Chapters 13

Digital Signatures

Dr. Shin-Ming Cheng

Content of this Chapter

- The principle of digital signatures
- Security services
- The RSA digital signature scheme
- The Digital Signature Algorithm (DSA)

Motivation

- Alice orders a pink car from the car salesmen Bob
- After seeing the pink car, Alice states that she has never ordered it
- How can Bob prove towards a judge that Alice has ordered a pink car? (And that he did not fabricate the order himself)
  - Symmetric cryptography fails because both Alice and Bob can be malicious
  - Can be achieved with public-key cryptography

Motivation

- Diffie-Hellman on its own is not enough
  - For example how does Alice know who she is agreeing a key with? Is it Bob or Eve?
- One way around is for
  - Alice to sign her message to Bob
  - Bob to sign his message to Alice
  - In this way both parties know who they are talking to
- Signature: An important concept of public key cryptography
  - Also were invented by Diffie and Hellman
  - But the first practical system was due to Rivest, Shamir and Adleman (RSA)

Public Key Encryption

- Recap: Public Key Encryption
  - Recall the basic idea
    - Message + Alice's Public Key = CipherText
    - CipherText + Alice's Private Key = Message
  - Hence anyone with Alice's public key can send Alice a secret message
  - Only Alice can decrypt the message
    - Since only Alice has the private key
  - All they need to do is to look up Alice's public key in some directory

Public Key Signature

- Public Key Signature
  - To obtain signatures, we swap things around
  - The basic idea is
    - Message + Alice's Private Key = Signature
    - Message + Signature + Alice's Public Key = YES / NO
  - Alice signs a message, which can only come from her
    - Since only Alice has access to the private key
  - Anyone can verify Alice's signature, since everyone can have her Public Key

