
Kriptográfia 4. gyakorlat

1) Példa az Enigma használatára:

Beállítások (a titkos kódkönyv alapján mai napra)

modell: M3 Wehrmacht/Luftwaffe

UKW (reflektor típusa): B

Walzenlänge (rotorok): II, IV, III

Ringstellung (huzalozásuk): U, N, F

Steckerverbindungen (kapcsolótábla):
AL CF DU HP RX

Titkosítsuk a következő üzenetet:

THIS X IS X A X MESSAGE

A titkosításhoz választott kezdeti pozíciónk legyen: **VTK**

A titkosításhoz választott kapcsolatkulcsunk legyen: **RHC**

Megoldás

U47 DE U112 1800 = 4 =

az üzenet egyéb adatai

start pozíció

VTK ZIW

a kapcsolatkulcs
a start pozícióból
titkosítva

ALNRI HWKVS BSGZK FDUMK

2) Ellenőrzésként fejtsük is vissza az üzenetet!

3) Fejtsük vissza ezt a másik, szintén a napi beállításban titkosított üzenetet:

..... = KEZ SAW

ZUYAB HEBWZ JFNJO

ENIGMA CHALLENGES

[http://users.telenet.be/d.rijmenants/
en/challenge.htm](http://users.telenet.be/d.rijmenants/en/challenge.htm)

4) A OTP titkosító szinkronizálása

- A OTP alkalmazása megköveteli, hogy az adó és a vevő tökéletesen szinkronban legyen, ne legyen kimaradt, vagy többször vett adásrészlet.
 - Ezt azonban a gyakorlatban nem mindig teljesül.
 - Javasolj erre a problémára valami megoldást!
-

Egy lehetséges megoldás

- Módosítsuk az átviteli protokollt úgy, hogy
 - Adjunk az üzenethez időnként egy **számlálót**.
 - Titkosítva küldjük-e a számlálót vagy ne?
 - Mi történik, ha a számláló ismertté válik? Hogyan érinti ez az üzenet tökéletes titkosságát?
 - Hogyan védekezhetünk a számláló szándékos megváltoztatása ellen?
-

5) Két OTP üzenet egy kulccsal

- Javasoljuk olyan módszert mellyel Alíz **két** üzenetet küldhet **egyetlen**, Béla számára **ismeretlen OTP kulccsal**.
- A két üzenet legyen:
 $M1$ és $M2$, mindkettő n bites
- A OTP kulcs legyen k , szintén n bites
- A rendszernek biztonságosan kell védenie külön-külön mindkét üzenetet.
- Persze, ha mindkét üzenetet lehallgatják, akkor a k kulcs és az üzenetek is felfedésre kerülnek.

Megoldás

- Alíz generál egy véletlen n bites bitsorozatot, legyen ez N .
- Ennek a szokásos neve „nonce” (egyszer)
nonce-word = alkalmi kifejezés
- $E_k(M1) = (M1 \oplus k) \parallel (N \oplus k)$ (\parallel az egymás után írás)
- $E'_k(M2) = (M2 \oplus k) \parallel N$
- Béla
 - $E'_k(M2)$ -ből leválasztja N -et,
 - majd $E_k(M1)$ -ből kiszámítja k -t
- Ugyanakkor, ha a támadó csak az egyik üzenetet hallgatja le, nem tud meg semmit:
 - $E_k(M1)$ -ben ugyan, kétszer van alkalmazva a k kulcs, nem „one-time”, de mivel N véletlen, az üzenet második n bitje bármi lehet, nem ad semmi információt.
 - $E'_k(M2)$ -ből ugyan megkapja N -et, de az önmagában semmit sem ér.

6) OTP, mely még az üzenet hosszát sem árulja el

- OTP helyes alkalmazása esetén az egyetlen információ, amit a lehallgató meg tud szerzi az üzenet hossza
- Hogyan lehetne ezen segíteni?
- Tegyük fel, hogy a küldendő üzenet 8,16,24 vagy 32 bit hosszú
- Azaz $P = \{0,1\}^8 \cup \{0,1\}^{16} \cup \{0,1\}^{24} \cup \{0,1\}^{32}$
- Tervezzünk olyan feltétlenül biztonságos (unconditionally secure) titkosítót, mellyel a támadó **semmit sem** tud meg a nyílt szövegről, **még annak hosszát sem**.
- Feltehetjük, hogy a kulcs valóban véletlen, tetszőleges hosszú kulcsot generálhatunk, de azért maradjunk hatékonyak. És persze a terv a OTP-n alapuljon.

