Adatbiztonság

Tóth Zsolt

Miskolci Egyetem

2014

Tartalomjegyzék

Bevezetés

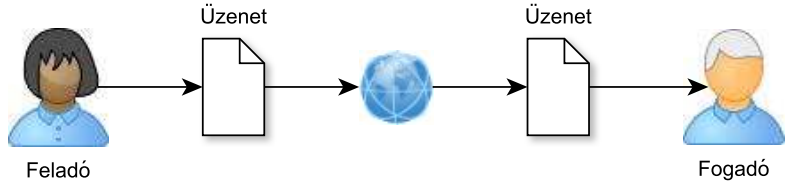
- Matematika háttér
  - Kommunikáció elmélet
  - Információ elmélet
  - Kód elmélet
  - Kriptográfia
- Megismerésük nem képezi a tárgy részét.

Célkitűzés

- Alapfogalmak megismerése
- Kapcsolódó .Net osztályok áttekintése

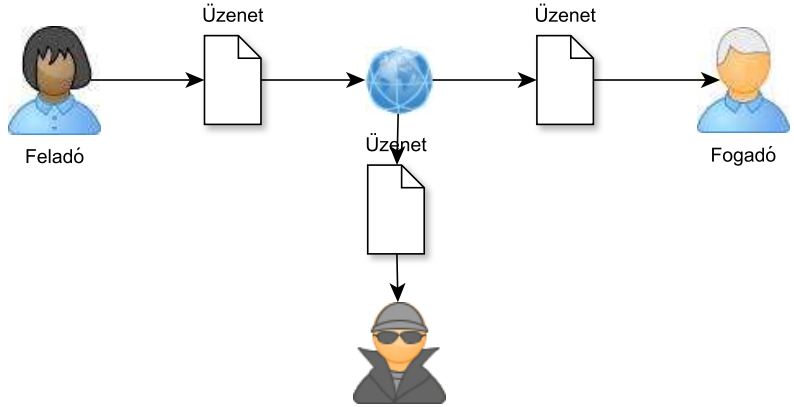
Kommunikáció

- Feladó
- Üzenet
  - Byte tömb
    - Szöveg
    - Hang
    - Videó
    - Állomány
- Csatorna
  - Hálózat
  - Fájl
  - Memória
- Vevő



Probléma

- Osztott csatorna
  - Üzenetszórás hálózaton
  - Elérhető állomány
- Bárki hozzáférhet
- Üzenet elérhető
- Bizalmas információ



Tartalomjegyzék

Titkosítás

- Üzenet kódolása
- Csak a Feladó és a Fogadó tudja olvasni
- Kulcs alapú
  - Egy kulcsos (Szimmetrikus)
  - Két kulcsos (Aszimmetrikus)

Alapfogalmak

Nyílt szöveg információt hordozza (*plain text*)

Titkosított szöveg kódolt dokumentum (*cypher text*)

