transposition cipher: Simple transposition. Product cipher.

4. Substitution ciphers: Simple substitution cipher. Caesar cipher. Vigener cipher.

5. Substitution ciphers: Mono and Poly alphabetic substitution. Playfair cipher.

6. Substitution ciphers: Rotor machines. The Enigma: a unique rotor machine.

7. **Data Encryption Standard (DES):** History of the DES. DES algorithm

8. Data Encryption Standard (DES): Weak and semi weak keys. Advanced DES versions.

9. Number theory: Prime numbers. Euler's function. Euler's theorem. Congruence.

"CRYPTOGRAPHY AND DATA SECURITY"

Part I

Prof. Dr. V.N.Yarmolik

Lecture course

10. Public Key Cipher: Principles of the public key cipher. One-way function. Deffie and Hellman algorithm.

11. Knapsack Cipher: The trapdoor knapsack. Practical aspects of the trapdoor knapsack.

12. RSA Cipher: Riverst, Shamir and Adleman public key cipher. Practical aspects.

13. Elliptic Cipher: Elliptic Curve Public key cryptosystems.

14. Linear Feedback Shift Register: Key encoding by LFSR. M-sequences.

15. Stream cipher: Synchronous stream ciphers. Self-synchronizing cipher.

Laboratory Projects:

1. Transposition cipher investigations.
2. Simple substitution cipher investigations.
3. Modern electronic Rotor machines investigations.
4. Data Encryption Standard (DES) software implementation and investigations.
5. RSA algorithms investigations.
6. Pseudorandom number generators implementations.
7. Stream cipher investigations.
8. Elliptic Cipher investigations.

1. Introduction.

1.1. Cryptography

Cryptography is the science and study of secret writing.

A ***cipher*** is a secret method of writing, whereby ***plaintext*** (or ***cleartext***) is transformed into ***ciphertext*** (***cryptogram***).

Encipherment (***encryption***) is the process of transforming plaintext into ciphertext.

Decipherment (***decryption***) is the reverse process of transforming ciphertext into plaintext.

Both encipherment and decipherment are controlled by a cryptographic ***key*** or ***keys.***

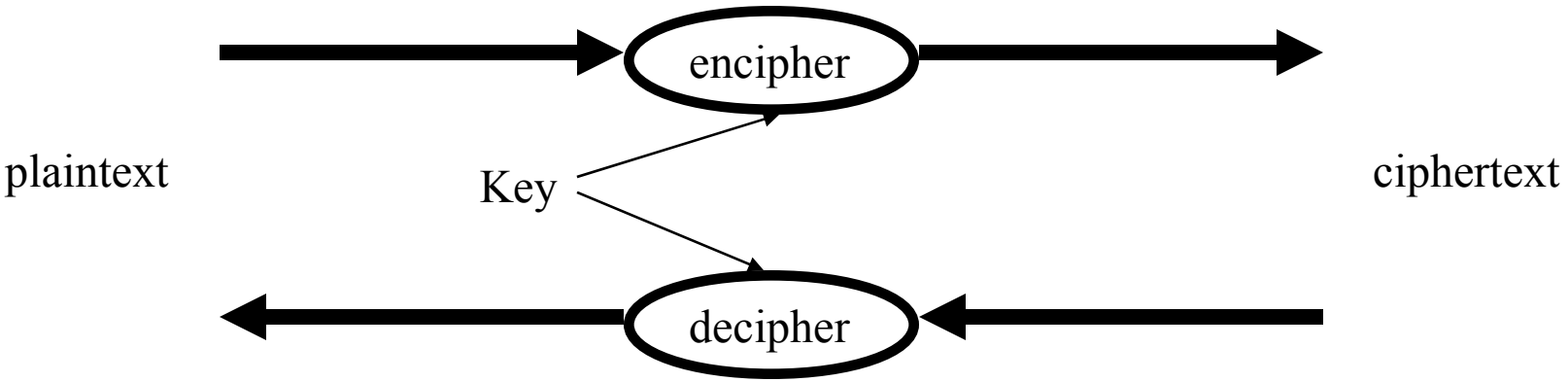

Fig.1.1. Secret writing

1. Introduction.

1.2. Transposition ciphers

There are two basic types of ciphers ***transpositions*** and ***substitutions.***

Transposition ciphers rearrange bits or characters in the data. With a "rail-fence" cipher, for example, the letters of plaintext message are written down in a pattern resembling a rail fence, and then removed by rows. The following simple example illustrate this method.

