6.857 Recitation 9 Apr 8

- ☐ One-Time Signature
- ☐ Key Encapsulation Mechanism (KEM)
- ☐ Functional Encryption (FE) for Inner Product
- ☐ Bilinear Map

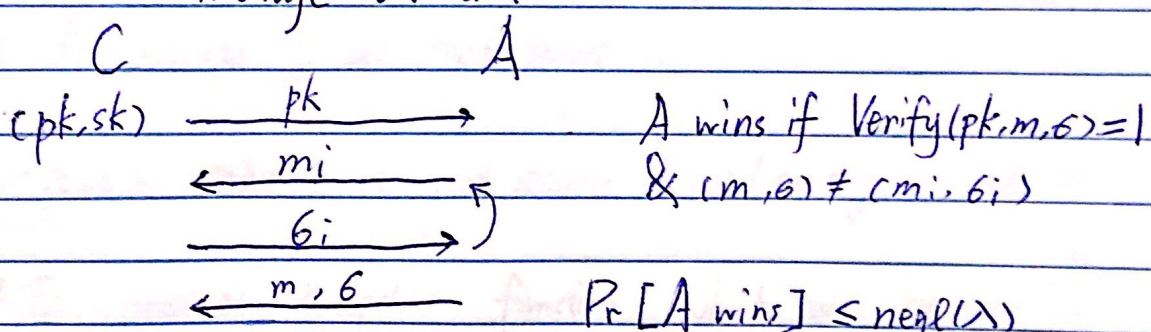
One-Time Signature

Recall definition of signature scheme

- $KeyGen(1^\lambda) \to (pk, sk)$
- $Sign(sk, m) \to \sigma$
- $Verify(pk, m, \sigma) \to 1/0$.

Correctness: $\forall m \quad Verify(pk, m, Sign(sk, m)) = 1$

Security: existentially unforgeability under an adaptive chosen message attack.



Signature scheme can be deterministic. Unlike encryption scheme, CPA secure encryption scheme must be randomized.

One-time security: A can only ask for one m_i.

Lamport's construction

f: one-way function

Key Gen: $x_{i,0}, x_{i,1} \leftarrow \{0,1\}^\lambda$

$y_{i,j} = f(x_{i,j})$.

$$SK = \begin{pmatrix} x_{1,0} & x_{2,0} & \cdots & x_{\ell,0} \\ x_{1,1} & x_{2,1} & \cdots & x_{\ell,1} \end{pmatrix} \quad PK = \begin{pmatrix} y_{1,0} & y_{2,0} & \cdots & y_{\ell,0} \\ y_{1,1} & y_{2,1} & \cdots & y_{\ell,1} \end{pmatrix}$$

Sign(m): $m = (m_1, \cdots, m_\ell)$.

Output $\sigma = (x_{1,m_1}, \cdots, x_{\ell,m_\ell})$

Verify(σ): $\sigma = (x_1, \cdots, x_\ell)$

Output 1 iff $f(x_i) = y_{i,m_i}$ $\forall i$.

The forge that A outputs must differ from the query A asks on at least one point. Then it must be the case that A inverts f at that point.

Lamport's scheme is not secure if used to sign multiple messages

PKE implies trapdoor function, which we believe to be strictly stronger than one-way function.
Surprisingly, Signature scheme can be built from OWF.

Public Key Cryptography	Private Key Cryptography
public key encryption	private key encryption
digital signature	MAC

Key Encapsulation Mechanism (KEM)

a.k.a Hybrid Encryption

Motivation: We have seen "Hash-and-Sign" Paradigm from the class to improve efficiency and shorten the signature. We can also use KEM to improve encryption scheme.

$\Pi = (Gen, Enc, Dec)$ expansive CPA secure public-key encryption scheme

$\Pi' = (Gen', Enc', Dec')$ cheap CPA secure private-key encryption scheme

Gen^{hybrid}: $(pk, sk) \leftarrow Gen$

$Enc^{hybrid}(m)$: $sk' \leftarrow Gen'$

Output $(c_1 = Enc(pk, sk'), c_2 = Enc'(sk', m))$

$Dec^{hybrid}_{(c_1, c_2)}$: $sk' = Dec(sk, c_1)$

$m = Dec'(sk', c_2)$

Π^{hybrid} is a CPA secure public-key encryption scheme with efficiency comparable to Π'.

For those who know Fully Homomorphic Encryption (FHE), hybrid encryption can make its ciphertext size as small as AES. (linear)

Bilinear Map

$e: G_1 \times G_1 \longrightarrow G_2$

DDH is easy in G_1. CDH is still hard.

Can be extended to multilinear map:

$e: G_1 \times \cdots \times G_1 \longrightarrow G_2$.

Application: • Non-interactive N-party key exchange.

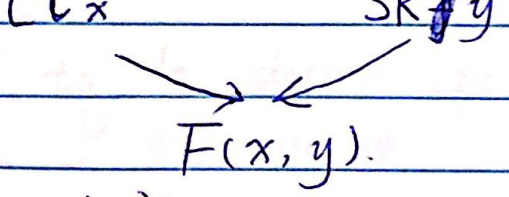
• Program Obfuscation. (can be used to construct almost all crypto primitives).

However, both candidate constructions are broken.

Functional Encryption for Inner Product

~~Origin~~ PKE: secret key. all-or-nothing.

FE: partial secret key.



Think of F as universal circuit. and y is some circuit.

- Setup$(1^\lambda) \longrightarrow (mpk, msk)$.
- Gen$(msk, y) \longrightarrow sk_y$.
- Enc$(mpk, x) \longrightarrow ct_x$
- Dec$(sk_y, ct_x) \longrightarrow F(x, y)$.

$F(x, y) = \langle x, y \rangle$

Setup: (G, p, g). (p: order, g: generator)

$s = (s_1, \cdots, s_l) \leftarrow \mathbb{Z}_p^l$.

$h_i = g^{s_i}$.

$msk = s \quad mpk = \{h_i\}$.

$\underline{Enc(x)}$: $x = (x_1, \cdots, x_\ell) \in \mathbb{Z}_p^\ell$.

$r \leftarrow \mathbb{Z}_p$.

$ct_0 = g^r$. $ct_i = h_i^r \cdot g^{x_i}$

$ct_x = (ct_0, \{ct_i\})$

$\underline{Gen(y)}$: $y = (y_1, \cdots, y_\ell) \in \mathbb{Z}_p^\ell$.

$sky = \langle y, s \rangle$.

Decrypt (ct_x, sky): Output $\frac{\prod_i ct_i^{y_i}}{ct_0^{sky}}$.

Correctness: $\frac{\prod_i ct_i^{y_i}}{ct_0^{sky}} = \frac{\prod_i g^{(sir + x_i) y_i}}{g^{r \sum y_i s_i}} = g^{\langle x, y \rangle}$.

① need to do discrete log.
Use Paillier group.

② selective secure.
tweak the scheme a little bit.

Linear functions are already very expressive: SVM.