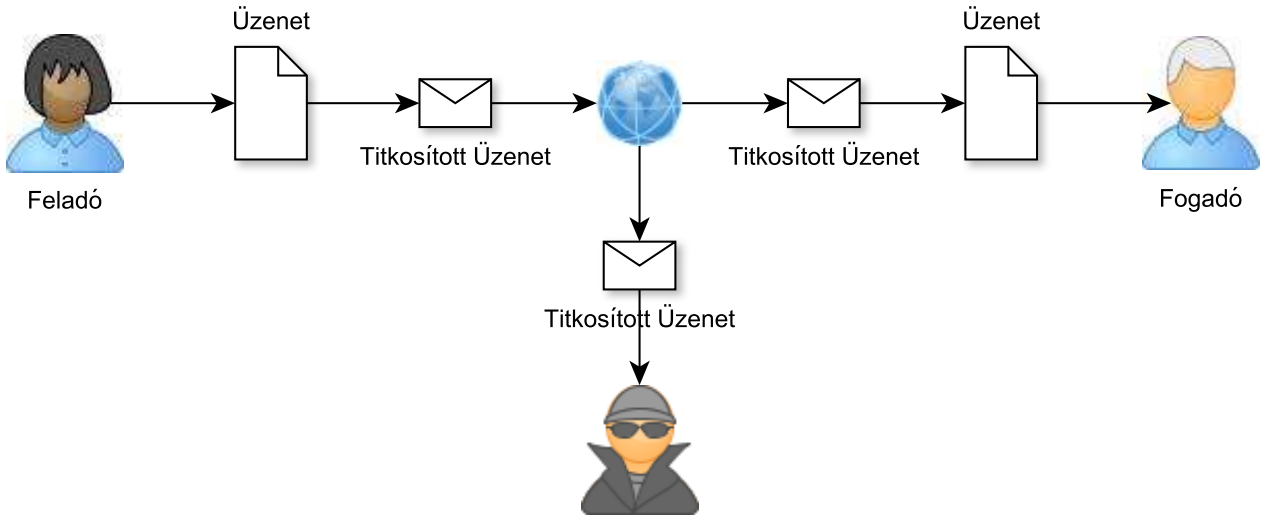
Kódolás Algoritmus alkalmazása a nyílt szövegre

Dekódolás Algoritmus a titkosított szövegből a nyílt szöveg meghatározására

Alkalmazások

- Kommunikáció
  - Számítógép–hálózatok
  - Internet
  - Mobiltelefonok
  - Wifi
- Gazdaság
  - Bankautomaták
  - e–Banking
- Szoftverek
  - Lemez titkosítás
  - Fájl, mappa titkosítás
  - Adatbázis titkosítás

Titkosító Algoritmusok

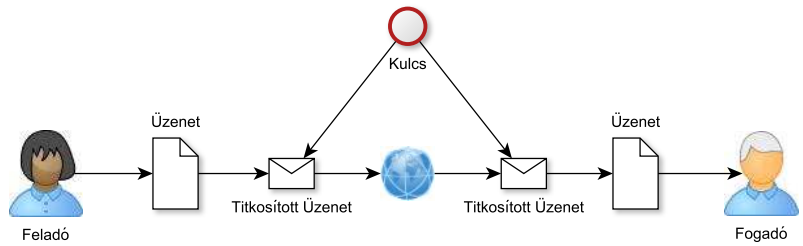


Szimmetrikus Algoritmusok
Egykulcsos rendszerek

Aszimmetrikus Algoritmusok
Nyílt kulcsos rendszerek

Szimmetrikus Algoritmusok

- Egy kulcs
  - Közösen ismert
  - Titkos
- Hagyományos módszer
- Kódolás, dekódolás a kulcs ismeretében lehetséges
- Problémák
  - Kulcsot védeni kell
  - Kulcsot el kell juttatni a Fogadónak



Aszimmetrikus Algoritmusok

Fő lépések

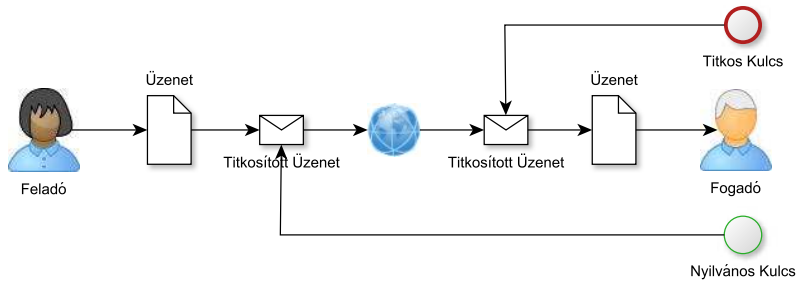
1. Feladó kódolja az üzenetet a Kulccsal
2. Elküldi a Titkosított üzenetet a Csatornán
3. Fogadó megkapja a Titkosított Üzenetet
4. Dekódolja a Titkosított Üzenetet
5. Olvassa az Üzenetet

