Medii vizuale de programare
Curs 10

Conf. dr.ing. GENGE Béla

Universitatea "Petru Maior", Departamentul de Informatica
Tîrgu Mureș, Romania
bela.genge@ing.upm.ro

Servicii criptografice în .NET

- În .NET serviciile criptografice sunt disponibile în cadrul spațiului de nume `System.Security.Cryptography`.
- Ce ne oferă .NET:
  - Criptografie simetrică.
  - Criptografie asimetrică.
  - Funcții hash și MAC.
  - Generatoare de numere aleatoare.
  - Semnături digitale.

Managed vs unmanaged cryptography

- Criptografia nativă (Win32/Win64) Windows este implementată în bibliotecile CryptoAPI - unmanaged code, din afara CLR. Main nou (după Windows Vista) în Cryptographic API Next Generation (CNG)
- .NET extinde CNG cu implementări managed pentru a face aplicațiile portabile.
- Dacă o aplicație .NET utilizează CNG, aceasta trebuie să fie disponibilă pe stațiile destinație.

Federal Information Processing Standards (FIPS)

- Nu doar algoritmii, dar și implementările algoritmilor criptografici trebuie utilizate cu grijă.
- În România: ORDIN nr. M.125 din 18 octombrie 2012 pentru stabilirea unor masuri in domeniul activitatii oficiale de criptologie
  - `www.monitoruljuridic.ro/act/ordin-nr-m-125-din-18-octombrie-2012-pentru-stabilirea-uno html`
- Federal Information Processing Standards - FIPS publication 140-2 este un standard al SUA pentru acreditarea modulelor criptografice.
- FIPS 140-2 definește cerințele și standardele pentru module criptografice.
- FIPS 140-2 definește 4 nivele crescătoare de securitate.

Nivele FIPS 140-2

- Nivelul 1: cerințe de bază de criptografie, cum ar fi existența a cel puțin un algoritm criptografic aprobat. Nu există cerințe pentru securitatea fizică a sistemului. Target: stații personale (PC).
- Nivelul 2: îmbunătățește nivelul 1 cu mecanisme pentru prevenirea accesului la parametrii critici de securitate (Critical Security Parameters - CSP) în scopul manipulării fizice. Exemple: carcase protectoare, sigilii, încuietori, etc.
- Nivelul 3: îmbunătățește nivelul 2 cu mecanisme care detectează tentativele de acces la CSP și reacționează corespunzător. Exemplu: circuit închis care șterge CSP, circuit care mută o mașină virtuală.
- Nivelul 4: cel mai înalt nivel de securitate. Se presupune că modulul e utilizat într-un mediu neprotejat. Modulul trebuie să monitorizeze o serie de parametrii proprii, e.g., nivele de tensiune, temperatură, pentru a detecta atacuri și trebuie să reacționeze pentru protejarea CSP. Modulele trec printr-o serie de teste pentru a verifica funcționarea în afara parametrilor normali.

Windows și FIPS 140-2

- Aplicațiile trebuie să demonstreze că respectă standarde de securitate. MS sprijină aplicațiile Windows cu module CNG validate FIPS 140-2:
  - Pe Windows Vista și Windows Server 2008: BCRYPT.DLL (user space), KSECDD.SYS (kernel space).
  - Pe Windows 7,8,10 și Windows Server 2008 R2: BCRYPTPRIMITIVES.DLL (user space), CNG.SYS (kernel space).
- Dovada certificării modulelor: `https://technet.microsoft.com/en-us/library/cc750357.aspx`

1329	Microsoft Corporation One Microsoft Way Redmond, WA 98052-6399 USA Tim Myers TEL: 800-MICROSOFT	**Microsoft Windows 7 Cryptographic Primitives Library (bcryptprimitives.dll)** (Software Version: 6.1.7600.16385 or 6.1.7601.17514) *(When operated in FIPS mode with Windows 7 Code Integrity (ci.dll) validated to FIPS 140-2 under Cert. #1327 operating in FIPSmode)*
1328	Microsoft Corporation One Microsoft Way Redmond, WA 98052-6399 USA Tim Myers TEL: 800-MICROSOFT	**Microsoft Windows 7 Kernel Mode Cryptographic Primitives Library (cng.sys)** (Software Versions: 6.1.7600.16385, 6.1.7600.16915, 6.1.7600.21092, 6.1.7601.17514, 6.1.7601.17919, 6.1.7601.17725, 6.1.7601.21861 and 6.1.7601.22076) *(When operated in FIPS mode with Windows 7 Winload OS Loader (winload.exe) validated to FIPS 140-2 under Cert. #1326 operating in FIPS mode)*

