Computer security

Protocols, cryptography 1

Literature

- https://www.cs.vsb.cz/ochodkova/courses/kpb/cryptography-and-network-security_-principles-and-practice-7th-global-edition.pdf
- Chapters 1.1, 1.6, 3.1

Introduction (1)

- Ensuring security requirements is not easy. The mechanisms used to meet them can be, and often are, quite complex.

- When developing a particular security mechanism or algorithm, one must always consider the potential attacks on them. In many cases, however, attacks exploit unexpected weaknesses in the mechanism.

- After designing the various security mechanisms, it is necessary to decide where to use them. This applies both in terms of physical location and in a logical sense.

- **Security mechanisms usually involve more than e.g. a specific algorithm, but also require that participants have some secret information (e.g. an encryption key) → need to address the creation, distribution and protection of this secret information.**

- The struggle is between the attacker trying to find weaknesses in the system and the designer or administrator trying to patch them:

  - An attacker only needs to find a single weakness in the system,

  - while the designer (administrator) must find and remove (treat) all of them.

Introduction (2)

- There are no security design and implementation techniques that systematically eliminate all security vulnerabilities and prevent all unauthorized activities. Therefore, there is a set of broadly agreed-upon principles that guide the development of security mechanisms.
    - **Economy of mechanism** - security measures included in both hardware and software should be as simple and minimal as possible.
    - **Fail-safe defaults** - the default situation is restricted access, and the protection scheme specifies the conditions under which access is allowed.
    - **Complete mediation** - each access must be checked within the access control mechanism. Systems should not rely on cached access decisions.
    - **Open design** means that the design of the security mechanism should be open. For example, the sub-algorithms can then be reviewed by many experts, and therefore can be highly trusted by users.
    - **Separation of privilege-** generally refers to the separation of users and processes based on different levels of trust, needs, and permission requirements.
    - **Least privilege** - each process and each user of the system should operate with the smallest set of privileges necessary to perform the task. A good example of using this principle is role-based access control. Any access control system should allow each subject only the privileges for which they are authorized.

Introduction (3)

- **Least common mechanism** - the design should minimize functions shared by different users and provide mutual security.
- **Psychological acceptability** - security mechanisms should not unduly interfere with users' work, be intrusive or burdensome.
- **Isolation** - publicly accessible systems should be isolated from critical resources (data, processes, etc.). In addition, processes and files of individual users should be isolated from each other, except where explicitly required. Finally, security mechanisms should be isolated in the sense of preventing access to these mechanisms.
- **Encapsulation** - Encapsulation is a specific form of isolation in OOP where data is only accessible from the outside through methods. Furthermore, encapsulation is encountered in computer networks.
- **Modularity** - in the context of security, it refers to both the development of security functions as separate protected modules and the use of a modular architecture for mechanism design and implementation.
- **Layering** - Layering refers to the use of multiple overlapping protective approaches focused on people, technology and operational aspects of information systems.
- **Least astonishment** - a program or user interface should always respond in a way that is least likely to surprise the user.

Protocols (1)

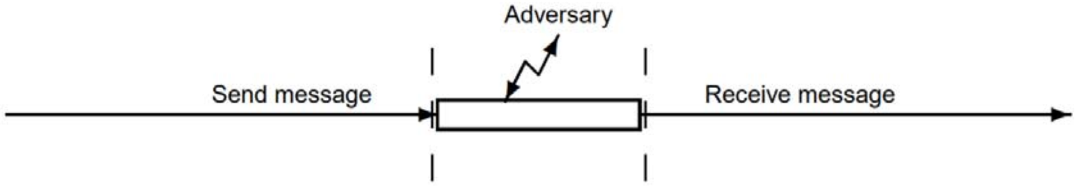
- A security protocol is essentially a communication protocol (an agreed sequence of actions performed by two or more communicating entities to achieve some mutually desired goal) that uses (cryptographic) techniques to enable the communicating entities to achieve a security goal.
- Cryptographic protocols typically use one or more cryptographic primitives and/or schemes.
    - An example would be transferring Bob's credit card number to Alice's e-commerce website.
    - Such a protocol may include a digital signature scheme (so Bob knows he is communicating with Alice) and a form of encryption (to ensure that Bob's credit card information is not intercepted in transit).
- Encryption algorithms, signing algorithms and hashing functions are the basis of security protocols.

Protocols (2)

- General protocols
  - Key agreement - The key agreement protocol allows two parties to agree on a shared secret key. Authentication is based on a public/private key pair. An important distinction is between key transfer, where one party generates a key and sends it to the other, and key agreement, where neither party has complete control over the key generation process.
  - Identification and Authentication Protocols - protocols for user identification (1:N) and authentication (1:1)
  - ...
- Specific protocols
  - TLS was designed to secure communication between the browser and the website
  - SSH 10 - used, for example, to provide a secure channel between two computers on a network for applications such as file transfer
  - IPSec - (Internet Protocol SECurity) is a complex set of protocols at the network layer that offers tunneling, encryption and authentication
  - Kerberos - a protocol that allows a client to authenticate against multiple services. Kerberos provides a centralized authentication server whose function is to authenticate users against servers and servers against users
  - ...
- Application-specific protocols
  - WEP/WPA - WEP/WPA protocols are used to protect communications on wireless networks
  - UMTS/LTE - The GSM, UMTS and LTE protocols are designed to secure communication between a mobile phone and an operator's base station
  - Bluetooth - Bluetooth is a technology for secure data exchange over short distances
  - ZigBee is a radio communication standard operating mainly at lower power and range than Bluetooth
  - ...

Why cryptography?

- Why cryptography? Cryptography is the basis of many technological solutions to computer security problems.
- Cryptography deals with algorithms that can be used to:
  - **confidentiality of the messages** (their content, not their existence),
  - **authentication** = traceability of the origin of the message, secure identification of the subject who
    - created the information,
    - accepts it,
    - operates it,
  - **Integrity checking** - information can only be modified/generated by an authorized entity,
  - to ensure the **nonrepuriation**
    - income,
    - delivery,
    - the origin of sensitive information.



Cryptology

- **Cryptology** is the science of information hiding, standing at the boundary between mathematics and computer science, with a greater overlap into mathematics, especially in the field of number theory and algebra. Cryptology includes cryptography and cryptanalysis
  - **Cryptography** is the "art" and science of translating information into a form in which the content of the information is hidden (even if this illegible information is disclosed to a third party). In the past, it was the science of encrypting data using mathematical methods, but today cryptography also involves signatures, hashing functions, etc.
  - **Cryptanalysis** is the art and science of breaking hidden information (ciphers or keys), it deals with the resistance of a cryptographic system.
    - The goal of cryptanalysis is to attack a cryptographic system, the goal is to uncover the message M without knowing the encryption key K (and thus uncovering it), and
    - Usually a mathematical analysis of an algorithm that tries to discover its weaknesses and exploit them.
    - Currently, it is mainly about assessing the security and efficiency of newly designed algorithms.

Cryptographic system

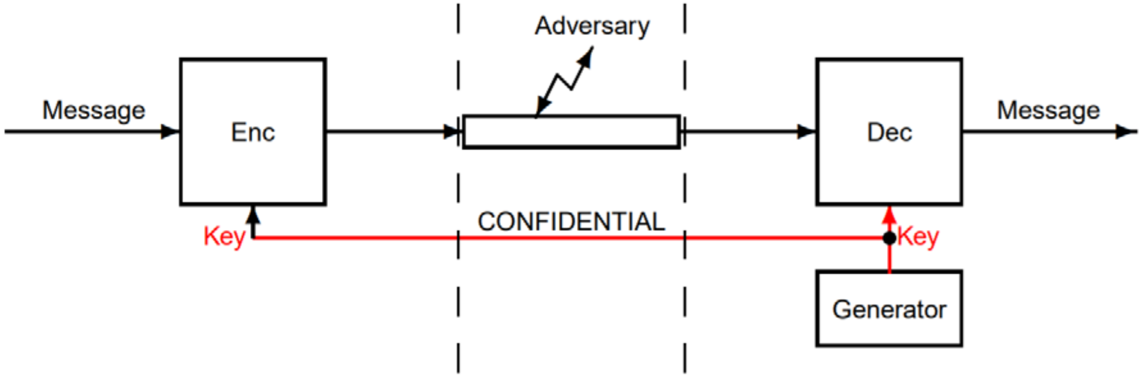
- **A cryptographic system** is a quintuple {M, C, K, E, D}, where:
- The symbol M denotes the so-called plaintext space, which is a finite set of elements called **plaintext** (message)
- The symbol C denotes the so-called ciphertext space. C is a finite set of elements called **ciphertext**
- K is a finite set of possible keys, the so-called key space, whose element is the **key**
- E is the set of encryption or signing functions (algorithms, rules). For example, **encryption** (enciphering) is the process by which we obtain the ciphertext from the plaintext (we will also use E for signing)
- D is the set of decryption functions or signature verification functions (algorithms, rules). For example, we call **decryption** (decciphering) the process by which we obtain the plaintext from the ciphertext (we will also use D for signature verification)

