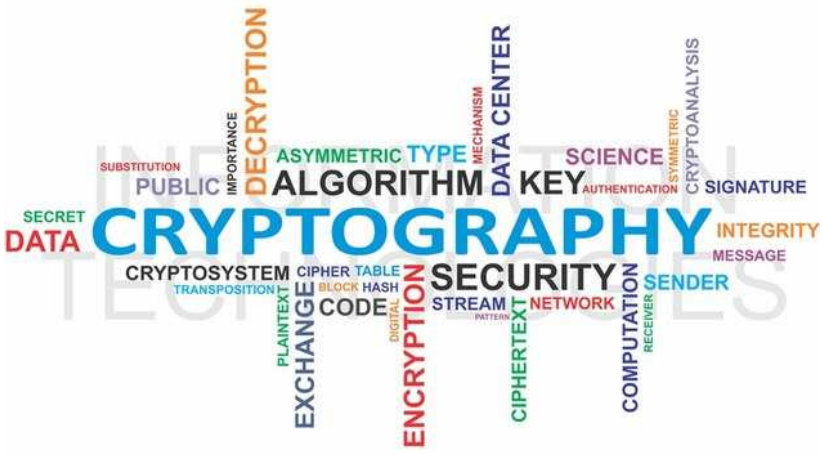
IT biztonság
2020/2021 tanév ősz



2020.10.04. ELTE IT BIztonság Speci

Tartalom

1. Alapfogalmak
2. Titkosítás
   - Szimmetrikus kulcsú sémák
   - Nyilvános kulcsú sémák
3. Hash függvények
   - Message Authentication Codes (MAC)
   - Digitális aláírás

3/71

Mi az a kriptográfia?

- **Cryptography** is the study of mathematical techniques related to aspects of information security such as confidentality, data integrity, entity authentication, and data origin authentication
- **Cryptanalysis** is the study of mathematical techniques for attempting to defeat cryptographic techniques, and, more generally, information security services
- **Cryptographic primitives** are tools used to provide information security
- **Cryptographic protocol** is a distributed algorithm defined by a sequence of steps to achieve a specific security objective
- **Cryptology** is the study of cryptography and cryptanalysis

2020.10.04. ELTE IT Biztonság Speci

A biztonságot érintő kriptográfia alapú szolgáltatások

- Hitelesítés (authentication)
  - Az eljárás, amely során ellenőrzésre kerül, hogy egy adott azonosító valóban a megfelelő entitáshoz tartozik-e. A hitelesítés kriptográfiai eljárások segítségével történik.
    - Társentitás hitelesítés
    - Adateredet hitelesítés
- Hozzáférés-ellenőrzés (access control)
  - Az eljárás, amely megakadályozza, hogy illetéktelenek hozzáférjenek az erőforrásokhoz.
- Titkosság/Bizalmasság kezelése (confidentiality)
  - Illetéktelenektől való adatvédelem
- Sértetlenség kezelése (integritás, integrity)
  - Az adatot csak arra jogosultak hozhatják létre, illetve módosíthatják.
- Letagadhatatlanság kezelése (non-repudiation)
  - Forrásigazolással, vagy kézbesítés-igazolással (esetleg együtt) történhet

5/71

1. Alapfogalmak

A támadási vektor

- A **támadás** egy szándékos kísérlet a védendő rendszer kompromittálására; rendszerint a rendszerterv, implementáció, működés vagy menedzsment hibáit próbálja kihasználni.
- Cél: a támadások megakadályozása, ha pedig ez nem teljesíthető, akkor a támadások felismerése és a támadás előtti állapot visszaállítása.
- Egy támadás lehet
  - Passzív
    - A támadó csak megfigyelőként van jelen, esetleg archiválja az adatokat, de nem végez módosításokat a kommunikációban. A támadó tehát információkat próbál nyerni a rendszerről, miközben nincs hatással a rendszer erőforrásaira.
    - Példák: üzenetek lehallgatása, forgalomelemzés.
    - Nehéz detektálni, a megakadályozásra kell törekedni
  - Aktív
    - A támadó saját maga által létrehozott, vagy az út során elkapott és módosított adatok juttat el az áldozathoz. A támadó meg próbálja változtatni a rendszer bizonyos adatait, hatással van a rendszer működésére.
    - Példák: hamisítás (spoofing: e-mail, IP, ARP), visszajátszásos támadás (replay attack), módosítás (helyettesítés, beszúrás, törlés), denial of service.
    - Nehéz megakadályozni, a detektálásra kell törekedni.
- A **támadási vektor** egy olyan módszer vagy útvonal, amelyet a támadó használ a célrendszer eléréséhez vagy behatoláshoz.

6/71

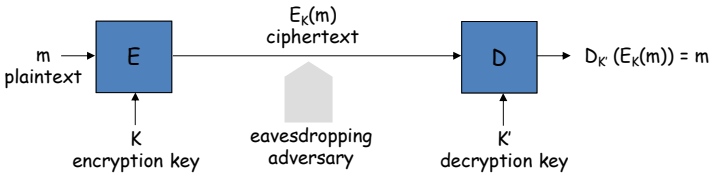
1. Alapfogalmak

A biztonságot érintő eljárások

- Titkosítás (encryption)
  - Szimmetrikus
  - Aszimmetrikus (nyilvános kulcsú)
- Digitális aláírás (digital signature)
- Hozzáférés-ellenőrzés sémák (access control schemes)
  - Access control lists, security labels, ...
- Adatsértetlenséget megőrző eljárások (data integrity mechanisms)
  - Message authentication codes (MAC), sorszámozás (sequence numbering), időbélyegző (time stamping)
- Hitelesítő protokollok (authentication protocols)
  - Passwords, cryptographic challenge-response protocols, biometrics
- Hamis forgalom (traffic padding)
  - A kommunikáció meghamisított eseteinek, hamis adatelemeknek és/vagy adatelemekben hamis adatoknak az előállítása.

