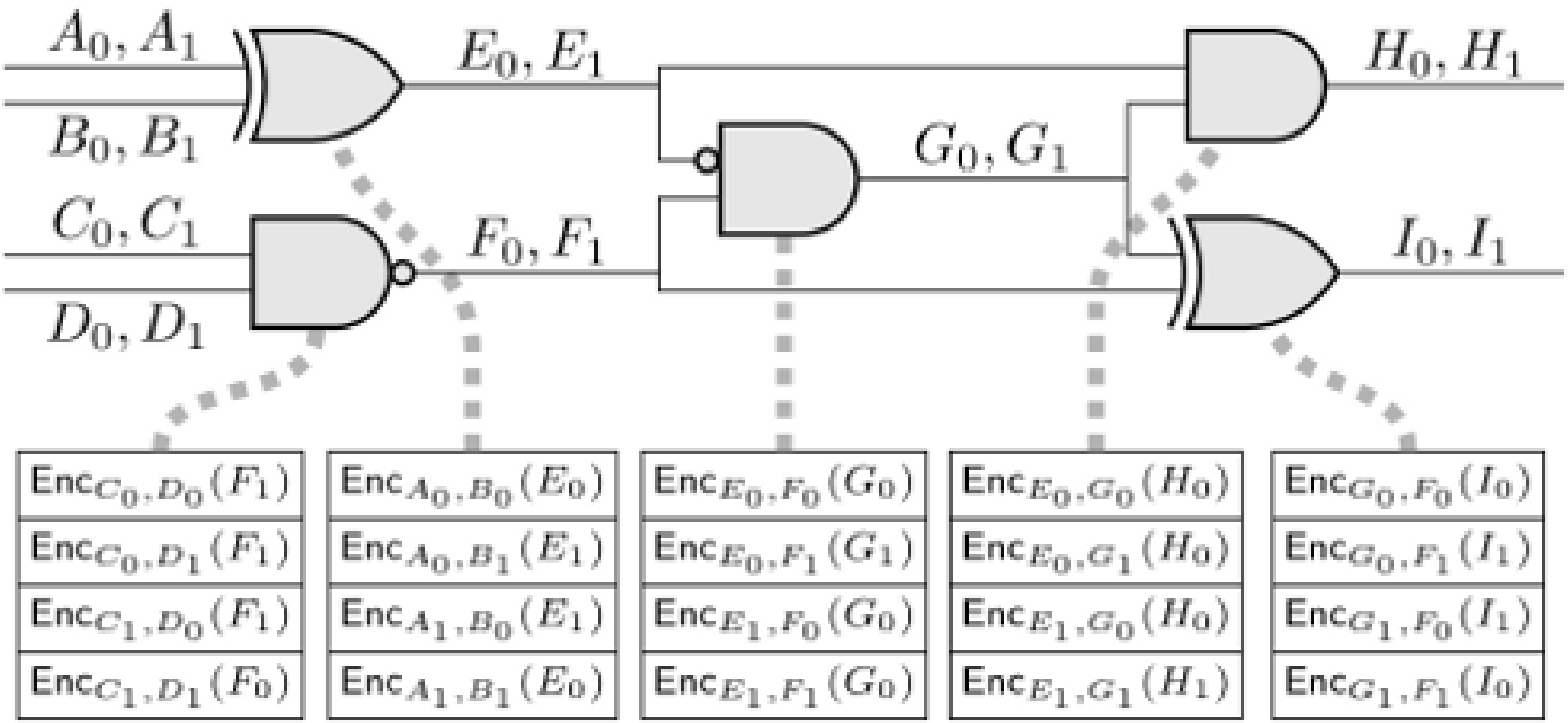
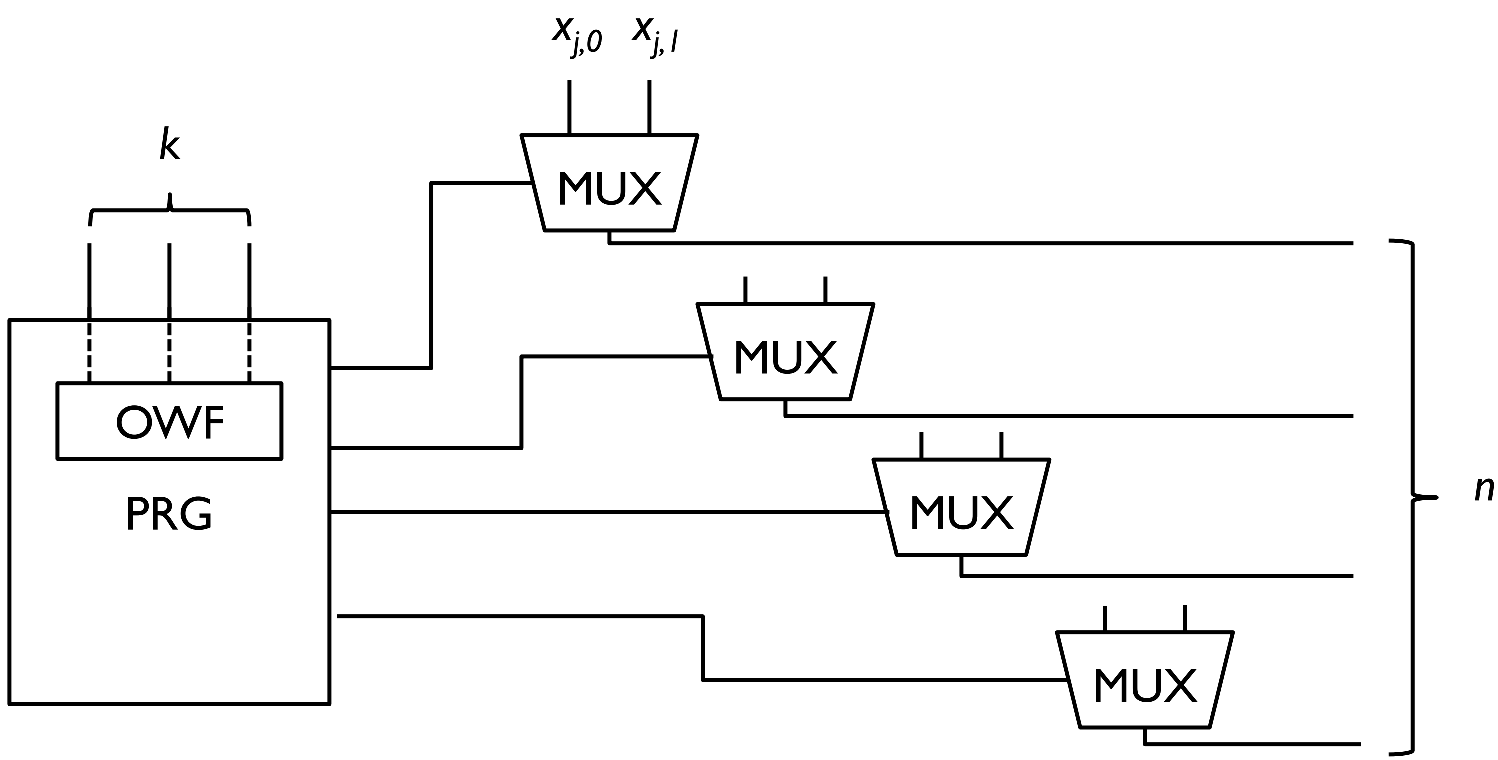
Survey of
Oblivious Transfer Extension

Natnatee Dokmai

Garbled Circuit



Beaver's Protocol



IKNP Protocol

INPUT OF SENDER $\mathcal{S}$: m pairs $(x_{j,0}, x_{j,1})$ of ℓ-bit strings for $1 \leq j \leq m$.
INPUT OF RECEIVER $\mathcal{R}$: m selection bits $\mathbf{r} = (r_1, ..., r_m)$.
COMMON INPUT: a security parameter k.
ORACLE: a random oracle H: $[m] \times \{0,1\}^k \to \{0,1\}^\ell$.

1. S initializes a random vector $\mathbf{s} \in \{0,1\}^k$ and $\mathcal{R}$ a random $m \times k$ bit matrix T. We have the following T:

$$T = \begin{bmatrix} t_1^1 & t_1^2 & \dots & t_1^k \\ t_2^1 & t_2^2 & \dots & t_2^k \\ \vdots & \vdots & \ddots & \vdots \\ t_m^1 & t_m^2 & \cdots & t_m^k \end{bmatrix} = \begin{bmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \\ \vdots \\ \mathbf{t}_m \end{bmatrix} = \begin{bmatrix} (\mathbf{t}^1)^T & (\mathbf{t}^2)^T & \cdots & (\mathbf{t}^k)^T \end{bmatrix}$$

2. The parties invoke OT_m^k primitive:
   - $\underline{\mathcal{R}}$ sends $(\mathbf{m}_{i,0}, \mathbf{m}_{i,1}) = (\mathbf{t}^i, \mathbf{r} \oplus \mathbf{t}^i)$, $1 \leq i \leq k$.
   - $\underline{\mathcal{S}}$ receives with input $\mathbf{s}$, i.e. $\mathcal{S}$ receives $\mathbf{m}_{i,s_i}$ for every i. (Note that $\mathbf{m}_{i,b} = b\mathbf{r} \oplus \mathbf{t}^i$.)

IKNP Protocol (2)

3. Let Q denote the $m \times k$ matrix of values received by $\mathcal{S}$ where

$$
\begin{aligned}
Q &= \left[(\mathbf{m}_{1,s_1})^T \quad (\mathbf{m}_{2,s_2})^T \quad \ldots \quad (\mathbf{m}_{k,s_k})^T\right] \\
&= \left[(s_1\mathbf{r} \oplus \mathbf{t}^1)^T \quad (s_2\mathbf{r} \oplus \mathbf{t}^2)^T \quad \ldots \quad (s_k\mathbf{r} \oplus \mathbf{t}^k)^T\right] \\
&= \begin{bmatrix} s_1r_1 \oplus t_1^1 & s_2r_1 \oplus t_1^2 & \ldots & s_kr_1 \oplus t_1^k \\ s_1r_2 \oplus t_2^1 & s_2r_2 \oplus t_2^2 & \ldots & s_kr_2 \oplus t_2^k \\ \vdots & \vdots & \ddots & \vdots \\ s_1r_m \oplus t_m^1 & s_2r_m \oplus t_m^2 & \ldots & s_kr_m \oplus t_m^k \end{bmatrix} \\
&= \begin{bmatrix} r_1\mathbf{s} \oplus \mathbf{t}_1 \\ r_2\mathbf{s} \oplus \mathbf{t}_2 \\ \vdots \\ r_m\mathbf{s} \oplus \mathbf{t}_m \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \vdots \\ \mathbf{q}_m \end{bmatrix}
\end{aligned}
$$

$\underline{\mathcal{S}}$ sends $(y_{j,0}, y_{j,1})$, for $1 \leq j \leq m$, where $y_{j,0} = x_{j,0} \oplus H(j, \mathbf{q}_j)$ and $y_{j,1} = x_{j,1} \oplus H(j, \mathbf{q}_j \oplus \mathbf{s})$. (Note that $y_{j,b} = x_{j,b} \oplus H(j, q_j \oplus b\mathbf{s})$)

IKNP Protocol (3)

4. $\underline{\mathcal{R}}$ receives $(y_{j,0}, y_{j,1})$ and recovers x_{j,r_j}, the messages he intends to receive, from the following steps:

$$\begin{aligned} z_{j,r_j} &= y_{j,r_j} \oplus H(j, \mathbf{t}_j) \\ &= \left[x_{j,r_j} \oplus H(j, q_j \oplus r_j \mathbf{s})\right] \oplus H(j, \mathbf{t}_j) \\ &= x_{j,r_j} \oplus H(j, (r_j \mathbf{s} \oplus \mathbf{t}_j) \oplus r_j \mathbf{s}) \oplus H(j, \mathbf{t}_j) \\ &= x_{j,r_j} \oplus H(j, \mathbf{t}_j), \oplus H(j, \mathbf{t}_j) \\ &= x_{j,r_j} \end{aligned}$$

For the messages $\mathcal{R}$ did not choose, they appear to be uniformly random bits because of the fact that $\mathcal{R}$ possesses no knowledge of $\mathbf{s}$:

$$\begin{aligned} z_{j,1-r_j} &= y_{j,1-r_j} \oplus H(j, \mathbf{t}_j) \\ &= \left[x_{j,1-r_j} \oplus H(j, q_j \oplus (1 - r_j)\mathbf{s})\right] \oplus H(j, \mathbf{t}_j) \\ &= x_{j,1-r_j} \oplus H(j, (r_j \mathbf{s} \oplus \mathbf{t}_j) \oplus (1 - r_j)\mathbf{s}) \oplus H(j, \mathbf{t}_j) \\ &= x_{j,1-r_j} \oplus H(j, \mathbf{s} \oplus \mathbf{t}_j), \oplus H(j, \mathbf{t}_j) \end{aligned}$$

NNOB & ZDE Protocol

$$
\begin{aligned}
Q &= \left[(\mathbf{m}_{1,s_1})^T \quad (\mathbf{m}_{2,s_2})^T \quad \ldots \quad (\mathbf{m}_{k,s_k})^T \right] \\
&= \left[(s_1\mathbf{r} \oplus \mathbf{t}^1)^T \quad (s_2\mathbf{r} \oplus \mathbf{t}^2)^T \quad \ldots \quad (s_k\mathbf{r} \oplus \mathbf{t}^k)^T \right] \\
&= \begin{bmatrix} s_1 r_1 \oplus t_1^1 & s_2 r_1 \oplus t_1^2 & \ldots & s_k r_1 \oplus t_1^k \\ s_1 r_2 \oplus t_2^1 & s_2 r_2 \oplus t_2^2 & \ldots & s_k r_2 \oplus t_2^k \\ \vdots & \vdots & \ddots & \vdots \\ s_1 r_m \oplus t_m^1 & s_2 r_m \oplus t_m^2 & \ldots & s_k r_m \oplus t_m^k \end{bmatrix} \\
&= \begin{bmatrix} r_1\mathbf{s} \oplus \mathbf{t}_1 \\ r_2\mathbf{s} \oplus \mathbf{t}_2 \\ \vdots \\ r_m\mathbf{s} \oplus \mathbf{t}_m \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \vdots \\ \mathbf{q}_m \end{bmatrix}
\end{aligned}
$$

KK Protocol

$\mathcal{R}$ *forms* $m \times k$ *matrices* T_0, T_1 *in the following way:*

– *Choose* $\mathbf{t}_{j,0}, \mathbf{t}_{j,1} \leftarrow \{0,1\}^k$ *at random such that* $\mathbf{t}_{j,0} \oplus \mathbf{t}_{j,1} = \mathbf{c}_{r_j}$.

Let $\mathbf{t}_0^i, \mathbf{t}_1^i$ *denote the i-th column of matrices* T_0, T_1 *respectively.*

$\mathcal{S}$ *forms* $m \times k$ *matrix* Q *such that the i-th column of* Q *is the vector* $\mathbf{q}^i$. *(Note* $\mathbf{q}^i = \mathbf{t}^i_{s_i}$.*) Let* $\mathbf{q}_j$ *denote the j-th row of* Q. *(Note* $\mathbf{q}_j = ((\mathbf{t}_{j,0} \oplus \mathbf{t}_{j,1}) \odot \mathbf{s}) \oplus \mathbf{t}_{j,0}$. *Simplifying,* $\mathbf{q}_j \oplus \mathbf{t}_{j,0} = \mathbf{c}_{r_j} \odot \mathbf{s}$.*)*

For $j \in [m]$ *and for every* $0 \le r < n$, $\mathcal{S}$ *sends* $y_{j,r} = x_{j,r} \oplus H(j, \mathbf{q}_j \oplus (\mathbf{c}_r \odot \mathbf{s}))$.

For $j \in [m]$, $\mathcal{R}$ *recovers* $z_j = y_{j,r_j} \oplus H(j, \mathbf{t}_{j,0})$.

Comparison

Protocol	Efficiency	Security Model	Assumptions
Beaver's	$\mathcal{O}(n \cdot poly(k))$ (for semi-honest)	semi-honest or malicious; static or adaptive	one-way functions
IKNP	$\mathcal{O}(n) + \mathcal{O}(k)$	static semi-honest	correlation-robust hash function
NNOB	$\mathcal{O}(n) + \mathcal{O}(k)$	static malicious	random oracle, IKNP
ZDE	$\mathcal{O}(n) + \mathcal{O}(k)$	static malicious	homomorphic hash function, IKNP
KK	$\mathcal{O}(n/\log(k)) + \mathcal{O}(k)$	static semi-honest	random oracle

Conclusion: Assumptions

- Assumptions
  - Information-theoretic extension is impossible
  - One-way function is the weakest possible assumption
  - Random oracle is most efficient assumption

Conclusion: Security Models

- Semi-honest vs Malicious
  - Semi-honest is highly efficient
  - Malicious is practical

- Static vs Adaptive
  - Static is highly efficient
  - Adaptive is impractical

Conclusion: Length

- Message Length
  - Possible to send shorter messages with improved efficiency

- Number of underlying Ots
  - Impossible to base extension on log(k) OTs