

Cryptography CS 555

Lecture 18



Digital Signatures Schemes and Elliptic Curves

Review

- Security of signature schemes
 - attack objectives
 - total break, selective forgery, and existential forgery
 - attack types
 - key only, known message, chosen message
- RSA Signatures
 - $h(M)^d \bmod n$
- ElGamal Signatures
 - Signature is: (r, s)
 - $r = g^k \bmod p$ $s = k^{-1}(h(M) - ar) \bmod (p-1)$
 - Verification: check $\beta^r r^s \equiv g^{h(M)} \pmod{p}$

Digital Signature Algorithm (DSA)

Specified as FIPS 186

Key generation

- Select a prime q of 160-bits
- Choose $0 \leq t \leq 8$
- Select $2^{511+64t} < p < 2^{512+64t}$ with $q \mid p-1$
- Let α be a generator of Z_p^* , and set
 $g = \alpha^{(p-1)/q} \bmod p$
- Select $1 \leq a \leq q-1$
- Compute $\beta = g^a \bmod p$

Public key: (p, q, g, β)

Private key: a

DSA

Signing message M:

- Select a random integer k , $0 < k < q$
- Compute
$$k^{-1} \bmod q$$
$$r = (g^k \bmod p) \bmod q$$
$$s = k^{-1} (h(M) + ar) \bmod q$$
- Signature: (r, s)

Note: FIPS recommends
the use of SHA-1 as hash function.



DSA

Signature: (r, s)

$$r = (g^k \bmod p) \bmod q$$

$$s = k^{-1} (h(M) + ar) \bmod q$$

Verification

- Verify $0 < r < q$ and $0 < s < q$, if not, invalid

- Compute

$$u_1 = h(M)s^{-1} \bmod q,$$

$$u_2 = rs^{-1} \bmod q$$

- Valid iff $r = (g^{u_1} \beta^{u_2} \bmod p) \bmod q$

$$g^{u_1} \beta^{u_2} = g^{h(M)s^{-1}} g^{ars^{-1}} = g^{(h(M)+ar)s^{-1}} = g^k \pmod{p}$$

Lecture Outline

- One-time signatures
- Blind signatures
- Undeniable signatures
- Elliptic curves



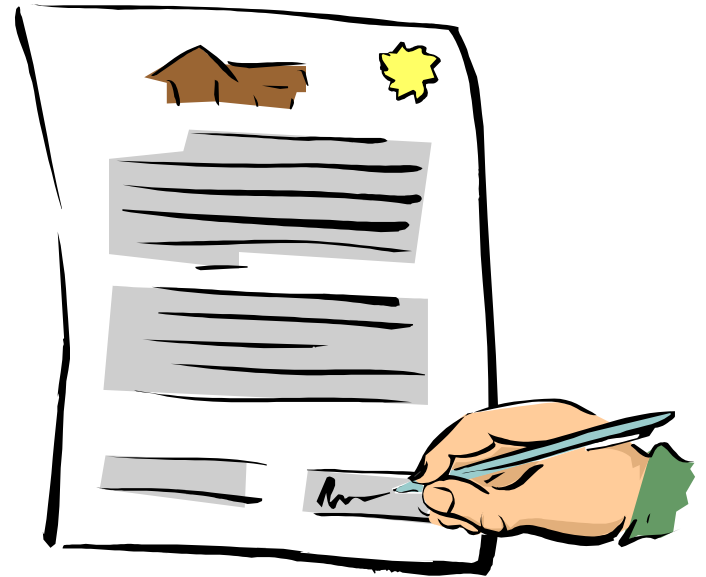
One-Time Digital Signatures

- One-time digital signatures: digital schemes used to sign, at most one message; otherwise signature can be forged.
- A new public key is required for each signed message.
- Advantage: signature generation and verification are very efficient and is useful for chipcards, where low computation complexity is required.

Lamport One-time Signature

To sign one bit:

- Choose as secret keys x_1, x_2
 - x_1 represents '0'
 - x_2 represents '1'
- public key:
 - $y_1 = h(x_1)$,
 - $y_2 = h(x_2)$.
- Signature is $h(x_1)$ if the message is x_1 or $h(x_2)$ for x_2



Merkle One-Time Signature

Key generation: to sign a n -bit message

- Select $t = n + 1 + \lfloor \lg n \rfloor$ and random secret strings k_1, k_2, \dots, k_t each of bitlength l

- Compute

$$v_i = h(k_i), \quad 1 \leq i \leq t,$$

h is a pre-image resistant hash:

$$\{0,1\}^* \rightarrow \{0,1\}^l$$

- Public key: (v_1, v_2, \dots, v_t)
- Private key: (k_1, k_2, \dots, k_t)

QuickTime™ and a TIFF (Uncompressed) decompressor are needed to see this picture.

Merkle One-Time Signature

Sign message m of bit-length n

- Compute c , the binary representation of number of 0's in m ;

$$w = m \parallel c = (a_1 a_2 \dots a_t)$$

- Determine the coordinate positions

$$i_1 < i_2 \dots < i_u \text{ in } w, \text{ s.t. } a_{i_j} = 1, 1 \leq j \leq u$$

- Let $s_j = k_{i_j}$, $1 \leq j \leq u$
- Signature is: (s_1, \dots, s_u)

Merkle One-Time Signature

Verification

- Compute c , the binary representation of number of 0's in m ;
 $w = m \parallel c = (a_1 a_2 \dots a_t)$
- Determine the coordinate positions $i_1 < i_2 \dots < i_u$ in w , s.t. $a_{i_j} = 1$
- Let $s_j = k_{i_j}$, $1 \leq j \leq u$
- Valid iff $v_{i_j} = h(s_j)$ for all j , $1 \leq j \leq u$

Blind Signature Schemes

- A wants B's signature on a message m , but doesn't want B to know the message m or the signature
- Applications: electronic cash
 - goal: anonymous spending
 - the bank signs a bank note, but A doesn't want B to know the note, as then B can associate the spending of B with A's identity

Chaum's Bind Signature Protocol Based on RSA

- Setup:
 - B has public key (n,e) and private key d
 - A has m
- Actions:
 - (blinding) A picks random $e \in \mathbb{Z}_n - \{0\}$ computes $m' = mk^e \pmod n$ and sends to B
 - (signing) B computes $s' = (m')^d \pmod n$ and sends to A
 - (unblinding) A computes $s = s'k^{-1} \pmod n$, which is B's signature on m

Undeniable Signatures

- Objectives:
 - signature verification must involve the participation of the signer
 - through a challenge-response verification protocol
 - the signer can also prove that a signature is not valid
 - through a disavowal protocol
- Applications:
 - software distribution

The Chaum-van Antwerpen Undeniable Signature Scheme

- Setup
 - Let $p=2q+1$ be a prime s.t. q is also prime
 - Let g be an element in Z_p^* with order q
 - Private key: $a \in Z_q^*$
 - Public key: $(p, g, \beta = g^a \text{ mod } p)$
- Signing algorithm:
 - To sign $x \in \langle g \rangle$, compute $y = x^a \text{ mod } p$

Verification Protocol

- Setup
 - Bob has x and s , wants to verify that s is a signature
- Interaction
 - Bob randomly chooses $e_1, e_2 \in \mathbb{Z}_q^*$
 - Bob computes $c = y^{e_1} \beta^{e_2} \pmod p$ and sends to Alice
 - Alice computes $d = c^{a^{-1} \pmod q} \pmod p$ and sends to Bob
 - Bob verifies that $d = x^{e_1} g^{e_2} \pmod p$

Correctness of the Verification Protocol

Theorem: if $y \neq x^a \pmod p$, then Bob accepts with prob. at most $1/q$

Proof. Given $c = y^{e_1} x^{e_2} \pmod p$, there are q pairs of (e_1, e_2) that are consistent with c .

Alice needs to return d s.t. $d = x^{e_1} y^{e_2} \pmod p$.

Let $c = g^i$, $d = g^j$, $x = g^k$, $y = g^w$

$$i \equiv w \cdot e_1 + a \cdot e_2 \pmod q \qquad j \equiv k \cdot e_1 + e_2 \pmod q$$

when $y = x^a \pmod p$, then $w = ka$, then $j = i \cdot a^{-1} \pmod q$

when $y \neq x^a \pmod p$, then the correct value of j depends on which pair of (e_1, e_2) is chosen

The Disavowal Protocol

- Interaction
 1. Bob randomly chooses $e_1, e_2 \in \mathbb{Z}_q^*$
 2. Bob computes $c_1 = y^{e_1} \beta^{e_2} \pmod p$ and sends to Alice
 3. Alice computes $d_1 = c_1^{a^{-1} \pmod q} \pmod p$ and sends to Bob
 4. Bob verifies that $d_1 \neq x^{e_1} g^{e_2} \pmod p$
 5. Bob randomly chooses $f_1, f_2 \in \mathbb{Z}_q^*$
 6. Bob computes $c_2 = y^{f_1} \beta^{f_2} \pmod p$ and sends to Alice
 7. Alice computes $d_2 = c_2^{a^{-1} \pmod q} \pmod p$ and sends to Bob
 8. Bob verifies that $d_2 \neq x^{f_1} g^{f_2} \pmod p$
 9. Bob verifies that $(d_1 g^{-e_2})^{f_1} \equiv (d_2 g^{-f_2})^{e_1} \pmod p$

Correctness of the Disavow Protocol

Theorem: If $y \neq x^a \pmod p$, and both parties follow the protocol, then Alice can convince Bob.

