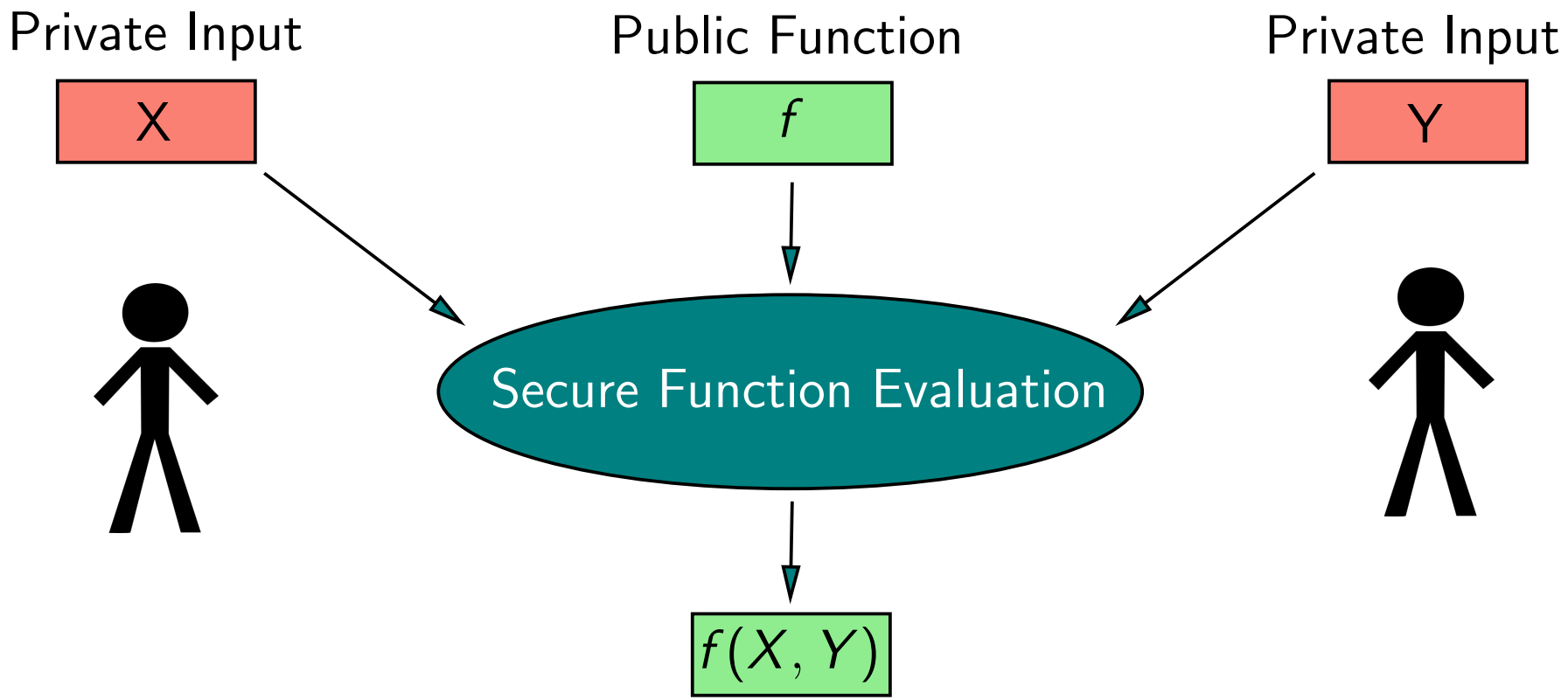
An Efficient Framework for Private Function Evaluation

Payman Mohassel Saeed Sadeghian

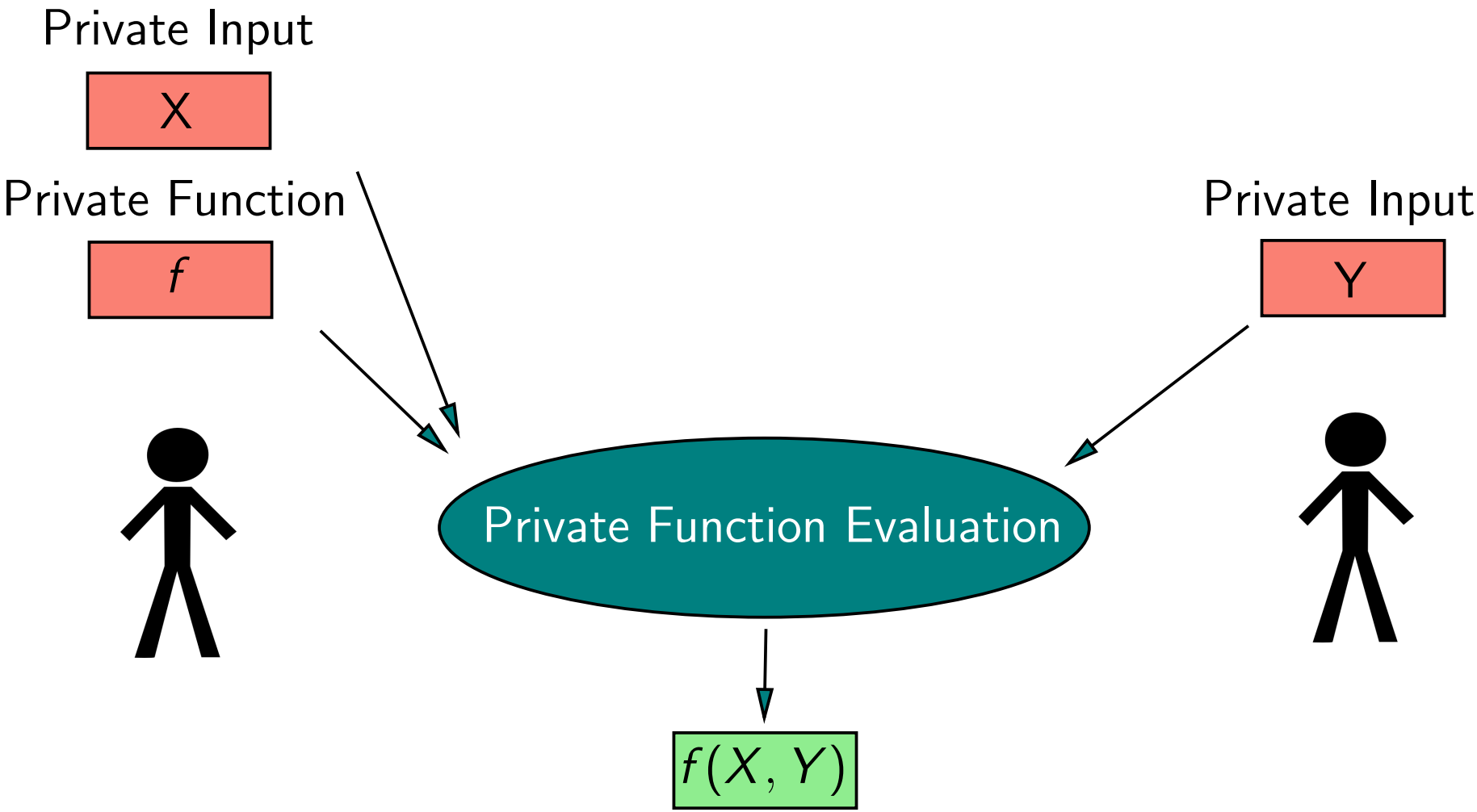
University of Calgary

Eurocrypt 2013

Secure Function Evaluation



Private Function Evaluation



Motivation

- Revealing the function can compromise privacy
- Revealing the function can reveal security vulnerability
- Intellectual property: Genetic tests, credit history check,...
- Leakage from output?

Private Function Evaluation: Universal Circuit approach

- General SFE: Convert function $f \rightarrow$ Circuit C and securely evaluate

Private Function Evaluation: Universal Circuit approach

- General SFE: Convert function $f \rightarrow$ Circuit C and securely evaluate
- General PFE: Build Universal circuit handling the circuits with the desired size and use general SFE
- Valiant [76]: universal circuit of size $O(g \log g)$ to handle circuits of size g.

Private Function Evaluation: Questions?

- *Can we achieve better concrete efficiency?*

Private Function Evaluation: Questions?

- *Can we achieve better concrete efficiency?*
- *Can we achieve the same asymptotic efficiency as SFE?*

Our Contribution

- A general framework for designing PFE
  - Modular design, based on abstract functionalities

Our Contribution

- A general framework for designing PFE
  - Modular design, based on abstract functionalities
- Applying our framework to several well-known MPC constructions

Our Contribution

- A general framework for designing PFE
  - Modular design, based on abstract functionalities
- Applying our framework to several well-known MPC constructions
- A novel protocol for *oblivious switching network evaluation* (OSN)
  - A generalization of the oblivious shuffling problem
  - Based on OT and non-crypto operations

Current state

Semi-honest

They all leak the circuit size and input length.

- Two-Party PFE: Linear solution of [KM11]
- Multi-party PFE: no customized construction, only universal circuit based approaches with at best $O(m^2 g \log g)$ complexity
- Arithmetic circuit PFE: no customized solution, except for arithmetic universal circuits with best known $O(g^5)$ gates

Current state

Semi-honest

They all leak the circuit size and input length.

- Two-Party PFE: Linear solution of [KM11]
- Multi-party PFE: no customized construction, only universal circuit based approaches with at best $O(m^2 g \log g)$ complexity
- Arithmetic circuit PFE: no customized solution, except for arithmetic universal circuits with best known $O(g^5)$ gates

Our work:

- Two-Party PFE: More efficient solutions.
- Multi-party PFE: We present the first linear multi-party PFE based on GMW.
- Arithmetic circuit PFE: We present the first linear arithmetic PFE.

Our Framework for PFE

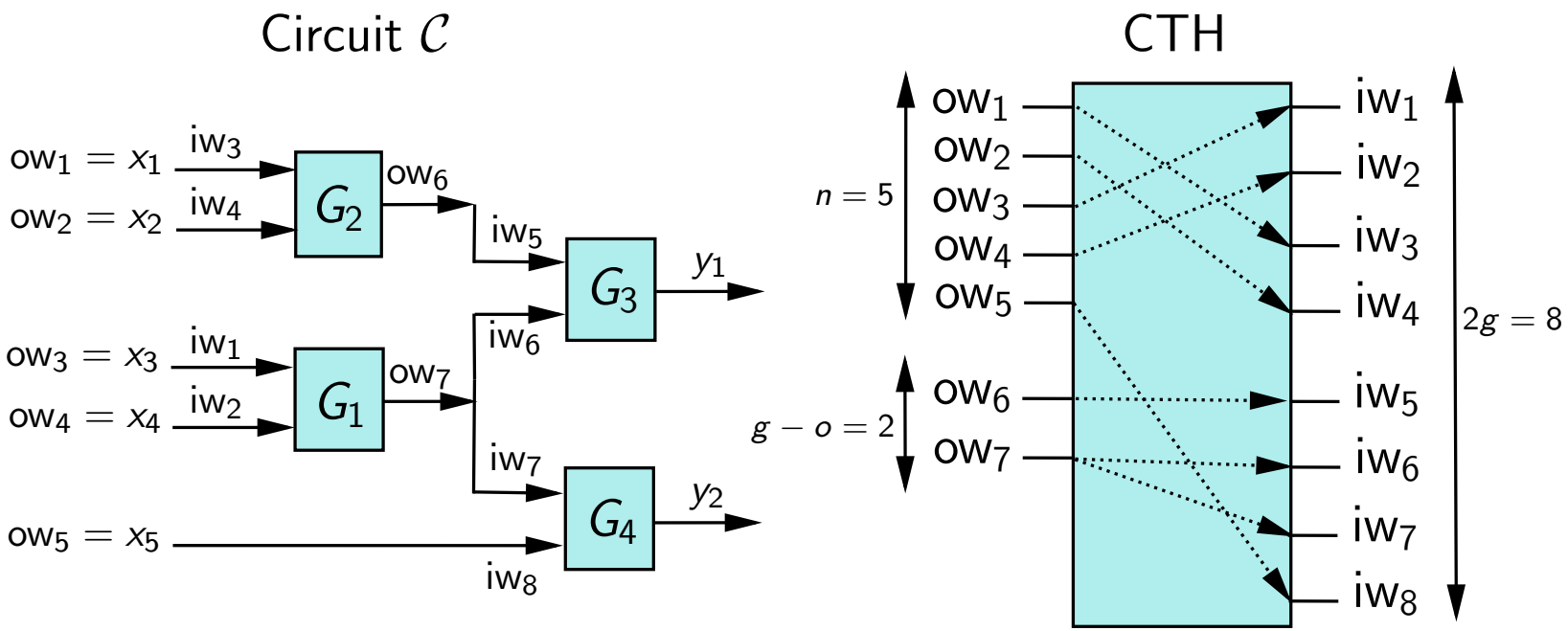
- We need to hide two types of information about the circuit:
  1. *Circuit topology*
  2. *Function of the gates* in the circuit

Our Framework for PFE

- We need to hide two types of information about the circuit:
  1. *Circuit topology*
  2. *Function of the gates* in the circuit
- We divide the task of private function evaluation into two different functionalities:
  1. The Circuit Topology Hiding (CTH) functionality
  2. The Private Gate Evaluation (PGE) functionality

Circuit Topology Hiding (CTH) functionality

- *Observation:* the topology of a circuit $\mathcal{C}$ can be fully described using a mapping $\pi_{\mathcal{C}} : \{1 \dots |\mathsf{OW}|\} \rightarrow \{1 \dots |\mathsf{IW}|\}$
  - OMAP query
  - Reveal query
  - We refer to this mapping as an *extended permutation* (EP)
- Inputs are shares and outputs are freshly generated shares.



The Private Gate Evaluation (PGE) functionality

- The PGE functionality :
  - PFE for a single gate
  - Shared inputs and output

Framework

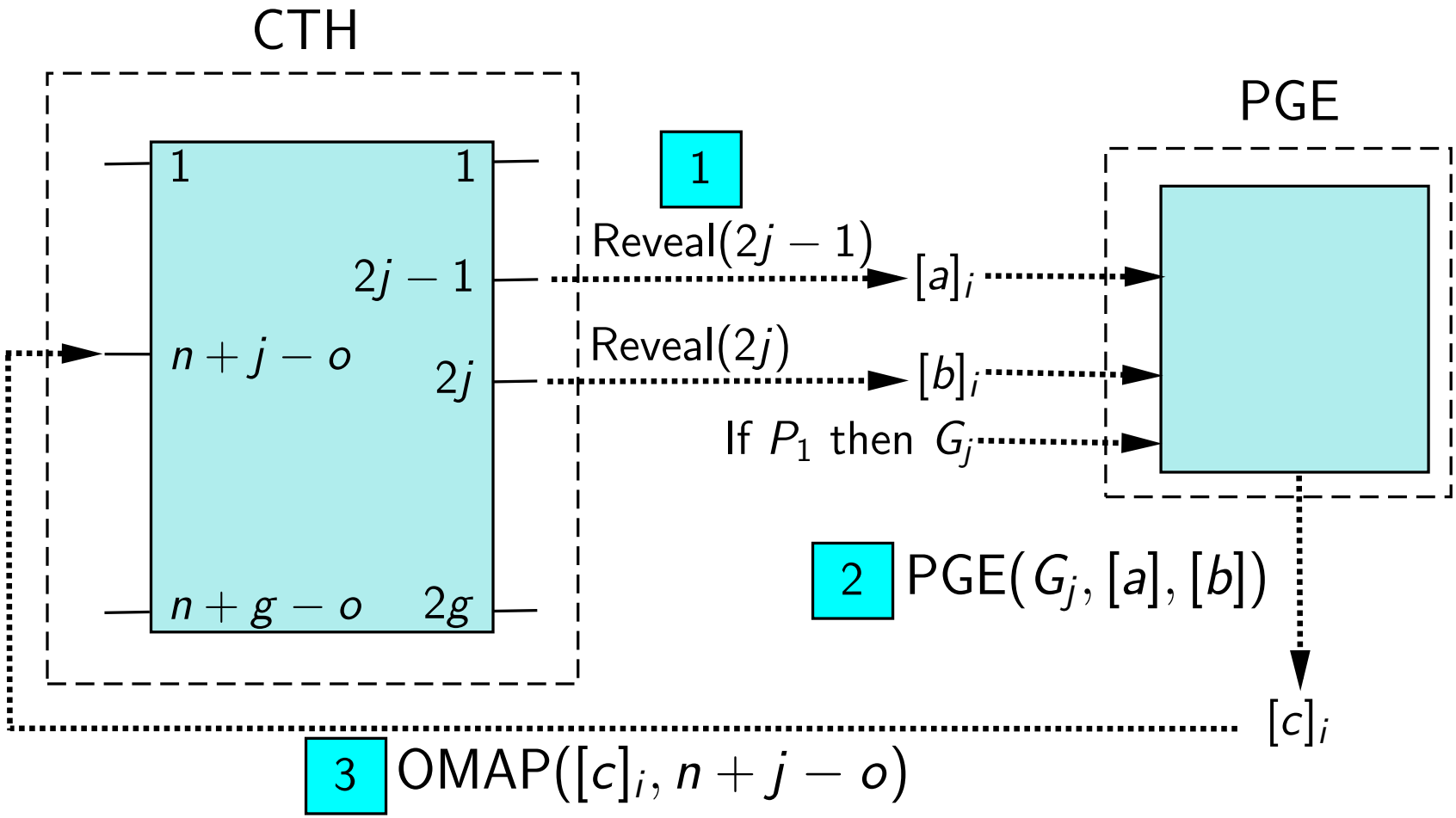
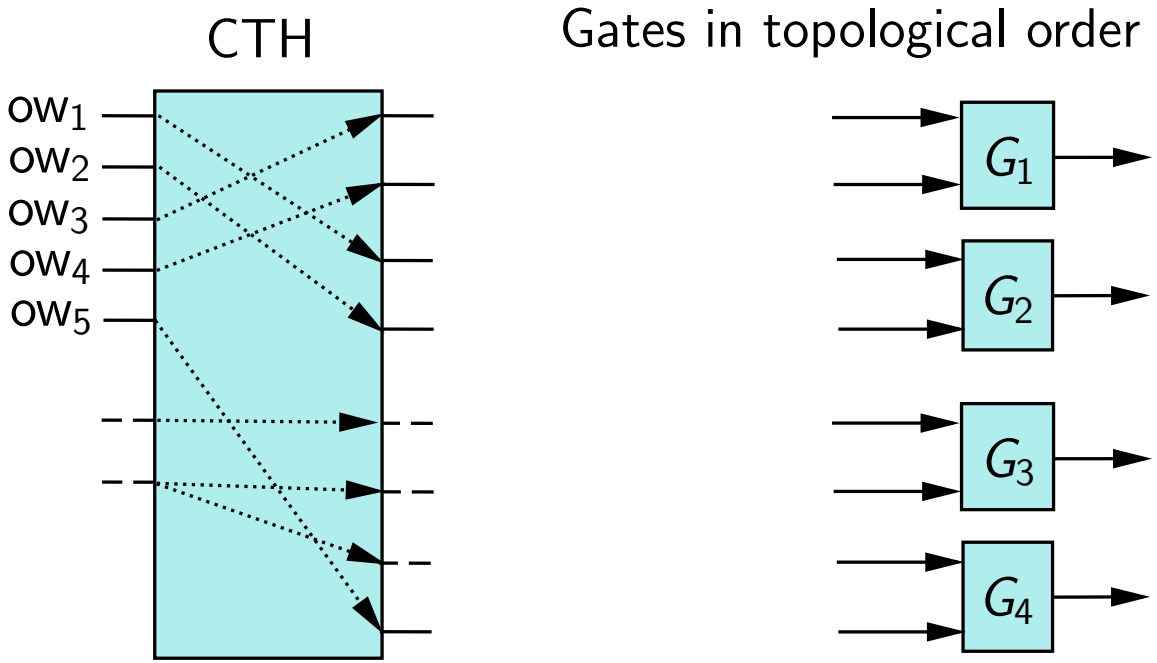
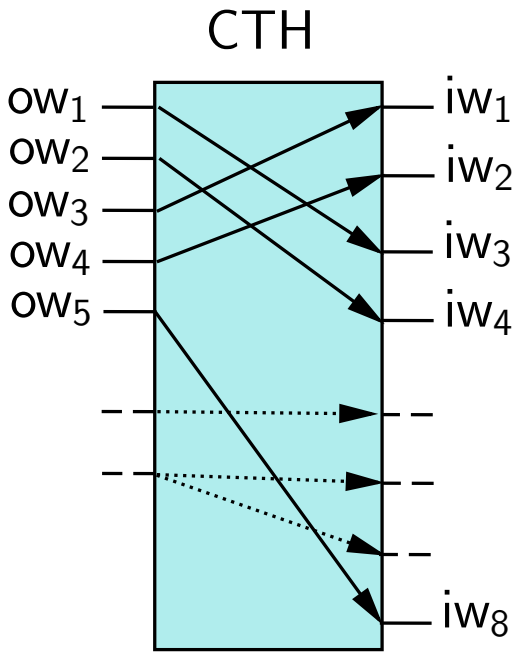


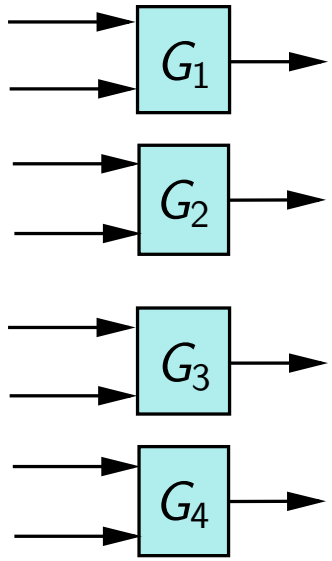
Figure : Steps of framework for party i and the jth gate in a topological order.

