LECTURE 7

Key management

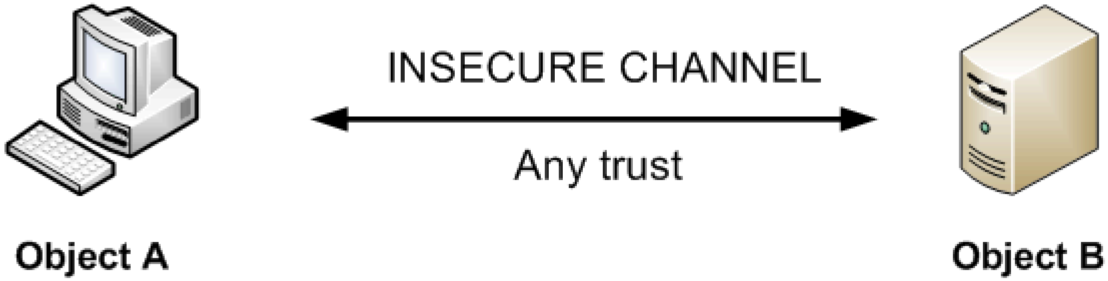
Telecommunication systems department

Lecturer: assistant professor Persikov Anatoliy Valentinovich

KEY AGREEMENT USING DIFFIE-HELLMAN PROCEDURE

The first published public-key algorithm appeared in the seminal paper by Diffie and Hellman that defined public-key cryptography and is generally referred to as Diffie-Hellman key exchange. A number of commercial products employ this key exchange technique.

The purpose of the algorithm is to enable two users to securely exchange a key that can then be used for subsequent encryption of messages. The algorithm itself is limited to the exchange of secret values.



KEY AGREEMENT USING DIFFIE-HELLMAN PROCEDURE

The Diffie-Hellman algorithm depends for its effectiveness on the difficulty of computing discrete logarithms. Briefly, we can define the discrete logarithm in the following way. First, we define a primitive root of a prime number p as one whose powers modulo p generate all the integers from 1 to $p-1$. That is, if a is a primitive root of the prime number p, then the numbers

$$a \bmod p,\ a^2 \bmod p, \ldots,\ a^{p-1} \bmod p$$

are distinct and consist of the integers from 1 through $p-1$ in some permutation.

For any integer b and a primitive root a of prime number p, we can find a unique exponent i such that

$$b \equiv a^i (\bmod p) \text{ where } 0 \leq i \leq (p-1)$$

The exponent i is referred to as the discrete logarithm of b for the base a, mod p. We express this value as $\mathrm{dlog}_{a,p}(b)$.

KEY AGREEMENT PROCEDURE

Global public elements

q	prime number
α	$\alpha < q$ and α a primitive root of q

User A key generation

Select private X_A	$X_A < q$
Calculate public Y_A	$Y_A = \alpha^{X_A} \bmod q$

User B key generation

Select private X_B	$X_B < q$
Calculate public Y_B	$Y_B = \alpha^{X_B} \bmod q$

KEY AGREEMENT PROCEDURE

Calculation of secret key by user A

$$K = (Y_B)^{X_A} \bmod q$$

Calculation of secret key by user B

$$K = (Y_A)^{X_B} \bmod q$$

For this scheme, there are two publicly known numbers: a prime number q and an integer that is a primitive root of q. Suppose the users **A** and **B** wish to exchange a key. User **A** selects a random integer $X_A < q$ and computes $Y_A = \alpha^{X_A} \bmod q$. Similarly, user **B** independently selects a random integer $X_B < q$ and computes $Y_B = \alpha^{X_B} \bmod q$. Each side keeps the X value private and makes the Y value available publicly to the other side. User **A** computes the key as $K = {Y_B}^{X_A} \bmod q$ and user **B** computes the key as $K = {Y_A}^{X_B} \bmod q$. These two calculations produce identical results:

$$K = {Y_B}^{X_A} \bmod q = (\alpha^{X_B} \bmod q)^{X_A}\, mod\, q = (\alpha^{X_B})^{X_A} \bmod q = \alpha^{X_B X_A} \bmod q =$$

$$= \alpha^{X_A X_B} \bmod q = (\alpha^{X_A})^{X_B} \bmod q = {Y_A}^{X_B} \bmod q$$

KEY AGREEMENT PROCEDURE STRENGTH

The result is that the two sides have exchanged a secret value. Furthermore, because X_A and X_B are private, an adversary only has the following ingredients to work with: q, α, Y_A, and Y_B. Thus, the adversary is forced to take a discrete logarithm to determine the key. For example, to determine the private key of user B, an adversary must compute

$$X_B = dlog_{\alpha,q}(Y_B)$$

The adversary can then calculate the key K in the same manner as user **B** calculates it.