.NET și FIPS 140-2

- .NET permite accesarea CNG, adică a criptografiei certificate FIPS 140-2 utilizând API-ul `unmanaged` (nativ).
  - Dezavantajul principal îl reprezintă dificultatea portării aplicațiilor pe platforme unde CNG nu e disponibil (e.g., Linux).
- .NET include și clase criptografice `managed`, dar acestea nu sunt certificate FIPS 140-2.
- API-ul `unmanaged` se distinge prin denumirea claselor ce include `CryptoServiceProvider` sau `CNG`.
- API-ul `managed` se distinge prin denumirea claselor ce includ `Managed`.

.NET și FIPS 140-2

- Atenție! NU toate clasele `unmanaged` sunt certificate FIPS 140-2. Certificarea se oferă pentru anumiți algoritmi și anumite configurații (identificate explicit în certificat).
- Pentru a activa per-sistem criptografia certificată FIPS 140-2:
  - Se lansează `gpedit.msc` cu drepturi de administrator. În **Local Group Policy Editor**, se selectează **Computer Configuration**, urmat de **Windows Settings** și **Security Settings**. În **Security Settings** se selectează **Local Policies** și **Security Options**. Se activează opțiunea: **System cryptography: Use FIPS-compliant algorithms for encryption, hashing, and signing**.
  - Sau din regiștrii: `HKLM/System/CurrentControlSet/Control/Lsa/FIPSAlgorithmPoli` `to 1`
  - Atenție! Activarea opțiunii poate avea repercursiuni grave asupra anumitor aplicații (de ex. SSL 2.0 și 3.0 nu sunt certificate FIPS 140-2).

.NET și FIPS 140-2

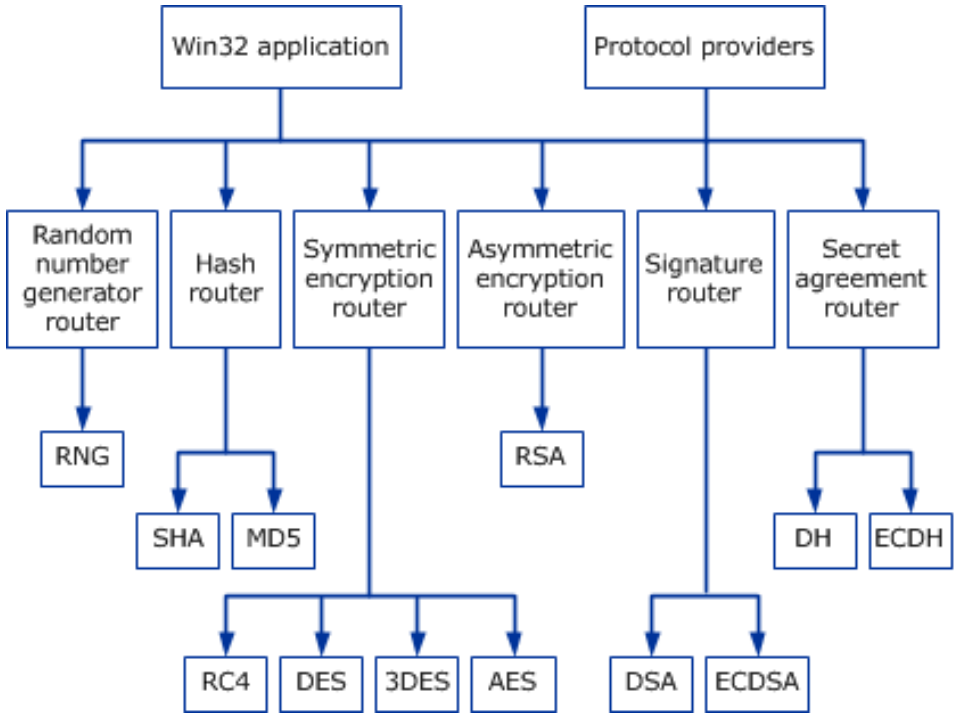
- Pentru a activa/dezactiva per-aplicație criptografia certificată FIPS 140-2 se folosește fișierul de configurare a MS Build (`msbuild.exe.config`).

```
<configuration>
    <runtime>
        <enforceFIPSPolicy enabled="false"/>
    </runtime>
</configuration>
```

