Az informatikai biztonság alapjai

Pintér-Huszti Andrea

2023. szeptember 17.

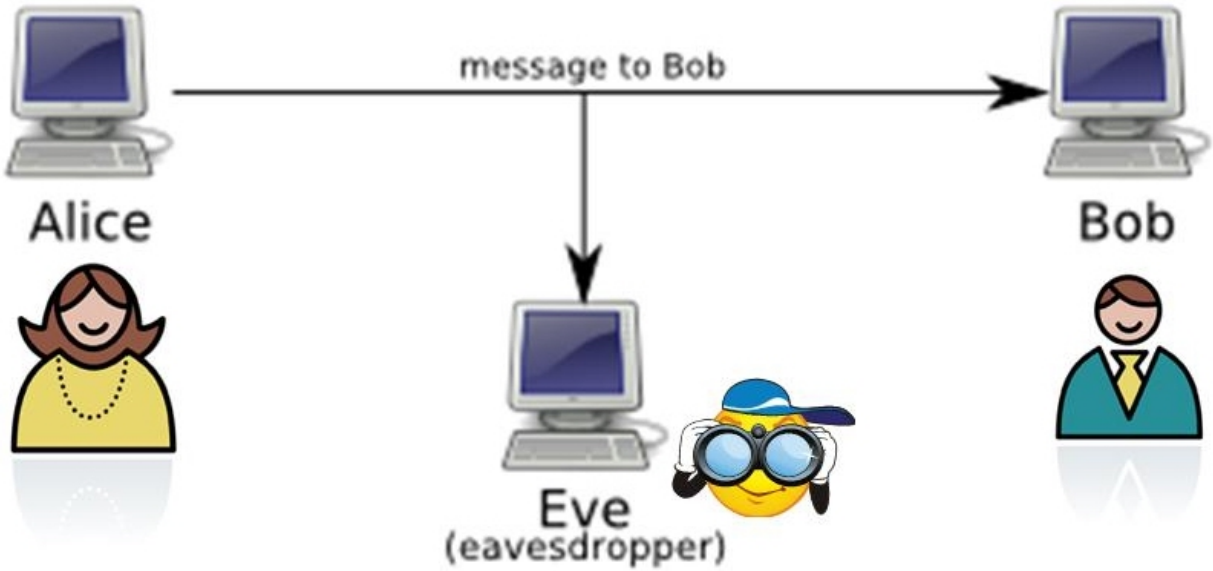
Tartalom

Titkosítási sémákról általában

Választott titkosított üzenet alapú támadás
Nem alkalmazkodó (Non-adaptive CCA)
A támadó előre kiválasztja a titkos üzenetet, és egyszerre kapja meg a dekódolás
Alkalmazkodó (Adaptive CCA)
A támadó lépésenként, az előző válaszok ismeretében választ új titkos üzeneteke
Szintén aktív támadás, nagyon erős modell.

Titkosítási sémák

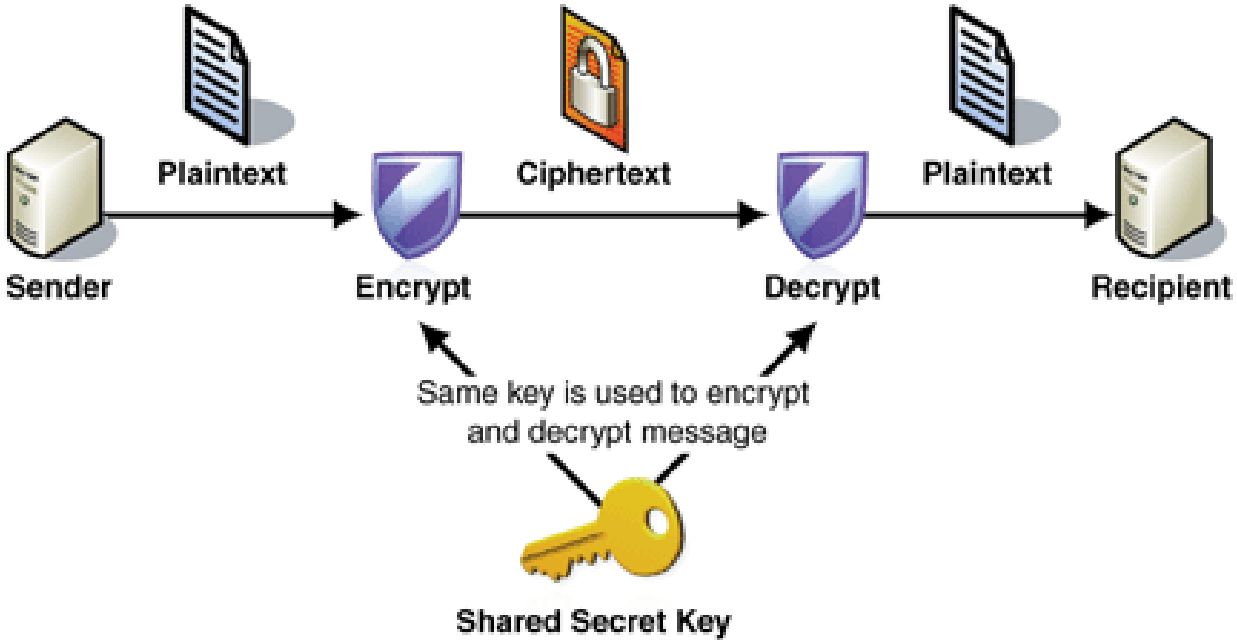
Alice szeretne egy bizalmas üzenetet küldeni Bobnak a nyílt csatornán.