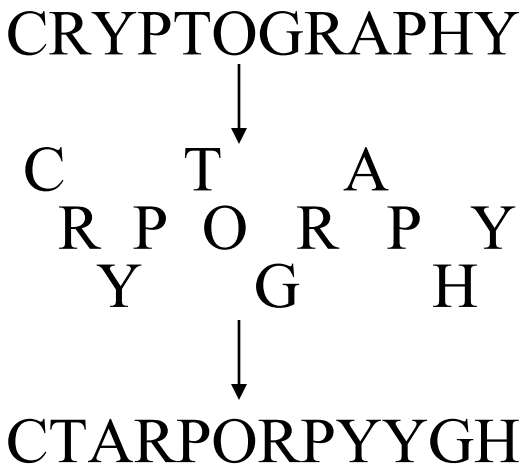

Fig.1.2. "Rail-fence" transposition cipher

CRYPTOGRAPHY - is the plaintext, and STARPORPYYGH is the corresponding ciphertext. The key to the cipher is given by the depth of the fence, which in this example is 3. To decipher the ciphertext the same key and inverse procedure have to be used.

1. Introduction.

1.3. Substitutions ciphers

Substitution ciphers replace bits, characters, or blocks of characters with substitutes. A simplest type of substitution cipher shifts each letter in the English alphabet forward by ***k*** positions cyclically (shifts past Z cycle back to A). ***k*** is the key to the cipher. This type of cipher is often called a Caesar cipher because Julius Caesar used it with ***k***=3. Next Fig.1.3 illustrate Caesar's method.

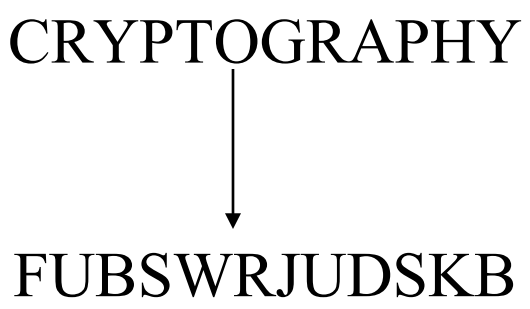

Fig.1.3. Caesar's substitution cipher

In practical applications, substitution is usually combined with transposition. The Data Encryption Standard (DES), for example, encipher 64-bit blocks using a combination of transposition and substitution.

A ***code*** is a special type of substitution cipher that uses a "code book" as the key. Plaintext words or phrases are entered into the code book together with their ciphertext substitutes, for example for the word CRYPTOGRAPHY the code is 7905.

1. Introduction.

1.4. Cryptanalysis

Cryptoanalysis is the science and study of methods of breaking cipher. A cipher is breakable if it is possible to determine the plaintext or key from the ciphertext, or to determine the key from plaintext-ciphertext pairs. There are three basic methods of attack: ***ciphertext-only***, ***known-plaintext***, and ***chosen-plaintext***.

Under a ***ciphertext-only*** attack, a cryptanalyst must determine the key solely from intercepted ciphertext, though the method of encryption, the plaintext language, the subject matter of the ciphertext, and certain probable words may be known. For example, a message describing the location of buried treasure would probably contain words such as BURIED, TREASURE, NORTH, TURN, RIGHT, MILES,... .

Under a ***known-plaintext attack,*** a cryptanalyst knows some plaintext-ciphertext pairs. As an example, suppose an enciphered message transmitted from a user's terminal to the computer is intercepted by a cryptanalyst who knows that the message begin with a standard header such as "LOGIN"

Under a ***chosen-plaintext attack***, a cryptoanalyst is able to acquire the ciphertext corresponding to selected plaintext. This is the most favorable case for the cryptoanalyst.