Framework: Initialization



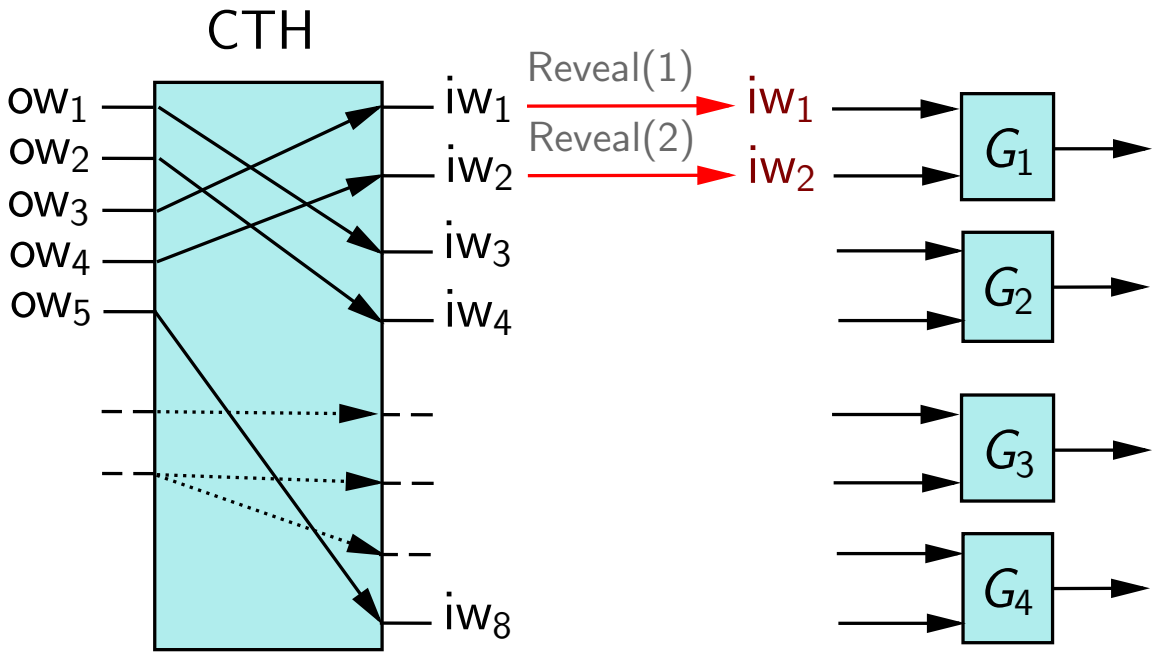
Framework: Initialization





$\mathrm{OMAP}([x_1]_i, 1)$
$\mathrm{OMAP}([x_2]_i, 2)$
$\vdots$
$\mathrm{OMAP}([x_5]_i, 5)$

Framework: Private Function Evaluation

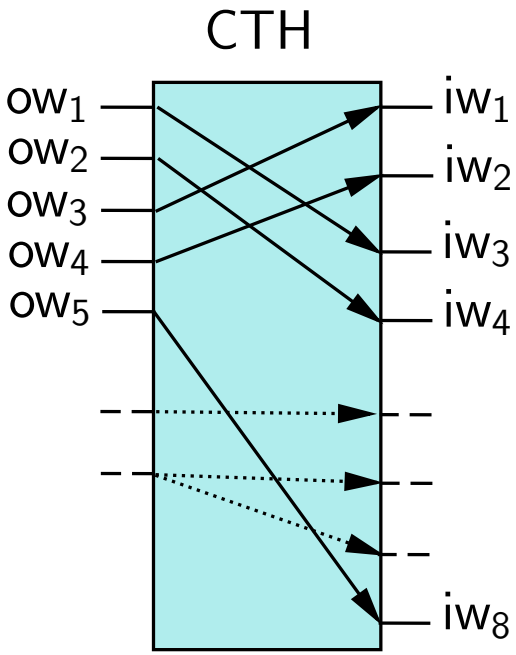


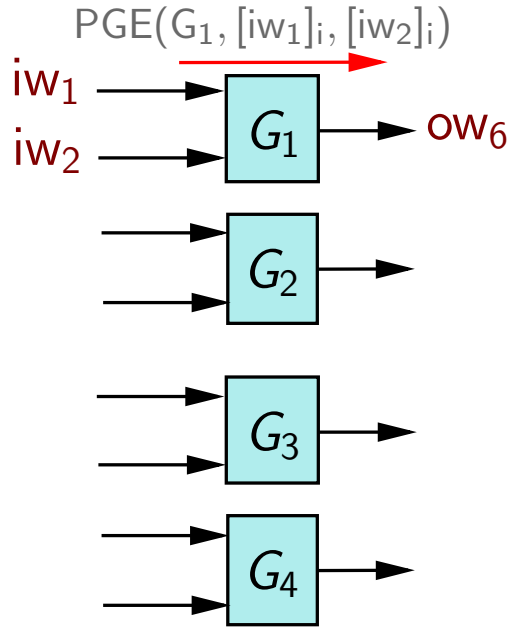
1 Reveal$(2j-1)$

2 PGE$(G_j, [a], [b])$

3 OMAP$([c]_i, n+j-o)$

Framework: Private Function Evaluation



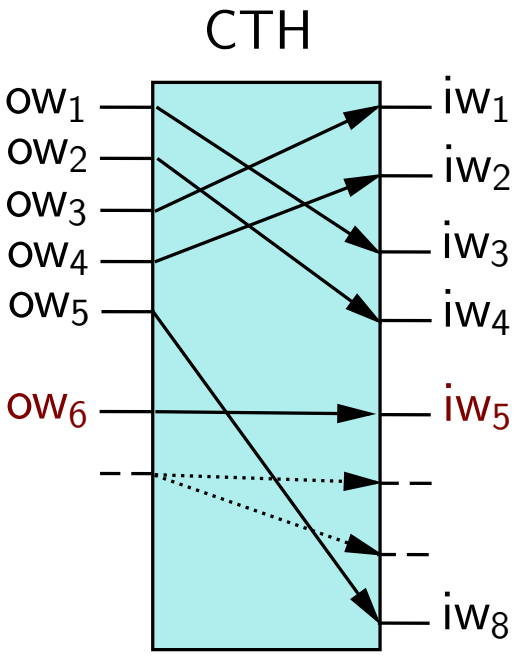


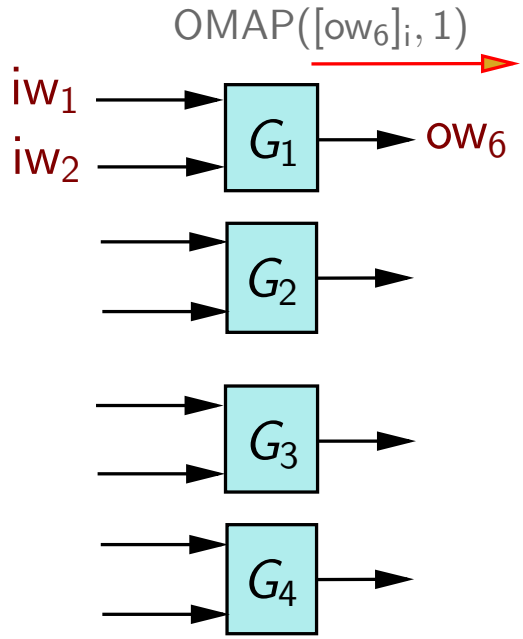
1 $\mathrm{Reveal}(2j-1)$

2 $\mathrm{PGE}(G_j, [a], [b])$

3 $\mathrm{OMAP}([c]_i, n+j-o)$

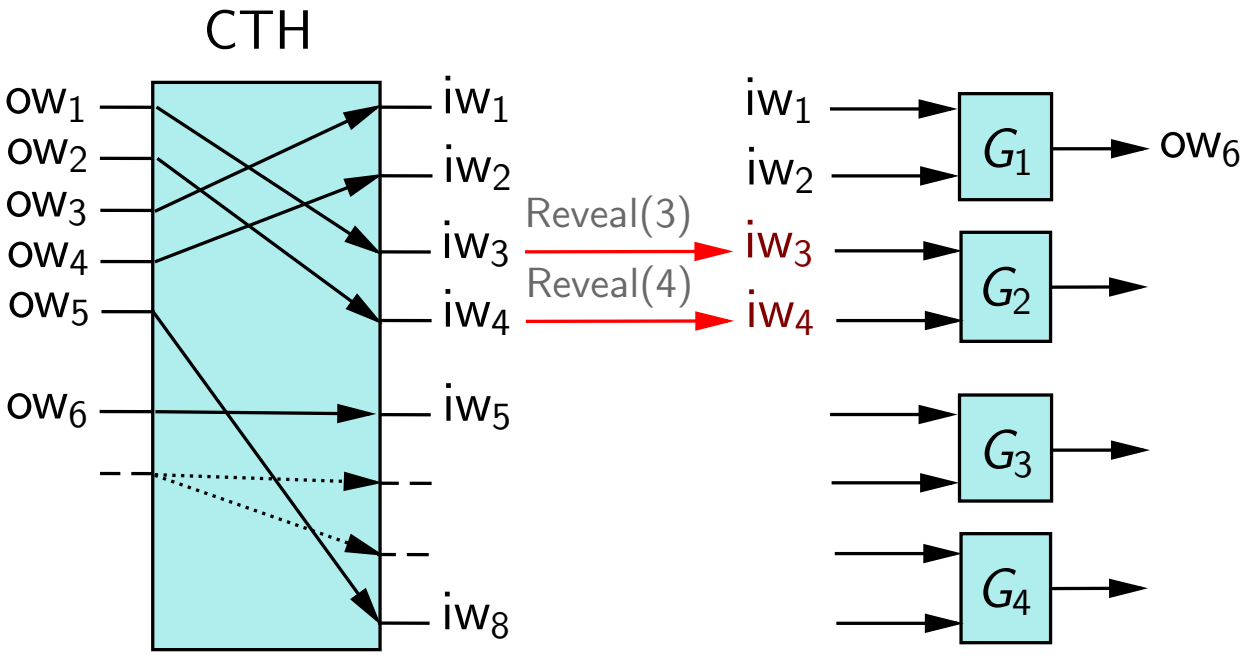
Framework: Private Function Evaluation





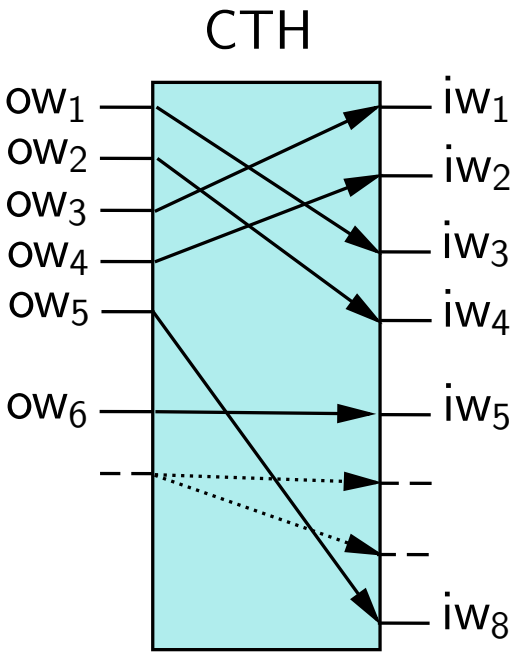
1. $\mathrm{Reveal}(2j-1)$
2. $\mathrm{PGE}(G_j, [a], [b])$
3. $\mathrm{OMAP}([c]_i, n+j-o)$