- $\mathcal{P}$ (plaintext space): nyílt üzenetek **véges** halmaza
- $\mathcal{C}$ (ciphertext space): titkosított üzenetek **véges** halmaza
- $\mathcal{K}$ (key space): a lehetséges kulcsok **véges** halmaza
- $m \in \mathcal{P}$ (plaintext): nyílt üzenet
- $c \in \mathcal{C}$ (ciphertext): titkosított üzenet

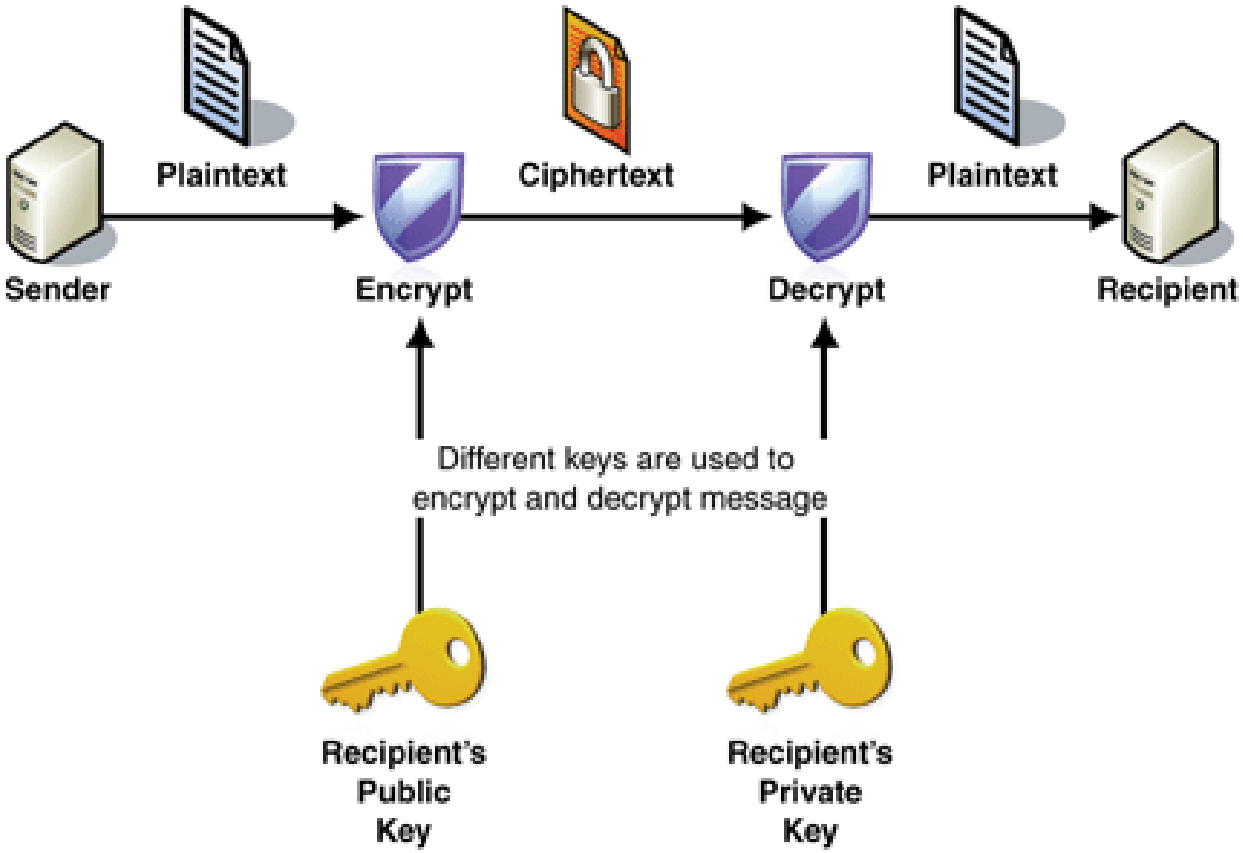
Szimmetrikus titkosítási sémák

A titkosító és visszafejtő kulcs megegyezik, vagy a visszafejtő a titkosító kulcsból *könnyen* (polinomiális időn belül) kiszámítható.



Aszimmetrikus titkosítási sémák

A titkosító és visszafejtő kulcs különbözik olyannyira, hogy a visszafejtő a titkosító kulcsból csak *nehezen* (nem ismerünk rá polinomiális idejű algoritmust) számítható ki.



Összehasonlítás

	Szimmetrikus	Aszimmetrikus
Kulcsok titkossága	kulcsok titkosak (K)	(PK, SK) nyilvános (public) és titkos (secret) kulcs
Kulcsok kezelése	kulcscsere algoritmusok	Nyilvános Kulcs Infrastruktúra (Public Key Infrastructure)
Időigény	gyors algoritmusok	lassú algoritmusok
Üzenetek mérete	nagy méretű	kis méretű
Példák	TDES, AES	RSA, ElGamal, elliptikus görbe titkosítás

Választott nyílt üzenet alapú támadás
Nem alkalmazkodó (Non-adaptive CPA)
A támadó előre kiválasztja az összes üzenetet, és egyszerre titkosíttatja őket.
Nincs lehetősége új kérdéseket feltenni az előző válaszok alapján.
Alkalmazkodó (Adaptive CPA)
