Cryptography

Cunsheng DING

HKUST, Hong Kong

Lecture 18

Lecture 18: The Digital Signature Standard

Main Topics of This Lecture

1. Recall the digital signature scheme with a public-key cipher and a hash function covered in Lecture 16.
2. Introduce the Digital Signature Standard (DSS, also called the Digital Signature Algorithm/DSA)

Digital Signature Scheme with a Public-Key Cipher

The scheme introduced in Lecture 16: It has two building blocks: a public key cipher and a hash function h.

Assume that Alice and Bob have exchanged their public keys:

$$\text{Alice} \quad \longrightarrow \quad m||D_{k_d^{(A)}}[h(m)] \quad \longrightarrow \quad \text{Bob}$$

Remark: If c_2 is Alice's digital signature on c_1, then $h(c_1) = E_{k_e^{(A)}}(c_2)$. This is the basis for signature verification.

The Digital Signature Standard (DSS): Its Framework

The DSS building blocks: a hash function h, a set of public parameters (called the global public key $k_e^{(G)}$), a signature function sig, and a verification function ver.

Each user A has a private key $k_d^{(A)}$ for signing, and a public key $k_e^{(A)}$ for verifying. The signature value and verification value are defined:

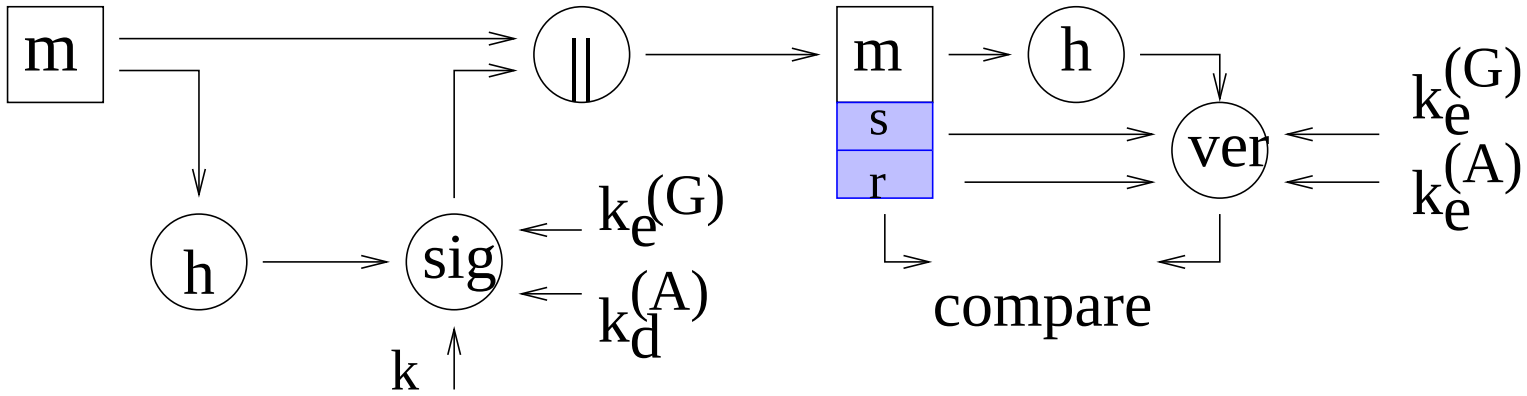
$$S := sig\left[k, k_e^{(G)}, k_d^{(A)}, h(m)\right],$$

$$V := ver\left[k_e^{(G)}, k_e^{(A)}, h(m), S\right],$$

where k is a random secret number generated by the signer for this m.

Remarks: The keys, the random number k, and the signature and verification functions will be defined shortly.