Tartalom

A klasszikus modell



- A lehallgató (támadó, adversary) célja:
  - Szisztematikusan felfedni a nyílt szöveget a titkos szövegből
  - A feloldó kulcs (decryption key) megfejtése
- **Kerckhoff-alapelv:**
  - A rendszer biztonsága nem épülhet arra a feltevésre, hogy a támadó nem ismeri E és D működését (security by obscurity)
- Bruce Schneier:
  - A rendszer biztonságát a feloldó kulcs védelmére kell építeni

Fogalmak

- A titkosítás hagyományosan az illetéktelen hozzáférés ellen szolgált
- Fogalmak:
  - **Nyílt szöveg** (plaintext, cleartext) – a titkositatlan információ, nem csak az arra feljogosított számára értelmezhető
  - **Titkos szöveg** (ciphertext) – a titkosított információ, ebben a formájában értelmezhetetlen
  - **Kulcs** – olyan információ, amely segítségével a titkosítási műveletek elvégezhetők
  - **Algoritmus** (módszer) - olyan eljárás, amely a nyílt és a titkos szöveg közötti konverziót (titkosítás – encryption; titkosítás feloldása - decryption) végzi a kulcs segítségével

Fogalmak

- **A csatorna** az információ továbbításának egy eszköze, amelyben egy entitás üzenetet továbbít egy másiknak
- **A fizikailag biztonságos csatorna** az, amely nem fizikailag hozzáférhető a támadó számára
- **A nem biztonságos csatorna** az, amelyben az érdekelteken kívül más felek is képesek az információt olvasni, átrendezni, törölni, vagy hozzáilleszteni
- **A biztonságos csatorna az**, amelyben a támadó nem képes az információt átrendezni, törölni, beilleszteni vagy olvasni

Támadási vektorok

- **Csak titkos szöveg alapú támadás** (ciphertext-only attack)
  - the adversary can only observe ciphertexts produced by the same encryption key
- **Ismert nyílt szöveg támadás** (known-plaintext attack)
  - the adversary can obtain corresponding plaintext-ciphertext pairs produced with the same encryption key
- **Választható nyílt szöveg támadás** (adaptive chosen-plaintext attack)
  - the adversary can choose plaintexts and obtain the corresponding ciphertexts
- **Választható titkos szöveg támadás** (adaptive chosen-ciphertext attack)
  - the adversary can choose ciphertexts and obtain the corresponding plaintexts
- **Rokon kulcs alapú támadás** (related-key attack)
  - the adversary can obtain ciphertexts, or plaintext-ciphertext pairs that are produced with different encryption keys that are related in a known way to a specific encryption key

A titkosítási sémák biztonsága

- Egy titkosítási séma biztonságos, ha adott támadási modell esetén a támadó tetszőleges informatikai erőforrás használata esetén sem képes a feloldó kulcsot meghatározni.
- A gyakorlatban sokféle séma létezik, sokuk biztonsága nem bizonyított
  - Hatékonyak, ellenállnak az ismert támadásoknak
- Néhány séma bizonyítottan biztonságos, de ezek gyakran nem hatékonyak

A titkosítási sémák osztályozása

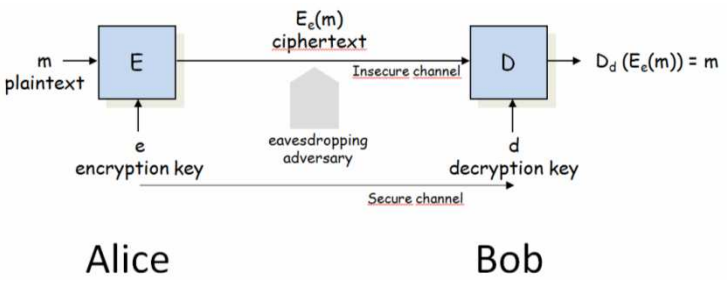
- Szimmetrikus kulcsú titkosítás (symmetric-key encryption)
  - K' könnyen számolható K ismeretében (és vice versa)
  - gyakran K' = K
  - Két fő típus
    - Blokktitkosítás (block ciphers) – karakterek blokkján dolgozik
    - Folyamtitkosítás (stream ciphers) – a nyílt szöveg különböző karakterein dolgozik
- Aszimmetrikus (nyilvános) kulcsú titkosítás (asymmetric-key encryption)
  - Nehéz (computationally infeasible) K' kiszámítása K-ból
  - K nyilvánossá tehető (→ public-key cryptography)

Tartalom

1. Alapfogalmak
2. Titkosítás
   - Szimmetrikus kulcsú sémák
   - Nyilvános kulcsú sémák
3. Hash függvények
   - Message Authentication Codes (MAC)
   - Digitális aláírás

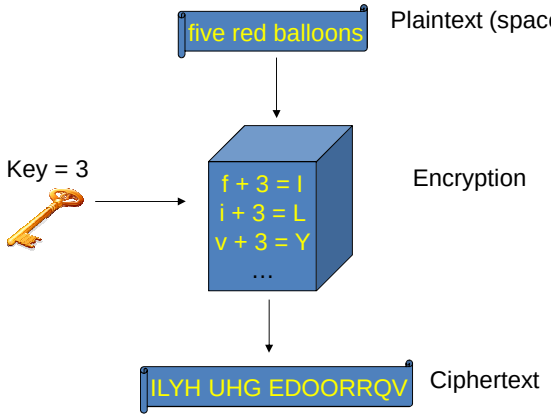
15/71

Szimmetrikus kulcsú sémák



- Mi történik, ha NINCS biztonságos csatorna?
- Kulcsmegosztás

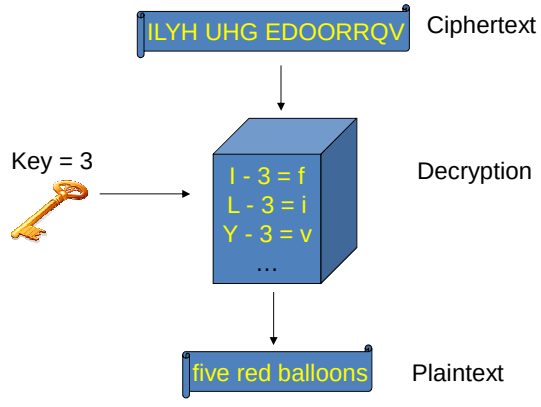
Az eltolásos titkosítás (Shift Cipher, Caesar)



17/71

2. Titkosítás

Az eltolásos titkosítás dekódolása



18/71

2. Titkosítás

Az eltolásos titkosítás problémája

- Túl kicsi a kulcstér!!
- If we shift a letter 26 times, we get the same letter back
  - A shift of 27 is the same as a shift of 1, etc.
  - So we only have 25 keys (1 to 25)
- Eve just tries every key until she finds the right one

A helyettesítéses titkosítás (Vigenere)

- Rather than having a fixed shift, change every plaintext letter to an arbitrary ciphertext letter

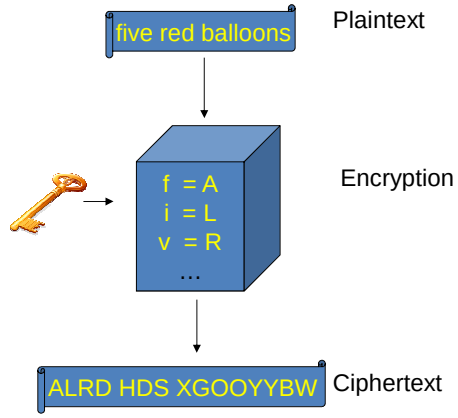
Plaintext	Ciphertext
a	G
b	X
c	N
d	S
e	D
...	...
z	Q

A helyettesítéses titkosítás (folyt.)

Key =

a	G
b	X
c	N
d	S
e	D
f	A
g	F
h	V
i	L
j	M
k	C
l	O
m	E

n	B
o	Y
p	Z
q	P
r	H
s	W
t	I
u	J
v	R
w	U
x	K
y	T
z	Q



A helyettesítéses titkosítás (folyt.)

- To decrypt we just look up the ciphertext letter in the table and then write down the matching plaintext letter
- How many keys do we have now?
  - A key is just a permutation of the letters of the alphabet
  - There are 26! permutations
    - 403291461126605635584000000

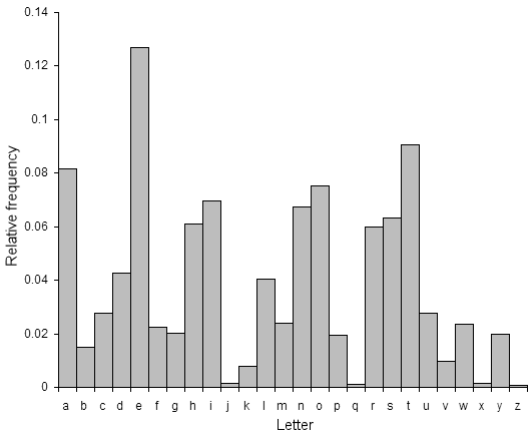
Támadás: gyakoriságvizsgálat

Letter based encriptions:

- In any language certain letters are used more often than others
- If we look at a ciphertext, certain ciphertext letters are going to appear more often than others
- It would be a good guess that the letters that occur most often in the ciphertext are actually the most common letters

Betűgyakoriság az angol abc-ben

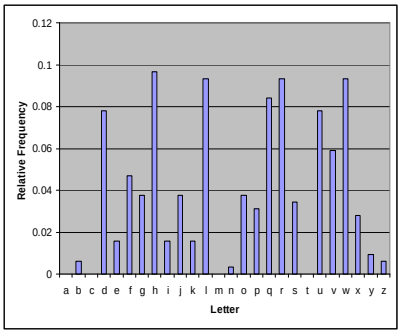
- This is the letter frequency for English
- The most common letter is ‘e’ by a large margin, followed by ‘t’, ‘a’, and ‘o’
- ‘J’, ‘q’, ‘x’, and ‘z’ hardly occur at all



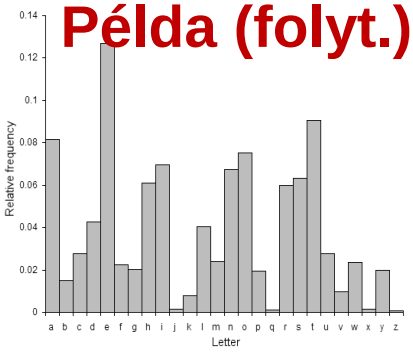
Példa

- Suppose this is our ciphertext
  - dq lqwurgxfwlrq wr frpsxwlqj surylglqj d eurdg vxuyhb ri wkh glvflsolqh dqg dq lqwurgxfwlrq wr surjudpplqj. vxuyhb wrslfv zloo eh fkrvhq iurp: ruljlqv ri frpsxwhuv, gdwd uhsuhvhqwdwlrq dqg vwrudjh, errohdq dojheud, gljlwdo orjlf jdwhv, frpsxwhu dufklwhfwxuh, dvvhpeohuv dqg frpslohuv, rshudwlqj vbvwhpv, qhwzrunv dqg wkh lqwhuqhw, wkhrulhv ri frpsxwdwlrq, dqg duwlilfldo lqwhooljhqfh.

Példa (folyt.)



Ciphertext distribution



English distribution

- In our ciphertext we have one letter that occurs more often than any other (h),
- There is a good chance that h corresponds to e, and d, l, q, r, u, and w correspond to the next 6 most common English letters

Példa (folyt.)