A támadó lépésről lépésre választhat új nyílt szövegeket az előző titkosítások eredménye alapján.
Ez egy aktív támadás, mert a támadó interaktívan „kérdezhet” a rendszerből.

Szimmetrikus titkosítási séma formális definíció

Definíció

A $SE = (Key, Enc, Dec)$ hármas egy szimmetrikus titkosítási séma, ha

- *Key: kulcsgeneráló algoritmus, mely egy k biztonsági paraméterhez (kulcs méretére utal) megad egy $K \in \mathcal{K}$ titkos kulcsot.*
- *Enc: titkosító algoritmus, mely $\forall m \in \mathcal{P}$ nyílt üzenethez és $\forall K \in \mathcal{K}$ titkos kulcshoz generál egy $c \in \mathcal{C}$ titkosított üzenetet.*

$$c = Enc_K(m)$$

- *Dec: visszafejtő algoritmus, mely egy $c \in \mathcal{C}$ titkosított üzenethez és egy adott $K \in \mathcal{K}$ kulcshoz megad egy $m \in \mathcal{P}$ nyílt üzenetet.*

$$m = Dec_K(c)$$

Szimmetrikus titkosítási séma

- Sok esetben a titkosítási algoritmus inputja egy r véletlen is. Így a titkosító algoritmus randomizált.
- A visszafejtő algoritmus determinisztikus.

Definíció

Az $SE = (Key, Enc, Dec)$ szimmetrikus titkosítási séma korrekt visszafejtést biztosít, ha $\forall m \in \mathcal{P}$ és $\forall K \in \mathcal{K}$ esetén

$$Dec_K(Enc_K(m)) = m.$$

A szimmetrikus titkosítási séma helyesen működik: minden üzenetet pontosan vissza lehet fejteni, ha ugyanazzal a kulccsal titkosítjuk és dekódoljuk.