The security of the Diffie-Hellman key exchange lies in the fact that, while it is relatively easy to calculate exponentials modulo a prime, it is very difficult to calculate discrete logarithms. For large primes, the latter task is considered infeasible.

BASIC SECURITY PROCEDURES

Authentication refers to the process where an entity's identity is authenticated, typically by providing evidence that it holds a specific digital identity such as an identifier and the corresponding credentials. Examples of types of credentials are passwords, one-time tokens, digital certificates, and phone numbers (calling/called).

The **authorization** function determines whether a particular entity is authorized to perform a given activity, typically inherited from authentication when logging on to an application or service. Authorization may be determined based on a range of restrictions, for example time-of-day restrictions, or physical location restrictions, or restrictions against multiple access by the same entity or user. Typical authorizations in everyday computer life is for example granting read access to a specific file for authenticated user. Examples of types of service include, but are not limited to: IP address filtering, address assignment, route assignment, Quality of Service/differential services, bandwidth control/traffic management, compulsory tunneling to a specific endpoint, and encryption.

BASIC SECURITY PROCEDURES

Accounting refers to the tracking of resource consumption by users for the purpose of capacity and trend analysis, cost allocation, billing. In addition, it may record events such as authentication and authorization failures, and include auditing functionality, which permits verifying the correctness of procedures carried out based on accounting data. Real-time accounting refers to accounting information that is delivered concurrently with the consumption of the resources. Batch accounting refers to accounting information that is saved until it is delivered at a later time. Typical information that is gathered in accounting is the identity of the user or other entity, the nature of the service delivered, when the service began, and when it ended, and if there is a status to report.

KEY TYPES AND OTHER INFORMATION

Several different types of keys are defined. The keys are identified according to their classification as public, private or symmetric keys, and as to their use. For public and private key agreement keys, their status as static or ephemeral keys is also specified.

1) **Private signature key**: Private signature keys are the private keys of asymmetric (public) key pairs that are used by public key algorithms to generate digital signatures with possible long-term implications. When properly handled, private signature keys can be used to provide authentication, integrity and non-repudiation.
2) **Public signature verification key**: A public signature verification key is the public key of an asymmetric (public) key pair that is used by a public key algorithm to verify digital signatures, either to authenticate a user's identity, to determine the integrity of the data, for non-repudiation, or a combination thereof.
3) **Symmetric authentication key**: Symmetric authentication keys are used with symmetric key algorithms to provide assurance of the integrity and source of messages, communication sessions, or stored data.

KEY TYPES AND OTHER INFORMATION

4) **Private authentication key**: A private authentication key is the private key of an asymmetric (public) key pair that is used with a public key algorithm to provide assurance as to the integrity of information, and the identity of the originating entity or the source of messages, communication sessions, or stored data.

5) **Public authentication key**: A public authentication key is the public key of an asymmetric (public) key pair that is used with a public key algorithm to determine the integrity of information and to authenticate the identity of entities, or the source of messages, communication sessions, or stored data.

6) **Symmetric data encryption key**: These keys are used with symmetric key algorithms to apply confidentiality protection to information.

7) **Symmetric key wrapping key**: Symmetric key wrapping keys are used to encrypt other keys using symmetric key algorithms. Key wrapping keys are also known as key encrypting keys.

8) **Symmetric and asymmetric random number generation keys**: These keys are keys used to generate random numbers.

KEY TYPES AND OTHER INFORMATION

9) **Symmetric master key**: A symmetric master key is used to derive other symmetric keys (e.g., data encryption keys, key wrapping keys, or authentication keys) using symmetric cryptographic methods.

10) **Private key transport key**: Private key transport keys are the private keys of asymmetric (public) key pairs that are used to decrypt keys that have been encrypted with the associated public key using a public key algorithm. Key transport keys are usually used to establish keys (e.g., key wrapping keys, data encryption keys or MAC keys) and, optionally, other keying material (e.g., Initialization Vectors).

11) **Public key transport key**: Public key transport keys are the public keys of asymmetric (public) key pairs that are used to encrypt keys using a public key algorithm. These keys are used to establish keys (e.g., key wrapping keys, data encryption keys or MAC keys) and, optionally, other keying material (e.g., Initialization Vectors).

12) **Symmetric key agreement key**: These symmetric keys are used to establish keys (e.g., key wrapping keys, data encryption keys, or MAC keys) and, optionally, other keying material (e.g., Initialization Vectors) using a symmetric key agreement algorithm.

KEY TYPES AND OTHER INFORMATION

13) **Private static key agreement key**: Private static key agreement keys are the private keys of asymmetric (public) key pairs that are used to establish keys (e.g., key wrapping keys, data encryption keys, or MAC keys) and, optionally, other keying material (e.g., Initialization Vectors).