A cipher is ***unconditionally secure*** if, no matter how much ciphertext is intercepted, there is not enough information in the ciphertext to determine the plaintext uniquely.

A cipher is ***computationally secure***, or ***strong***, if it cannot be broken by systematic analysis with available resources.

1. Introduction.

1.5. Data Security

Modern cryptography protects data transmitted over high-speed electronic lines or stored in computer systems. There are two principle objectives: ***secrecy*** (or ***privacy***), to prevent the unauthorized disclosure of data; and ***authenticity*** or ***integrity***), to prevent the unauthorized modification of data.

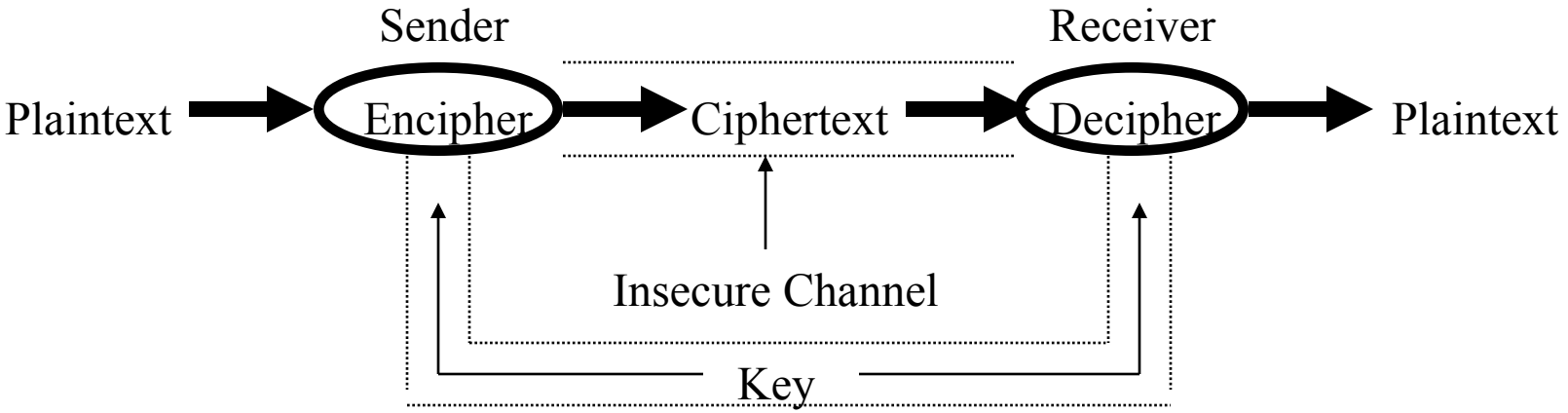

Fig.1.4. Classical information channel

Information transmitted over insecure channel is vulnerable to passive wiretapping, which threatens secrecy, and to active wiretapping, which threatens authenticity. ***Passive wiretapping*** (***eavesdropping***) refers to the interception of messages, usually without detection. ***Active wiretapping*** (***tampering***) refer to deliberate modifications made to the message stream.

1. Introduction.

1.6. Cryptographic Systems

A ***cryptographic system*** (or ***cryptosystem*** for short) has five components:

1. A ***plaintext message space***, *M.*
2. A ***cipher message space***, *C.*
3. A ***key space***, *k.*
4. A family of ***enciphering transformations***, E_k: *M* --> *C.*
5. A family of ***deciphering transformations***, D_k: *C* --> *M.*

Each enciphering transformation E_k is defined by an enciphering algorithm ***E***, which is common to every transformation in the family. and a key ***k***, which distinguishes it from the other transformations. Similarly, each deciphering transformation $\boldsymbol{D_k}$ is defined by a deciphering algorithm ***D*** and a key ***k.*** For a given ***k***, $\boldsymbol{D_k}$ is inverse of $\boldsymbol{E_k}$; that is $\boldsymbol{D_k\,(E_k\,(M))=M}$ for every plaintext message ***M***. In a given cryptographic system, the transformations $\boldsymbol{E_k}$ and $\boldsymbol{D_k}$ are described by parameter derived from ***k*** (or directly by ***k***).