Cryptographic system

- It applies that:
  - For $\forall k \in K$ there is an encryption algorithm $e_k \in E$ and its corresponding decryption algorithm $d_k \in D$.
  - Each $e_k : m \rightarrow c$ and $d_k : c \rightarrow m,\ k \in K$ are functions for which the **correctness condition** applies
    - ***$d_k (e_k (m)) = m$, for $\forall m \in M,\ \forall k \in K$.***
- The encryption key *k* need not be the same as the decryption key *k'*, then we speak of a key pair ($k \neq k'$).
- Cryptographic algorithms
  - Conventional:
    - Symmetric encryption algorithms
    - Hash function
    - Message authentication codes
  - Asymmetric
    - Asymmetric algorithms
    - Digital signature

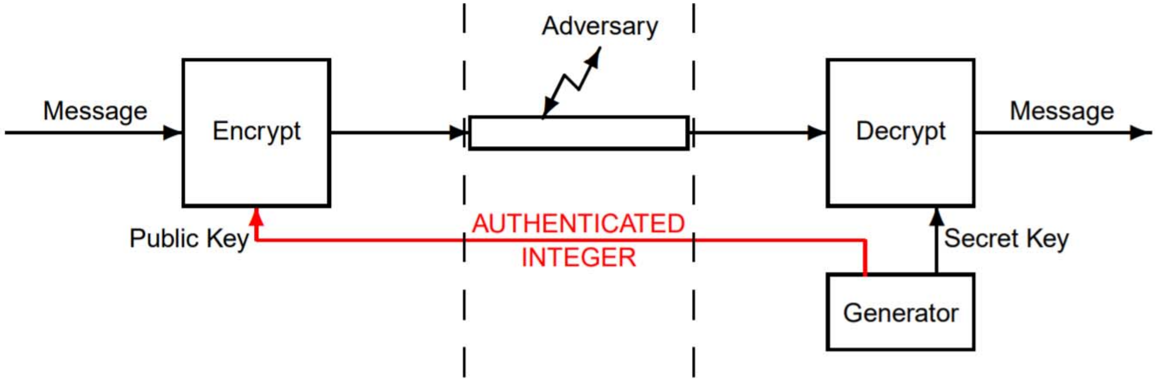
Symmetric algorithms

- **Symmetric cryptography** (secret key cryptography) uses the same key, *k = k'*, for encryption and decryption. Symmetric (cryptographic) algorithms are divided into:
  - **block ciphers** (algorithms) that operate on an *n-bit* block of data (plaintext), typically n≥128b, (AES, 3DES, Twofish, ...)
  - **stream ciphers** (algorithms) - the plaintext is processed bit by bit or byte by byte (RC4, GSM-A5/1, Bluetooth-E0, ...).



Asymmetric algorithms

- **Asymmetric cryptography** (public key cryptography) uses a key pair $k \neq k'$, called
  - private key and
  - public key.
  - Each sender and recipient owns his/her key pair and its use varies depending on whether you want to encrypt or sign the message.