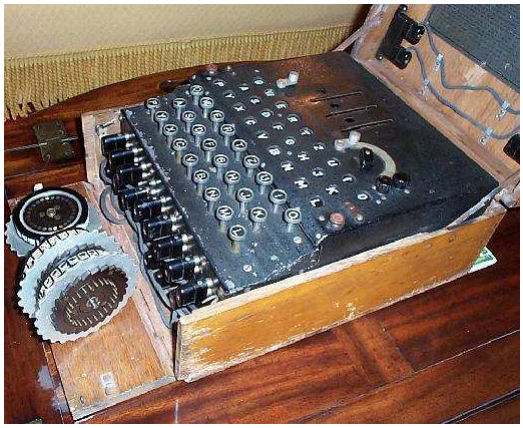
- If we replace ‘e’ with ‘h’ and the 6 next most common letters with their matches, the ciphertext becomes
  - an intro???tion to ?o?p?tin? pro?i?in? a ?roa? ??r?e? o? t?e ?i??ip?ine an? an intro???tion to pro?ra??in?. ??r?e? topi?? ?i?? ?e ??o?en ?ro?: ori?in? o? ?o?p?ter?, ?ata repre?entation an? ?tora?e, ?oo?ean a??e?ra, ?i?ita? ?o?i? ?ate?, ?o?p?ter ar??ite?t?re, a??e???er? an? ?o?pi?er?, operatin? ???te??, net?or?? an? t?e internet, t?eorie? o? ?o?p?tation, an? arti?i?ia? inte??i?en?e.

Klasszikus és modern kriptográfia

- Classical cryptography
  - Encryption/decryption done by hand
- Modern cryptography
  - Computers to encrypt and decrypt
  - Same principles, but automation allows ciphers to become much more complex

Az Enigma

- German encryption and decryption machine used in WW-II
- Essentially a complex, automated substitution cipher

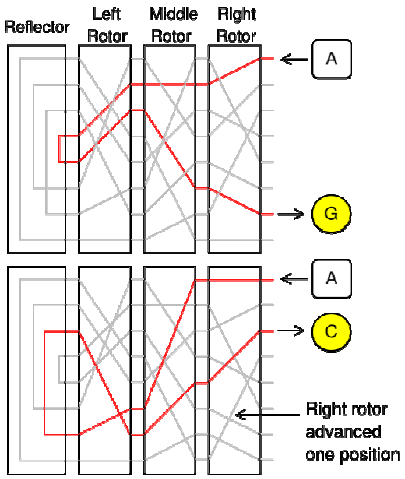


29/71

2. Titkosítás

Az Enigma elve

- Rotors have different wiring connecting input to output
- Rotors move after each keypress
- The key is the initial position of the three rotors
- Britain consistently broke German codes throughout the war

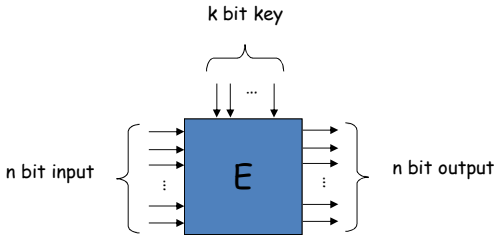


30/71

2. Titkosítás

Blokk titkosítás (block ciphers)

Egy *n* bites blokk titkosító egy olyan E: $\{0, 1\}^n$ x $\{0, 1\}^k \rightarrow \{0, 1\}^n$ függvény, amelyre minden $K \in \{0, 1\}^k$ esetén $E(x, K) = E_K(x)$ injektív



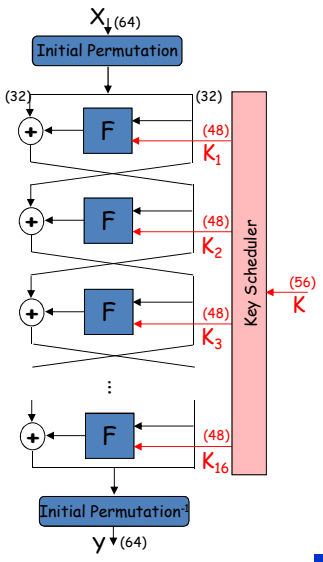
Blokk titkosítás tervezési kritériumai

- Teljesség (completeness)
  - Each bit of the output block should depend on each bit of the input block and on each bit of the key
- Lavinahatás (avalanche effect)
  - Changing one bit in the input block should change approximately half of the bits in the output block
  - Similarly, changing one key bit should result in the change of approximately half of the bits in the output block
- Statisztikai függetlenség (statistical independence)
  - Input and output should appear to be statistically independent

Hogyan teljesíthetők a tervezési kritériumok?

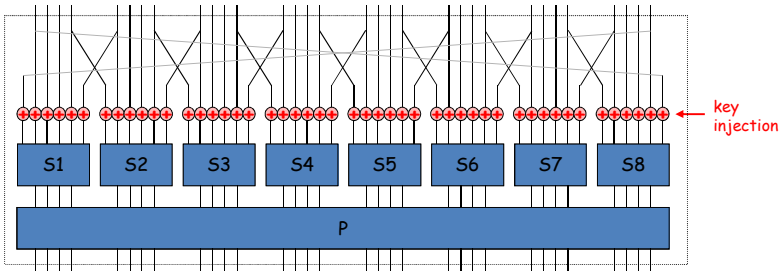
- Összetett titkosítás egyszerű műveletek kompozíciójával is megvalósítható. Ezek leginkább kiegészítik a védekezési mechanizmusokat.
- Példa egyszerű műveletekre:
  - elementary arithmetic operations
  - logical operations (e.g., XOR)
  - modular multiplication
  - transpositions
  - substitutions
  - ...
- Két vagy több transzformáció oly módon is kombinálható, hogy az eredményül kapott titkosítás biztonságosabb, mint komponensei külön-külön.