Proof:

$$d_1 = c_1^{a^{-1}} = (y^{e_1} \beta^{e_2})^{a^{-1}} = y^{e_1 \cdot a^{-1}} g^{e_2}$$

therefore, $d_1 = x^{e_1} g^{e_2}$ iff. $y^{e_1 \cdot a^{-1}} = x^{e_1}$ iff. $y = x^a$

step 4 verifies, similarly, step 8 verifies

$$(d_1 g^{-e_2})^{f_1} = (y^{e_1 \cdot a^{-1}} g^{e_2} g^{-e_2})^{f_1} = y^{e_1 \cdot a^{-1} \cdot f_1}$$

$$d_2 = c_2^{a^{-1}} = (y^{f_1} \beta^{f_2})^{a^{-1}} = y^{f_1 \cdot a^{-1}} g^{f_2}$$

$$(d_2 g^{-f_2})^{e_1} = (y^{f_1 \cdot a^{-1}} g^{f_2} g^{-f_2})^{e_1} = y^{f_1 \cdot a^{-1} \cdot e_1}$$

therefore, step 9 verifies

Correctness of the Disavow Protocol

Theorem: If $y=x^a \pmod p$, and Bob follows the protocol, then Alice can convince Bob with prob. $1/q$.

Proof:

$$\begin{array}{ll} \text{we have } y=x^a & (d_1g^{-e_2})^{f_1} = (d_2g^{-f_2})^{e_1} \\ d_1 \neq x^{e_1}g^{e_2} & d_2 \neq x^{f_1}g^{f_2} \end{array}$$

$$(d_1g^{-e_2})^{f_1} = (d_2g^{-f_2})^{e_1} \quad \text{iff. } d_2 = (d_1^{1/e_1}g^{-e_2/e_1})^{f_1}g^{f_2}$$

Let $x_0 = d_1^{1/e_1}g^{-e_2/e_1}$, which can be computed after step 4

Steps 5, 6, 7, 9 is the verification protocol that y is x_0 's signature, i.e., with prob. $1-1/q$, $y=x_0^a$, which implies

$$x = x_0 = d_1^{1/e_1}g^{-e_2/e_1}, \text{ which contradicts with } d_1 \neq x^{e_1}g^{e_2}$$

Elliptic Curves

Definition: A **non-singular elliptic curve** is the set E of solutions $(x,y) \in \mathbb{R} \times \mathbb{R}$ to the equation

$$y^2 = x^3 + ax + b$$

together with a special point O called the point at infinity, where a, b are real numbers s.t.

$$4a^3 + 27b^2 \neq 0$$

Explanation:

- $x^3 + ax + b = 0$ has 3 distinct solutions iff. $4a^3 + 27b^2 \neq 0$
- if $4a^3 + 27b^2 = 0$, then it is a singular elliptic curve

Elliptic Curves

- Defining a binary operation $+$ on a curve E to make the curve a group
 - Given two points $P=(x_1, y_1), Q=(x_2, y_2)$ in E
 - $P+O=O+P=P$
 - for $P+Q$, there are 3 cases
 1. $x_1 \neq x_2$: $P+Q = (x_3 = \lambda^2 - x_1 - x_2, y_3 = \lambda(x_1 - x_3) - y_1)$, and $\lambda = (y_2 - y_1) / (x_2 - x_1)$
 2. $x_1 = x_2 \wedge y_1 = -y_2$: $P+Q=O$
 3. $x_1 = x_2 \wedge y_1 = y_2$: $P+Q = (x_3 = \lambda^2 - x_1 - x_2, y_3 = \lambda(x_1 - x_3) - y_1)$, and $\lambda = (3x_1^2 + a) / (2y_1)$
- $(E, +)$ is a group

Elliptic Curves Modulo a Prime

- Instead of using real numbers, use numbers modulo a prime
 - the operation can be defined using the same formulas
- An elliptic curve defined over Z_p ($p > 3$ is a prime) will have roughly p points on it, more precisely

$$p + 1 - 2\sqrt{p} \leq \#E \leq p + 1 + 2\sqrt{p}$$

Summary

- One-time signatures
 - Lamport
 - Merkle
- Blind signatures
- Undeniable signatures
- Elliptic Curves



Next ...

- Entity authentication and identification protocols