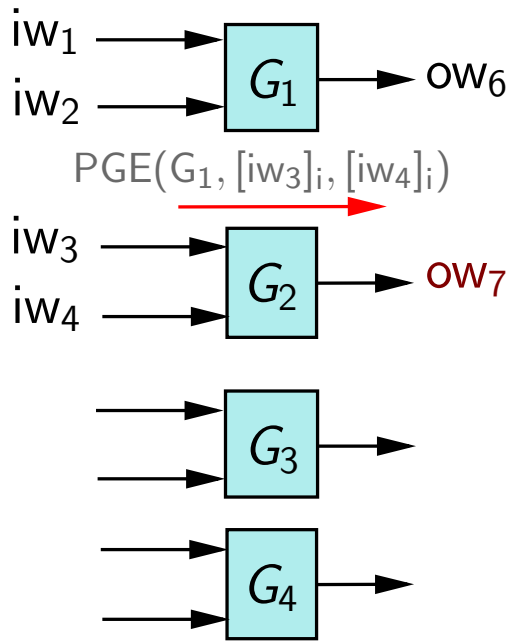
Framework: Private Function Evaluation



1. Reveal$(2j-1)$
2. PGE$(G_j, [a], [b])$
3. OMAP$([c]_i, n+j-o)$

Framework: Private Function Evaluation



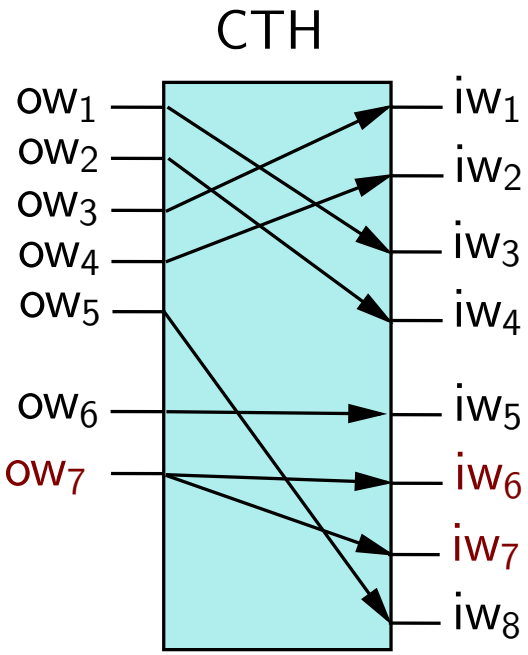


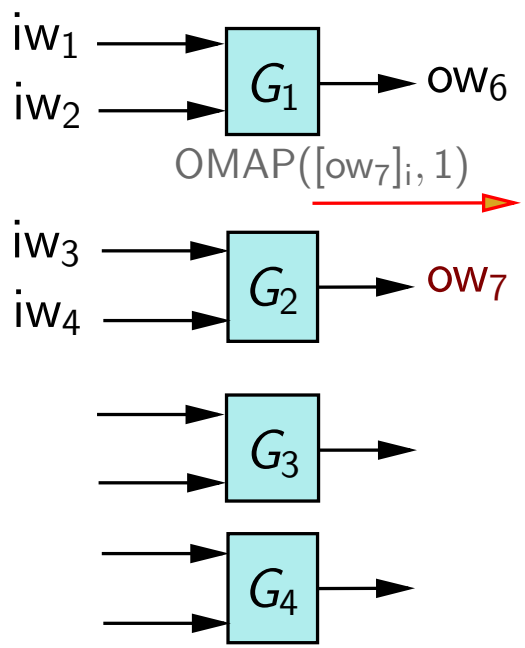
1 Reveal$(2j-1)$

2 PGE$(G_j, [a], [b])$

3 OMAP$([c]_i, n+j-o)$

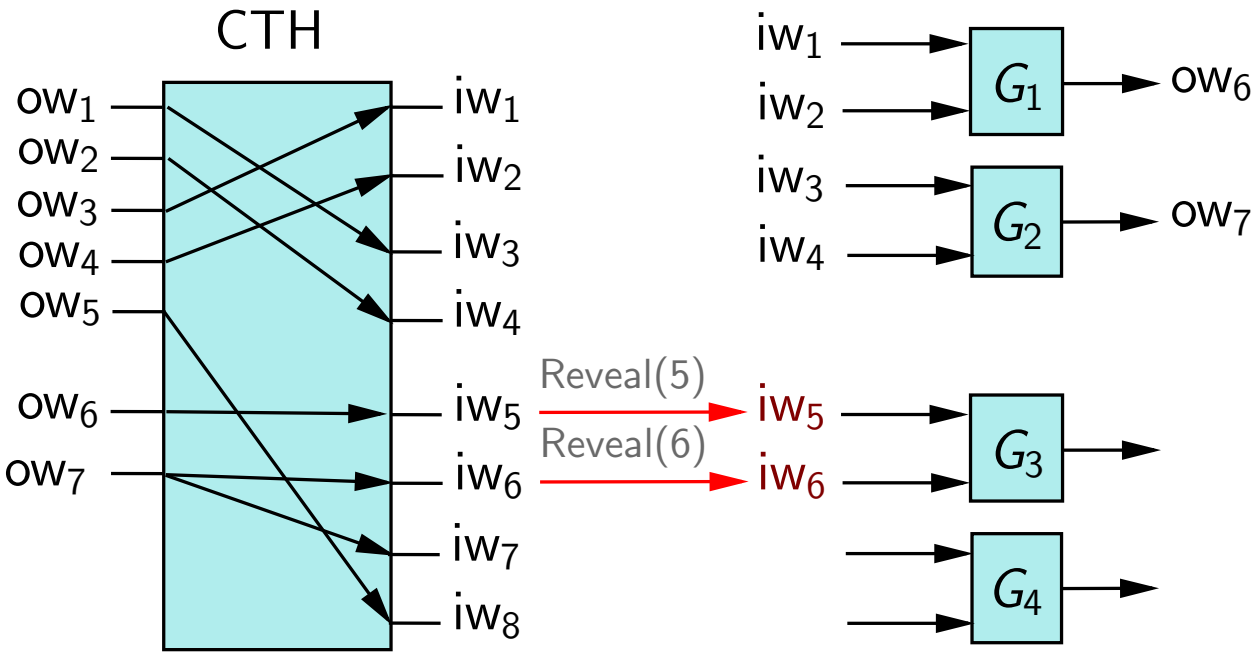
Framework: Private Function Evaluation





1. $\mathrm{Reveal}(2j-1)$
2. $\mathrm{PGE}(G_j, [a], [b])$
3. $\mathrm{OMAP}([c]_i, n+j-o)$

Framework: Private Function Evaluation

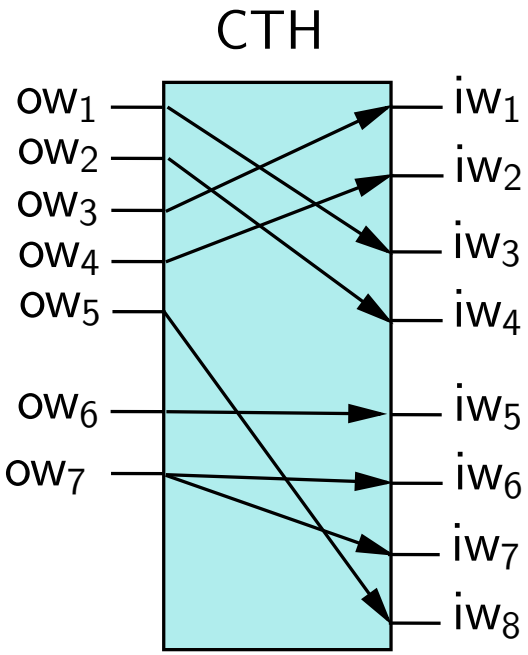


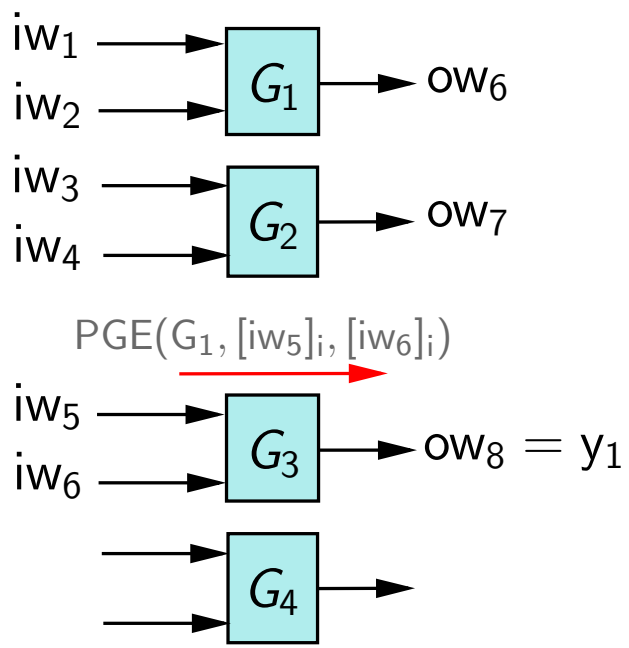
1 Reveal$(2j-1)$

2 PGE$(G_j, [a], [b])$

3 OMAP$([c]_i, n+j-o)$

Framework: Private Function Evaluation



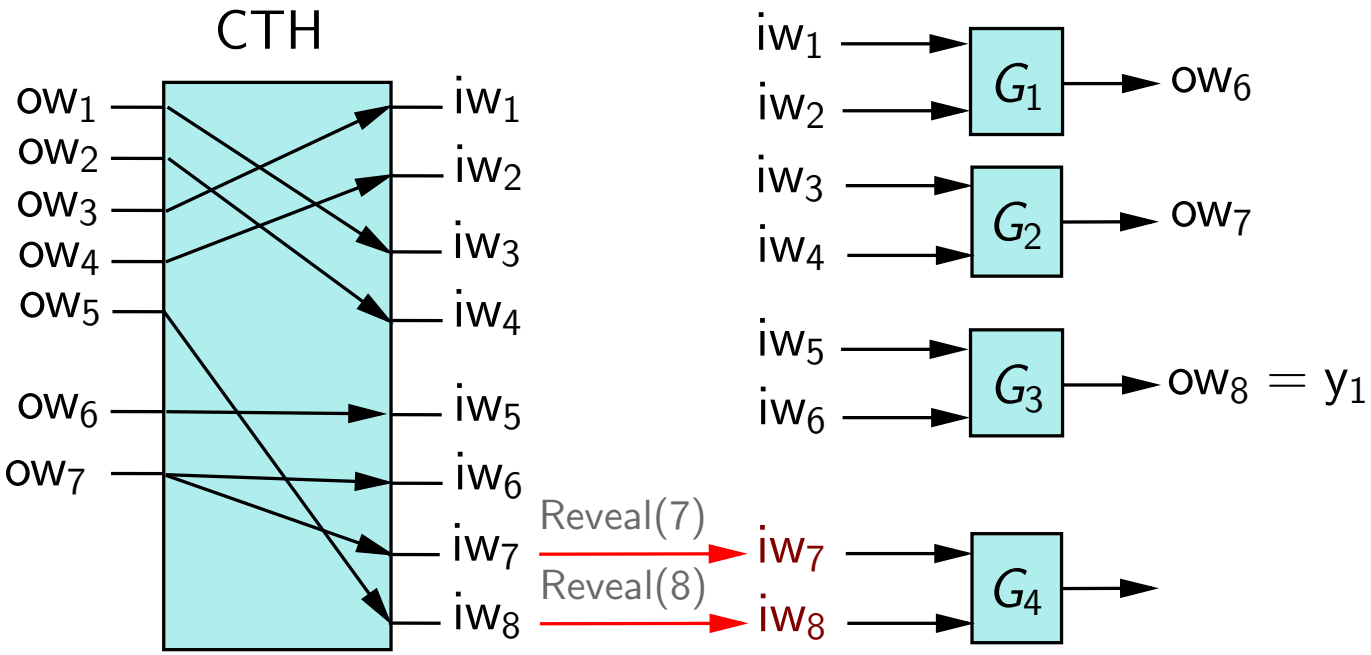


1 Reveal$(2j-1)$

2 PGE$(G_j, [a], [b])$

3 OMAP$([c]_i, n+j-o)$

Framework: Private Function Evaluation

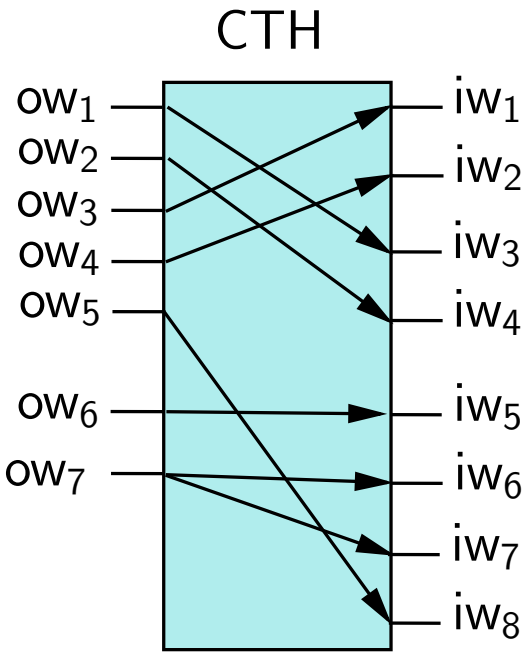


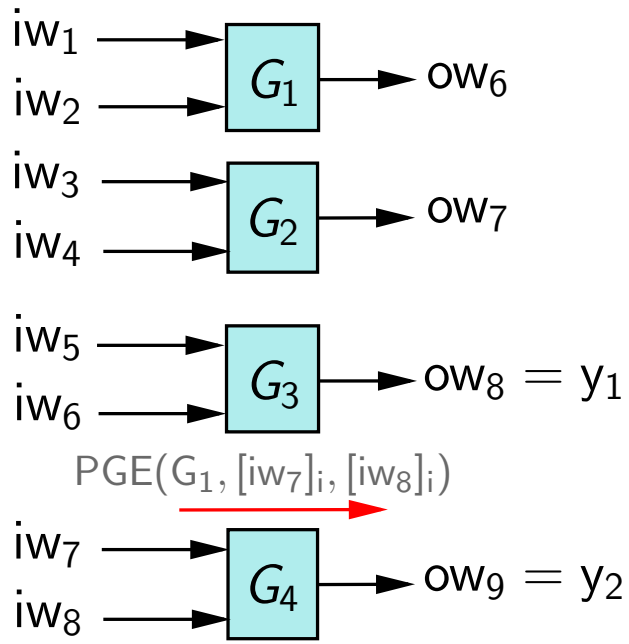
1 Reveal$(2j-1)$

2 PGE$(G_j, [a], [b])$

3 OMAP$([c]_i, n+j-o)$

Framework: Private Function Evaluation





1 Reveal$(2j-1)$

2 PGE$(G_j, [a], [b])$

3 OMAP$([c]_i, n+j-o)$

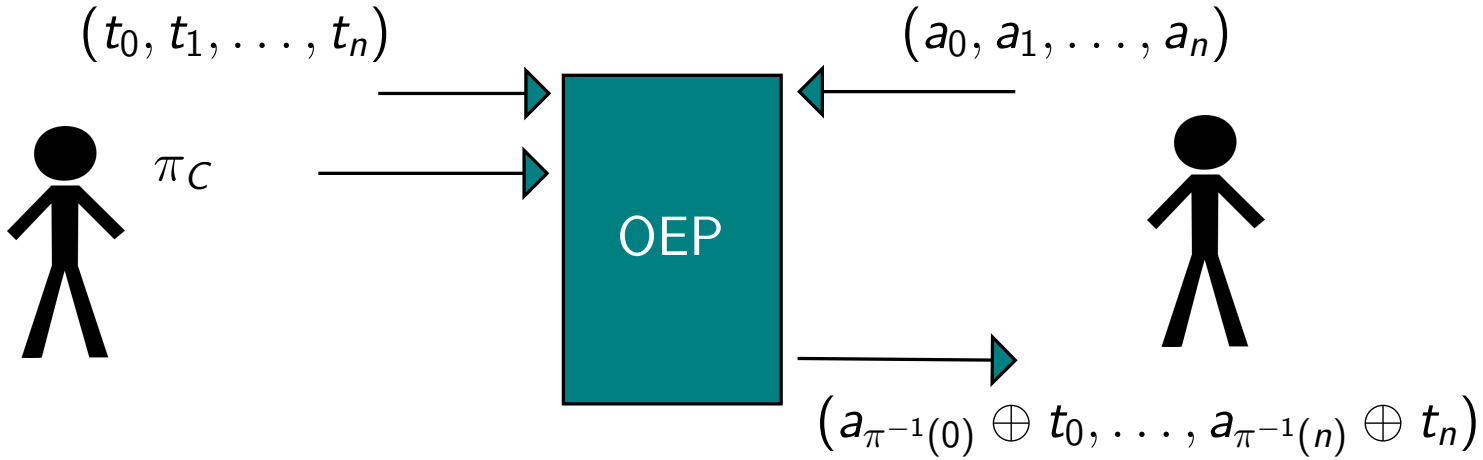
Realization of CTH functionality

- As discussed we refer to the mapping $\pi_{\mathcal{C}} : \{1 \dots |\mathsf{OW}|\} \rightarrow \{1 \dots |\mathsf{IW}|\}$ an *extended permutation* (EP)

Realization of CTH functionality

- As discussed we refer to the mapping $\pi_{\mathcal{C}} : \{1 \dots |\mathrm{OW}|\} \rightarrow \{1 \dots |\mathrm{IW}|\}$ an *extended permutation* (EP)
- A main component of our $\mathcal{F}_{\mathcal{CTH}}$ realization is a protocol for *oblivious evaluation* of this extended permutation (OEP)

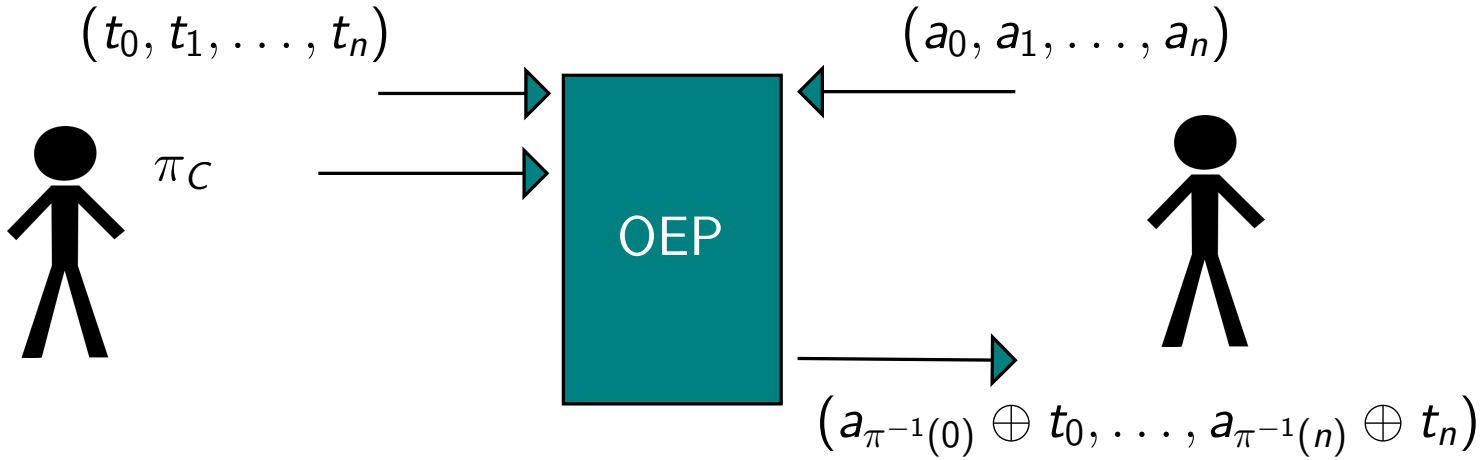
OEP Protocol



Realization of CTH functionality

- As discussed we refer to the mapping $\pi_{\mathcal{C}} : \{1 \dots |\mathsf{OW}|\} \rightarrow \{1 \dots |\mathsf{IW}|\}$ an *extended permutation* (EP)
- A main component of our $\mathcal{F}_{\mathcal{CTH}}$ realization is a protocol for *oblivious evaluation* of this extended permutation (OEP)

OEP Protocol



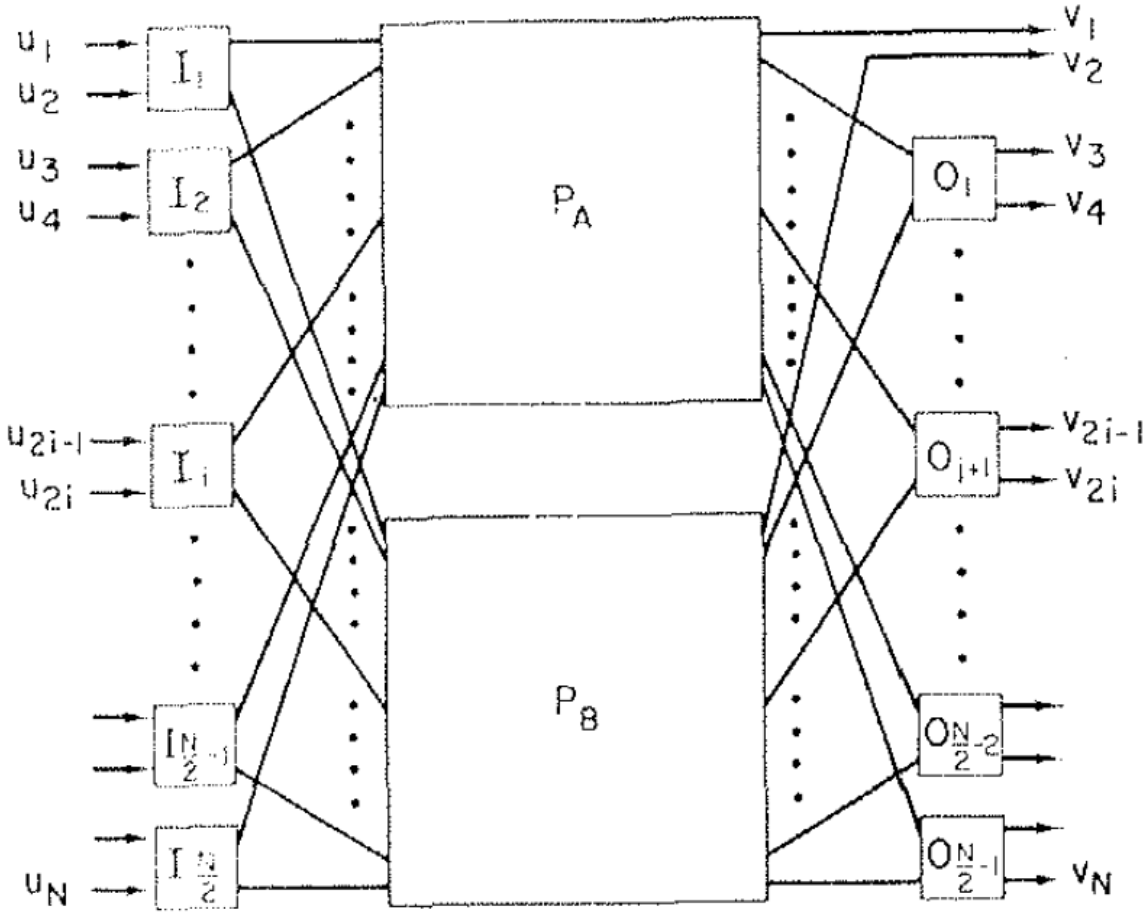
- HE-OEP: Can be realized using a singly homomorphic encryption

SN-OEP: OEP via Generalized Switching Networks

- Implement an extended permutation using a generalized switching network SN

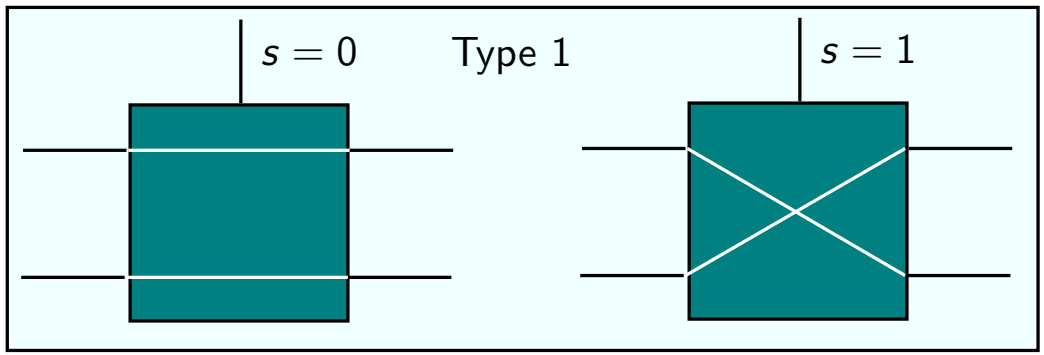
SN-OEP: OEP via Generalized Switching Networks

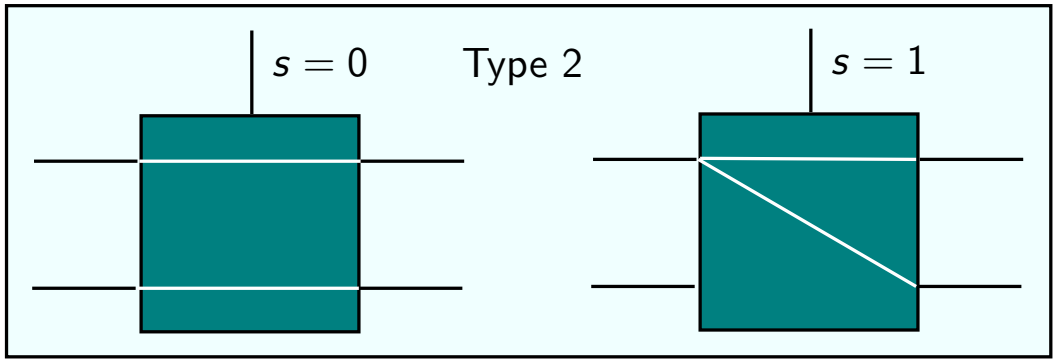
- Implement an extended permutation using a generalized switching network SN
- Obliviously evaluate the generalized switching network



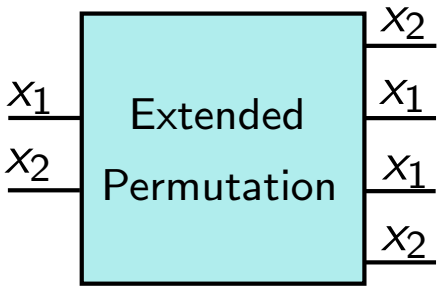
Generalized Switch

- In our generalized notion of a switch, each of the two output strings can take the value of each of the two input strings.
- We use the following types of switches in our EP construction:

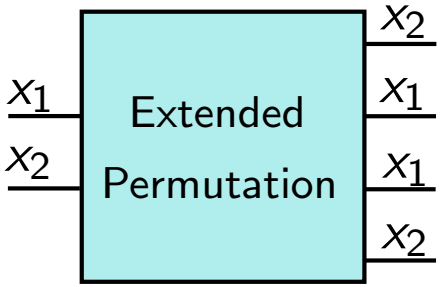


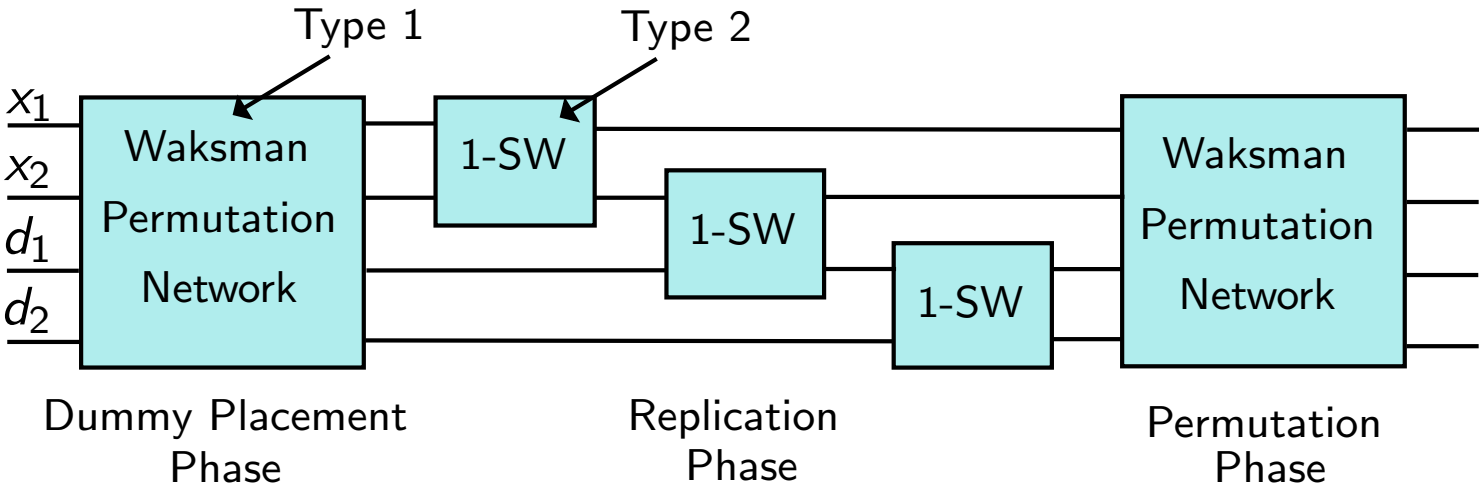


Extended Permutation Network

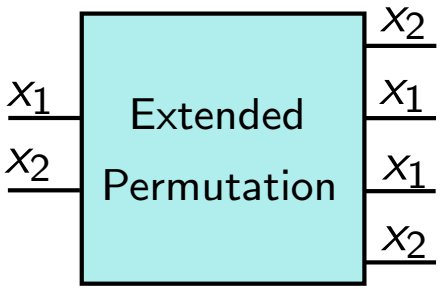


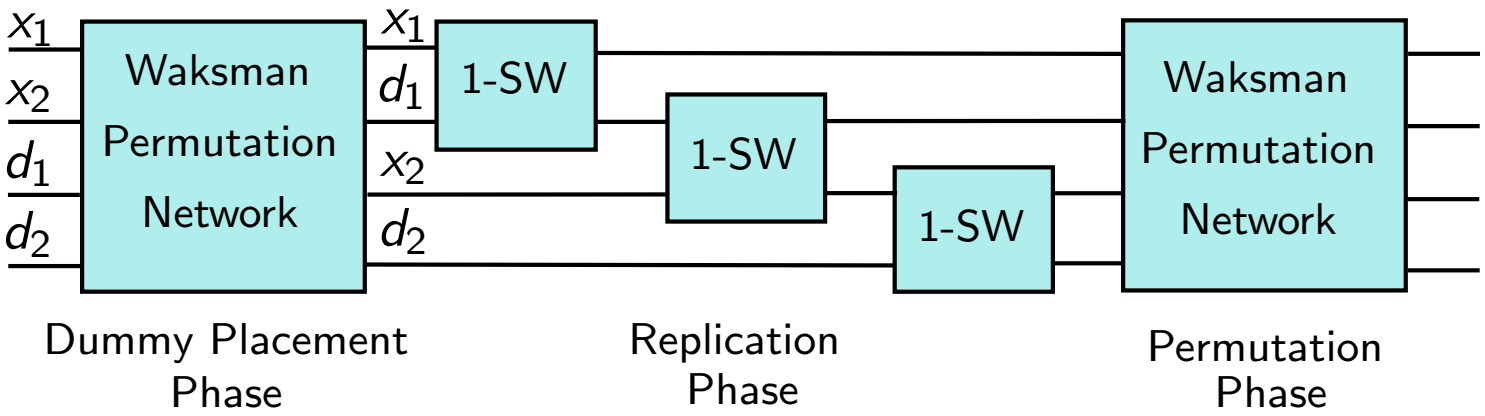
Extended Permutation Network



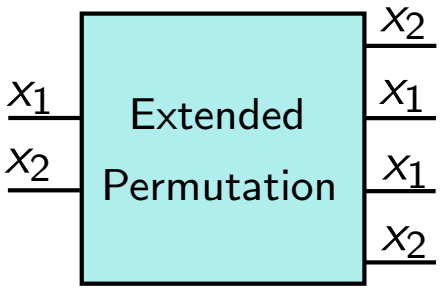


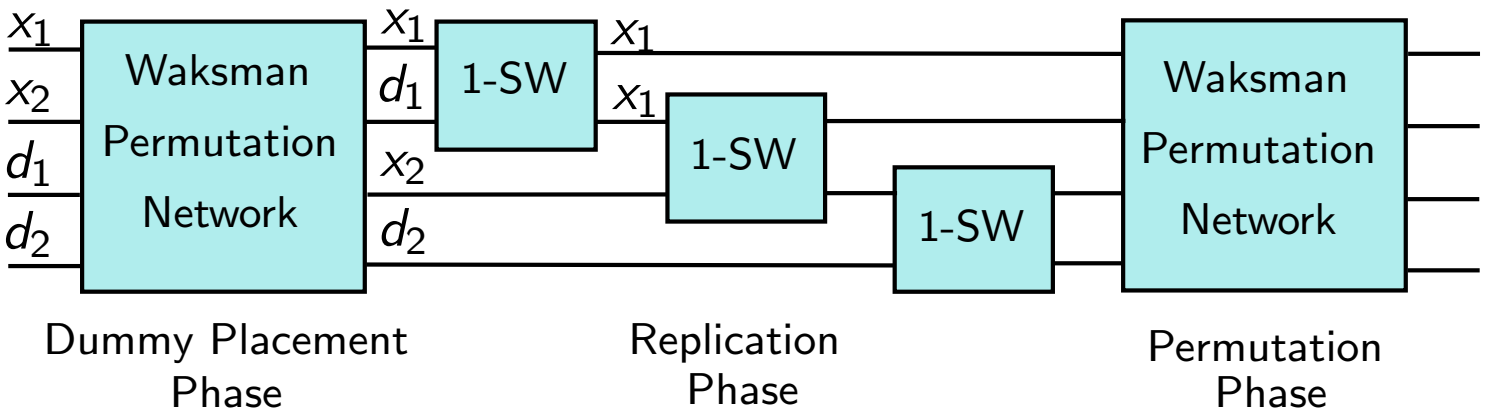
Extended Permutation Network



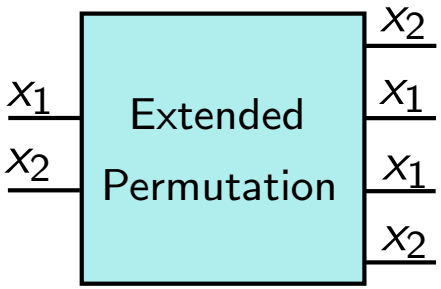


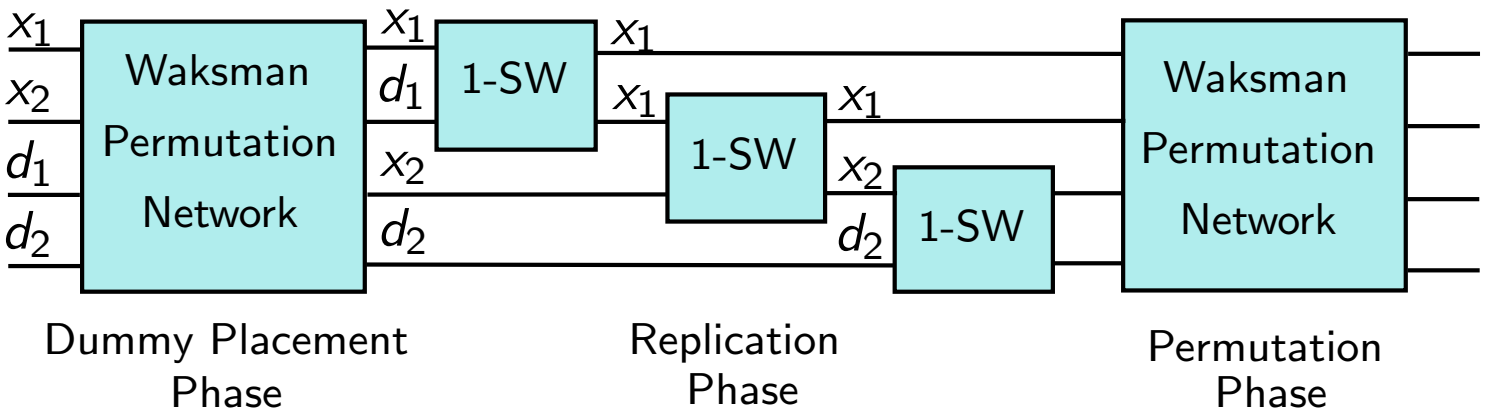
Extended Permutation Network



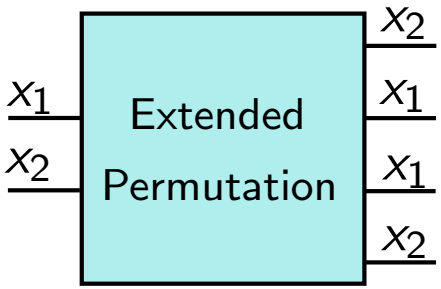


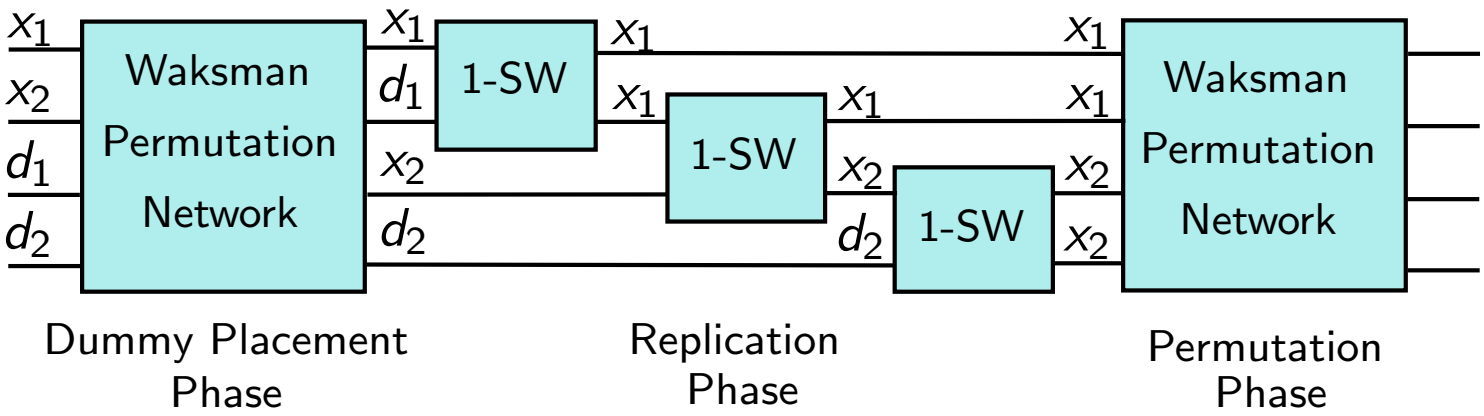
Extended Permutation Network



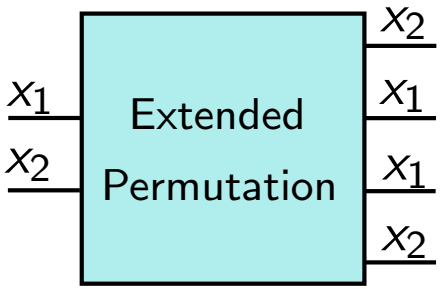


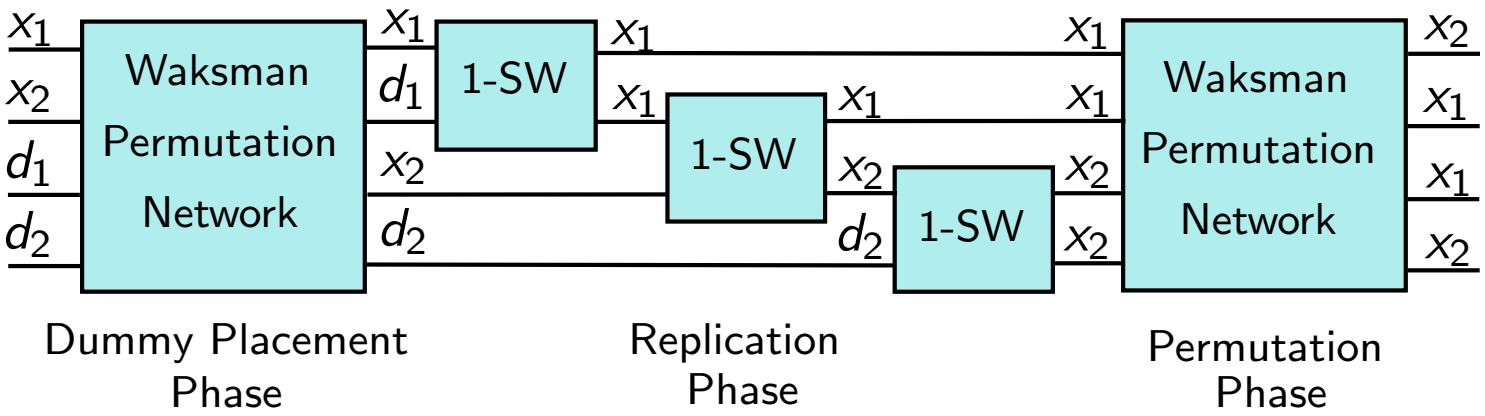
Extended Permutation Network



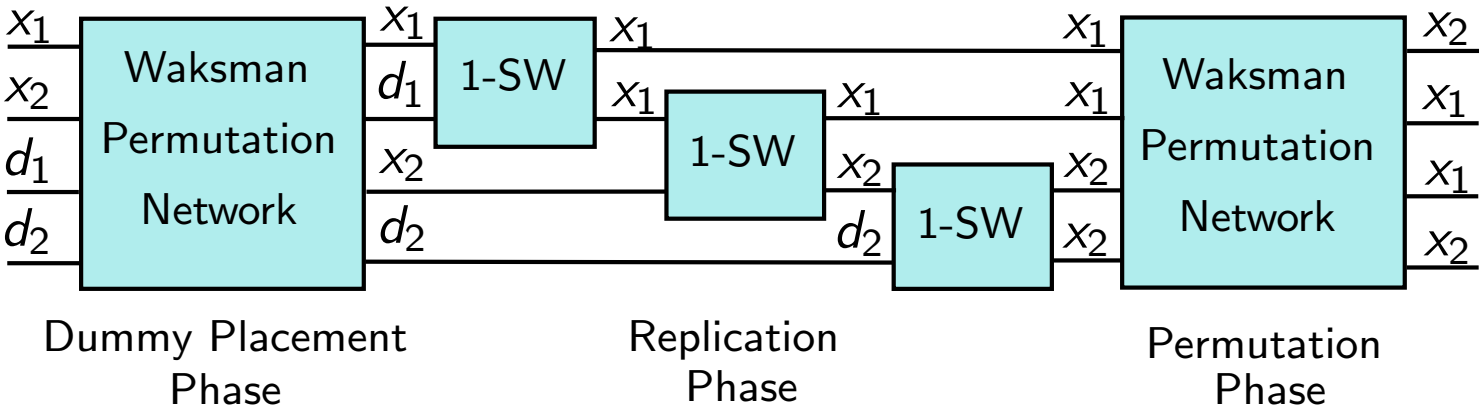


Extended Permutation Network



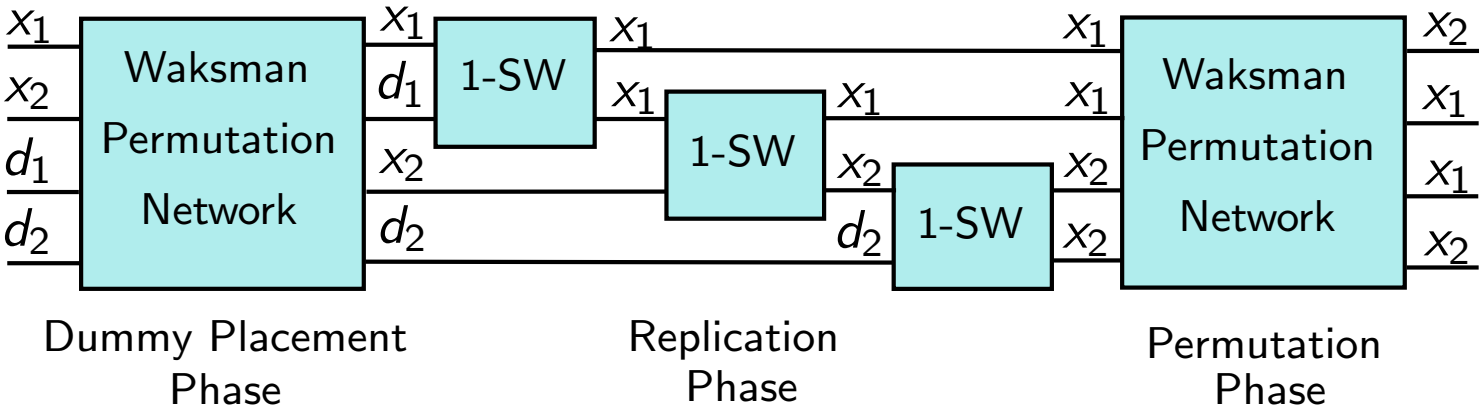


Extended Permutation Network



- Size of Waksman network: $N \log N - N + 1$.

