Advanced Topics in Cryptography

Lecture 2

Benny Pinkas

1-out-of-2 Oblivious Transfer

- Two players: sender and receiver.
  - Sender has two inputs, x_0, x_1.
  - Receiver has an input $j \in \{0,1\}$.
- Output:
  - Receiver learns x_j and nothing else.
  - Sender learns nothing about j.

- Depending on the OT variant, the inputs x_0,x_1 could be strings or bits.

Security Definitions for OT

- It appeared to be quite hard to design an OT protocol that is secure against malicious adversaries in the sense of comparison to the ideal model.
  - Only recently were efficient such protocols designed.

- Therefore looser security definitions were used
  - These definitions ensure privacy but not correctness.
  - Namely, they do not ensure that the output is that of an OT functionality, or ensure independence of inputs.

Security Definitions for OT

- Defining what is means to protect the receiver's privacy is easy, since the sender receives no output in the ideal model and should therefore learn nothing about the receiver's input.

- Receiver's privacy – indistinguishability
  - For any values of the sender's inputs x_0, x_1, the sender cannot distinguish between the case that the receiver's input is 0 and the case that it is 1.

Security Definitions for OT

- Definition of sender's security:
  - This case is harder since the receiver does learn something about the sender's input

Security Definitions for OT

- Definition of sender's security:
  - For every algorithm A' that the receiver might run in the real implementation of oblivious transfer
  - there is an algorithm A" that the receiver can run in the ideal implementation
  - such that for any values of x_0,x_1 the outputs of A' and A" are indistinguishable.
  - Namely, the receiver in the real implementation does not learn anything more than the receiver in the ideal implementation.

- This definition does not handle delicate issues, such as whether the receiver "knows" j or the sender "knows" x_0,x_1

The Even-Goldreich-Lempel 1-out-of-2 OT construction (providing security only against semi-honest adversaries)

- Setting:
  - Sender has two inputs, x_0, x_1.
  - Receiver has an input $j \in \{0,1\}$.
- Protocol:
  - Receiver chooses a random public/private key pair (E,D).
  - It sets $PK_j=E$, and chooses PK_{1-j} at random from the same distribution as that of public keys*. It then sends (PK_0,PK_1) to the sender.
  - The sender encrypts x_0 with PK_0, and x_1 with PK_1, and sends the results to the receiver.
  - The receiver decrypts x_j.
  - Why is this secure against semi-honest adversaries?
- (*) It is required that it is possible to sample items with the exact distribution of public keys, and do this without knowing how to decrypt the resulting ciphertexts.

The Bellare-Micali Construction
(providing security against malicious adversaries)

- Preliminaries:
  - G_q is a subgroup of order q of Z_p^*, where p is prime and $p=2q+1$.
  - The OT protocol is secure assuming that the Computational Diffie-Hellman assumption holds for G_q.
  - The Computational Diffie-Hellman assumption (CDH) is that the following problem is hard:
    - The input to the problem is a generator g and values g^a ,g^b generated with random $a,b \in [1,q]$.
    - The task is to find $z=g^{a \cdot b}$.
  - (There is no need to use here the Decisional Diffie-Hellman problem)

The Bellare-Micali Construction

- Initialization: The sender chooses a random C in G_q.
- Protocol: (slightly modified)
  - The receiver picks a random $k \in [1,q]$, sets public keys $PK_j = g^k$, and $PK_{1-j} = C/PK_j$. It sends PK_0 to the sender.
  - The sender computes $PK_1 = C/PK_0$. Chooses a random r.
  - Generates El Gamal encryptions:
    - $E_0 = (g^r, H((PK_0)^r) \oplus x_0)$, $E_1 = (g^r, H((PK_1)^r) \oplus x_1)$, and sends them to the receiver.
  - The receiver computes $H((PK_j)^r)$ and decrypts E_j.
- Security:
  - Sender cannot learn anything about j (unconditionally).
  - The receiver cannot compute the discrete logs of both PK_0 and PK_1. (why?) (why does this imply security? $\Rightarrow$)

Security of the Bellare-Micali Construction

- The receiver cannot compute the discrete logs of both PK_0 and PK_1.

- The Computational Diffie-Hellman assumption implies that it cannot compute both $(PK_0)^r$ and $(PK_1)^r$:
  - Computing both $(PK_0)^r$ and $(PK_1)^r$, implies that the receiver can also compute C^r.
  - CDH: $(g,g^a,g^b) \rightarrow g^{ab}$ is hard
  - The receiver only knows g,C,g^r (for random C and r), and CDH implies that it cannot compute C^r.
- There is therefore an index i such that the receiver does not know $(PK_i)^r$
  - If we assume that $H()$ is a random function (a random oracle) then the receiver cannot distinguish $H((PK_i)^r)$ from a random string.