Példa: DES (Data Encryption Standard)



- Input size: 64 bits
- Output size: 64 bits
- Key size: 56 bits
- 16 rounds
- Feistel structure

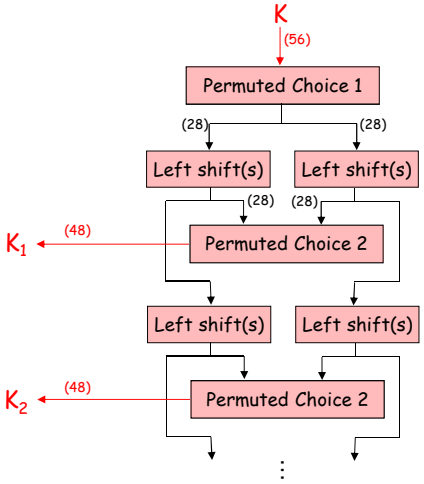
Példa: DES round function F



Si – Substitution box (S-box)

P – Permutation box (P-box)

Példa: DES key scheduler



Each key bit is used in around 14 out of 16 rounds

Blokktitkosító üzemmódok (általában 64 bites blokkok)

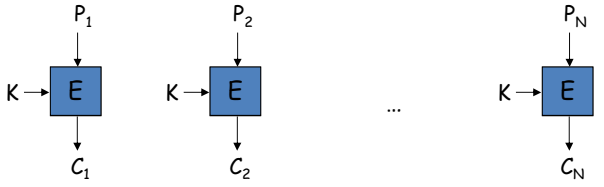
- ECB – Electronic Codebook (elektronikus kódkönyv)
  - Used to encipher a single plaintext block (e.g., a DES key)
- CBC – Cipher Block Chaining (titokblokk-láncolás)
  - Repeated use of the encryption algorithm to encipher a message consisting of many blocks
- CFB – Cipher Feedback (titokblokk-visszacsatolás)
  - Used to encipher a stream of characters, dealing with each character as it comes
- OFB – Output Feedback (output-visszacsatolás)
  - Another method of stream encryption, used on noisy channels
- CTR – Counter (számláló)
  - Simplified OFB with certain advantages

37/71

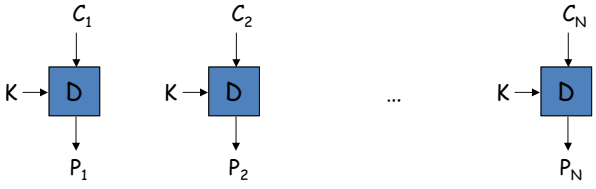
2. Titkosítás

ECB mód

- Encrypt



- Decrypt

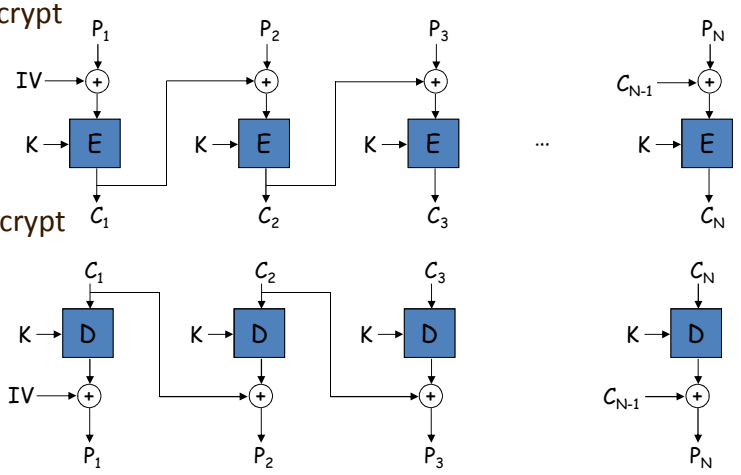


38/71

2. Titkosítás

CBC mód

- Encrypt
- Decrypt



A CBC mód tulajdonságai

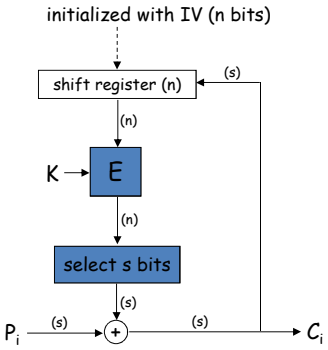
- IV (initialization vector) usually appended as C_0
- Popular mode (e.g. AES-128 CBC was mandatory in TLS 1.2, RFC 5246)
- Parallelizable decryption but not encryption
- Bit change in plaintext or IV propagates to the rest of the ciphertext
- "self-synchronizing" after losing a ciphertext block
- Data leak:

$$C_i = C_j \Rightarrow E_k(P_i \oplus C_{i-1}) = E_k(P_j \oplus C_{j-1})$$
$$P_i \oplus P_j = C_{i-1} \oplus C_{j-1}$$

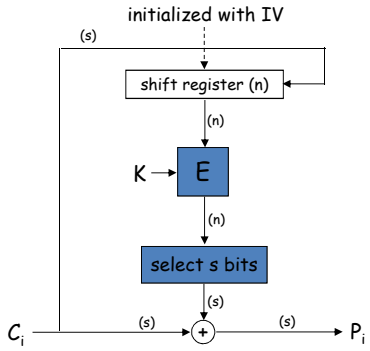
- Sweet32 attack (2016): ciphers with block length 64 bits and large amount of data encrypted using the same key (TLS, OpenVPN)

CFB mód (rövidebb, pld. 8 bites blokkokra)

- Encrypt

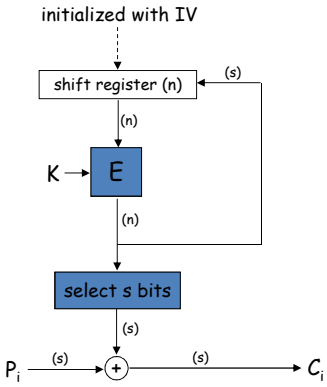


- Decrypt

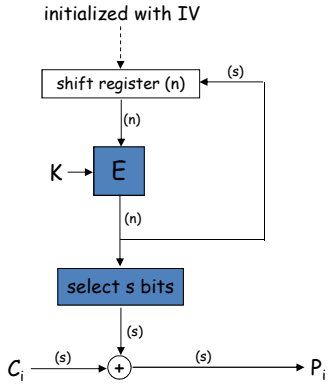


OFB mód (s bites visszacsatolás)

- Encrypt

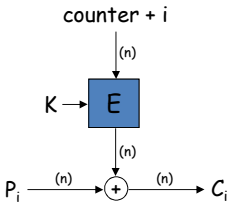


- Decrypt

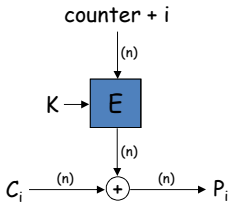


CTR mode

- Encrypt



- Decrypt



Folyamtitkosítás (stream ciphers)

- While block ciphers simultaneously encrypt groups of characters, stream ciphers encrypt individual characters
  - May be better suited for real time applications
- Stream ciphers are usually faster than block ciphers in hardware (but not necessarily in software)
- Limited or no error propagation
  - may be advantageous when transmission errors are probable
- NOTE: the distinction between stream ciphers and block ciphers is not definitive
  - Stream ciphers can be built out of block ciphers using CFB, OFB, or CTR modes
  - A block cipher in ECB or CBC mode can be viewed as a stream cipher that operates on large characters

A Vernam titkosítás és a one-time pad

- Vernam titkosítás
  - $c_i = p_i \oplus k_i$ for i = 1, 2, …
    where p_i are the plaintext digits, k_i are the key stream digits, c_i are the ciphertext digits, and $\oplus$ is the bitwise XOR operation
- Véletlen átkulcsolás (one-time pad)
  - A Vernam cipher where the key stream digits are generated independently and uniformly at random
  - The one-time pad is unconditionally secure [Shannon, 1949]
    - Impractical because of key management problems

45/71

2. Titkosítás

Példák szimmetrikus titkosításra

DES - 56 bit key length, designed by US security service
3DES - effective key length 112 bits
AES (Advanced Encryption Standard) - 128 to 256 bit key length
RC5 variable block size (32, 64 or 128 bits), key size (0 to 2040 bits)
Blowfish - 128 bits, optimized for fast operation on 32-bit microprocessors
IDEA - 128 bits, patented (requires a licence for commercial use)

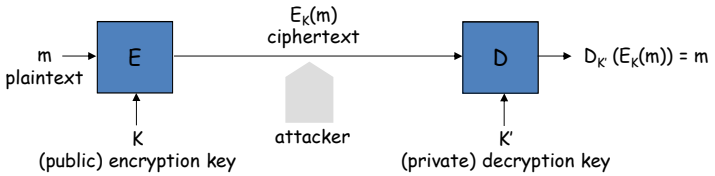
2. Titkosítás

Szimmetrikus titkosítás összegzés

- Fast to encrypt and decrypt, suitable for large volumes of data
- A well-designed cipher is only subject to brute-force attack; the strength is therefore directly related to the key length
- Current recommendation is a key length of at least 128 bits
  i.e. to be fairly sure that your data will be safe for at least 10 years
- Problem - how do you distribute the keys?

Tartalom

Nyilvános kulcsú titkosítás



- Asymmetric-key encryption
  - It is hard (computationally infeasible) to compute K' from K
- K can be made public (public-key cryptography)
  - no need for key setup before communication
- Public-keys are not confidential but they must be authentic!
- The security of asymmetric-key encryption schemes is usually based on some well-known or widely believed hard problems

Példák „nehéz” problémákra

- Factoring problem
  - Given a positive integer n, find its prime factors
    - True complexity is unknown
    - It is believed that it does not belong to P
- Discrete logarithm problem
  - Given a prime p, a generator g of Z_p^*, and an element y in Z_p^*, find the integer x, $0 \le x \le p-2$, such that $g^x \bmod p = y$
    - True complexity is unknown
    - It is believed that it does not belong to P
- Diffie-Hellman problem
  - Given a prime p, a generator g of Z_p^*, and elements $g^x \bmod p$ and $g^y \bmod p$, find $g^{xy} \bmod p$
    - True complexity is unknown
    - It is believed that it does not belong to P

Az RSA séma

- Key generation
  - select p, q large primes (at least 1024 bits each)
  - n = pq, φ(n) = (p-1)(q-1)
  - select e such that $1 < e < \varphi(n)$ and gcd(e, φ(n)) = 1
  - compute d such that ed mod φ(n) = 1 (this is easy if φ(n) is known)
  - the public key is (e, n)
  - the private key is d
- Encryption
  - represent the message as an integer m in [0, n-1]
  - compute $c = m^e \bmod n$
- Decryption
  - compute $m = c^d \bmod n$

Példa

- Choose p = 7 and q = 11 → n = p*q = 77
- Compute encryption key e : (p-1)*(q-1) = 6*10 = 60 → chose e = 13 (13 and 60 are relatively prime numbers)
- Compute decryption key d such that 13*d = 1 mod 60 → d = 37 (37*13 = 481)
- n = 77; e = 13; d = 37
- Send message block m = 7
- Encryption: $c = m^e \bmod n = 7^{13} \bmod 77 = 35$
- Decryption: $m = c^d \bmod n = 35^{37} \bmod 77 = 7$

Miért nehéz feltörni? (kapcsolat az egész faktorizálással)

- The problem of computing d from (e, n) is computationally equivalent to the problem of factoring n
  - If one can factor n, then one can easily compute d
  - If one can compute d, then one can efficiently factor n
- The problem of computing m from c and (e, n) (called the RSA problem) is believed to be computationally equivalent to factoring
  - If one can factor n, then one can easily compute m from c and (e, n)

Sózás (salting)

