The Simplest Protocol for Oblivious Transfer

Tung Chou

Technische Universiteit Eindhoven, The Netherlands

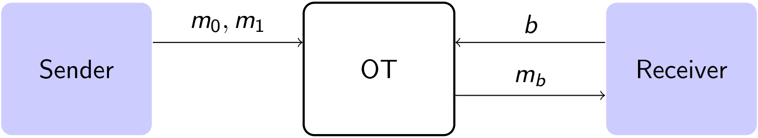
August 24, 2015

Latincrypt 2015, Guadalajara, Mexico

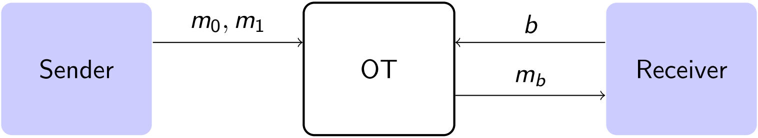
Joint work with Claudio Orlandi

$\binom{2}{1}$ OTs

$\binom{2}{1}$ OTs



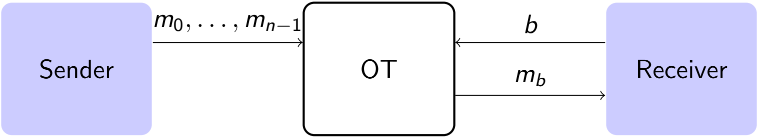
$\binom{2}{1}$ OTs



The **Receiver** should learn only m_b

The **Sender** should learn nothing

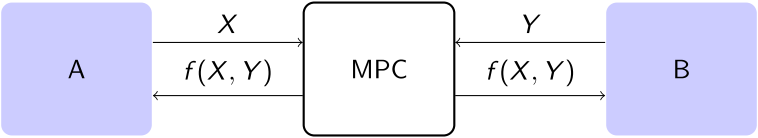
$\binom{n}{1}$ OTs



The **Receiver** should learn only m_b

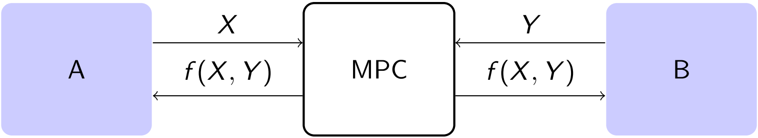
The **Sender** should learn nothing

Secure Multiparty Computation



The parties should learn no more than $f(X, Y)$

Secure Multiparty Computation



The parties should learn no more than $f(X, Y)$

"OT is complete for secure multiparty computation."

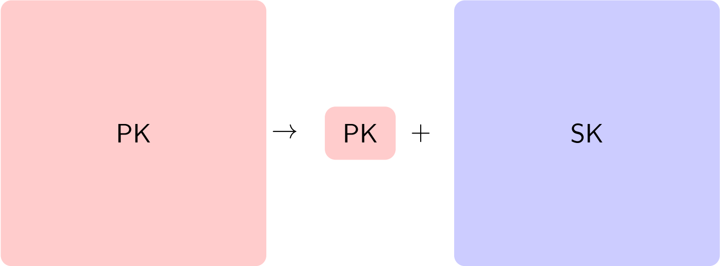
OT Extension

PK

OT Extension

PK $\rightarrow$ PK $+$ SK

OT Extension



- Similar to hybrid encryption
- Still we need base OTs

Diffie-Hellman

random x

$$\xrightarrow{\quad xB \quad}$$

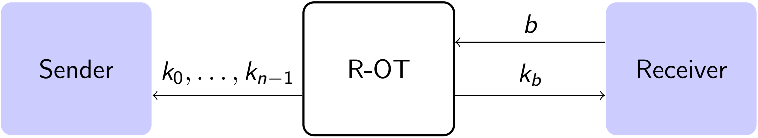
random y

$$\xleftarrow{\quad yB \quad}$$

$x(yB) = xyB$

$y(xB) = xyB$

Random-OT



The **Receiver** should learn only k_b

The **Sender** gets all k_i but nothing about b

Our Random-OT construction

random x

$$\xrightarrow{\quad S = xB \quad}$$

random y

$$\xleftarrow{\quad R = yB + \mathbf{bS} \quad}$$

$k_i \leftarrow \mathcal{H}\left(x(R - \mathbf{iS})\right), \forall i$

$k \leftarrow \mathcal{H}\left(yS = xyB\right)$

Our Random-OT construction

random x

$$\xrightarrow{\quad S = xB \quad}$$

random y

$$\xleftarrow{\quad R = yB + \mathbf{bS} \quad}$$

$k_i \leftarrow \mathcal{H}\left(x(R - \mathbf{iS})\right), \forall i$

$k \leftarrow \mathcal{H}\left(yS = xyB\right)$

- R uniformly random: privacy for Receiver

Our Random-OT construction

$$
\begin{array}{ccc}
\text{random } x & & \\
 & \xrightarrow{\quad S = xB \quad} & \\
 & & \text{random } y \\
 & \xleftarrow{\quad R = yB + \mathbf{bS} \quad} & \\
k_i \leftarrow \mathcal{H}\left(x(R - \mathbf{iS})\right), \forall i & & k \leftarrow \mathcal{H}\left(yS = xyB\right)
\end{array}
$$