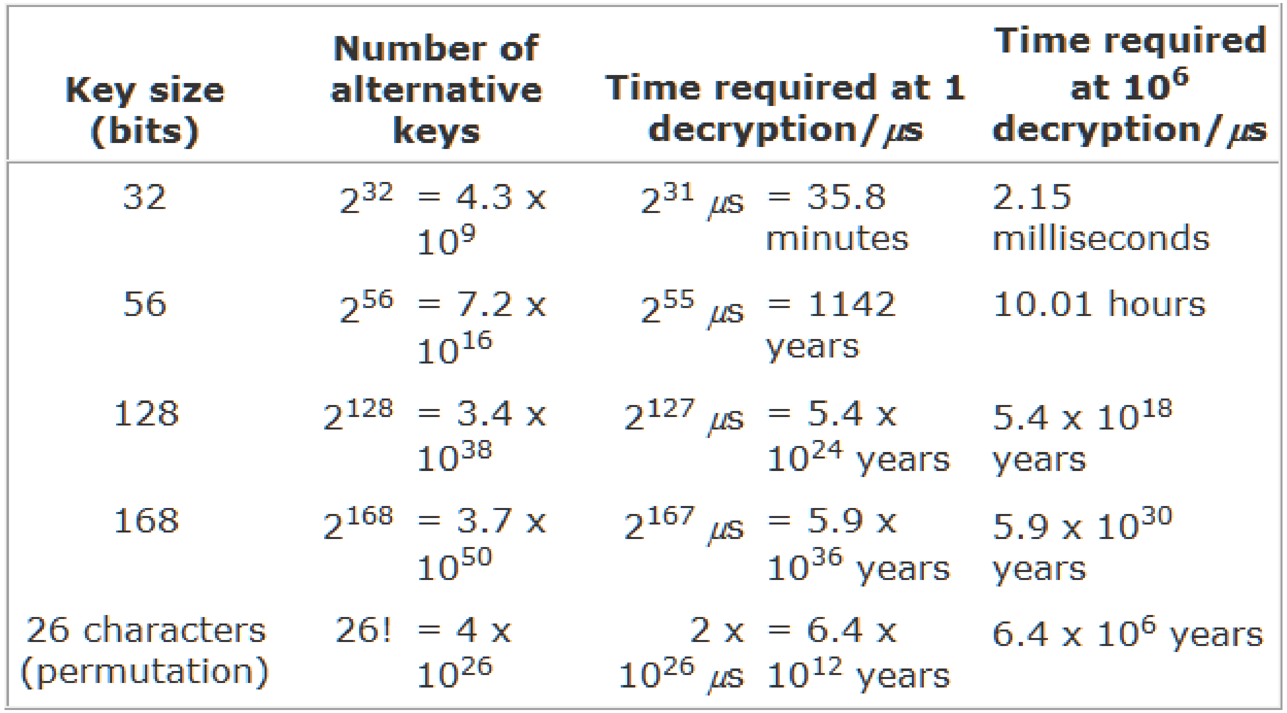
Kerckhoffs principle

- In 1883, Auguste Kerckhoffs formulated the first principles of cryptographic engineering: **'The security of an encryption system must not depend on the secrecy of the algorithm, but on the secrecy of the key'.**
- Modern cryptography generally uses key-dependent algorithms (the so-called ***Kerckhoffs principle*** is applied). That is, cryptographic algorithms **are publicly published** and knowledge of them must not help the cryptanalyst in his attempt to overcome secrecy - it must be fully key-dependent (which the cryptanalyst does not know).
- The security of any algorithm that uses a key lies in the number of possible keys.