Security of the Bellare-Micali Construction

- To complete the proof, based on the observations given in the previous slide, we must show a proof of security by simulation, namely show that:
  - For every algorithm A' that the receiver might run in the real implementation of oblivious transfer
  - there is an algorithm A" that the receiver can run in the ideal implementation
  - such that for any values of x_0, x_1 the outputs of A' and A" are indistinguishable.

OT secure against malicious adversaries, without random oracles [NP]

- Security is based on the DDH assumption alone.
  - Security is proven according to the definition given before, ensuring only privacy, rather than proving full security.

- The Decisional Diffie-Hellman assumption (DDH)
  - The following problem is hard:
  - The input to the problem is
    - a generator g
    - values g^a, g^b generated with random $a,b \in [1,q]$
    - and a value g^c where with probability ½, c was chosen at random in $[1,q]$, *and* with probability ½, $c=ab$.
  - The task is to decide whether $c= ab$, or is random.

OT secure against malicious adversaries, without random oracles [NP]

- Security is based on the DDH assumption alone.
- Z_p^*, q, and sender's and receiver's inputs are as before.
- Receiver
  - chooses random $a,b,c_{1-j} \in [1,q]$, and defines $c_j = ab \pmod q$.
  - It sends to the sender $(g^a, g^b, g^{c0}, g^{c1})$.
- The sender
  - Certifies that $g^{c0} \neq g^{c1}$. Chooses random $s_0,r_0,s_1,r_1 \in [1,q]$.
  - Defines $w_0 = (g^a)^{s0} g^{r0}$. Encrypts x_0 with the key $(g^{c0})^{s0}(g^b)^{r0}$.
  - Defines $w_1 = (g^a)^{s1} g^{r1}$. Encrypts x_1 with the key $(g^{c1})^{s1}(g^b)^{r1}$.
  - Sends w_0, w_1 and the encryptions to receiver.
- Receiver computes $(w_j)^b$ which is the key with which x_j was encrypted. It uses it to and decrypt x_j.

Properties

- Correctness
  - Suppose j=0. R sends (g^a, g^b, g^{ab}, g^c).
  - S defines $w_0=(g^a)^{u0}g^{v0}$.
  - S encrypts x_0 with $k_0=(g^{ab})^{u0}(g^b)^{v0}$.
    - Note that encryption key is equal to $(w_0)^b$.
  - R computes $k_0=(w_0)^b$ and uses it for decryption.

- Overhead:
  - R computes 5 exponentiations.
  - S computes 8 exponentiations.

Privacy – malicious sender

- Receiver's security
  - Based on the DDH assumption
  - Must show that sender's view is indistinguishable regardless of receiver's input.
    - Sender receives either (g^a, g^b, g^{ab}, g^c) or (g^a, g^b, g^c, g^{ab}).
    - Suppose that it can distinguish between the two cases.

    - We can construct a distinguisher for the DDH problem, which distinguishes between (g^a,g^b,g^{ab}) and (g^a,g^b,g^c):
    - The distinguisher receives (g^a,g^b,X) and (g^a,g^b,Y), and sends (g^a,g^b,X,Y) to S.

Privacy – malicious receiver

- The security of the server is unconditional.
  - Does not depend on any cryptographic assumption.
- Suppose that j=0.
- Regarding x_1, the server sends
  - $w_1=(g^a)^{u_1}g^{v_1}$.
  - x_1 is then encrypted with the key $k_1=(g^c)^{u_1}(g^b)^{v_1}$.
  - The values u_1,v_1 were chosen at random, and $ab \neq c_1$.
  - **Claim:** (w_1,k_1) are uniformly distributed.
  - Therefore the message (w_1,k_1) sent by S about x_1 can be easily simulated.

Privacy – malicious receiver

- Proof of claim:
  - $w_l=(g^a)^{u_l}g^{v_l}=g^{a\cdot u_l+v_l}$.
  - $k_l=(g^c)^{u_l}(g^b)^{v_l}=g^{c\cdot u_l+b\cdot v_l}=(g^{(c/b)\cdot u_l+v_l})^b$.
  - Define $F(x)=u_l\cdot x+v_l$. F(x) is pair-wise independent:
    - $\forall x,y,s,t$ Prob(F(x)=s & F(y)=t) = $1/|G|^2$
  - $w_l=g^{F(a)}$.
  - $k_l=(g^{F(c/b)})^b$.
  - $c\neq ab$ and therefore F(a) and F(c/b) are uniformly distributed.
  - $\Rightarrow (w_l,k_l)$ are uniformly distributed.