Cryptosystems General Requirements

1. The system must be easy to use.
2. The enciphering and deciphering transformations must be efficient for all keys.
3. The security of the system should depend only on the secrecy of the keys and not on the secrecy of the algorithms ***E*** and ***D***.

1. Introduction.

1.7. Requirement for secrecy and authenticity

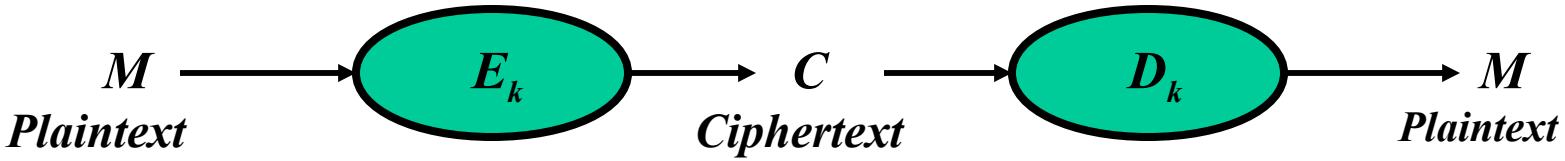

Fig.1.5. Cryptographic System.

There are specific requirements for secrecy and authenticity.

Secrecy Requirements

1. It should be computationally infeasible for a cryptanalyst to systematically determine the deciphering transformation D_k from intercepted ciphertext ***C***, even if the corresponding plaintext ***M*** is known.

2. It should be computationally infeasible for a cryptanalyst to systematically determine plaintext ***M*** from intercepted ciphertext ***C***.

Authenticity Requirements

1. It should be computationally infeasible for a cryptanalyst to systematically determine the enciphering transformation E_k given ***C*** even if the corresponding plaintext ***M*** is known.

2. It should be computationally infeasible for a cryptanalyst to systematically find ciphertext ***C'*** such that D_k ***(C')*** is valid plaintext in the set ***M***.

1. Introduction.

1.7. Simmons Cryptosystems Classifications

Simmons classifies cryptosystems as ***symmetric*** (***one-key***) and ***asymmetric*** (***two-key***). In ***symmetric*** or ***one-key*** cryptosystems the enciphering and deciphering key are the same (or easily determined from each other). This means the transformations E_k and D_k are also easily derived from each other. Until recently, all cryptosystems were one-key systems. Thus, one-key systems are also usually referred to as ***conventional*** (or ***classical***) systems. The DES is a conventional system.

One-key systems provide an exellent way of enciphering user's privite files. Each user A has privite transformations E_k and D_k for enciphering and deciphering files.