- R uniformly random: privacy for Receiver
- Square DH: privacy for Sender

Our Random-OT construction

random x

$$\xrightarrow{\quad S = xB \quad}$$

random y

$$\xleftarrow{\quad R = yB + \mathbf{bS} \quad}$$

$$k_i \leftarrow \mathcal{H}\left(x(R - \mathbf{iS})\right), \forall i \qquad\qquad k \leftarrow \mathcal{H}\left(yS = xyB\right)$$

- R uniformly random: privacy for Receiver
- Square DH: privacy for Sender
- Sender precomputes $T = xS$

Our Random-OT construction

random x

$$\xrightarrow{\quad S = xB \quad}$$

random y

$$\xleftarrow{\quad R = yB + \mathbf{bS} \quad}$$

$k_i \leftarrow \mathcal{H}\left(x(R - \mathbf{iS})\right), \forall i$ $\qquad$ $k \leftarrow \mathcal{H}\left(yS = xyB\right)$

- R uniformly random: privacy for Receiver
- Square DH: privacy for Sender
- Sender precomputes $T = xS$
- $\mathcal{H}$ is modeled as RO

Our Real-OT Construction

random OT

$$\xrightarrow{c_i = \mathcal{E}_{k_i}(m_i),\ \forall i}$$

$$m_b = \mathcal{D}_k(c_b)$$

Our Real-OT Construction

random OT

$$\xrightarrow{c_i = \mathcal{E}_{k_i}(m_i),\ \forall i}$$

$$m_b = \mathcal{D}_k(c_b)$$

- Encryption scheme:

$$\mathcal{E}_k(m) = k \oplus (m|0^\lambda)$$

Our Real-OT Construction

random OT

$$\xrightarrow{c_i = \mathcal{E}_{k_i}(m_i),\ \forall i}$$

$$m_b = \mathcal{D}_k(c_b)$$

- Encryption scheme:

$$\mathcal{E}_k(m) = k \oplus (m|0^\lambda)$$

$$\mathcal{D}_k(c = (m'|t) \oplus k) = \begin{cases} m' & \text{if } t = 0^\lambda \\ \textit{FAIL} & \text{otherwise} \end{cases}$$

The Naor-Pinkas OT

- #exponentiations: n vs. 2 offline (3 online)

The Naor-Pinkas OT

- #exponentiations: n vs. 2 offline (3 online)

$$
\begin{tikzcd}
 & R_{b=0} \arrow[dl, "s_1"] \arrow[d, "s_2"] \arrow[dr, "s_3"] & \\
R_{b=1} & R_{b=2} & R_{b=3}
\end{tikzcd}
$$

The Naor-Pinkas OT

- #exponentiations: n vs. 2 offline (3 online)

$$
\begin{tikzcd}
 & R_{b=0} \arrow[dl, "s_1"'] \arrow[d, "s_2"] \arrow[dr, "s_3"] & \\
R_{b=1} & R_{b=2} & R_{b=3}
\end{tikzcd}
$$

$$R_{b=0} \xrightarrow{s} R_{b=1} \xrightarrow{s} R_{b=2} \xrightarrow{s} R_{b=3}$$

The Naor-Pinkas OT

- #exponentiations: n vs. 2 offline (3 online)

$$\begin{tikzcd} & R_{b=0} \arrow[dl, "s_1"] \arrow[d, "s_2"] \arrow[dr, "s_3"] & \\ R_{b=1} & R_{b=2} & R_{b=3} \end{tikzcd}$$

$$R_{b=0} \xrightarrow{s} R_{b=1} \xrightarrow{s} R_{b=2} \xrightarrow{s} R_{b=3}$$

- Game-based proof vs. simulation-based proof (UC)

The Encryption Scheme

$\mathcal{E}, \mathcal{D}$ needs to satisfy

- Robustness: Given a set of random keys, it is hard for $\mathcal{A}$ to generate a ciphertext that can be decrypted with more than one key.
- Non-committing: it is possible for a simulator to come up with a ciphertext which can later be explained as an encryption of any message

Base-OT Implementation

- [ALSZ13]: based on MIRACL, used in the SCAPI library

Base-OT Implementation

- [ALSZ13]: based on MIRACL, used in the SCAPI library

	Our work	[ALSZ13]
Curve	Curve25519	NIST K-283
Constant-time	Yes	No
Million Cycles/OT	0.23	2.47

Base-OT Implementation

- [ALSZ13]: based on MIRACL, used in the SCAPI library

	Our work	[ALSZ13]
Curve	Curve25519	NIST K-283
Constant-time	Yes	No
Million Cycles/OT	0.23	2.47

- code available at orlandi.dk/simpleOT