Алгоритмы шифрования и их применение в .Net приложениях для защиты данных

Radislav Kerimhanov
rkerimhanov@codemastersintl.com

Содержание

- Что такое защита информации?
- Уязвимые места Web приложений
- Виды атак на Web приложения
- Алгоритмы шифрования и их сравнение
- Шифрование в .Net
- Примеры
- Шифрование данных в БД
- Шифрование строк подключения к БД
- Пример шифрования строк подключения к БД
- Заключение

Что такое защита информации?

- Информация – это любые сведения, которые интересуют конкретного человека в конкретной ситуации.
- Защита информации - совокупность мероприятий, методов и средств, обеспечивающих:
  - исключение НСД к ресурсам ЭВМ, программам и данным;
  - проверку целостности информации;
  - исключение несанкционированного использования программ (защита программ от копирования).

Уязвимые места Web приложений

- URL
- Поля ввода данных
- HTTP и HTTPS

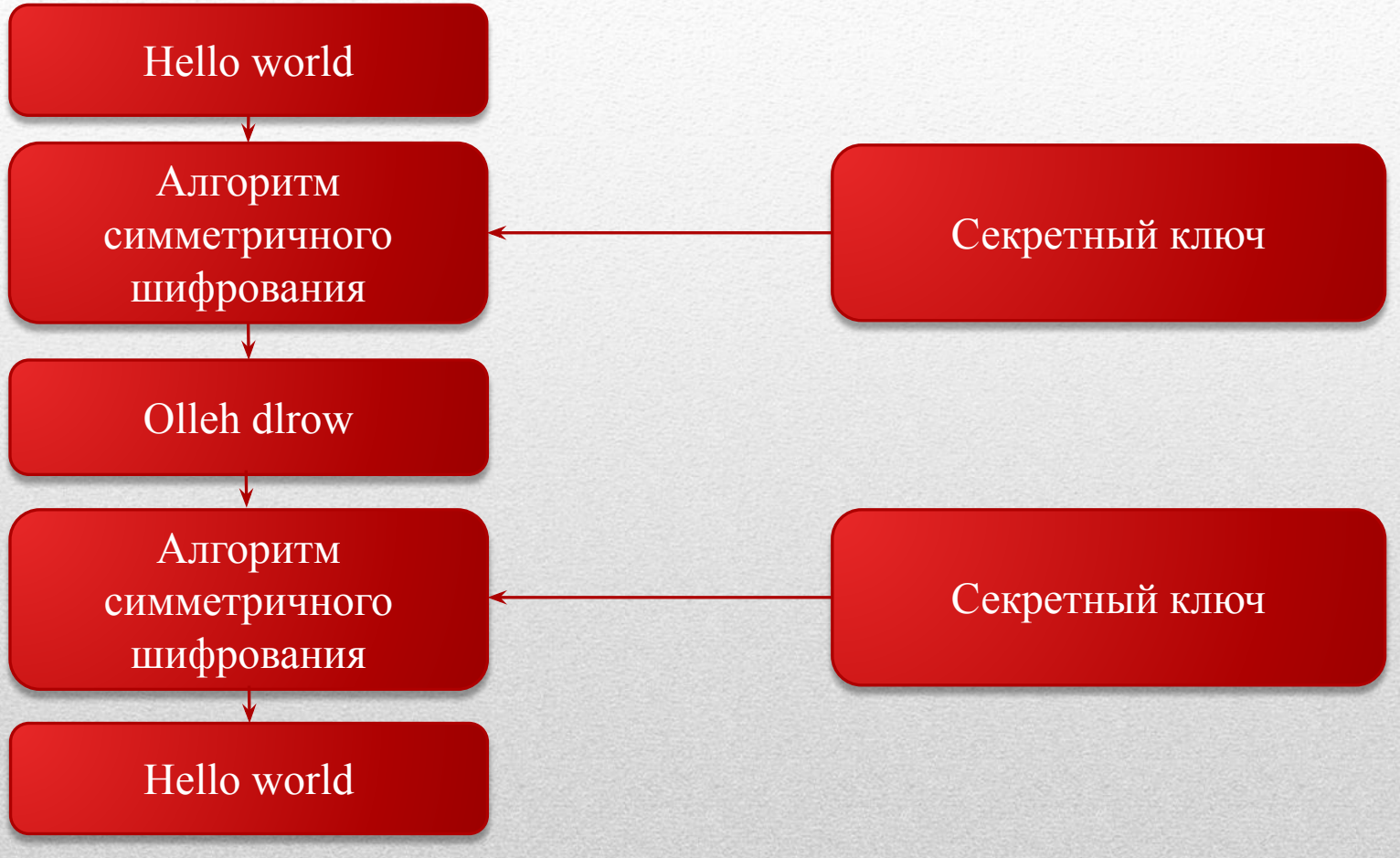
Виды атак на Web приложения

- **Подбор (Brute Force)**
- **Недостаточная аутентификация (Insufficient Authentication)**
- **Небезопасное восстановление паролей (Weak Password Recovery Validation)**
- **Предсказуемое значение идентификатора сессии (Credential/Session Prediction)**
- **Подмена содержимого (Content Spoofing)**
- **Межсайтовое выполнение сценариев (Cross-site Scripting, XSS)**
- **Расщепление HTTP-запроса (HTTP Response Splitting)**
- **Атака на функции форматирования строк (Format String Attack)**
- **Внедрение операторов SQL (SQL Injection)**

