CS590 Applied Cryptography
Oblivious Transfer

Chenghong Wang
What is Oblivious Transfer (OT)

1. Sender holds two inputs $x_0, x_1$ and wish to send exactly one of its input, say $x_b$.
2. Receiver has the choice bit $b$.
3. From protocol we seek (i) $x_{1-b}$ remains hidden from the Receiver and (ii) the choice bit $b$ remains hidden from the Sender.

$$(\bot, x_b) \leftarrow f_{OT}((x_0, x_1), b)$$
1. Receiver sends the choice bit \( b \) to the TTP.
2. TTP generates public encryption key pairs, and let these be \((pk_0, sk_0), (pk_1, sk_1)\), and send \( pk_0, pk_1 \) to the Sender and \( sk_b \) to the Receiver.
3. After that, the Sender sends \( c_0 = Enc(x_0, pk_0), c_1 = Enc(x_1, pk_1) \) to the Receiver.
4. Receiver decrypts \( x_b \leftarrow Dec(c_b, sk_b) \) to obtain the message.
1-2 Oblivious Transfer without Trusted party

1. \((pk_b, sk_b) \leftarrow Gen(1^k)\)
2. \((pk_{1-b}) \leftarrow z \leftarrow Rd\ Sample\)
3. \(pk_b, pk_{1-b} \leftarrow z\)
4. \(c_b \leftarrow Enc(pk_b, x_b), c_{1-b} \leftarrow Enc(z, x_{1-b})\)
5. \(x_b \leftarrow Dec(sk_b, c_b)\)
Security Analysis

• (i) **Receiver’s Security against unbounded Sender**: No efficient honest-but-curious sender does learn the choice bit $b$ of the receiver. This holds even for computationally unbounded adversary.

• (ii) **Sender’s Security against Computationally bounded Receiver**: Computationally bounded receiver learns no information about $x_{1-b}$ except what is revealed by $x_b$. 
Receiver’s Security – Game based Proof

\[ \text{Sender} \] \quad \text{Game 0} \quad \text{Receiver} \quad \begin{align*} \text{Sender} & \quad \text{Game 1} \quad \text{Receiver} \\ \text{pk}_0 \leftarrow \text{Gen}(1^k), \text{pk}_1 \leftarrow z & \quad \text{pk}_0 \leftarrow z, \text{pk}_1 \leftarrow \text{Gen}(1^k) \end{align*} \]

\[ \text{pk} \approx z \to \Pr[A(\text{Game 0}) = 1] \approx \Pr[A(\text{Game 1}) = 1] \]

No efficient honest-but-curios receiver will be able to distinguish between game 0 and game 1 with probability higher than \(1/2 + \text{negl}(k)\) probability.
Sender’s Security – Game based Proof

Game 0

Sender

$p k_b, p k_1-b$

c_b ← $Enc(pk_b, x_b)$,

c_{1-b} ← $Enc(pk_{1-b}, x_{1-b})$

Game 1

Sender

$p k_b, p k_1-b$

c_b ← $Enc(pk_b, x'_b)$,

c_{1-b} ← $Enc(pk_{1-b}, x'_{1-b})$

$Pr[A(Game 0) = 1] \approx Pr[A(Game 1) = 1]$
Sender’s Security – Game based Proof

Suppose the receiver is able to distinguish between the game 0 and game 1 with probability greater than $1/2 + \text{negl}(k)$, then there exists a p.p.t. adversary $A_{pk}$ that runs OT internally that can be used to break the employed public key encryption scheme.
Simulation based Proof

• The parties learn nothing from the protocol execution beyond what they can derive from their input and prescribed output.

\[
\Pr[A(S_x(In_x, Out_x)) = 1] - \Pr[A(View^T_x) = 1] \leq negl(k)
\]
Simulation based Proof

Sender’s Inputs: \((x_0, x_1, 1^k)\)
Sender’s outputs: nothing
Sender’s view: \((x_0, x_1, pk_0, pk_1, 1^k, \bot)\)

1. \(x_0, x_1, 1^k\) are identically distributed in both cases.
2. \(pk_0, pk_1\) and \(pk_0', pk_1'\) also have identical distribution as they all generated from the same key generation function.
Simulation based Proof

