CS655: Cryptography **Feb 20, 2007**

Lecture Outline for Secure Function Evaluation

1 Overview of Secure Function Evaluation

Problem Definition. There are u users, each user has $x_i \in \{0,1\}^n$ and a function $F_i : \{0,1\}^{n,u} \rightarrow \{0,1\}^m$. The goal is build a protocol such that at the end of the protocol, each user i has $F_i(x_1, \cdots, n_u)$, but knows nothing about $x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_u$ (beyond what can be inferred from its input x_i and output $F_i(x_1, \cdots, n_u)$).

The *ideal world* is have a Trust Third Party (TTP). Each user sends input x_i to the TTP, and the TTP returns $F_i(x_1, \cdots, n_u)$ to the i'th user.

The goal is to achieve the effect of the ideal world without using a TTP.

Security Models.

1. *Honest but curious*: (aka. semi-honest): all u parties follow protocols honestly.

   A protocol is t-*private* is any t parties collude still learn nothing from their transcript beyond their own inputs/outputs.

   Proof method: build a simulator, given inputs/outputs of t colluding parties, generate t transcripts that are indistinguishable from those in the actual protocol.

2. *Malicious*: The adversary gets to control a fixed set of t users. At start, the adversary chooses t parties to corrupt, remaining $n - t$ parties are honest.

   A protocol is t-*secure* if the adversary learns nothing about inputs of the other $u - t$ users beyond the outputs of t corrupt parties.

   Goal: t-secure, t'-private for some $t' \geq t$.

   Proof method: **For each** possible adversary behavior, build a simulator, given inputs/outputs of t colluding parties, generate t transcripts that are indistinguishable from those seen by the adversary.

3. *Dynamic (adaptive) adversary*: At any time period, the adversary can corrupt any t users.

   Security against such kind of adversary is called t-dynamic security.

Possible research topic: Honest but curious is too weak, and malicious is too strong. Any meaningful middle ground? One possibility is "Apparently honest".

Communication model. assume authentic and private communication channels between any two parties.

Example 1 *There are* 3 *users, having inputs* $x_1, x_2, x_3 \in \mathbb{Z}_p$*,* $F_1 = F_2 = F_3 = x_1 + x_2 + x_3$*.*

How to be 2-private? *(Degenerated case.)*

How to be 1-private?

Applications. (Pretty much) All crypto protocols.

1. *Identification*: Given a public one-way function f. B has y and A wants to prove to B that A knows x such that $f(x) = y$.

   A inputs x and B inputs y: $F_A = 0$ and $F_B(x, y) = \begin{cases} 1 & y = f(x) \\ 0 & y \neq f(x) \end{cases}$.

   Using SFE, no information about x is leaked.

2. *Private voting*: $x_i \in \{0, 1\}$ for $i = 1, \cdots, u$

   $F_1 = F_2 = \cdots = F_u = \mathsf{MAJ}(x_1, \cdots, x_u)$

   Using SFE, voter's privacy is preserved.

3. *Threshold cryptography*: Secret signing key is shared among u users such that signatures can be generated without reconstructing the private key.

   Let PK and SK be the public/private key for some signature scheme. Let $SK = SK_1 \oplus SK_2 \oplus \cdots \oplus SK_u$. Then let $x_i = SK_i$ and $F_1 = \cdots = F_u = \mathsf{Sign}(SK_1 \oplus \cdots \oplus SK_u, M)$.

   Using SFE, can sign without reconstructing the signing key.

4. *Private auctions*: $x_1, \cdots, x_u$ are distinct bids. Let s be the second highest bid. The define

$$F_i = \begin{cases} 0 & \text{if } x_i \text{ is not the higest bid} \\ s & \text{otherwise} \end{cases}$$

5. *Privacy preserving data mining*: $\cdots$

Summary of Results

1. 2-party SFE: Yao 82, Yao 86, GMW 87

2. BGW 87: n-party for $n > 2$: secure against $\lfloor \frac{n}{2} \rfloor - 1$ honest-but-curious adversaries.

3. Generic compiler that compiles any protocol secure against a honest-but-curious adversary to be secure against a malicious adversary

2 Oblivious Transfer (OT)

Oblivious Transfer (OT) is one specific SFE problem. However, it turns out to be a fundamental problem. It is used in Yao's 2-party SFE protocol.

1-out-of-n OT Party A has a list $x_1, \cdots, x_n$, and B has $i \in \{1, \cdots, n\}$. They want to compute: $F_A = 0$, $F_B = x_i$. That is: B learns nothing other than x_i, and A learns nothing about i.

Private information retrieval (PIR) is essentially OT, but the database is not private.

Kilian showed that 1-out-of-2 OT is universal for 2-party SFE. That is, given 1-out-of-2 OT, can do any SFE. Yao's construction uses 1-out-of-2 OT and block cipher (PRP).

The Bellare-Micali (92) construction for 1-out-of-2 OT . Let G be a group of prime order q, $g \in G$ a generator. Let $H : G \to \{0,1\}^n$ be a cryptographic hash function.

A has $x_0, x_2 \in \{0,1\}^n$ and B has $b \in \{0,1\}$.

The protocol is as follows:

1. A publishes $c \in G$.
2. B chooses $k \xleftarrow{\$} \mathbb{Z}_q$, sets $PK_b = g^k$ and $PK_{1-b} = c/g^k$, and sends both PK_0 and PK_1 to A.
3. A first checks that $PK_0 \cdot PK_1 = c$.
   A encrypts x_0 using PK_0: $C_0 = [g^{r_0}, H(PK_0^{r_0}) \oplus x_0]$
   A encrypts x_1 using PK_1: $C_1 = [g^{r_1}, H(PK_0^{r_1}) \oplus x_1]$
   A sends C_0, C_1 to B
4. X decrypts C_b using k as follows: Let $C_b = [v_1, v_2]$, calculate $H(v_1^k) \oplus v_2$.

Properties of the protocol:

- The encryption uses El Gamal encryption.
- A cannot get b; this is information theoretical secure against A.
- Computational secure against B: if B can break El Gamal encryption, B can get both x_0 and x_1.
- When B is honest but curious, secure under DDH assumption.
- When B is malicious, secure under the CDH assumption assuming that H is a random oracle.

1-out-of-2 OT implies 1-out-of-N OT A has $M_0, \cdots, M_N \in \{0,1\}^n$, and B has $t \in \{0, \cdots, N\}$. Assume $N = 2^\ell - 1$.

The protocol is as follows:

1. Let $F : \{0,1\}^m \times \{0,1\}^\ell \to \{0,1\}^n$ be a PRF.
   A prepares 2ℓ random keys $(k_1^0, k_1^1), \cdots, (k_\ell^0, k_\ell^1)$ for F.
2. A sends to B: $(C_0, \cdots, C_N)$ where
   $$C_I = M_I \oplus \bigoplus_{i=1}^{\ell} F_{k_i^{I_i}}(I)$$
   for $I = 0, 1, \cdots, N$ and I_i denotes the i'th bit of I.
3. Let the binary representation of t be $t_0, \cdots, t_\ell$. B does ℓ 1-out-of-2 OT, where in the j'th OT:
   A has (k_j^0, k_j^1), and B has $t_j \in \{0,1\}$.
4. B now has $k_1^{t_1}, \cdots, k_\ell^{t_\ell}$ and can decrypt C_t to get M_t.

In summary, using $\log N$ 1-out-of-2 OT, one can do 1-out-of-N OT, where the communication is $O(Nn)$.