14) **Public static key agreement key**: Public static key agreement keys are the public keys of asymmetric (public) key pairs that are used to establish keys (e.g., key wrapping keys, data encryption keys, or MAC keys) and, optionally, other keying material (e.g., Initialization Vectors).

15) **Private ephemeral key agreement key**: Private ephemeral key agreement keys are the private keys of asymmetric (public) key pairs that are used only once to establish one or more keys (e.g., key wrapping keys, data encryption keys, or MAC keys) and, optionally, other keying material (e.g., Initialization Vectors).

16) **Public ephemeral key agreement key**: Public ephemeral key agreement keys are the public keys of asymmetric key pairs that are used in a single key establishment transaction 11 to establish one or more keys (e.g., key wrapping keys, data encryption keys, or MAC keys) and, optionally, other keying material (e.g., Initialization Vectors).

KEY TYPES AND OTHER INFORMATION

17) **Symmetric authorization key**: Symmetric authorization keys are used to provide privileges to an entity using a symmetric cryptographic method. The authorization key is known by the entity responsible for monitoring and granting access privileges for authorized entities and by the entity seeking access to resources.

18) **Private authorization key**: A private authorization key is the private key of an asymmetric (public) key pair that is used to provide privileges to an entity.

19) **Public authorization key**: A public authorization key is the public key of an asymmetric (public) key pair that is used to verify privileges for an entity that knows the associated private authorization key.

OTHER CRYPTOGRAPHIC OR RELATED INFORMATION

Other information used in conjunction with cryptographic algorithms and keys also needs to be protected.

1) **Domain Parameters**: Domain parameters are used in conjunction with some public key algorithms to generate key pairs or to create digital signatures or to establish keying material.

2) **Initialization Vectors**: Initialization vectors (IVs) are used by several modes of operation for encryption and decryption and for the computation of MACs using block cipher algorithms.

3) **Shared Secrets**: Shared secrets are generated during a key establishment process. Shared secrets shall not exist outside the cryptographic boundary of the cryptomodule. If a FIPS 140-2 validated cryptomodule is being used, then protection of the shared secrets is provided by the cryptomodule.

4) **RNG seeds**: RNG seeds are used in the generation of deterministic random numbers and shall remain secret (e.g., used to generate keying material that must remain secret or private).

5) **Other public information**: Public information (e.g., a nonce) is often used in the key establishment process.

OTHER CRYPTOGRAPHIC OR RELATED INFORMATION

6) **Intermediate results**: The intermediate results of cryptographic operations using secret information shall be protected. Intermediate results shall not exist outside the cryptographic boundary of the cryptomodule.

7) **Key control information**: Information related to the keying material (e.g., the identifier, purpose, or a counter) shall be protected to ensure that the associated keying material can be correctly used.

8) **Random numbers**: The random numbers created by a random number generator should be protected when retained. When used directly as keying material, the random numbers shall be protected.

9) **Passwords**: A password is used to acquire access to privileges. As such, it is used as an authentication mechanism.

10) **Audit information**: Audit information that contains key management events shall be integrity protected.

KEY USAGE

In general, a single key should be used for only one purpose (e.g., encryption, authentication, key wrapping, random number generation, or digital signatures). There are several reasons for this:

1) The use of the same key for two different cryptographic processes may weaken the security provided by one or both of the processes.
2) Limiting the use of a key limits the damage that could be done if the key is compromised.
3) Some uses of keys interfere with each other. For example, consider a key pair used for both key transport and digital signatures. In this case the private key is used as both a private key transport key to decrypt data encryption keys and a private signature key to apply digital signatures. It may be necessary to retain the private key transport key beyond the cryptoperiod of the corresponding public key transport key in order to decrypt the data encryption keys needed to access encrypted data. On the other hand, the private signature key should be destroyed at the expiration of its cryptoperiod to prevent its compromise. In this example, the longevity requirements for the private key transport key and the private digital signature key contradict each other.

This principle does not preclude using a single key in cases where the same process can provide multiple services.

CRYPTOPERIODS

A **cryptoperiod** is the time span during which a specific key is authorized for use by legitimate entities, or the keys for a given system will remain in effect. A suitably defined cryptoperiod:

1) limits the amount of information protected by a given key that is available for cryptanalysis,
2) limits the amount of exposure if a single key is compromised,
3) limits the use of a particular algorithm to its estimated effective lifetime,
4) limits the time available for attempts to penetrate physical, procedural, and logical access mechanisms that protect a key from unauthorized disclosure
5) limits the period within which information may be compromised by inadvertent disclosure of keying material to unauthorized entities, and
6) limits the time available for computationally intensive cryptanalytic attacks (in applications where long-term key protection is not required).