- Această configurare supra-definește setările sistem.
- Verificarea utilizării FIPS 140-2 se poate face din aplicație prin valoarea variabilei `CryptoConfig.AllowOnlyFipsAlgorithms` (`True/False`).
- Atenție! Dacă CNG FIPS 140-2 e activat și se instanțiază un algoritm ne-certificat (i.e., managed), se va genera o excepție `InvalidOperationException`!

CNG

- Modul de funcționare a CNG (sursa: MSDN):
  - Fiecare algoritm criptografic are propriul set de funcții și propriul API.
  - Pentru a facilita utilizarea unitară a mai multor algoritmi aplicațiile interacționează cu 'routere'. Acestea expun aceeași interfață pentru o clasă de algoritmi.
  - Routerele sunt responsabile pentru apelul metodelor corespunzătoare fiecărui algoritm în parte.

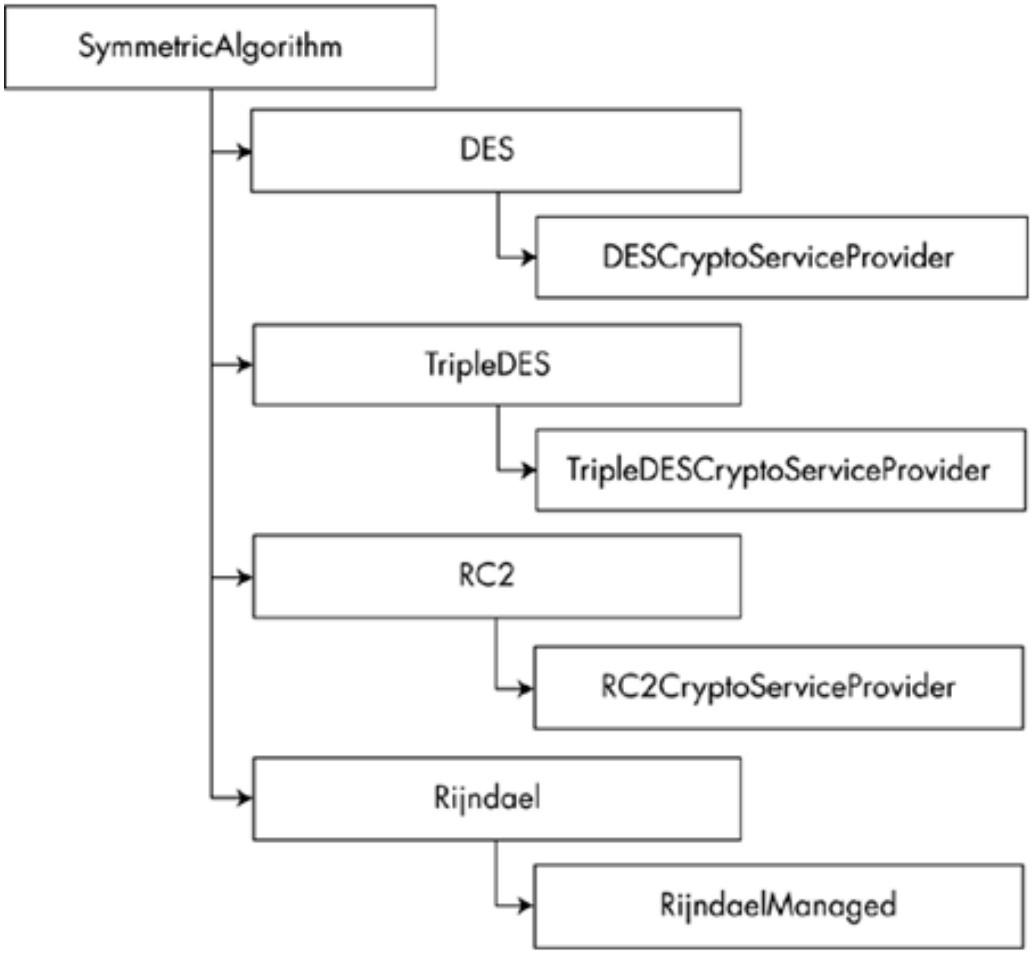


Spațiul de nume System.Security.Cryptography

- Conține clasele `managed` și `unmanaged` pentru implementarea aplicațiilor criptografice.
- Vizualizare clase din MSDN: `https://msdn.microsoft.com/en-us/library/system.security.cryptography(v=vs.110).aspx`

Criptografia simetrică în .NET

- Clasa de bază: `SymmetricAlgorithm`.
- Ierarhia de clase (parțială):



Proprietăți și metode ale clasei `SymmetricAlgorithm`

- Proprietăți (get/set pentru câmpurile clasei):
  - `BlockSize`.
  - `FeedbackSize`.
  - `IV`.
  - `Key`.
  - `KeySize`.
  - `LegalBlockSizes`.
  - `LegalKeySizes`.
  - `Mode`.
  - `Padding`.

Câmpuri și metode ale clasei `SymmetricAlgorithm`

- Metode principale:
  - `Clear()`: eliberarea tuturor resurselor.
  - `CreateDecryptor()`: crează un obiect decriptor cu cheia și IV curente.
  - `CreateDecryptor(Byte[], Byte[])`: crează un obiect decriptor cu cheia și IV date ca parametru.
  - `CreateEncryptor()`: crează un obiect encriptor cu cheia și IV curente.
  - `CreateEncryptor(Byte[], Byte[])`: crează un obiect encriptor cu cheia și IV date ca parametru.
  - `GenerateKey()`: generează o nouă cheie.
  - `GenerateIV()`: generează un nou IV.

Criptarea

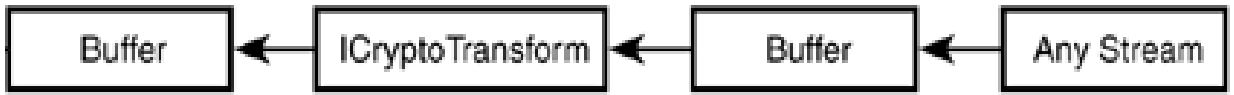
- Alegerea algoritmului. AES (standardul utilizat astăzi), DES, RC2, Rijndael, TripleDES.
- Diferența dintre AES și Rijndael:
  - Rijndael: propunerea pentru standardizare, mai flexibilă, cu mai multe opțiuni.
  - AES: versiunea Rijndael standardizată - utilizată în practică.
- Două clase disponibile pentru AES (ambele moștenesc clasa `System.Security.Cryptography.Aes`):
  - `AesCryptoServiceProvider`.
  - `AesManaged`.

Clasa `AesCryptoServiceProvider`

- Un singur constructor, fără parametrii: `AesCryptoServiceProvider()`.
- Implementează metodele clasei `SymmetricAlgorithm` descrise anterior.
- Aspecte importante ce trebuie luate în calcul înainte de criptare:
  - Dimensiunea cheii.
  - Generarea cheii și IV.
  - Modul de cifrare.
- Implicit, la instanțierea clasei se generează un nou IV și o nouă cheie pe 256 biți.
- Implicit, modul de cifrare este CBC.
- Criptarea/decriptarea necesită crearea de obiecte specifice prin apelul: `CreateEncryptor()`, `CreateDecryptor()`.

Clasa `CryptoStream`

- Criptarea/decriptarea se realizează pe un flux de date (octeți).
- Clasa `CryptoStream` leagă un obiect flux (FileStream, MemoryStream) de un obiect criptografic.
- Constructorul primește 3 parametrii:
  - Obiect stream destinație.
  - Obiect criptografic (criptare/decriptare).
  - Operația dorită: Read/Write.