Aszimmetrikus Algoritmusok

- Két kulcsos rendszer

  Nyilvános Kulcs Bárki hozzáférhet

  Titkos Kulcs Csak a tulajdonos ismeri
- Nyilvános kulcs célja a titkosítás
- A titkosított üzenet csak a Titkos Kulccsal visszafejthető



Aszimmetrikus Algoritmusok

Fő lépések

1. Feladó titkosítja az Üzenetet a Fogadó Nyilvános Kulcsával
2. Titkosított Üzenetet elküldi a Csatornán keresztül
3. Fogadó megkapja a Titkosított Üzenetet
4. Dekódolja a Titkosított Üzenetet a Titkos Kulcsával
5. Fogadó olvassa az Üzenetet

Tartalomjegyzék

System.Security.Cryptography

- Adatbiztonság
- Névtér
- Osztályok
- Algoritmusok
- Szolgáltatások

Célok

- Bizalmasság (olvasás)
- Adat integritás (módosítás)
- Hitelesítés (forrás)

Cryptography alapelemek

Alapelem	Használata
Egy kulcsos titkosítás	Átalakítja az adatot, hogy harmadik fél ne olvashassa. Egykulcsos titkosítást használ.
Nyílt kulcsos titkosítás	Átalakítja az adatot, hogy harmadik fél ne olvashassa. Két kulcsú titkosítást használ.
Digitális aláírás	Segít megerősíteni, hogy az adat egy adott féltől származik. Digitális aláírást készít ami egyedi.
Kriptongrafikus hash	Az adathoz egy adott hosszúságú byte tömböt rendel. Az adat módosulásával a hash érték változik

SymmetricAlgorithm

- abstract
- Ősosztály
- Szimmetrikus titkosítások

Metódusok

- Create
- GenerateKey
- GenerateIV
- CreateEncryptor
- CreateDecryptor

Propterties

- Key
- IV (Initialization Vector)

Leszármazottak

- Aes
- DES
- RC2
- Rijndael
- TripleDES

AsymmetricAlgorithm

- abstract
- Ősosztály
- Aszimmetrikus titkosítások
- Algoritmus leírás
- CryptoProvider

Leszármazottak

- DSA
- ECDiffieHellman
- ECDsa
- RSA

ICryptoTransform, ICspAsymmetricAlgorithm

ICryptoTransform

- Interface
- Egyszerű műveletek
- TransformBlock

Megkapható

- CreateEncryptor
- CreateDecryptor

ICspAsymmetricAlgorithm

- Crypto Service Provider
- Interface
- Hozzáférés a kulcs konténerhez
- Kulcsok
  - Importálása
  - Exportálása

CryptoStream

- Titkosított adatfolyam
- Stream

```
CryptoStream
    cryptoStream =
new CryptoStream(
stream,
transform,
mode);
```

Paraméterek

- Stream
- ICryptoTransform
- CryptoStreamMode
  - enumeráció
    - ★ Read
    - ★ Write

HashAlgorithm

- Őszosztály
- abstract
- Minden Hash algoritmus őse
- ComputeHash(Byte[])
- ComputeHash(Stream)
- A hash érték numerikus reprezentációja az adatnak!

Leszármazottak

- KeyedHashAlgorithm
- MD5
- RIPEMD160
- SHA1
- SHA256
- SHA384
- SHA512

Hash Algoritmusok Használata

1. A Feladó és a Fogadó közös Hash függvényt használnak.
2. A Feladó küldi az Üzenetet és a Hash kódot a Fogadónak.
3. A Fogaód kiszámolja a fogadott Üzenet Hash értékét.
4. Ha megegyezik a számított és a kapott érték akkor
   - az Üzenet eredeti.

1. Feladó küldi az Üzenetet titkosítás nélkül és a Hash értéket titkosítva.
2. A Fogadó kiszámolja az Üzenet Hash értékét és összehasonlítja a kapott Hash–sel.
3. Ha megegyeznek akkor tudja, hogy:
   - Az Üzenet nem változott.
   - A Feladótól jött az Üzenet.