Aszimmetrikus titkosítási séma formális definíció

Definíció

A $AE = (Key, Enc, Dec)$ hármas egy aszimmetrikus titkosítási séma, ha

- *Key: kulcsgeneráló algoritmus, mely egy k biztonsági paraméterhez (kulcs méretére utal) megad egy $(PK, SK) \in \mathcal{K}$ nyilvános és titkos kulcsból álló párt.*
- *Enc: titkosító algoritmus, mely $\forall m \in \mathcal{P}$ nyílt üzenethez és PK nyilvános kulcshoz generál egy $c \in \mathcal{C}$ titkosított üzenetet.*

$$c = Enc_{PK}(m)$$

- *Dec: visszafejtő algoritmus, mely egy $c \in \mathcal{C}$ titkosított üzenethez és egy adott SK kulcshoz megad egy $m \in \mathcal{P}$ nyílt üzenetet.*

$$m = Dec_{SK}(c)$$

Aszimmetrikus titkosítási séma

- Sok esetben a titkosítási algoritmus inputja egy r véletlen is. Így a titkosító algoritmus randomizált.
- A visszafejtő algoritmus determinisztikus.
- A kulcsgeneráló algoritmus outputja meghatározza a $\mathcal{P}, \mathcal{C}, \mathcal{K}$ halmazokat.

Definíció

Az $AE = (Key, Enc, Dec)$ aszimmetrikus titkosítási séma korrekt visszafejtést biztosít, ha $\forall m \in \mathcal{P}$ és $\forall (PK, SK) \in \mathcal{K}$ esetén

$$Dec_{SK}(Enc_{PK}(m)) = m.$$

ha minden üzenet és minden kulcspár esetén az üzenet titkosítása a publikus kulccsal, majd visszafejtése a titkos kulccsal pontosan visszaadja az eredeti üzenetet.

Passzív támadások

A támadó célja:

- A titkos visszafejtő kulcs megszerzése
- Egy adott titkosított üzenethez tartozó nyílt üzenet megszerzése

Támadási módok:

- Csak a titkosított üzenet ismert (Ciphertext Only Attack)
- Ismert nyílt üzenet alapú támadás (Known Plaintext Attack)
- Választott nyílt üzenet alapú (Chosen Plaintext Attack)
  - Nem alkalmazkodó (Non-adaptive)
  - Alkalmazkodó (Adaptive) (aktív támadás)
- Választott titkosított üzenet alapú (Chosen Ciphertext Attack)
  - Nem alkalmazkodó (Non-adaptive)
  - Alkalmazkodó (Adaptive) (aktív támadás)

COA

Csak a titkosított üzenet ismert (Ciphertext Only Attack) A támadó rendelkezésére áll egy ugyanazon kulccsal titkosított üzenetekből álló lista:

$Enc_K(m_1)$	$Enc_{PK}(m_1)$
$Enc_K(m_2)$	$Enc_{PK}(m_2)$
$\vdots$	$\vdots$
$Enc_K(m_n)$	$Enc_{PK}(m_n)$

A támadó csak a titkosított szöveget (ciphertext) ismeri, és ebből próbálja megfejteni az üzenetet vagy a kulcsot. Legnehezebb támadási forma.

KPA

Ismert nyílt üzenet alapú támadás (Known Plaintext Attack) A támadó rendelkezésére áll egy ugyanazon kulccsal titkosított nyílt és titkosított üzenetekből álló lista:

$(m_1, Enc_K(m_1))$	$(m_1, Enc_{PK}(m_1))$
$(m_2, Enc_K(m_2))$	$(m_2, Enc_{PK}(m_2))$
$\vdots$	$\vdots$
$(m_n, Enc_K(m_n))$	$(m_n, Enc_{PK}(m_n))$

A cél vagy a titkos kulcs, vagy a listán nem szereplő titkosított üzenethez tartozó nyílt üzenet megszerzése.

A támadó ismeri a nyílt szöveg egy részét és a hozzátartozó titkosított szöveget is. Ebből próbálja kitalálni a kulcsot vagy a titkosítási módot.

CPA

Választott nyílt üzenet alapú (Chosen Plaintext Attack) A támadó rendelkezésére áll egy ugyanazon kulccsal titkosított nyílt és titkosított üzenetekből álló lista, ahol a nyílt üzenetek a támadó által választottak:

A támadó tetszőleges nyílt szövegeket választhat, és megnézheti, hogyan titkosítja azokat a rendszer. Ezáltal mintázatokat keres a titkosított kimenetekben.