Алгоритмы шифрования

- **Симметричные алгоритмы шифрования (SymmetricAlgorithm)**
  - DES
  - TripleDES
  - Rijndael
  - RC2
- **Ассиметричные алгоритмы шифрования (AsymmetricAlgorithm)**
  - RSA
  - DSA
- **Хэш алгоритмы шифрования (HashAlgorithm)**
  - MD5
  - SHA1
  - SHA256
  - SHA512

Симметричные алгоритмы



Асимметричные алгоритмы

Шифрование в .Net

Абстрактный алгоритм	Реализация по умолчанию	Длина ключа	Максимальная длина ключа
DES	DESCryptoServiceProvider	64	64
TripleDES	TripleDESCryptoServiceProvider	128,192	192
RC2	RC2CryptServiceProvider	40-128	128
Rijndael	RijndaelManaged	128,192,256	256
RSA	RSACryptoServiceProvider	384–16384 (с увеличением на 8 бит)	1024
DSA	DSACryptoServiceProvider	512–1024 (с увеличением на 64 бита)	1024

Шифрование данных в БД

Шифрование с помощью хэш алгоритма	
Логин	Пароль
JoePresident	AEB0FC9FCEA137CF9BBC594BBC97991C10CD9138
MaryDeveloper	DEFAFC9FCEA137C12345694BBC97991C10123456
TomNewHire	AEB0FC9FCEA137CF9BBC594BBC97991C10CD9138

Шифрование с помощью хэш алгоритма + SALT		
Логин	Пароль	Salt
JoePresident	876ABD9FCEA137CF9BBC594 BBC97991C10CD9138	5d8a2052-1f3e-4fe1-9ca 4-7c891a980592
MaryDeveloper	ADCAD79FCEA137C12345694B BC97991C10987650	38011b98-3ce5-4f62-a95 9-415ce93ed7bb
TomNewHire	BC89129FCEA137CF9BBC594B BC97991C10BDCA09	fd05e0fe-d17c-4d6c-8a5 5-06445bb5613c

Шифрование строк подключения к БД

```
<system.webServer>
  <validation validateIntegratedModeConfiguration="false" />
  <modules runAllManagedModulesForAllRequests="true" />
</system.webServer>
<runtime>
  <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
    <dependentAssembly>
      <assemblyIdentity name="System.Web.Mvc" publicKeyToken="31bf3856ad364e35" />
      <bindingRedirect oldVersion="1.0.0.0-2.0.0.0" newVersion="3.0.0.0" />
    </dependentAssembly>
  </assemblyBinding>
</runtime>
<connectionStrings>
  <add name="TestConnectionString" connectionString="Data Source=.\;Initial Catalog=TulaDev; UserID=User; Password=Password;"
    providerName="System.Data.EntityClient" />
</connectionStrings>
```

```
<connectionStrings configProtectionProvider="RsaProtectedConfigurationProvider">
  <EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element"
    xmlns="http://www.w3.org/2001/04/xmlenc#">
    <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#tripledes-cbc" />
    <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
      <EncryptedKey xmlns="http://www.w3.org/2001/04/xmlenc#">
        <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5" />
        <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
          <KeyName>Rsa Key</KeyName>
        </KeyInfo>
        <CipherData>
          <CipherValue>EeKubIRZLuxfGqMQpG7Ou41BGnphHJfKnXfeCs38qdshesyo7RMZYgNGWHaTd/77v9lcqdhvg7VYPMJ0I1mN+tDzgFdUiiZzvtbTWiOurGoevFiOyGjFFnJXFnZfYWGy7h5+PmCC2RjHFfPZvC4E1wkarlfuNDjrl00d2gqo6G4=</CipherValue>
        </CipherData>
      </EncryptedKey>
    </KeyInfo>
    <CipherData>
      <CipherValue>s2Klniq1fC5RsNoQwRIbIbeZfJWtXZNZyUHQl8WZNAgFitK4fgHsis6DmTbW7umdQHat3TFKEoAnGyIheB0fi62tbW+gWo+89tIPYqHVVWVqdo83mTZYvpbaj6t4GRsmPiOCb2H9HlGOpzblG2whq4eNFojSpWPDiAOnG2CZ6sJuIC0iUHR02F3wbYQf
        oW+tegH0VoDt1PYHh67XZkqjt/SRc4JUmX7xepkcOnCqPV0=</CipherValue>
      </CipherData>
    </EncryptedData>
  </connectionStrings>
</configuration>
```

Заключение

- Достоинства симметричных алгоритмов по сравнению с асимметричными:
  - скорость (по данным Applied Cryptography — на 3 порядка выше)
  - простота реализации (за счёт более простых операций)
  - меньшая требуемая длина ключа
- Недостатки:
  - сложность управления ключами в большой сети
  - сложность обмена ключами

«Тот, кто владеет информацией, владеет миром»