Filip Paul

Počítačové a komunikační sítě – MKC-PKS-21-22: Samostatná práce č. 5

Šifrovací metody, integrita, certifikace, SSL

1. Vysvětlete princip symetrického šifrování a šifrování s veřejným klíčem. (1b)

Při symetrickém šifrování znají obě strany, které mezi sebou komunikují jeden tajný klíč K, kterým šifrují i dešifrují zprávy. Bezpečnost tak stojí na utajení klíče K.

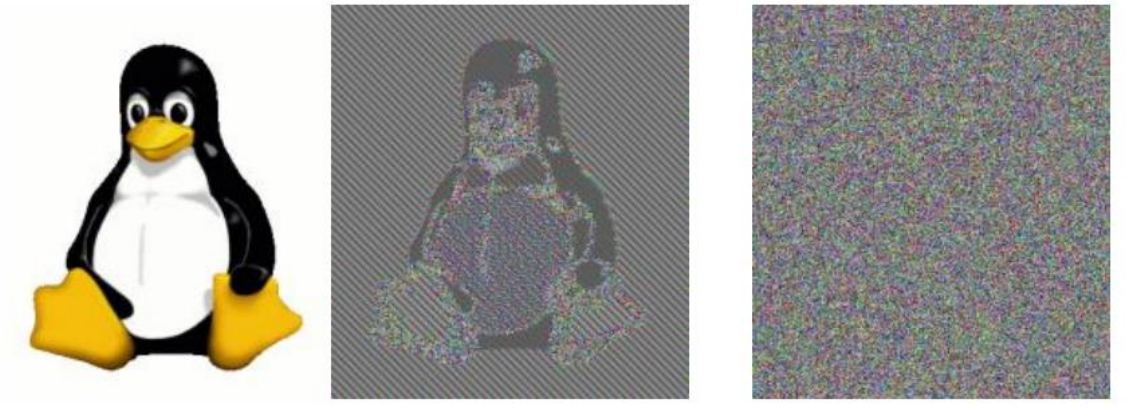
Při šifrování s veřejným klíčem má každá strana svůj soukromý a veřejný klíč.

Když například Alice chce poslat Bobovi zprávu, tak nejprve použije veřejný klíč Boba (ten znají všichni) k šifrování zprávy. Následně Bob dešifruje obdrženou svým soukromým klíčem (ten zná jen Bob).

Bezpečnost systému tak stojí na utajování soukromých klíčů a na „víře“, že nelze odvodit tyto soukromé klíče ze znalosti klíčů veřejných popř veřejného klíče i přenášené zprávy.

2. Jaký problém řeší mód CBC (Cipher Block Chaining) u blokových šifer ve srovnání s módem ECB (Electronic Code Book)? (1b)

Tady si půjčím obrázek z přednášky…



Problémem ECB šifrování je že, že nemá paměť a tak vždy stejný vstup převede na stejný výstup, proto je například z obrázku patrný obsah obrázku, protože obsahuje velké plochy stejných hodnot.

CBC šifrování tento problém řeší tím, že každý šifrovaný blok je závislý na šifrování předchozího bloku. Tím přidává do systému paměť a je tak schopný lépe šifrovat data.

3. Co se rozumí pod pojmem „bezkoliznost“ hašovací funkce? (1b)

Hashovací funkce je bezkolizní, pokud nelze vygenerovat 2 zprávy (nějak systematicky), tak aby měli po průchodu hashovací funkcí stejný otisk.

4. Pomocí programu Wireshark analyzujte komunikaci protokolem https vašeho prohlížeče při otevření stránky www.vutbr.cz. Z hlaviček protokolu TLS určete, jaké šifrování zvolil server, a dále určete, jakou autoritou je podepsán certifikát serveru a jaký algoritmus pro kryptografický otisk používá. (2b)

Nejsem si úplně jistý, zda vyčítám správný výpis z Wiresharku, protože když si zobrazím certifikát v prohlížeči dostanu jiné informace. Například, že vystavitelem certifikátu je GEANT OV RSA CA4. Kde vystavitelem ve Wireshark je pravděpodobně AzureCA06 OSCP CERT??

Výpis z Wireshark:

```
Transport Layer Security
  TLSv1.2 Record Layer: Handshake Protocol: Multiple Handshake Messages
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 7100
    > Handshake Protocol: Server Hello
    > Handshake Protocol: Certificate
    v Handshake Protocol: Certificate Status
        Handshake Type: Certificate Status (22)
        Length: 1745
        Certificate Status Type: OCSP (1)
        OCSP Response Length: 1741
      v OCSP Response
          responseStatus: successful (0)
        v responseBytes
            ResponseType Id: 1.3.6.1.5.5.7.48.1.1 (id-pkix-ocsp-basic)
          v BasicOCSPResponse
            > tbsResponseData
            > signatureAlgorithm (sha1WithRSAEncryption)
              Padding: 0
              signature: 7dce9d4b6d86356a418ec7ebece121403d76c3bff10d4bd1346f70207d1cde7407c51346…
            v certs: 1 item
              v Certificate (id-at-commonName=AzureCA06 OSCP Cert)
                v signedCertificate
                    version: v3 (2)
                    serialNumber: 0x330037e9883abc50925e0738bb00000037e988
                  v signature (sha384WithRSAEncryption)
                      Algorithm Id: 1.2.840.113549.1.1.12 (sha384WithRSAEncryption)
                  > issuer: rdnSequence (0)
                  v validity
                    v notBefore: utcTime (0)
                        utcTime: 2022-04-22 20:54:51 (UTC)
                    v notAfter: utcTime (0)
                        utcTime: 2022-04-29 20:54:51 (UTC)
                  v subject: rdnSequence (0)
                    > rdnSequence: 1 item (id-at-commonName=AzureCA06 OSCP Cert)
                  v subjectPublicKeyInfo
                    v algorithm (rsaEncryption)
                        Algorithm Id: 1.2.840.113549.1.1.1 (rsaEncryption)
                    v subjectPublicKey: 3082010a0282010100924a65480a4c9ee42b7be33386d01cc4797cdcb34527cf4c2c992e…
                        modulus: 0x00924a65480a4c9ee42b7be33386d01cc4797cdcb34527cf4c2c992ed37eab9aeab325fa…
                        publicExponent: 65537
                  > extensions: 8 items
                v algorithmIdentifier (sha384WithRSAEncryption)
                    Algorithm Id: 1.2.840.113549.1.1.12 (sha384WithRSAEncryption)
                Padding: 0
                encrypted: 443327aee68a939d4d66e7b3c8143a2e83a51919940b7796bf0daa18e1b45db4922bf9ff…
    v Handshake Protocol: Server Key Exchange
        Handshake Type: Server Key Exchange (12)
        Length: 361
      v EC Diffie-Hellman Server Params
          Curve Type: named_curve (0x03)
          Named Curve: secp384r1 (0x0018)
          Pubkey Length: 97
          Pubkey: 042ec463ffa599a82cbcaa964ef48615dae7d46751edfccda6f25c42c3ca10e1af1c1338…
        v Signature Algorithm: rsa_pkcs1_sha256 (0x0401)
            Signature Hash Algorithm Hash: SHA256 (4)
            Signature Hash Algorithm Signature: RSA (1)
          Signature Length: 256
          Signature: a056a8de4d4678b6aafca238ee4d8a8108b0b3210e20ecfcaf3fcbebcc32acf7d25e060e…
    > Handshake Protocol: Server Hello Done
```

Výpis z prohlížeče: