Generic
Fully Simulatable Adaptive OT

Kaoru KUROSAWA (Ibaraki Univ., Japan)

Ryo NOJIMA (NICT, Japan)

Le Trieu PHONG (NICT, Japan)

Outline

- **Oblivious Transfer (OT)**
  - Adaptive OT
  - Fully-simulatable security
- **Known results**
- **Our proposal**
  - DDH Linear assumptions.
  - QR, DCR assumptions.

Adaptive k-out-of-n OT

Sender $M_1, \ldots, M_n$

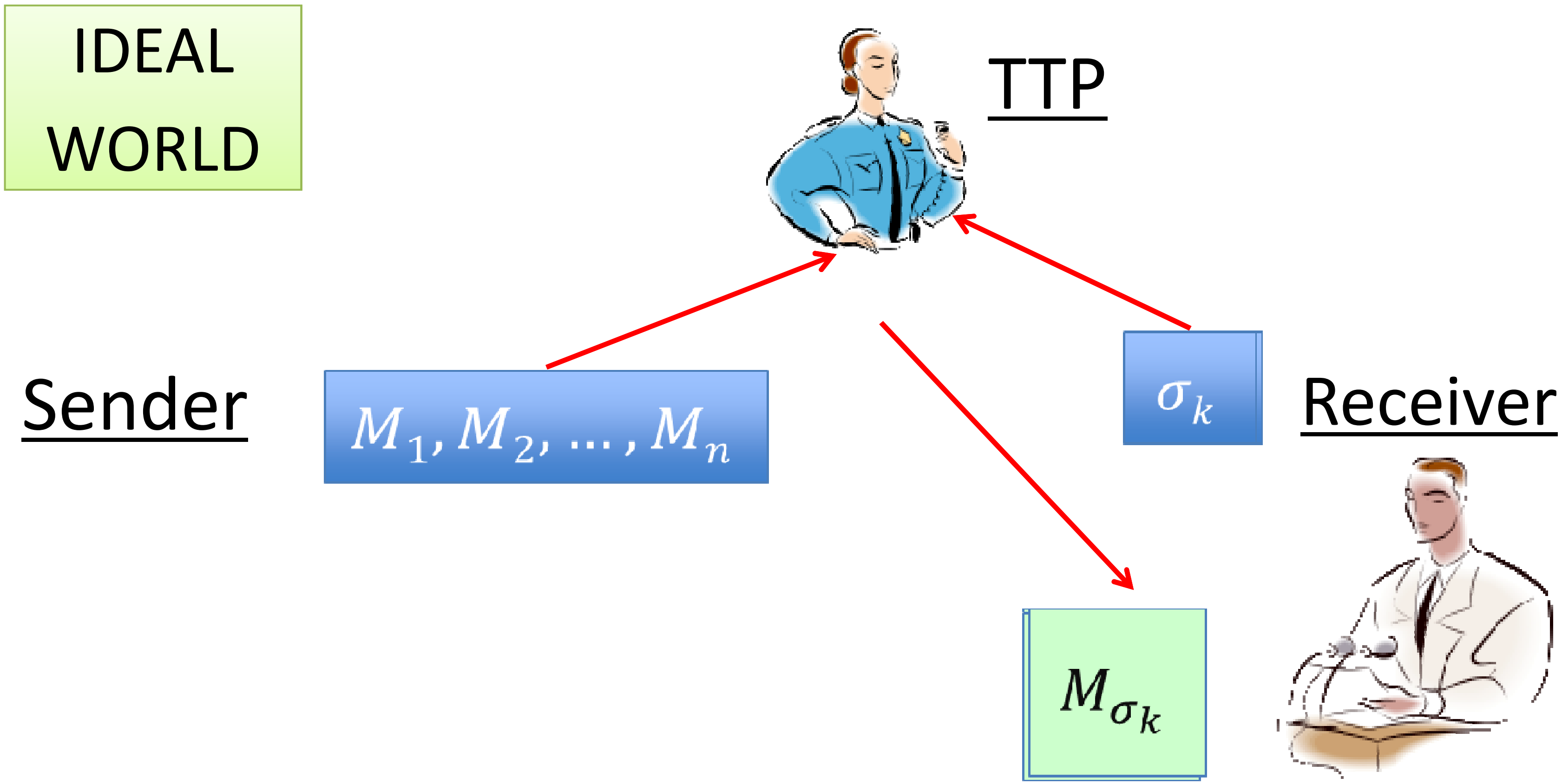
σ_k Receiver

Protocol

Receiver obtains M_{σ_k}

Applications: privacy-enhanced databases.

Fully-simulatable security



Fully simulatable: OT Protocol ≈ Ideal World.

Brief history of adaptive OT

- **Concept:**

  Naor-Pinkas (1999) *not* fully simulatable.

- **Ogata-Kurosawa** (2004): ROM, using blind signatures.

- **Camenisch, Neven, Shelat** (2007): *fully simulatable* adaptive OT, extending Ogata-Kurosawa + a standard model scheme.

Standard model schemes

Initialization cost = $O(n)$ for all

Protocols	Assumption	Comm. Cost (each transfer)
CNS (EC '07)	q-strong DH & q-PDDH	$O(1)$
GH (AC '07)	q-hidden LRSW (UC-secure)	$O(1)$
JL (TCC '09)	q-DHI	$O(1)$
KN (AC '09)	DDH	$O(n)$
GH (TCC '10)	3DDH (pairing)	$O(1)$
KNP (SCN '10)	DDH (no pairing)	$O(1)$
This work	DDH, Linear, QR, DCR	$O(1)$

A simplification

Threshold ElGamal in $G =< g >$
Public key $pk = g^x$ for $x = x_S + x_R$
Encryption:$(A_i, B_i) = (g^{r_i}, pk^{r_i} \cdot M_i)$
Partial decryption: $\mu_S = A_i^{x_S}, \mu_R = A_i^{x_R}$

Sender $x_S, M_1, \ldots, M_n$ ← pk → **Receiver** σ, x_R

Hide σ

$C_\sigma = \mathbf{Enc}(M_\sigma)$

$$\mathbf{Enc}(M_i) = (A_i, B_i) \forall 1 \leq i \leq$$

$$C_\sigma = \mathbf{Rnd}(A_\sigma, B_\sigma) = (A_\sigma, B_\sigma) \cdot (g^u, pk^u) \quad u \in_R \mathbf{Z}_q$$

$$\mu_S = C_\sigma[1]^{x_S}$$

Partial decryption

Compute μ_R using x_R

$$M_\sigma = C_\sigma[2]/(\mu_S \mu_R)$$

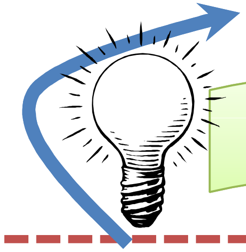
Adding ZKIP for full simutability

Sender

Initialization Phase

Receiver

$(A_i, B_i)\forall 1 \leq i \leq n, PoK\{r_i = \mathbf{dlog}_g A_i\}$



$\boldsymbol{O(n^2)}$

(Each) Transfer Phase

Comm. cost $\boldsymbol{O(n)}$.
k times $\Rightarrow \boldsymbol{O(kn)}$

$C_\sigma = \mathbf{Rnd}(A_\sigma, B_\sigma) = (A_\sigma, B_\sigma) \cdot (g^u, pk^u)$

$PoK\{C_\sigma = \mathbf{Rnd}(A_1, B_1) \vee \cdots \vee \mathbf{Rnd}(A_n, B_n)\}$

$\mu_S = C_\sigma[1]^{x_S}, PoK\{x_S\}$

Compute μ_R using x_R

$M_\sigma = C_\sigma[2]/(\mu_S \mu_R)$

$O(n^2) \rightarrow O(n)$ by **shuffle protocol**

Sender

Initialization Phase

Receiver

$(A_i, B_i) \forall 1 \le i \le n$, $PoK\{r_i = \mathbf{dlog}_g A_i\}$ →

Exactly Groth-Lu's shuffle protocol, **cost $O(n)$**

Permutation π over $\{1, \ldots, n\}$

Random $u_1, \ldots, u_n \in \mathbf{Z}_q$

$\forall i, C_i = \mathbf{Rnd}(A_{\pi(i)}, B_{\pi(i)}) = (A_{\pi(i)}, B_{\pi(i)}) \cdot (g^{u_i}, pk^{u_i})$

$PoK\{\pi, u_1, \ldots, u_n\}$ ←

Shuffling

Hide π

$O(1)$

(Each) Transfer Phase

$C_{\pi^{-1}(\sigma)} \in \{C_1, \ldots, C_n\}$ ←

$\mu_S = C_{\pi^{-1}(\sigma)}[1]^{x_S}$, $PoK\{x_S\}$ →

Basing on Linear Assumption

- Use the scheme of Naor-Segev (Crypto 2009).

$$sk \in \mathbf{Z}_q^{(d+1)\times 1}, pk = (\phi, \phi \cdot sk) \text{ for } \phi \in G^{d\times(d+1)}$$

$$\mathbf{Enc}(M) = (R \cdot \phi, R \cdot (\phi \cdot sk) \cdot M) \text{ for } R \in Z_q^{1\times d}$$

- Homomorphic, semantically-secure under d-linear assumption. ($d = 2 \rightarrow \mathrm{DLIN}$)
- Groth-Lu's shuffle protocol works well again.

OT based on QR, DCR

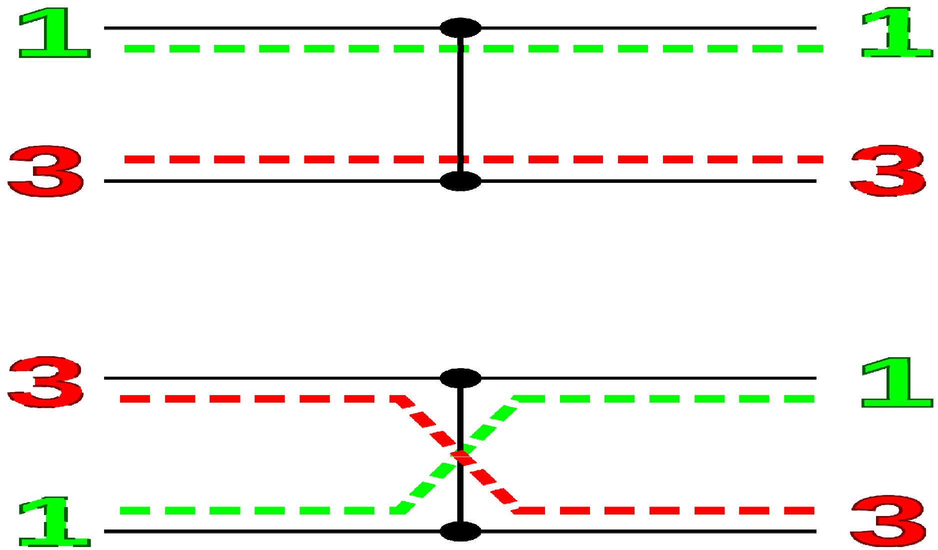
- Groth-Lu shuffle only works on group with known order (ElGamal, Linear).

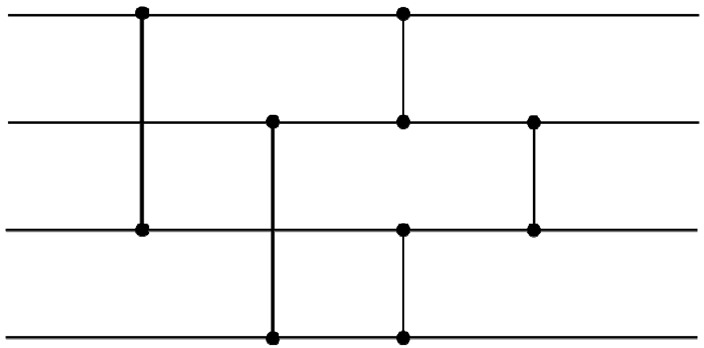
- But cannot work with un-known order groups

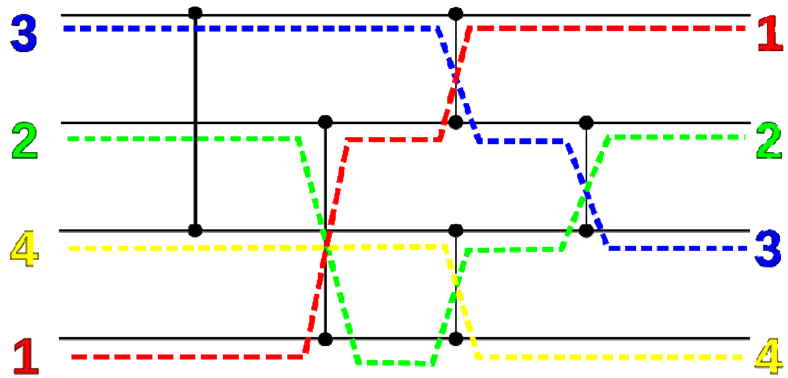
(QR, DCR).