Sometimes cryptoperiods are defined by an arbitrary time period or maximum amount of data protected by the key. However, trade-offs associated with the determination of cryptoperiods involve the risk and consequences of exposure, which should be carefully considered when selecting the cryptoperiod.

RISK FACTORS AFFECTING CRYPTOPERIODS

Among the factors affecting the risk of exposure are:

1) the strength of the cryptographic mechanisms (e.g., the algorithm, key length, block size, and mode of operation),
2) the embodiment of the mechanisms,
3) the operating environment (e.g., secure limited access facility, open office environment, or publicly accessible terminal),
4) the volume of information flow or the number of transactions,
5) the security life of the data,
6) the security function (e.g., data encryption, digital signature, key production or derivation),
7) the re-keying method (e.g., keyboard entry, re-keying using a key loading device where humans have no direct access to key information, remote re-keying within a PKI),
8) the key update or key derivation process,
9) the number of nodes in a network that share a common key,
10) the number of copies of a key and the distribution of those copies, and
11) the threat to the information (e.g., who the information is protected from, and what are their perceived technical capabilities and financial resources to mount an attack).

RISK FACTORS AFFECTING CRYPTOPERIODS

In general short cryptoperiods enhance security. For example, some cryptographic algorithms might be less vulnerable to cryptanalysis if the adversary has only a limited amount of information encrypted under a single key. On the other hand, where manual key distribution methods are subject to human error and frailty, more frequent key changes might actually increase the risk of exposure. In these cases, especially when very strong cryptography is employed, it may be more prudent to have fewer, well-controlled manual key distributions rather than more frequent, poorly controlled manual key distributions.

In general, where strong cryptography is employed, physical, procedural, and logical access protection considerations often have more impact on cryptoperiod selection than do algorithm and key size factors. In the case of Approved algorithms, modes of operation, and key sizes, adversaries may be able to access keys through penetration or subversion of a system with less expenditure of time and resources than would be required to mount and execute a cryptographic attack.

RISK FACTORS AFFECTING CRYPTOPERIODS

Key Type	Cryptoperiod	
	Originator Usage Period (OUP)	Recipient Usage Period
1. Private Signature Key	1-3 years	
2. Public Signature Key	Several years (depends on key size)	
3. Symmetric Authentication Key	≤ 2 years	≤ OUP + 3 years
4. Private Authentication Key	1-2 years	
5. Public Authentication Key	1-2 years	
6. Symmetric Data Encryption Keys	≤ 2 years	≤ OUP + 3 years
7. Symmetric Key Wrapping Key	≤ 2 years	≤ OUP + 3 years
8. Symmetric and asymmetric RNG Keys	Upon reseeding	
9. Symmetric Master Key	About 1 year	
10. Private Key Transport Key	≤ 2 years	
11. Public Key Transport Key	1-2 years	
12. Symmetric Key Agreement Key	1-2 years	
13. Private Static Key Agreement Key	1-2 years	
14. Public Static Key Agreement Key	1-2 years	
15. Private Ephemeral Key Agreement Key	One key agreement transaction	
16. Public Ephemeral Key Agreement Key	One key agreement transaction	
17. Symmetric Authorization Key	≤ 2 years	
18. Private Authorization Key	≤ 2 years	
19. Public Authorization Key	≤ 2 years	

RECOMMENDATIONS FOR OTHER KEYING MATERIAL

Other keying material does not have well-established cryptoperiods, per se. The following recommendations are offered regarding the disposition of this other keying material:

1) Domain parameters remain in effect until changed.

2) An IV is associated with the information that it helps to protect, and is needed until the information and its protection are no longer needed.

3) Shared secrets shall be destroyed as soon as they are no longer needed to derive keying material.

4) RNG seeds should be destroyed immediately after use.

5) Other public information should not be retained longer than needed for cryptographic processing.

6) Intermediate results shall be destroyed immediately after use.

RECOMMENDATIONS FOR OTHER KEYING MATERIAL

Bits of security	Symmetric key algorithms	FFC (e.g., DSA, D-H)	IFC (e.g., RSA)	ECC (e.g., ECDSA)

80	2TDEA	L = 1024
		N = 160
			k = 1024	f = 160-223

112	3TDEA	L = 2048
		N = 224
			k = 2048	f = 224-255

128	AES-128	L = 3072
		N = 256
			k = 3072	f = 256-383

192	AES-192	L = 7680
		N = 384
			k = 7680	f = 384-511

256	AES-256	L = 15360
		N = 512
			k = 15360	f = 512+

FFC – finite field cryptography

IFC – integer factorization cryptography

ECC – elliptic curve cryptography

THANKS FOR ATTENTION