Attack by brute force

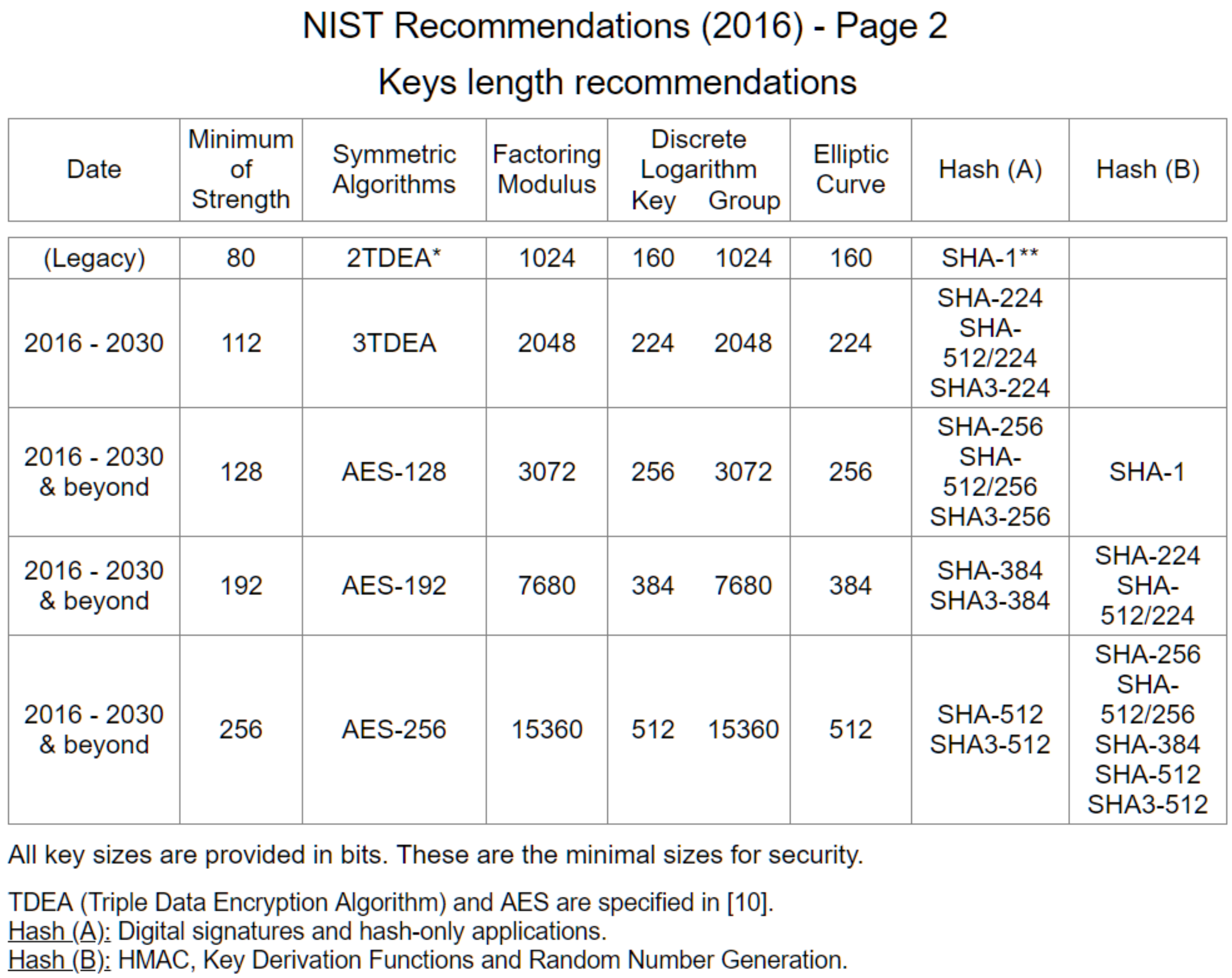
- **Brute Force Attack** (exhaustive search) is an attack where the attacker tries all possible keys one by one. He tries to see if applying them to a known ciphertext C yields a meaningful message M (even if we know the original M for this C, we wait for a key K to generate the corresponding M).
  - The size of the key and therefore the size of the key space is important. Every cryptographic system is breakable by this attack, but for how long?
    - This is just a matter of money (the attack is usually distributed) and time (the duration of the attack usually depends on the length of the key used).
    - As the length of the key increases, the time required to attack increases exponentially and quickly exceeds the physically possible limits.
  - Brute-force attack can be successfully used against algorithms with weak (short) keys.
  - All new algorithms must be resistant to brute force attacks!

Attack by brute force



Key size (bits)	Number of alternative keys	Time required at 1 decryption/μs	Time required at 10^6 decryption/μs
32	$2^{32} = 4.3 \times 10^9$	2^{31} μs = 35.8 minutes	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	2^{55} μs = 1142 years	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	2^{127} μs = 5.4×10^{24} years	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	2^{167} μs = 5.9×10^{36} years	5.9×10^{30} years
26 characters (permutation)	$26! = 4 \times 10^{26}$	2×10^{26} μs = 6.4×10^{12} years	6.4×10^6 years

Recommended key sizes and other parameters



NIST Recommendations (2016) - Page 2

Keys length recommendations

Date	Minimum of Strength	Symmetric Algorithms	Factoring Modulus	Discrete Logarithm		Elliptic Curve	Hash (A)	Hash (B)
				Key	Group			
(Legacy)	80	2TDEA*	1024	160	1024	160	SHA-1**	
2016 - 2030	112	3TDEA	2048	224	2048	224	SHA-224 SHA-512/224 SHA3-224	
2016 - 2030 & beyond	128	AES-128	3072	256	3072	256	SHA-256 SHA-512/256 SHA3-256	SHA-1
2016 - 2030 & beyond	192	AES-192	7680	384	7680	384	SHA-384 SHA3-384	SHA-224 SHA-512/224
2016 - 2030 & beyond	256	AES-256	15360	512	15360	512	SHA-512 SHA3-512	SHA-256 SHA-512/256 SHA-384 SHA-512 SHA3-512

All key sizes are provided in bits. These are the minimal sizes for security.

TDEA (Triple Data Encryption Algorithm) and AES are specified in [10].
Hash (A): Digital signatures and hash-only applications.
Hash (B): HMAC, Key Derivation Functions and Random Number Generation.