- We overcome the problem by making use of permutation network for shuffling.

Permutation network







$O(n^2) \rightarrow O(nlogn)$ by **permutation network**

Sender | **Receiver**

Initialization Phase

$\forall i\ , A_i = \mathbf{E}(k_i, r_i) = y^{k_i} r_i^2 \ mod\ N, B_i = k_i \oplus M_i$ →

Permutation π over $\{1, \ldots, n\}$
Random $u_i, s_i \in \boldsymbol{Z}_q$

$\forall i, C_i = \mathbf{Rnd}\left(A_{\pi(i)}\right) = A_{\pi(i)} \cdot \mathbf{E}(u_i, s_i)\ mod\ N$
$PoK\{\pi, u_i, s_i\}$ ←

(Each) Transfer Phase

$C_{\pi^{-1}(\sigma)}$ ←

$\mu_S = \mathbf{D}\ (C_{\pi^{-1}(\sigma)})$, ZKIP →

$$PoK\{\pi, u_i, s_i : C_i = A_{\pi(i)} \cdot \mathbf{E}(u_i, s_i) \forall 1 \leq i \leq n\}$$

- n = #Messages = 2
- *PoK* of π, u_1, u_2, s_1, s_2:
  $C_1 = A_{\pi(1)}\mathbf{E}(u_1, s_1) \wedge C_2 = A_{\pi(2)}\mathbf{E}(u_2, s_2)$

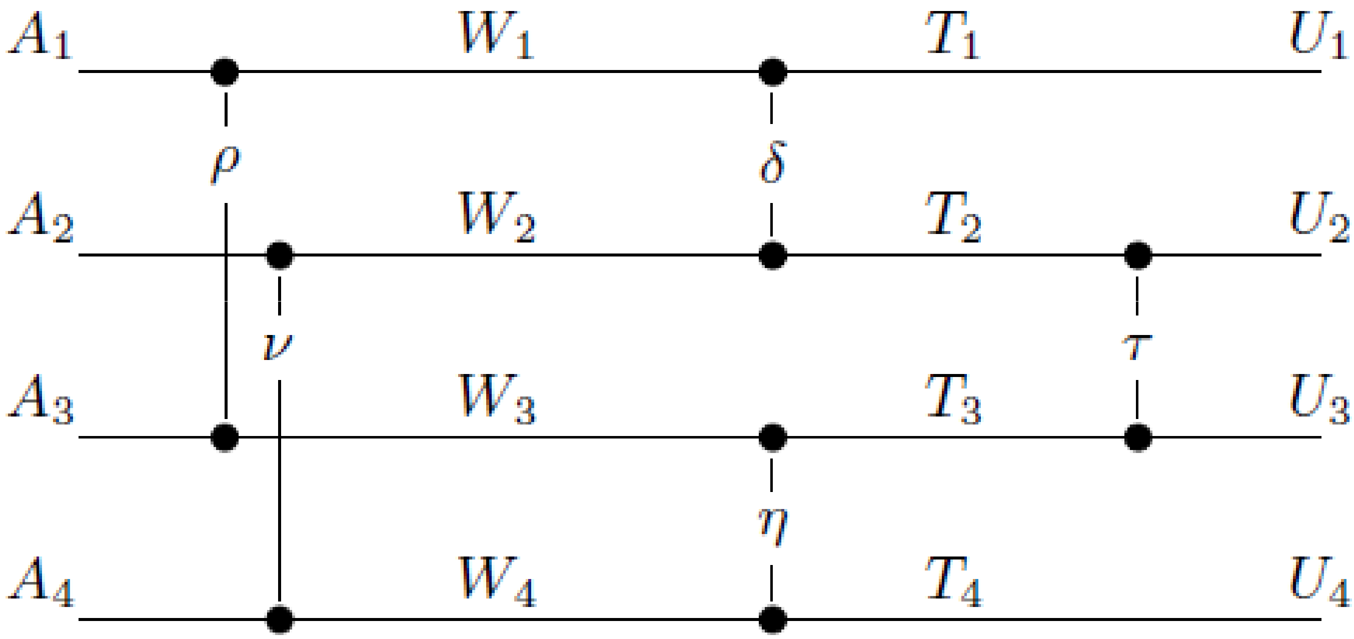
$$\Leftrightarrow \quad [C_1 = A_1\mathbf{E}(u_1, s_1) \wedge C_2 = A_2\mathbf{E}(u_2, s_2)]$$
$$\vee [C_1 = A_2\mathbf{E}(u_1, s_1) \wedge C_2 = A_1\mathbf{E}(u_2, s_2)]$$