Receiver’s Inputs: \((b, 1^k)\)
Receiver’s outputs: \((x_b)\)
Receiver’s view: \((c_0, c_1, b, 1^k, x_b)\)

Input

1. \((sk_b, pk_b) \leftarrow Gen(1^k)\)
2. \((pk_{1-b}) \leftarrow RdGen\)
3. \((c_b') \leftarrow Enc(pk_b, x_b)\)
4. \((c_{1-b}') \leftarrow Enc(pk_{1-b}, 0^n)\)

Output

\((c_0', c_1', b, 1^k, x_b)\)

1. \(b, 1^k, x_b\), are identically distributed in both cases.
2. \(c_b, c_b'\) are computational indistinguishable.
3. \(c_{1-b}, c_{1-b}'\) are computational indistinguishable?
   If there exists any adversary that can distinguish between \(c_{1-b}, c_{1-b}'\), then for any message \(x\), this adversary is able to distinguish between \(Enc(x), Enc(0^n)\), where \(x \in \{0,1\}^n\) which breaks CPA secure.
Practice for Simulation based Proof
The Simplest Protocol for Oblivious Transfer

Sender $x_0, x_1$

1. $a \leftarrow Z_p$

2. $A \leftarrow g^a$

$B$

1. $c \leftarrow Z_p$

2. $A \leftarrow g^a$

3. If $b = 0$, $B \leftarrow g^c$, else $B \leftarrow Ag^c$

$B$

4. $B$

5. $k_0 \leftarrow B^a; k_1 \leftarrow \left(\frac{B}{A}\right)^a$

6. $c_0 \leftarrow Enc(k_0, x_0) \leftarrow B^ax_0$

$C$

7. $x_b \leftarrow Dec(k_b, c_b) \leftarrow c_b/A^c$

$C$

6. $c_1 \leftarrow Enc(k_1, x_1) \leftarrow \left(\frac{B}{A}\right)^a x_1$

7. $x_b \leftarrow Dec(k_b, c_b) \leftarrow c_b/A^c$
**Simulation based Proof**

Receiver’s Inputs: \((b, c, g, Z_p)\)

Receiver’s outputs: \((x_b)\)

Receiver’s view: \((b, c, g, Z_p, A, c_0, c_1, x_b)\)

1. \(b, c, g, Z_p, x_b\) are identically distributed in both cases.
2. \(A\) and \(A'\) are identically distributed in both cases.
3. \(c_b\) has the same distribution in both cases. As \((g^c)^z\) and \((g^c)^a\) are identically distributed, as long as \(c\), \(a\) are randomly sampled from \(Z_p\).
4. \(c_{1-b}\) has the same distribution in both cases. If there exists p.p.t. adversary can distinguish \(c_{1-b}\) in both cases then it can distinguish between \(g^z x\) and \(g^z x'\), where \(x \neq x'\), and \(z\) is random sampled from \(Z_p\) and unknown to the adversary.

---

**Input**

\((b, c, g, Z_p, x_b)\)

1. RdGen: \(z \leftarrow Z_p, A' \leftarrow g^z, B \leftarrow g^c\)

2. If \(b = 0\),
   \[c_0 \leftarrow (g^c)^z x_0, c_1 \leftarrow \left(\frac{g^c}{g^z}\right)^z (0^n)\]

   Else if \(b = 1\),
   \[c_0 \leftarrow g^z g^c (0^n), c_1 \leftarrow (g^c)^z (x_1)\]

**Output**

\((b, c, g, Z_p, A', c_0, c_1, x_b)\)
Simulation based Proof

Sender’s Inputs: \((x_0, x_1, a, g, Z_p)\)
Sender’s outputs: nothing
Sender’s view: \((x_0, x_1, a, g, Z_p, B, \bot)\)

1. \(x_0, x_1, a, g, Z_p\) are identically distributed in both cases.
2. \(B, B'\) also have identical distribution.

\[
\begin{align*}
\text{Input} & \quad \rightarrow \quad (x_0, x_1, a, g, Z_p) \\
1. \text{RdGen: } z & \leftarrow Z_p \\
2. B' & \leftarrow g^z g^a \\
\text{Output} & \quad \leftarrow \quad (x_0, x_1, a, g, Z_p, B', \bot)
\end{align*}
\]