$(m_1, Enc_K(m_1))$	$(m_1, Enc_{PK}(m_1))$
$(m_2, Enc_K(m_2))$	$(m_2, Enc_{PK}(m_2))$
$\vdots$	$\vdots$
$(m_n, Enc_K(m_n))$	$(m_n, Enc_{PK}(m_n))$

Nem alkalmazkodó esetben a támadó előre kiválasztja a nyílt üzeneteket, míg alkalmazkodó esetben a kapott titkosított üzenetek alapján választja ki a következő nyílt üzenetet.

A cél vagy a titkos kulcs, vagy a listán nem szereplő titkosított üzenethez tartozó nyílt üzenet megszerzése.

CCA

Választott titkosított üzenet alapú (Chosen Ciphertext Attack) A támadó rendelkezésére áll egy ugyanazon kulccsal titkosított nyílt és titkosított üzenetekből álló lista, ahol a titkosított üzenetek a támadó által választottak:

$(c_1, Dec_K(c_1))$	$(c_1, Dec_{SK}(c_1))$
$(c_2, Dec_K(c_2))$	$(c_2, Dec_{SK}(c_2))$
$\vdots$	$\vdots$
$(c_n, Dec_K(c_n))$	$(c_n, Dec_{SK}(c_n))$

Nem alkalmazkodó esetben a támadó előre kiválasztja a titkosított üzeneteket, míg alkalmazkodó esetben a kapott nyílt üzenetek alapján választja ki a következő titkosított üzenetet.

A cél vagy a titkos kulcs, vagy a listán nem szereplő titkosított üzenethez tartozó nyílt üzenet megszerzése.

Mi az? A támadó tetszőleges ciphertextet adhat a dekódoló rendszernek, és kap visszajelzést (pl. siker/hiba, vagy a dekódolt plaintext). Nagyon erős modell (pl. padding oracle támadások).

Biztonsági kérdések

- Feltétel nélküli biztonság: A támadó korlátlan számítási kapacitással rendelkezik. Nehéz a gyakorlatban megvalósítani.
- Feltételes biztonság: A támadó korlátos számítási kapacitással rendelkezik, polinom idejű algoritmusokat használ.
- Kerckhoff-elv: Azaz a biztonság egyedül a kulcsnak, és nem magának az algoritmusnak a titkosságán alapuljon. Feltesszük, hogy a támadó a rendszert ismeri. Mert:
  - tömeges méretű alkalmazásoknál úgy sem lehetne az algoritmust titokban tartani
  - az algoritmus az implementációkból visszafejthető
  - egy nyilvános, tesztelt módszer nagyobb bizalmat érdemel, mint egy soha nem látott "szupertitkos"

Vernam One Time Pad

Gilbert Vernam (1917) $\mathcal{P} = \mathcal{C} = \mathcal{K} = \mathbb{Z}_2^n$

- *Key*: $K = K_1 K_2 \dots K_n \in \mathbb{Z}_2^n$ véletlenül választott
- *Enc*: $\forall m = m_1 m_2 \dots m_n \in \mathbb{Z}_2^n$ esetén
  $Enc_K(m) = m_1 + K_1 \ m_2 + K_2 \dots m_n + K_n \pmod 2$
- *Dec*: $\forall c = c_1 c_2 \dots c_n \in \mathbb{Z}_2^n$ esetén
  $Dec_K(c) = c_1 + K_1 \ c_2 + K_2 \dots c_n + K_n \pmod 2$

Jellemzők:

- mod 2 összeadás: XOR ($\oplus$)
- Korrekt visszafejtést biztosít, hiszen a bitenkénti XOR asszociatív művelet, azaz $\forall m, K \in \mathbb{Z}_2^n$
  $(m \oplus K) \oplus K = m \oplus (K \oplus K) = m$
- az nyílt üzenet mérete határozza meg a kulcs méretét
- a kulcsnak valódi véletlennek kell lennie
- kétszer nem használható ugyanaz a kulcs
  $(m \oplus c = m \oplus (m \oplus K) = K)$-> sok kulcscsere