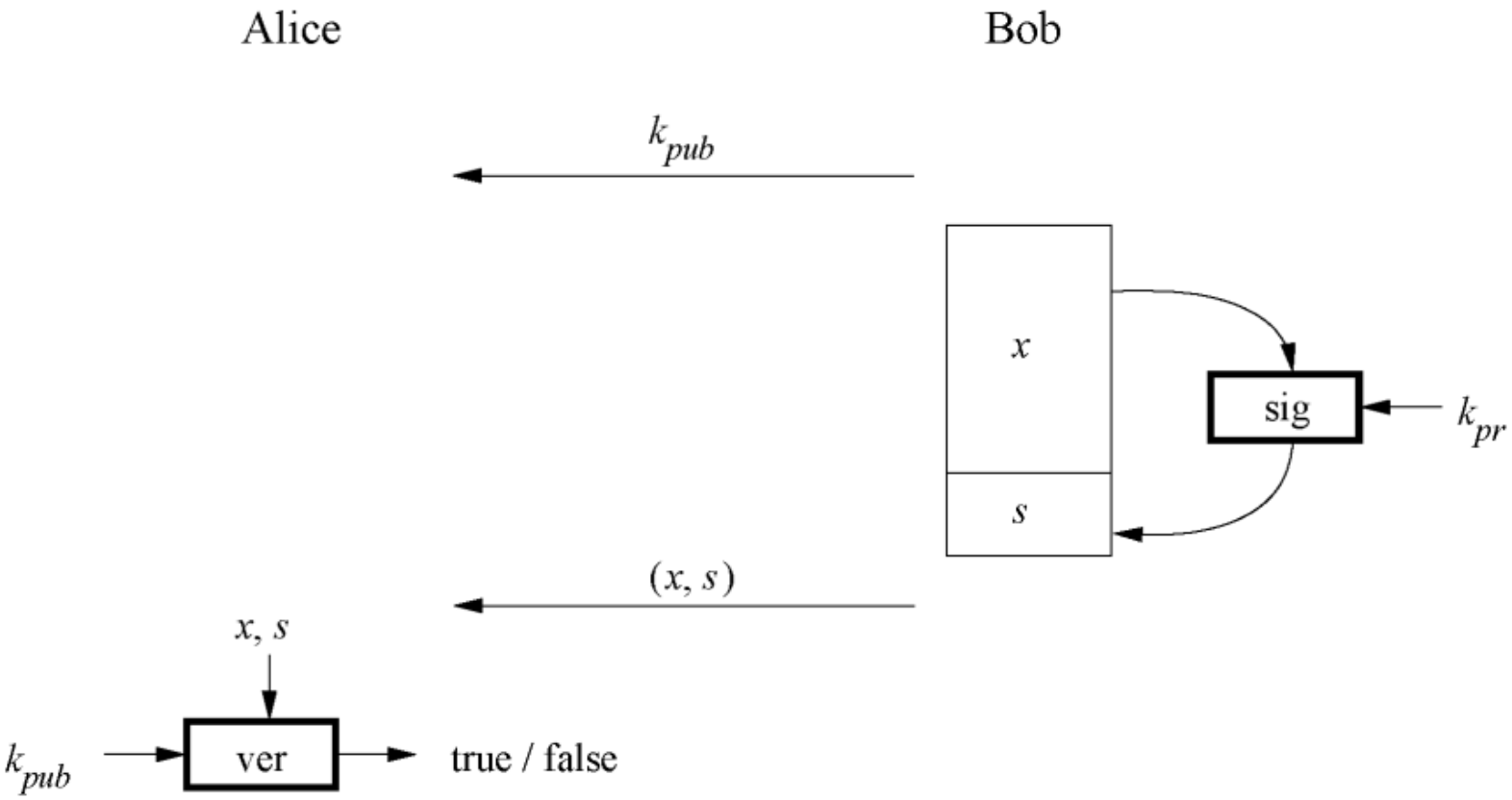
Public Key Signature

- Another variant is a signature scheme with message recovery
  - The message is recovered from a signature
- The basic idea is
  - Message + Alice's Private Key = Signature
  - Signature + Alice's Public Key = Message / INVALID

Authentic Public Key

- The main problem is how the public keys to be trusted?
- How to know a certain key is associated to a given entity?
  - You may think a public key belongs to Alice, but it belongs to Eve
  - Then Eve can sign checks, and you will think they come from Alice
- We seem to have the same key management problem as in symmetric cryptosystems
  - We have replaced the problem of
    - Confidential distribution of secret keys
  - with the problem of
    - Authentic distribution of public keys

Basic Principle of Digital Signatures



Main idea

- For a given message x, a digital signature is appended to the message (just like a conventional signature)
- Only the person with the private key should be able to generate the signature
- The signature must change for every document
  - The signature is realized as a function with the message x and the private key as input
  - The public key and the message x are the inputs to the verification function

Core Security Services

The objectives of a security systems are called security services.

- **Confidentiality**: Information is kept secret from all but authorized parties.
- **Integrity**: Ensures that a message has not been modified in transit.
- **Message Authentication**: Ensures that the sender of a message is authentic. An alternative term is data origin authentication.
- **Non-repudiation**: Ensures that the sender of a message can not deny the creation of the message.
  - order of a pink car

Additional Security Services

- **Identification/entity authentication:** Establishing and verification of the identity of an entity,
  - A person, a computer, or a credit card.
- **Access control:** Restricting access to the resources to privileged entities.
- **Availability:** The electronic system is reliably available.
- **Auditing:** Provides evidences about security relevant activities
  - by keeping logs about certain events.
- **Physical security:** Providing protection against physical tampering and/or responses to physical tampering attempts.
- **Anonymity:** Providing protection against discovery and misuse of identity.

Main Idea of the RSA Signature Scheme

- To generate the private and public key:
  - Use the same key generation as RSA encryption.
- To generate the signature:
  - "encrypt" the message x with the private key

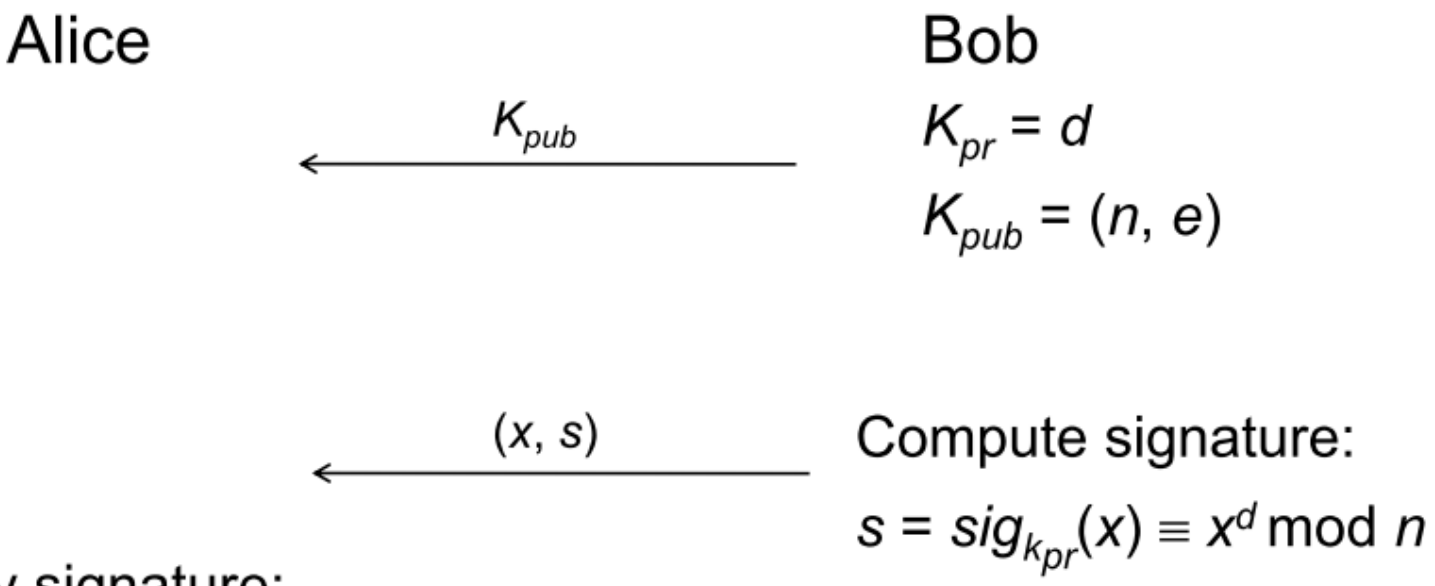
$$s = sig_{Kpr}(x) = x^d \mod n$$

  - Append s to message x
- To verify the signature:
  - "decrypt" the signature s with the public key

$$x' = ver_{Kpub}(s) = s^e \mod n$$

  - If $x = x'$, the signature is valid

The RSA Signature Protocol



Security and Performance of the RSA Signature Scheme

- Security:
  - The same constrains as RSA encryption: n needs to be at least 1024 bits to provide a security level of 80 bit.
  - The signature, consisting of s, needs to be at least 1024 bits long.
- Performance:
  - The signing process is an exponentiation with the private key and the verification process an exponentiation with the public key e.
  - Signature verification is very efficient as a small number can be chosen for the public key.

Existential Forgery Attack against RSA Digital Signature

Alice Oscar Bob

$K_{pr} = d$

$K_{pub} = (n, e)$

(n, e) ← Bob

(n, e) ← Oscar

1. Choose signature:
   $s \in \mathbf{Z}_n$
2. Compute message:
   $x \equiv s^e \bmod n$

(x, s) ←

Verification:

$s^e \equiv x' \bmod n$

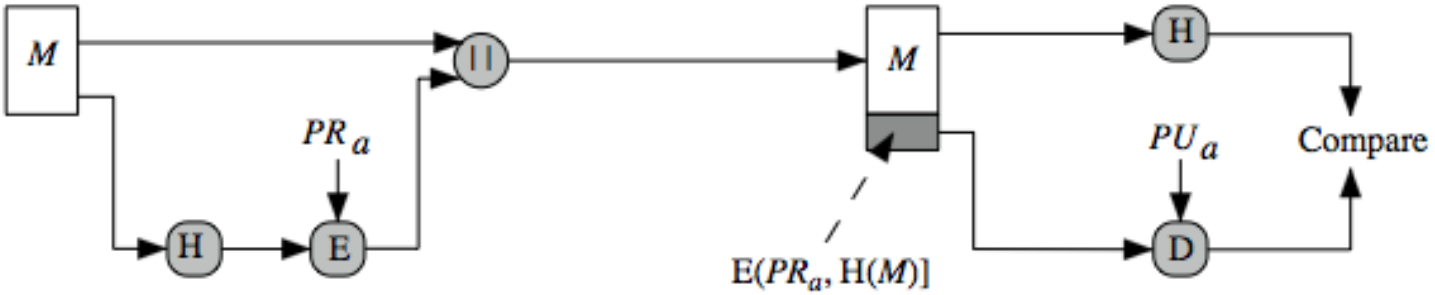
since $s^e = (x^d)^e \equiv x \bmod n$

→ Signature is valid

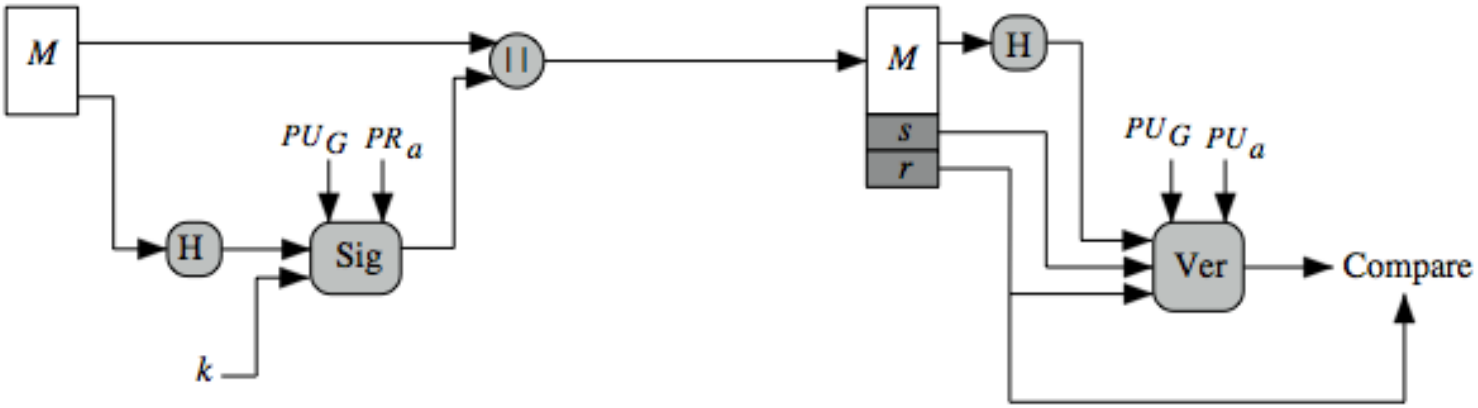
Existential Forgery and Padding

- An attacker can generate valid message-signature pairs (x, s)
- But an attack can only choose the signature s and NOT the message x
  - Attacker cannot generate messages like "Transfer $1000 into Oscar's account"
- Formatting the message x according to a padding scheme can be used to make sure that an attacker cannot generate valid (x, s) pairs.

RSA vs DSA Signatures



(a) RSA Approach



(b) DSS Approach

Facts about the Digital Signature Algorithm (DSA)

- Federal US Government standard for digital signatures (DSS)
- Proposed by the National Institute of Standards and Technology (NIST)
- DSA is based on the Elgamal signature scheme
  - Signature is only 320 bits long
  - Signature verification is slower compared to RSA
    - e in RSA is small
    - Signature verification is performed at devices with high-computing power

The Digital Signature Algorithm (DSA)

- Key generation of DSA:
  - Generate a prime p with $2^{1023} < p < 2^{1024}$
  - Find a prime divisor q of $p-1$ with $2^{159} < q < 2^{160}$
  - Find an integer α with $\mathrm{ord}(\alpha) = q$
  - Choose a random integer d with $0 < d < q$
  - Compute $\beta \equiv \alpha^d \mod p$
- The keys are:
  - $k_{pub} = (p, q, \alpha, \beta)$
  - $k_{pr} = (d)$

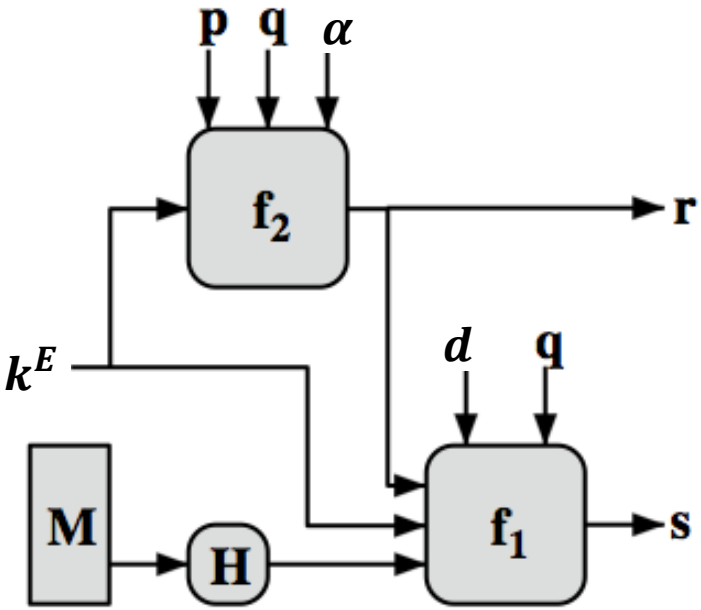
The Digital Signature Algorithm
SA signature generation