$$\Leftrightarrow (C_1 = A_1\mathbf{E}(u_1, s_1) \vee C_1 = A_2\mathbf{E}(u_1, s_1)) \wedge$$
$$(\cdot \vee \cdot) \wedge (\cdot \vee \cdot) \wedge (\cdot \vee \cdot)$$

- Totally, 4 OR-proofs. Can be realized efficiently

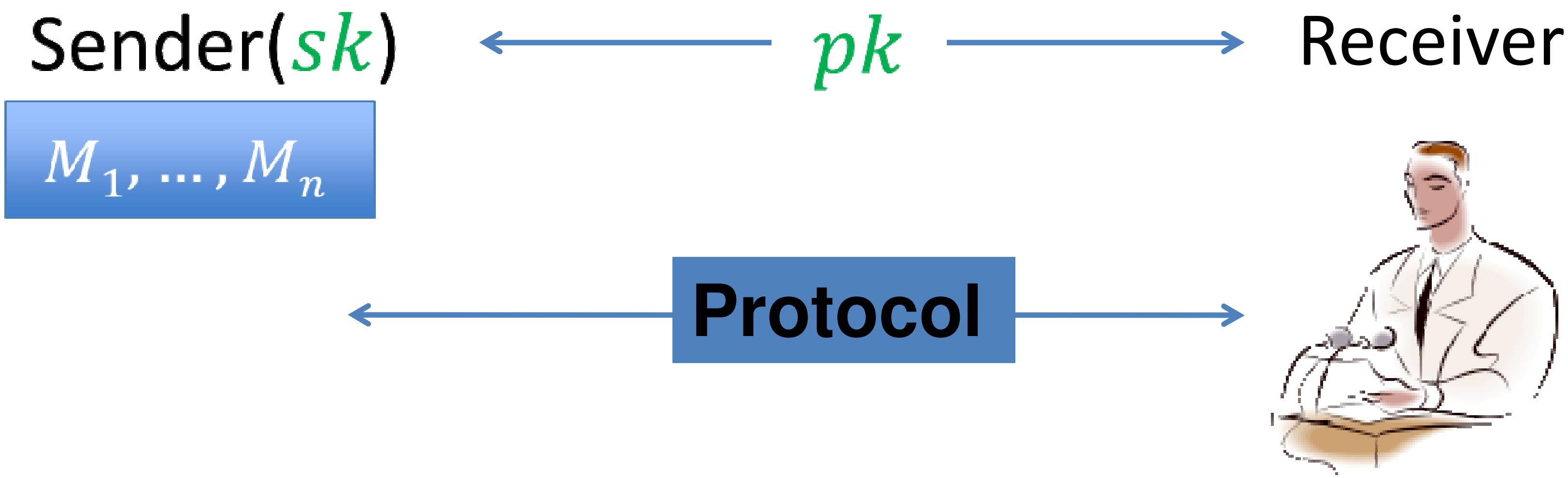
Going from $n = 2$ to $n = 4$

For permutations $\rho, \nu, \delta, \eta, \tau$, applying the case $n = 2$



Going from $n = 2$ to general n: use general permutation network with $O(n\log n)$ switches.

Leakage-resilient OT



- sk may be leaked by side-channel attacks.
- If we use leakage-resilient encryption, our protocols remain secure even sk is leaked.

Conclusion *Thank you!*

Initialization cost = $O(n)$ for all

Protocols	Assumption	Comm. Cost (each transfer)
CNS (EC '07)	q-strong DH & q-PDDH	$O(1)$
GH (AC '07)	q-hidden LRSW (UC-secure)	$O(1)$
JL (TCC '09)	q-DHI	$O(1)$
KN (AC '09)	DDH	$O(n)$
GH (TCC '10)	3DDH (pairing)	$O(1)$
KNP (SCN '10)	DDH (no pairing)	$O(1)$
This work	DDH, Linear, QR, DCR	$O(1)$