- Let us assume that the adversary observes a ciphertext $c = E_K(m)$
- Let the set of possible plaintexts be M
- If M is small, then the adversary can try to encrypt every message in M with the publicly known key K until she finds the message m that maps into c
- The usual way to prevent this attack is to randomize the encryption
  - Some random bytes are added to the plaintext message before encryption through the application of the PKCS #1 formatting rules
  - When the message is decrypted, the recipient can recognize and discard these random bytes

Az ElGamal titkosítási séma

- Key generation
  - Generate a large random prime p and choose generator g of the multiplicative group $Z_p^* = \{1, 2, \ldots, p-1\}$
  - Select a random integer a, $1 \le a \le p-2$, and compute $A = g^a \bmod p$
  - The public key is (p, g, A)
  - The private key is a
- Encryption
  - Represent the message as an integer m in [0, p-1]
  - Select a random integer r, $1 \le r \le p-2$, and compute $R = g^r \bmod p$
  - Compute $C = m \times A^r \bmod p$
  - The ciphertext is the pair (R, C)
- Decryption
  - Compute $m = C \times R^{p-1-a} \bmod p$
- Proof of decryption

  $C \times R^{p-1-a} \equiv m \times A^r \times R^{p-1-a} \equiv m \times g^{ar} \times g^{r(p-1-a)} \equiv m \times (g^{p-1})^r \equiv m \pmod p$

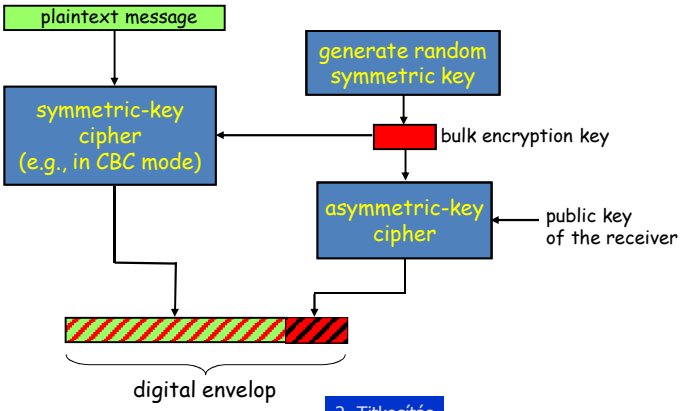
Az ElGamal séma biztonsága

- The security of the ElGamal scheme is said to be based on the discrete logarithm problem in Z_p^*, although equivalence has not been proven yet
- Recovering m given p, g, A, R, and C is equivalent to solving the Diffie-Hellman problem

A digitális boríték

- Most popular public-key encryption methods are several orders of magnitude slower than the best known symmetric key schemes

→ Public-key encryption is used together with symmetric-key encryption; the technique is called digital enveloping



57/71

2. Titkosítás

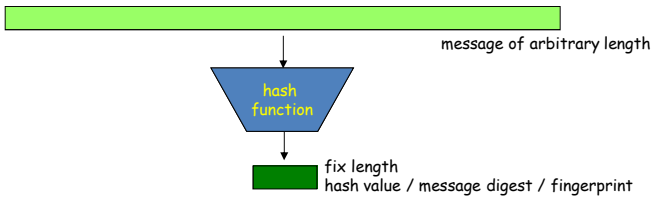
Tartalom

1. Alapfogalmak
2. Titkosítás
   - Szimmetrikus kulcsú sémák
   - Nyilvános kulcsú sémák
3. Hash függvények
   - Message Authentication Codes (MAC)
   - Digitális aláírás

58/71

Hash függvények

- A hash function maps bit strings of arbitrary finite length to bit strings of fixed length (n bits)
- Many-to-one mapping → collisions are unavoidable
- However, finding collisions are difficult → the hash value of a message can serve as a compact representative image of the message (similar to fingerprints)



Elvárt tulajdonságok

- Ease of computation
  - Given an input x, the hash value h(x) of x is easy to compute
- One-way property (preimage resistance)
  - Given a hash value y (for which no preimage is known), it is computationally infeasible to find any input x s.t. h(x) = y
- Weak collision resistance (2nd preimage resistance)
  - Given an input x, it is computationally infeasible to find a second input x' such that h(x') = h(x)
- Strong collision resistance (collision resistance)
  - it is computationally infeasible to find any two distinct inputs x and x' such that h(x) = h(x')

A születésnapi paradoxon (Birthday Paradox)

- Given a set of N elements, from which we draw k elements randomly (with replacement). What is the probability of encountering at least one repeating element?
- First, compute the probability of no repetition:
  - The first element x_1 can be anything
  - When choosing the second element x_2, the probability of $x_2 \neq x_1$ is 1-1/N
  - When choosing x_3, the probability of $x_3 \neq x_2$ and $x_3 \neq x_1$ is 1-2/N
  - …
  - When choosing the k-th element, the probability of no repetition is 1-(k-1)/N
  - The probability of no repetition is (1 - 1/N)(1 - 2/N)…(1 – (k-1)/N)
  - When x is small, $(1-x) \approx e^{-x}$
  - $(1 - 1/N)(1 - 2/N)\ldots(1 – (k-1)/N) = e^{-1/N}e^{-2/N} \ldots e^{-(k-1)/N} = e^{-k(k-1)/2N}$
- The probability of at least one repetition after k drawing is

$$1 – e^{-k(k-1)/2N}$$

A születésnapi paradoxon (folyt.)