Extended Permutation Network

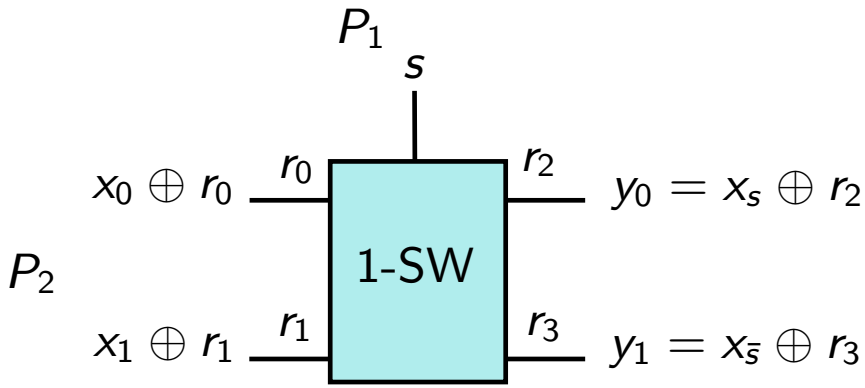


- Size of Waksman network: $N \log N - N + 1$.
- Size of EP network: $2(N \log N - N + 1) + N - 1 = 2N \log N - N + 1$

Oblivious Evaluation of Switching Network

Oblivious Switching Network Evaluation Protocol

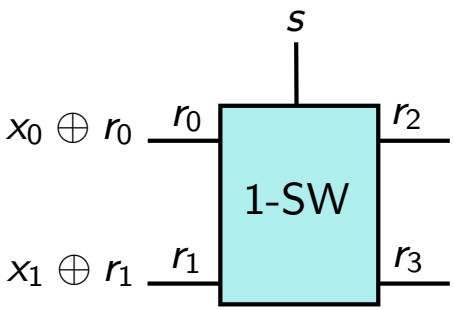
- Evaluate each switch



- Extend to the whole network

Oblivious Evaluation of Switch

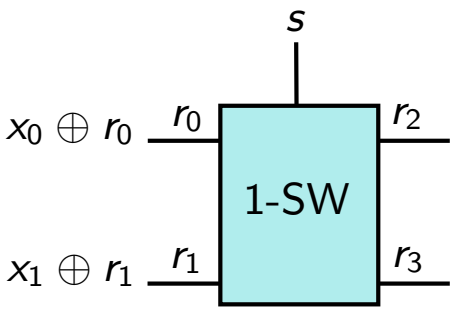
- P_2 generates random values for each wire and prepares the following table and inputs. He then sends the inputs to P_1.



s		
0	$r_0 \oplus r_2$	$r_1 \oplus r_3$
1	$r_1 \oplus r_2$	$r_0 \oplus r_3$

Oblivious Evaluation of Switch

- P_2 generates random values for each wire and prepares the following table and inputs. He then sends the inputs to P_1.

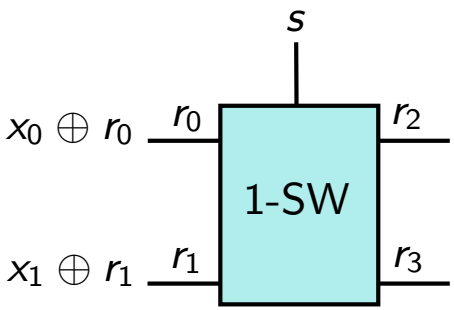


s		
0	$r_0 \oplus r_2$	$r_1 \oplus r_3$
1	$r_1 \oplus r_2$	$r_0 \oplus r_3$

- P_1 and P_2 engage in 1-out-of-2 oblivious transfer protocol where P_1 acts as a receiver with input s, and P_2 acts as a sender with his table as inputs.

Oblivious Evaluation of Switch

- P_2 generates random values for each wire and prepares the following table and inputs. He then sends the inputs to P_1.



s		
0	$r_0 \oplus r_2$	$r_1 \oplus r_3$
1	$r_1 \oplus r_2$	$r_0 \oplus r_3$

- P_1 and P_2 engage in 1-out-of-2 oblivious transfer protocol where P_1 acts as a receiver with input s, and P_2 acts as a sender with his table as inputs.
- P_1 applies the switch and sends back the received inputs XORed with result of OT and his blinding. For $s = 0$ and y_0:

$$y_0 = (x_0 \oplus r_0) \oplus (r_0 \oplus r_2) = x_0 \oplus r_2$$

Improved Oblivious Shuffling

- OSN is a generalization of the previously studied problems such as oblivious shuffling [HEK12].

Improved Oblivious Shuffling

- OSN is a generalization of the previously studied problems such as oblivious shuffling [HEK12].
- A subprotocol used for private set intersection

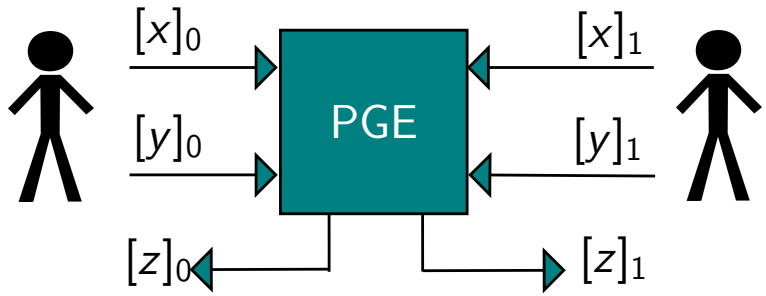
Oblivious Shuffling Prot.	Asymptotic Complexity	Eff. Gain
HE-Based	$O(N)$ Asym.	175
GC-Based [HEK12]	$(\frac{4\ell(N \log N - N + 1)}{3} + 2N\ell)$ Sym. $+ O(k)$ Asym.	25
OSN-Based (This work)	$(2N \log N - 2N + 2)$ Sym. $+ O(k)$ Asym.	1

Multi-Party PFE

- CTH Realization: Multi-party variant of OEP (parallel instances between P_1 and P_i)

Multi-Party PFE

- CTH Realization: Multi-party variant of OEP (parallel instances between P_1 and P_i)
- PGE Realization: Multi-party OT (e.g. [FGM07])
  - For two-party case:

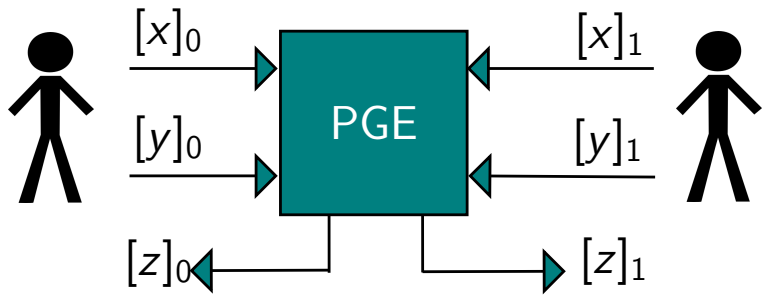


  - Create this table and use OT with the shares:

$[x]_1$	$[y]_1$	Output
0	0	$[z]_0 \oplus f([x]_0 \oplus 0, [y]_0 \oplus 0)$
0	1	$[z]_0 \oplus f([x]_0 \oplus 0, [y]_0 \oplus 1)$
1	0	$[z]_0 \oplus f([x]_0 \oplus 1, [y]_0 \oplus 0)$
1	1	$[z]_0 \oplus f([x]_0 \oplus 1, [y]_0 \oplus 1)$

Multi-Party PFE

- CTH Realization: Multi-party variant of OEP (parallel instances between P_1 and P_i)
- PGE Realization: Multi-party OT (e.g. [FGM07])
  - For two-party case:



  - Create this table and use OT with the shares:

$[x]_1$	$[y]_1$	Output
0	0	$[z]_0 \oplus f([x]_0 \oplus 0, [y]_0 \oplus 0)$
0	1	$[z]_0 \oplus f([x]_0 \oplus 0, [y]_0 \oplus 1)$
1	0	$[z]_0 \oplus f([x]_0 \oplus 1, [y]_0 \oplus 0)$
1	1	$[z]_0 \oplus f([x]_0 \oplus 1, [y]_0 \oplus 1)$

- Plug into the framework

Multi-Party PFE

Multi-Party PFE	Complexity
[KS08] Universal Circuits	$O(m^2 g \log^2 g)$ Sym. + $O(k)$ Asym.
[Val76] Universal Circuits	$O(m^2 g \log g)$ Sym. + $O(k)$ Asym.
GMW-PFE (SN-OEP)	$O(m^2 g + mg \log g)$ Sym. + $O(k)$ Asym.
GMW-PFE (HE-OEP)	$O(m^2 g)$ Sym. + $O(mg)$ HE. + $O(k)$ Asym.

Constant round Two-Party PFE

2-Party PFE	Complexity
[KS08]	$(1.5g \log^2 g)$ sym. + $O(k)$ Asym.
[Val76]	$(19g \log g)$ sym. + $O(k)$ Asym.
[KM11]	$O(g)$ Sym. + $O(g)$ (HE+HM+HA) + $O(k)$ Asym.
Yao-PFE (HE-OEP)	$O(g)$ Sym. + $O(g)$ (HE+HA) + $O(k)$ Asym.
Yao-PFE (SN-OEP)	$(8g \log 2g)$ Sym. + $O(k)$ Asym.

PFE for Arithmetic Circuits

- CTH Realization: HE-OEP

PFE for Arithmetic Circuits

- CTH Realization: HE-OEP
- PGE Realization: Our PGE realization based on singly homomorphic encryption (requires a constant number of public-key operations)

PFE for Arithmetic Circuits

- CTH Realization: HE-OEP
- PGE Realization: Our PGE realization based on singly homomorphic encryption (requires a constant number of public-key operations)
- Plug into the framework

PFE for Arithmetic Circuits

- CTH Realization: HE-OEP
- PGE Realization: Our PGE realization based on singly homomorphic encryption (requires a constant number of public-key operations)
- Plug into the framework

Efficiency

$O(g)$ public key operations

PFE for Arithmetic Circuits

- CTH Realization: HE-OEP
- PGE Realization: Our PGE realization based on singly homomorphic encryption (requires a constant number of public-key operations)
- Plug into the framework

Efficiency

$O(g)$ public key operations

Universal Arithmetic Circuits

While universal arithmetic circuits exist, their sizes are too large for any practical purpose (e.g. as high as $O(g^5)$).

Conclusion

- Message of the talk: A framework for constructing PFE protocols from SFE protocols.

Conclusion

- Message of the talk: A framework for constructing PFE protocols from SFE protocols.
- Open Problems
  - Linear solution with practical efficiency
  - Extending our result to be secure against covert and malicious adversaries
  - Hiding the size of circuit without FHE

Conclusion

- Message of the talk: A framework for constructing PFE protocols from SFE protocols.
- Open Problems
  - Linear solution with practical efficiency
  - Extending our result to be secure against covert and malicious adversaries
  - Hiding the size of circuit without FHE
- Thank you !


Ronald Cramer, Ivan Damgård, and Jesper Buus Nielsen.
Multiparty computation from threshold homomorphic encryption.
In *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques: Advances in Cryptology*, EUROCRYPT '01, pages 280–299, London, UK, UK, 2001. Springer-Verlag.


M. Franklin, M. Gondree, and P. Mohassel.
Multi-party indirect indexing and applications.
Advances in Cryptology–ASIACRYPT 2007, pages 283–297, 2007.


Craig Gentry.
Fully homomorphic encryption using ideal lattices.
In *Proceedings of the 41st annual ACM symposium on Theory of computing*, STOC '09, pages 169–178, New York, NY, USA, 2009. ACM.


Yan Huang, David Evans, and Jonathan Katz.
Private set intersection: Are garbled circuits better than custom protocols?

In *Proceedings of 19th Network and Distributed Security Symposium (NDSS)*, 2012.

Jonathan Katz and Lior Malka.
Constant-round private function evaluation with linear complexity.
In Dong Lee and Xiaoyun Wang, editors, *Advances in Cryptology ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 556–571. Springer Berlin / Heidelberg, 2011.

Vladimir Kolesnikov and Thomas Schneider.
Financial cryptography and data security.
chapter A Practical Universal Circuit Construction and Secure Evaluation of Private Functions, pages 83–97. Springer-Verlag, Berlin, Heidelberg, 2008.

R. Raz.
Elusive functions and lower bounds for arithmetic circuits.
In *Proceedings of the 40th annual ACM symposium on Theory of computing*. ACM, 2008.


A. Shpilka and A. Yehudayoff.
Arithmetic circuits: A survey of recent results and open questions.
2010.


L.G. Valiant.
Universal circuits (preliminary report).
In *Proceedings of the eighth annual ACM symposium on Theory of computing*, pages 196–203. ACM, 1976.