- Given: message x, private key d and public key (p, q, α, β)
  - Choose an integer as random ephemeral key k_E with $0 < k_E < q$
    - Comparing with the k_E in ElGamal
  - Compute $r \equiv (\alpha^{k_E} \bmod p) \bmod q$
  - Computes $s \equiv (\mathrm{SHA}(x) + d \times r) k_E^{-1} \bmod q$
    - SHA denotes the hash function SHA-1 which computes a 160-bit fingerprint of message x.
- The signature consists of (r, s)
  - r and s are both 160 bits

The Digital Signature Algorithm
Signature verification

- Given: message x, signature (r, s) and public key (p, q, α, β)
  - Compute auxiliary value $w \equiv s^{-1} \bmod q$
  - Compute auxiliary value $u_1 \equiv w \times \mathrm{SHA}(x) \bmod q$
  - Compute auxiliary value $u_2 \equiv w \times r \bmod q$
  - Compute $v \equiv (\alpha^{u1} \times \beta^{u2} \bmod p) \bmod q$
- If $v \equiv r \bmod q \rightarrow$ signature is valid
- If $v \equiv r \bmod q \rightarrow$ signature is invalid

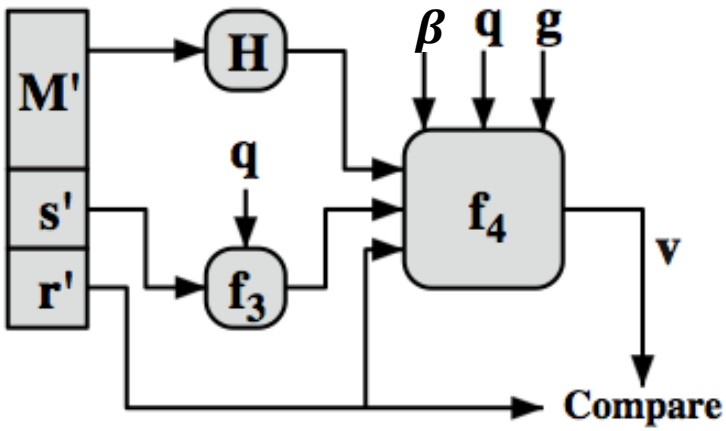
The Digital Signature Algorithm



$$s \equiv (\mathrm{SHA}(x) + d \times r) k_E^{-1} \bmod q$$

$$r \equiv (\alpha^{k_E} \bmod p) \bmod q$$

(a) Signing



$$w \equiv s^{-1} \bmod q$$

$$v \equiv (\alpha^{w\mathrm{SHA}(x) \bmod q} \times \beta^{wr \bmod q} \bmod p) \bmod q$$

(b) Verifying

Proof of DSA:

- We need to show that the signature (r, s) in fact satisfied the condition $r \equiv v \bmod q$:

$s \equiv \left(\mathrm{SHA}(x)\right) + d \times r)\, k_E^{-1} \bmod q$

$\Leftrightarrow k_E \equiv s^{-1}\, \mathrm{SHA}(x) + d \times s^{-1}\, r \bmod q$

$\Leftrightarrow k_E \equiv u_1 + d \times u_2 \bmod q$

- We can raise α to either side of the equation if we reduce modulo p:

$\Leftrightarrow \alpha^{k_E} \bmod p \equiv \alpha^{u1+d\times u_2} \bmod p$

- Since $\beta \equiv \alpha^d \bmod p$ we can write:

$\Leftrightarrow \alpha^{k_E} \bmod p \equiv \alpha^{u1}\beta^{u2} \bmod p$

- We now reduce both sides of the equation modulo q:

$\Leftrightarrow (\alpha^{k_E} \bmod p) \bmod q \equiv (\alpha^{u1}\beta^{u2} \bmod p) \bmod q$

- Since $r \equiv \alpha^{k_E} \bmod p \bmod q$ and $v \equiv (\alpha^{u1}\beta^{u2} \bmod p) \bmod q$, this expression is identical to:

$\Leftrightarrow r \equiv v$

Example

Alice

Bob

Key generation:

1. choose $p = 59$ and $q = 29$
2. choose $\alpha = 3$
3. choose private key $d = 7$
4. $\beta = \alpha^{\beta} = 3^7 \equiv 4 \bmod 59$

$(p, q, \alpha, \beta) = (59, 29, 3, 4)$ ←

$(x, (r, s)) = (x, 20, 5)$ ←

Sign:

Compute hash of message $H(x) = 26$

1. choose ephermal key $k_E = 10$
2. $r = (3^{10} \bmod 59) \equiv 20 \bmod 29$
3. $s = (26 + 7 \cdot 20) \cdot 3) \equiv 5 \bmod 29$

Verify:

$w \equiv 5^{-1} \equiv 6 \bmod 29$

$u_1 \equiv 6 \cdot 26 \equiv 11 \bmod 29$

$u_2 \equiv 6 \cdot 20 \equiv 4 \bmod 29$

$v = (3^{11} \cdot 4^4 \bmod 59) \bmod 29 = 20$

$v \equiv r \bmod 29 \rightarrow$ valid signature

Security of DSA

› To solve the discrete logarithm problem in p the powerful index calculus method can be applied. But this method cannot be applied to the discrete logarithm problem of the subgroup q. Therefore q can be smaller than p.

p	q	Hash output (min)	Security level
1024	160	160	80
2048	224	224	112
3072	256	256	128

Elliptic Curve Digital Signature Algorithm (ECDSA)

- Based on Elliptic Curve Cryptography (ECC)
- Bit lengths in the range of $160 - 256$ bits can be chosen to provide security equivalent to $1024 - 3072$ bit RSA ($80 - 128$ bit symmetric security level)
- One signature consists of two points, hence the signature is twice the used bit length (i.e., $320 - 512$ bits for $80 - 128$ bit security level)
- The shorter bit length of ECDSA often result in shorter processing time

Signed Diffie-Hellman

- Assuming the following security sizes
  - 1024 bits for RSA
  - 1024 bits for the prime in DSA
  - 160 bits for the group order in both DSA and EC-DSA
- Then we obtain the following message sizes for our signed Diffie-Hellman protocol
  - DH + DSA = 1024 (DH) + 320 (Signature) = 1344
  - DH + RSA = 1024 (DH) +1024 (Signature) = 2048
  - ECDH + RSA =160 (DH) + 1024 (Signature) = 1184
  - ECDH + ECDSA = 160 (DH) + 320 (Signature) = 480

NSA Suite B

- **Suite B** is a set of cryptographic algorithms promulgated by the National Security Agency as part of its Cryptographic Modernization Program
- Announced in February 2005
- Unpublished **Suite A** is intended for highly sensitive communication and critical authentication systems
- In December 2006, NSA submitted an Internet Draft on implementing Suite B as part of IPsec (Internet Protocol Security), and this draft has been accepted for publication by IETF (Internet Engineering Task Force) as RFC 4869

NSA Suite B

- Components of NSA Suite B
  - Symmetric Encryption: AES with key sizes of 128 and 256 bits, operated with Galois/Counter Mode (GCM)
  - Digital Signatures: ECDSA (P-256 and P-384)
  - Key Agreement: ECDH (P-256 and P-384)
  - Message Digest: SHA-2 (SHA-256 and SHA-384)
- Secret Information
  - 256-bit elliptic curve (FIPS 186-3), SHA-256, and AES with 128-bit keys are necessary
- Top Secret information
  - 384-bit elliptic curve (FIPS 186-3), SHA-384, and AES with 256-bit keys are necessary

Lessons Learned

- Digital signatures provide message integrity, message authentication and non-repudiation
- RSA is currently the most widely used digital signature algorithm
- Competitors are the Digital Signature Standard (DSA) and the Elliptic Curve Digital Signature Standard (ECDSA)
- RSA verification can be done with short public keys e. Hence, in practice, RSA verification is usually faster than signing
- DSA and ECDSA have shorter signatures than RSA
- In order to prevent certain attacks, RSA should be used with padding
- The modulus of DSA and the RSA signature schemes should be at least 1024 bits long. For true long-term security, a modulus of length 3072 bits should be chosen. In contrast, ECDSA achieves the same security levels with bit lengths in the range 160– 256 bits