The Digital Signature Standard (DSS): Its Framework



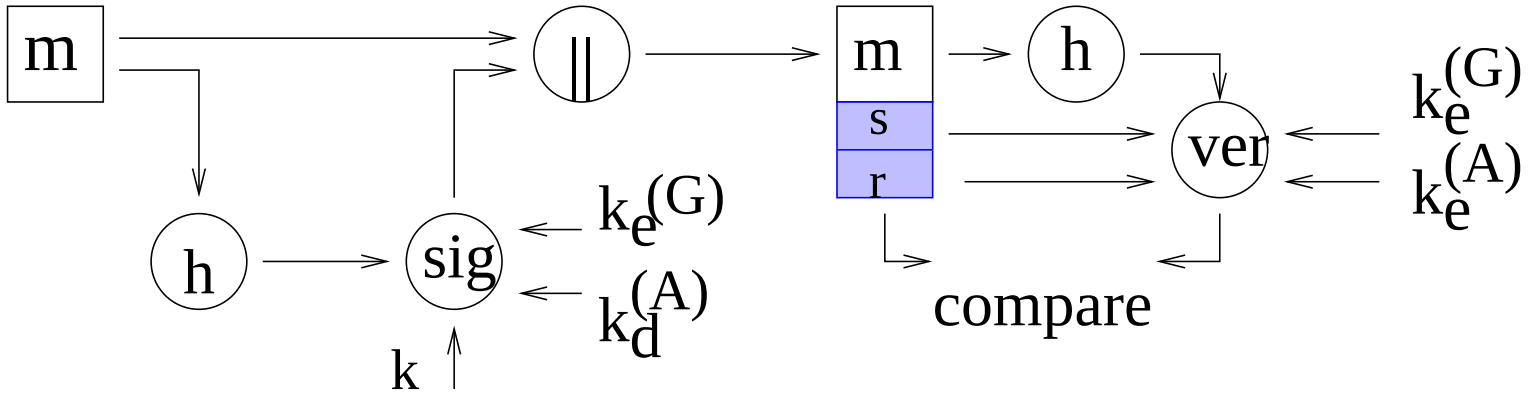
(b) DSS Approach

Signing: To sign, A generates a random number k for this message m and computes the signature:

$$(s, r) = sig\left[h(m), k, k_e^{(G)}, k_d^{(A)}\right].$$

A will then send $m||s||r$ to the receiver.

The Digital Signature Standard (DSS): Its Framework



(b) DSS Approach

Verifying: After partitioning a received text into $m'||s'||r'$, where r' and s' have the same length as a prime q (a parameter of the global public key to be defined later), the receiver uses $k_e^{(G)}$, $k_e^{(A)}$, and h to compute

$$v' := ver\left[h(m'), r', s', k_e^{(G)}, k_e^{(A)}\right].$$

The receiver then verifies the signature by checking if $v' = r'$.

DSS: Its Building Blocks

Hash function h: The original version of DSS uses SHA-1.

Global public key: $k_e^{(G)} = \{p, q, g\}$

p**:** prime number, where $2^{L-1} < p < 2^L$ for $512 \leq L \leq 1024$ and L a multiple of 64.

q**:** prime divisor of $(p-1)$, where $2^{159} < q < 2^{160}$, i.e., bit length of 160.

g**:** $= e^{(p-1)/q} \bmod p$, where e is any integer with $1 < e < (p-1)$ such that $e^{(p-1)/q} \bmod p > 1$.

By Fermat's Theorem, $g^q \bmod p = e^{p-1} \bmod p = 1$.

Remark: p and q are selected in such a range, as SHA-1 was used in this version of DSS and its hash values have 160 bits.

The signature and verification functions: To be defined later.

DSS: Description of the Parameters of a User

User's parameters

User's private key: x, a random integer with $0 < x < q$.

User's public key: $y = g^x \bmod p$.

DSS: The Signing Process in Detail

A uses her private key x, the public parameters (p, q, g), and a private random integer k with $0 < k < q$ to compute

- $r := (g^k \bmod p) \bmod q$.
- $s := [k^{-1}(h(m) + xr)] \bmod q$,

where k is used for this message m once.

If $s = 0$, then A has to choose another random k and recompute the signature so that $s \neq 0$. Note that $\Pr(s = 0) = 2^{-160}$.

The signature of m created by the signer is (s, r), **which depends on the private random number** k (only the signer knows k).

Remark: These two formulas for computing the (s, r) define the signature function sig.

A Relation Between A Message and Its Signature

The objective of this part: To find a relation satisfied by a message m and its digital signature (s, r) created by the signer, which will be a basis for signature verification.

Lemma 1: $k = (h(m) + xr)s^{-1} \bmod q$.

Proof: Note that $s \neq 0$ and $s < q$. By definition

$$s = [k^{-1}(h(m) + xr)] \bmod q.$$

Multiplying both sides with $s^{-1}k$ followed by the modulo-q operation gives

$$k = [s^{-1}(h(m) + xr)] \bmod q.$$

A Relation Between A Message and Its Signature

Lemma 2: $g^{x(rs^{-1} \bmod q)} \bmod p = g^{xrs^{-1} \bmod q} \bmod p$, s^{-1} is the inverse of s modulo q.

Proof: By Fermat's theorem, $g^q \bmod p = e^{p-1} \bmod p = 1$.

Let $rs^{-1} = qu + R$ for some u and $0 \leq R < q$.

Let $xR = qw + Z$ for some w and $0 \leq Z < q$. Then

$$g^{x(rs^{-1} \bmod q)} \bmod p = g^{xR} \bmod p = (g^q)^w g^Z \bmod p = g^Z \bmod p$$

and

$$g^{xrs^{-1} \bmod q} \bmod p = g^{(xqu+xR) \bmod q} \bmod p = g^{xR \bmod q} \bmod p = g^Z \bmod p.$$

The desired conclusion then follows.