- How many drawings do you need, if you want the probability of at least one repetition to be ε?
- Solve the following for k:

$$\varepsilon = 1 – e^{-k(k-1)/2N}$$
$$k(k-1) = 2N \ln(1/1-\varepsilon)$$
$$k \approx \text{sqrt}(2N \ln(1/1-\varepsilon))$$

- Examples:
  - $\varepsilon = ½ \rightarrow k \approx 1.177 \text{ sqrt}(N)$
  - $\varepsilon = ¾ \rightarrow k \approx 1.665 \text{ sqrt}(N)$
  - $\varepsilon = 0.9 \rightarrow k \approx 2.146 \text{ sqrt}(N)$
- Origin of the name “Birthday Paradox”:
  - Elements are dates in a year (N = 365)
  - Among $1.177 \text{ sqrt}(365) \approx 23$ randomly selected people, there will be at least two that have the same birthday with probability ½

A hash függvény kimeneti mérete

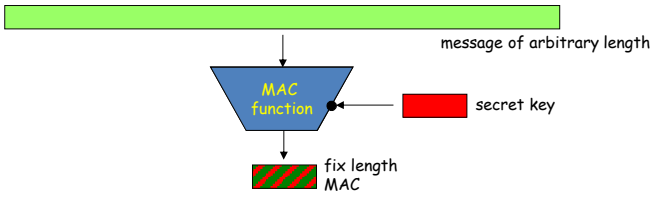
- the Birthday Paradox have a profound impact on the design of hash functions (and other cryptographic algorithms and protocols)!
  - Let n be the output size of a hash function
  - Among ≈sqrt(2^n) = $2^{n/2}$ randomly chosen messages, with high probability, there will be a collision pair
  - It is easier to find collisions than to find preimages or 2^{nd} preimages for a given hash value

  → in order to resist birthday attacks, $2^{n/2}$ should be sufficiently large (e.g., n = 160 bits)

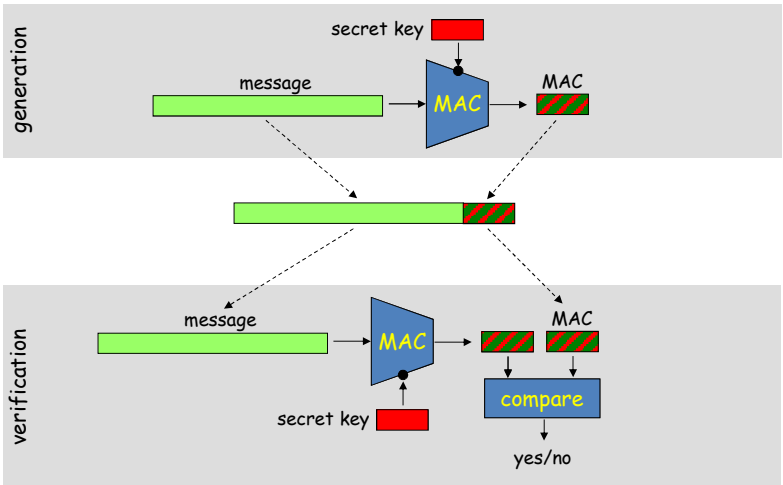
Tartalom

Message Authentication Codes (MACs)

- MAC functions can be viewed as hash functions with two functionally distinct inputs: a **message and a secret key**
- they produce a fixed size output (say n bits) called the MAC
- practically it should be infeasible to produce a correct MAC for a message without the knowledge of the secret key
- MAC functions can be used to implement data integrity and message origin authentication services



MAC generálás and verifikáció



A MAC elvárt tulajdonságai

- Ease of computation
  - Given an input x and a secret key k, it is easy to compute $MAC_k(x)$
- Key non-recovery
  - it is computationally infeasible to recover the secret key k, given one or more text-MAC pairs $(x_i, MAC_k(x_i))$ for that k
- Computation resistance
  - Given zero or more text-MAC pairs $(x_i, MAC_k(x_i))$, it is computationally infeasible to find a text-MAC pair $(x, MAC_k(x))$ for any new input $x \neq x_i$
  - computation resistance implies key non-recovery but the reverse is not true in general
- Problems
  - Establishment of shared secret
  - Inability to provide non-repudiation

67/71

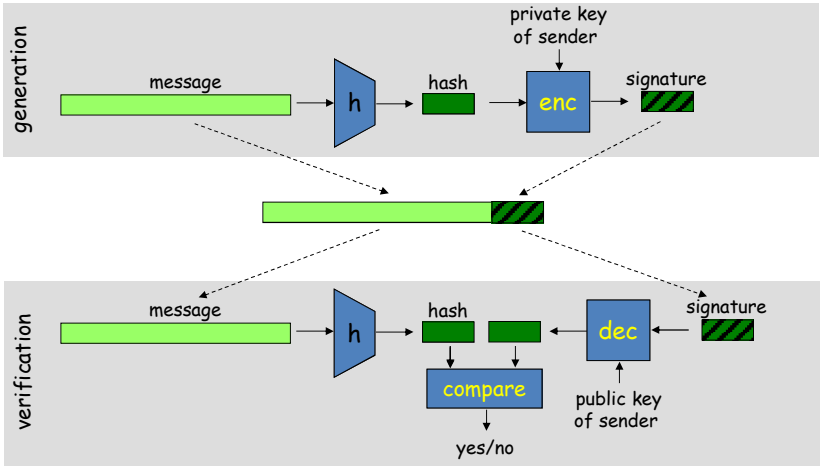
Message authentication codes

Tartalom

1. Alapfogalmak
2. Titkosítás
   - Szimmetrikus kulcsú sémák
   - Nyilvános kulcsú sémák
3. Hash függvények
   - Message Authentication Codes (MAC)
   - Digitális aláírás

68/71

Digitális aláírás



69/71

Digitális aláírás

Digitális aláírás össszegzés

- A **digitális aláírás** a nyilvános kulcsú titkosító rendszerek egy lehetősége, amellyel a hagyományos aláírást tudjuk helyettesíteni az informatika világában
- Igazolni tudjuk az aláíró személyét, és azt, hogy a dokumentum az aláírás óta nem változott meg
- A jó digitális aláírás a hagyományos aláírás minden jó tulajdonságát hordozza, sőt ki is egészíti őket

Köszönöm a figyelmet
Szép napot!