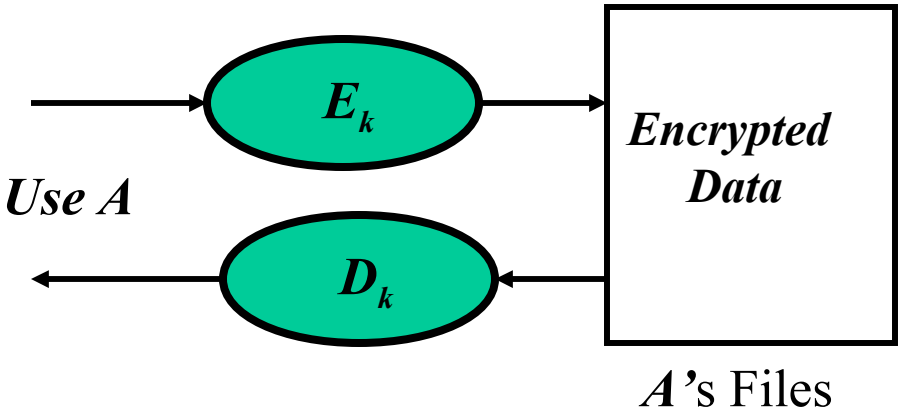

Fig.1.6. Single-key encryption of private files

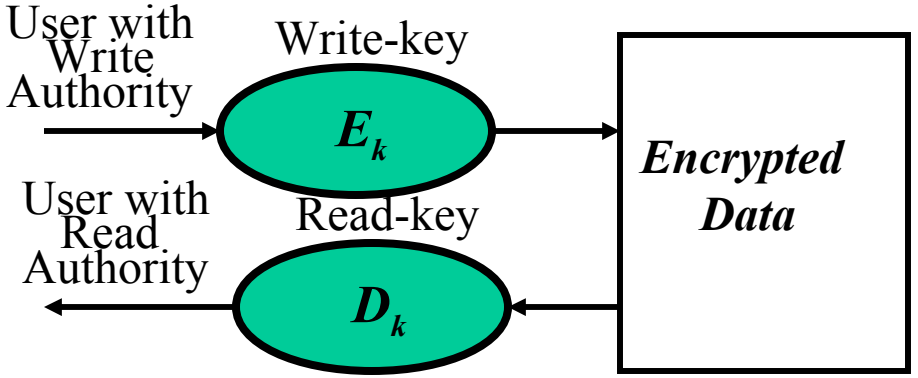

Fig.1.7. File encryption with separate Read/Write keys

1. Introduction.

1.8. Public-Key Systems

In a public-key system, each user *A* has a ***public enciphering transformation*** E_A, which may be registered with a public directory, and a ***private deciphering transformation*** D_A, which is known only to that user. The private transformation D_A is described by a private key, and the public transformation E_A by a public key derived from the private key by one-way transformation. It must be computational infeasible to determine D_A from E_A (or even to fined a transformation equivalent to D_A).

In a public-key system, secrecy and authenticity are provided by the separate transformations. Suppose user *A* wishes to send a message ***M*** to another user *B*. If *A* knows *B*'s public transformation E_B, *A* can transmit ***M*** to *B* in secrecy by sending the ciphertext $C=E_B(M)$. On receipt, *B* deciphers *C* using *B*'s private transformation , getting

$$D_B(C)=D_B(E_B(M))=M.$$

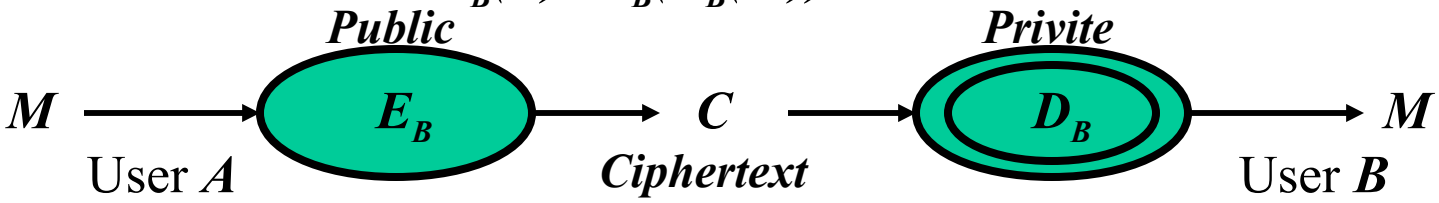

Fig.1.8. Secrecy in public-key system

1. Introduction.

1.8. Public-Key Systems

For authenticity, ***M*** must be transformed by *A*'s own private transformation D_A. Ignoring secrecy for the moment, *A* sends $C=D_A(M)$ to *B*. On receipt, *B* uses *A's* public transformation E_A to compute

$$E_A(C)=E_A(D_A(M))=M$$

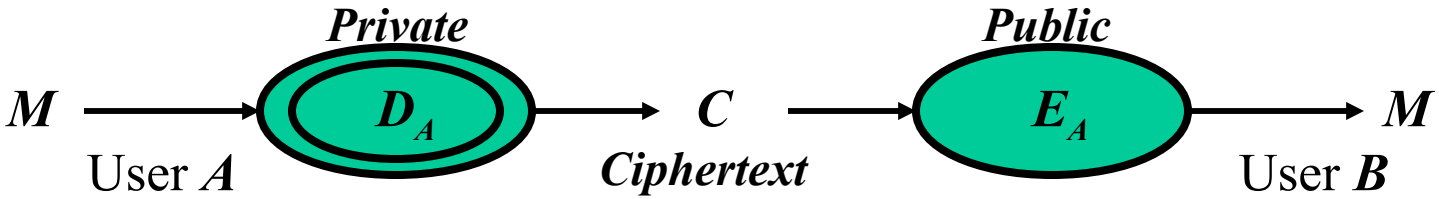

Fig.1.9. Authenticity in public-key system

To achieve both secrecy and authenticity, the sender and receiver must each apply two sets of transformations. Sender *A* generates a ciphertext $C=E_B(D_A(M))$, and *B* recovers M according to

$$E_A(D_B(C))=E_A(D_B(E_B(D_A(M))))= E_A(D_A(M))=M.$$

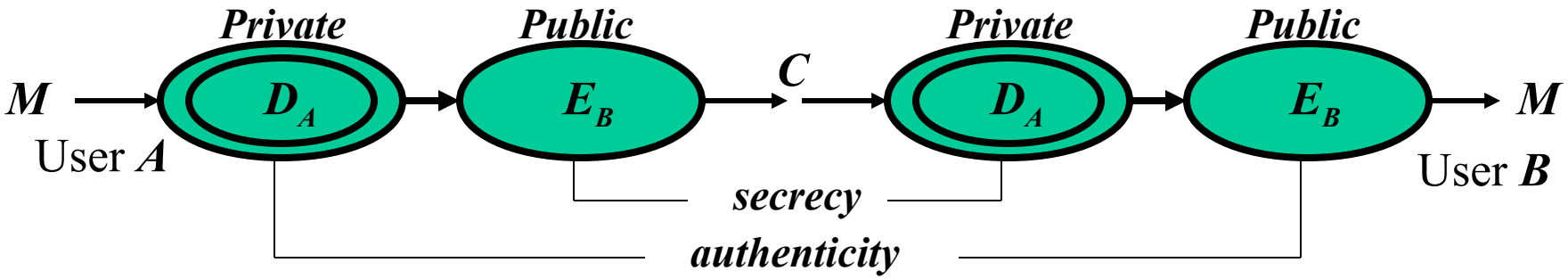

Fig.1.10. Secrecy and authenticity in public-key system

1. Introduction.

1.9. Digital Signatures

A ***digital signature*** is a property private to a user or process that is used for signing message. Let ***B*** be the recipient of a message ***M*** signed by *A*. Then ***A***'s signature must satisfy these requirements:

1. ***B*** must be able to validate ***A***'s signature on ***M***.

2. It must be impossible for anyone, including B, to ***forge A***'s signature.

3. In a case ***A*** should disavow signing a message ***M***, it must be possible for a judge or third party to resolve a dispute arising between ***A*** and ***B***.

A digital signature, therefore, establishes ***sender authenticity***; it is analogous to an ordinary written signature. By condition (2), it also establishes ***data authenticity***.

Public-key authentication systems provide a simple scheme for implementing digital signatures. Because the transformation D_A is private to ***A***, D_A serves as ***A***'s digital signature. The recipient ***B*** of a message ***M*** signed by ***A*** (i.e., transformed by D_A) is assured of both sender and data outhenticity. It is impossible for B or anyone else to forge A's signature on another message, and impossible for A to disclaim a signed document (assuming D_A has not bee lost or stolen). Because the inverse transformation E_A is public, the receiver ***B*** can readily validate the signature, and a judge can settle any disputes arising between ***A*** and ***B***. Summarizing,

1. ***A*** signs ***M*** by computing $C = D_A(M)$.

2. ***B*** validates ***A***'s signature by checking that $E_A(C)$ restores ***M***.

3. A judge resolve a dispute arising between ***A*** and ***B*** by checking