The Relation Between A Message and Its Signature

Theorem 3: Let

$$v := [g^{(h(m)s^{-1}) \bmod q} y^{rs^{-1} \bmod q} \bmod p] \bmod q.$$

Then $v = r$.

Remark: The signature verification uses this result and checks whether $v = r$.

Remark: This formula for computing v defines the verification function ver.

Attention: The verification value v does not involve the random number k generated by the signer.

The Relation Between A Message and Its Signature

Proof of Theorem 3: By Lemmas 1 and 2 and by definition

$$
\begin{aligned}
v &= [(g^{[h(m)s^{-1}] \bmod q} \times y^{rs^{-1} \bmod q}) \bmod p] \bmod q \\
&= [(g^{[h(m)s^{-1}] \bmod q} \times g^{x(rs^{-1} \bmod q)}) \bmod p] \bmod q \\
&= [(g^{[h(m)s^{-1}] \bmod q} \times g^{xrs^{-1} \bmod q}) \bmod p] \bmod q \text{ (by Lemma 2)} \\
&= [g^{[h(m)s^{-1}+xrs^{-1}] \bmod q} \bmod p] \bmod q \\
&= [g^{[h(m)+xr]s^{-1} \bmod q} \bmod p] \bmod q \\
&= [g^k \bmod p] \bmod q \text{ (by Lemma 1)} \\
&= r.
\end{aligned}
$$

DSS: the Signature Verifying Process in Detail

The receiver partitions the received data into $m'||s'||r'$ such that s' and r' both have 160 bits. The receiver uses A's public key y and the public parameters (p, q, g) to compute

$$v' := [g^{(h(m')(s')^{-1}) \bmod q} y^{r'(s')^{-1} \bmod q} \bmod p] \bmod q.$$

Finally, test whether $v' = r'$. If $v' = r'$, the receiver accepts the message and the digital signature. Otherwise, the receiver rejects the received data.

Remark: The signature verification is based on Theorem 3.

Security of the Digital Signature Standard

Question: Is it possible to derive the private key x from the public parameters and the public key y?

Answer: The public parameters and public key are

$$(p, q, g), y.$$

The private key x is only related to those parameters by

$$y = g^x \bmod p.$$

So one has to solve this discrete-logarithm-like problem, which is believed to be hard in general. Note that p and q are very large. Notice that g may be or may not be a primitive root of p.

Security of the Digital Signature Standard

Question: Is it possible to derive the private key x from some $m||s||r$?

Solving the system of equations:

- $r = (g^k \bmod p) \bmod q$.
- $s = [k^{-1}(h(m) + xr)] \bmod q$.

Note that the random integer k is used only for one message m.

Observation: The system of two equations has two unknowns k and x. It may contain enough information on k and x. But it looks hard to compute k and x in the computational sense.

Approach 1: Solve the first one to find k, then solve the second to find x. Solving the first equation directly is to solve a discrete-logarithm-like problem, which is believed to be hard.

Security of the Digital Signature Standard

Solving the system of equations:

- $r = (g^k \bmod p) \bmod q$.
- $s = [k^{-1}(h(m) + xr)] \bmod q$.

Approach 2: By Lemma 1, $k = (h(m) + xr)s^{-1} \bmod q$. Plugging k into the first equation yields

- $r = (g^{s^{-1}h(m)}(g^{s^{-1}r})^x \bmod p) \bmod q$.

Hence, this is to solve a discrete-logarithm-like problem, which is believed to be hard.

Security of the Digital Signature Standard

Question: After receiving $m||s||r$ from Alice, Bob has successfully verified the signature. Today Bob has found a weak collision m' of m, i.e., $h(m) = h(m')$. Is $s||r$ the signature of m' by Alice?

Answer: It is left as an exercise.

Question: Should the underlying hash function h have the weak collision resistance property?

Answer: It depends on the answer to the previous question.

Historical Development of the DSS

- It was adopted as a standard on December 1 of 1994 by the NIST.
- It makes use of SHA-1.
- In new versions of the DSS, new versions of SHA are used. In such case, the sizes of p and q are increased accordingly.
- It is quite different from the digital signature system based on the RSA public-key cipher, as the digital signature depends on a random number k.
- DSS is also called the **Digital Signature Algorithm (DSA)**.

Remark: DSS is used in the **Pretty Good Privacy (PGP)** for email security.

Online Demo of the DSS/DSA

In the following URL, you find the demo of the Digital Signature Algorithm:

`https://8gwifi.org/dsafunctions.jsp`