Exemplu criptare fișier

Exemplu

```
FileStream fin = new FileStream("D:/source.pdf",
    FileMode.Open, FileAccess.Read);
FileStream fout = new FileStream("D:/dest.enc",
    FileMode.OpenOrCreate, FileAccess.Write);
AesCryptoServiceProvider cryptoProvider = new
    AesCryptoServiceProvider();
ICryptoTransform encryptor =
    cryptoProvider.CreateEncryptor();
CryptoStream stream = new CryptoStream(fout, encryptor,
    CryptoStreamMode.Write);
byte[] input = new byte[128];
int inLen = -1;
while ((inLen = fin.Read(input, 0, 128)) > 0) {
    stream.Write(input, 0, inLen);}
stream.Close(); fout.Close(); fin.Close();
```

Exemplu decriptare fișier

Exemplu

```
FileStream fin = new FileStream("D:/dest.enc",
    FileMode.Open, FileAccess.Read);
FileStream fout = new FileStream("D:/source.dec.pdf",
    FileMode.OpenOrCreate, FileAccess.Write);
AesCryptoServiceProvider cryptoProvider = new
    AesCryptoServiceProvider();
ICryptoTransform decryptor =
    cryptoProvider.CreateDecryptor(encKey, encIV);
CryptoStream stream = new CryptoStream(fout, decryptor,
    CryptoStreamMode.Write);
byte[] input = new byte[128];
int inLen = -1;
while ((inLen = fin.Read(input, 0, 128)) > 0) {
    stream.Write(input, 0, inLen);}
stream.Close(); fout.Close(); fin.Close();
```

Exemplu criptare string

Exemplu

```
string s = "String exemplu";
byte[] benc = null;
AesCryptoServiceProvider cryptoProvider = new
    AesCryptoServiceProvider();
ICryptoTransform encryptor =
    cryptoProvider.CreateEncryptor();
MemoryStream memStream = new MemoryStream();
CryptoStream cryptStream = new CryptoStream(memStream,
    encryptor, CryptoStreamMode.Write);
StreamWriter wrStream = new StreamWriter(cryptStream);
wrStream.Write(s);
wrStream.Close();
cryptStream.Close();
memStream.Close();
benc = memStream.ToArray();
```

Exemplu decriptare string

Exemplu

```
byte[] bdec = null;
AesCryptoServiceProvider cryptoProvider = new
    AesCryptoServiceProvider();
ICryptoTransform decryptor =
    cryptoProvider.CreateDecryptor(encKey, encIV);
MemoryStream memStream = new MemoryStream();
CryptoStream cryptStream = new CryptoStream(memStream,
    decryptor, CryptoStreamMode.Write);
cryptStream.Write(encData, 0, encData.Length);
cryptStream.Close();
memStream.Close();
bdec = memStream.ToArray();
MessageBox.Show("Dec: " +
    System.Text.Encoding.UTF8.GetString(bdec));
```

Alegerea unui alt algoritm

- Cunoscând pașii anteriori, utilizarea unui alt algoritm criptografic e deosebit de intuitiv.
- Singura modificare este în clasa instanțiată.
- De exemplu, dacă în loc de AES se dorește utilizarea 3DES, atunci se va instanția clasa `TripleDESCryptoServiceProvider` în locul `AesCryptoServiceProvider`.

FIPS 140-2 și alte biblioteci criptografice

- OpenSSL, cel mai răspândit modul criptografic, nu este certificat FIPS 140-2. Motivele (conform `https://www.openssl.org/docs/fipsnotes.html`):
  - Costuri ridicate: 50.000USD pentru o validare de bază. O singură modificare a sursei va necesita o re-certificare.
  - Timpii de certificare: durată lungă între 6 și 12 luni.
- În schimb, OpenSSL pune la dispoziție OpenSSL FIPS Object Module 2.0, o versiune diferită de biblioteca standard OpenSSL, ce conține algoritmi și implementare certificate FIPS 140-2.