Megoldás

- Minden üzenet 4 bájt + 2 redundáns bit legyen
- A 2 redundáns bit adja meg a nyílt szöveg hosszát (1, 2, 3 vagy 4 bájt binárisan)
- Ezt az üzenet elejére vagy végére tehetjük.
- A nemhasznált bájtok helyére válasszunk véletlen biteket.
- Vajon most kódolnunk, kell a hosszát jelző két bájtot?
- IGEN.

7) OTP rövidebb kulccsal?

- Dr. Rövidítő professzor az alábbi ötlettel állt elő a klasszikus OTP kulcshosszának csökkentésére:

Tegyük fel, hogy a nyíltzöveg páros sok, mondjuk $2l$ bitből áll: $m = m_1 m_2 \dots m_{2l}$. Használjunk hozzá csak fele olyan hosszú, azaz l bites véletlen kulcsot $k = k_1 k_2 \dots k_l$ -t, így:

$$\begin{aligned} E_k(m) := & m_1 \oplus k_1 \parallel m_2 \oplus m_1 \oplus k_1 \parallel \\ & m_3 \oplus k_2 \parallel m_4 \oplus m_3 \oplus k_2 \parallel \\ & \dots \qquad \qquad \qquad \dots \\ & m_{2l-3} \oplus k_{l-1} \parallel m_{2l-2} \oplus m_{2l-3} \oplus k_{l-1} \parallel \\ & m_{2l-1} \oplus k_l \parallel m_{2l} \oplus m_{2l-1} \oplus k_l \end{aligned}$$

Vajon beválik-e ez az újítás?

Megoldás

- Írjuk fel a megfejtést a kódszöveg bitjeit $E_k(m)=c=c_1c_2\dots c_{2l}$ -vel jelölve:
- $m=D_k(c) = c_1 \oplus k_1 \parallel c_2 \oplus c_1$
 $c_3 \oplus k_2 \parallel c_4 \oplus c_3$
...
 $c_{2l-1} \oplus k_l \parallel c_{2l} \oplus c_{2l-1}$
- Ez eddig rendben. Mi akkor a probléma a módszerrel?
- A páros sorszámú bitek a kulcs nélkül is fejthetők, így az üzenet „fele” elolvasható, nagyon nem biztonságos.

8) OTP rövidebb kulccsal II?

- Dr. Rövidítő professzor újabb ötlete a következő:
- Tegyük fel, hogy a nyíltzöveg páros sok, mondjuk $2l$ bitből áll: $m=m_1m_2\dots m_{2l}$. Használjunk hozzá csak fele olyan hosszú, azaz l bites véletlen kulcsot $k=k_1k_2\dots k_l$ -t, így:

$$E_k(m) := m_1 \oplus k_1 \parallel m_2 \oplus k_1 \parallel$$
$$m_3 \oplus k_2 \parallel m_4 \oplus k_2 \parallel$$
$$\dots \qquad \dots$$
$$m_{2l-3} \oplus k_{l-1} \parallel m_{2l-2} \oplus k_{l-1} \parallel$$
$$m_{2l-1} \oplus k_l \parallel m_{2l} \oplus k_l$$

- Ez a módszer feltétlenül biztonságos-e?

Megoldás

- Legyen $E_k(m) = c = c_1 c_2 \dots c_{2l}$.

- Ekkor

$$c_1 \oplus c_2 = m_1 \oplus k_1 \oplus m_2 \oplus k_1 = m_1 \oplus m_2,$$

$$c_3 \oplus c_4 = m_3 \oplus k_2 \oplus m_4 \oplus k_2 = m_3 \oplus m_4, \dots$$

$$c_{2l-1} \oplus c_{2l} = m_{2l-1} \oplus k_l \oplus m_{2l} \oplus k_l = m_{2l-1} \oplus m_{2l}.$$

- Vagyis a kulcs nélkül $m_1 \oplus m_2, m_3 \oplus m_4, \dots, m_{2l-1} \oplus m_{2l}$ kiszámítható.
- Így, ha az üzenet minden második bitjét tudjuk, vagy megsejtjük, a maradék kiszámítható.
- Nem feltétel nélkül biztonságos.