Private Set Intersection:

Are Garbled Circuits Better than Custom Protocols?

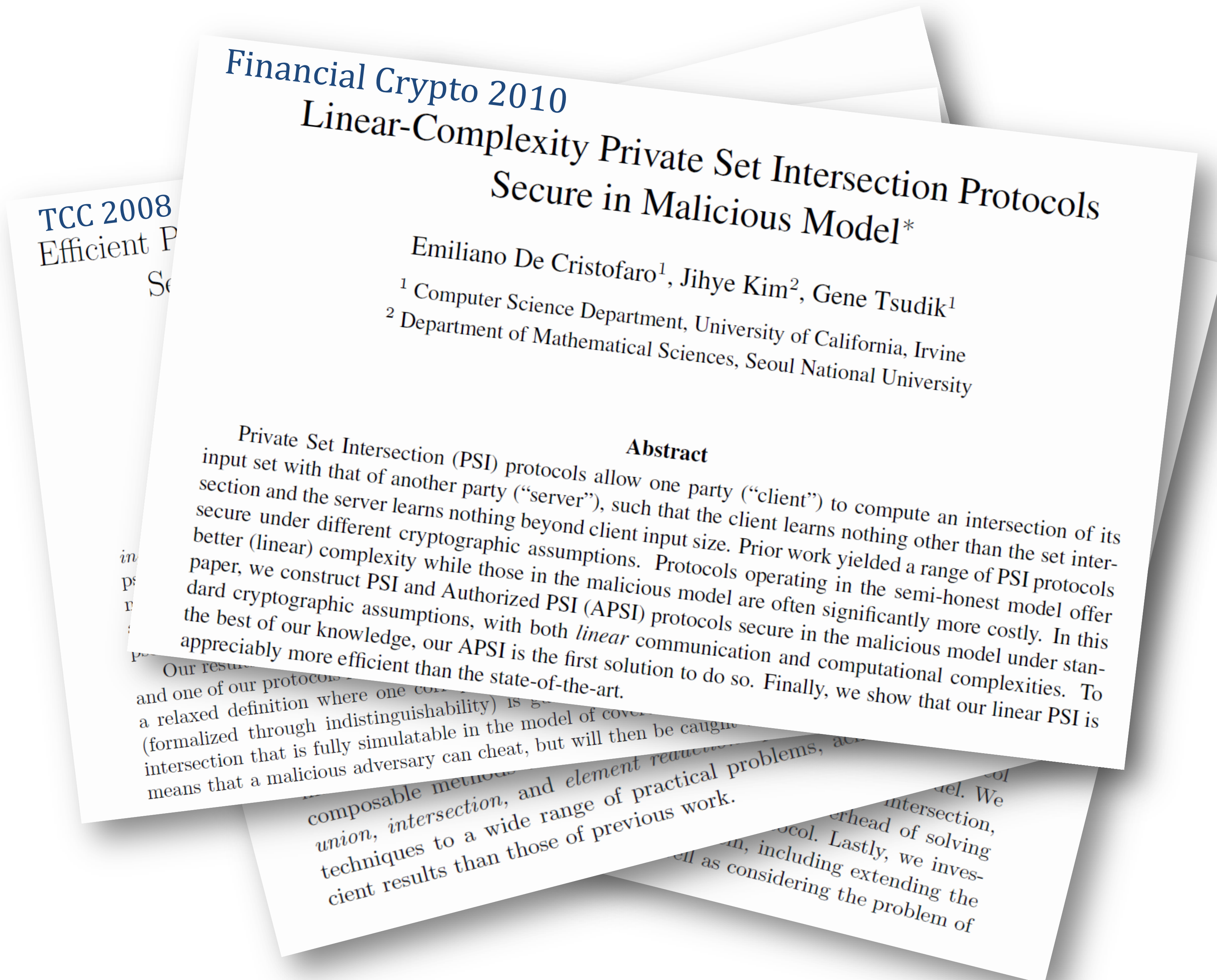
Yan Huang, David Evans, Jonathan Katz

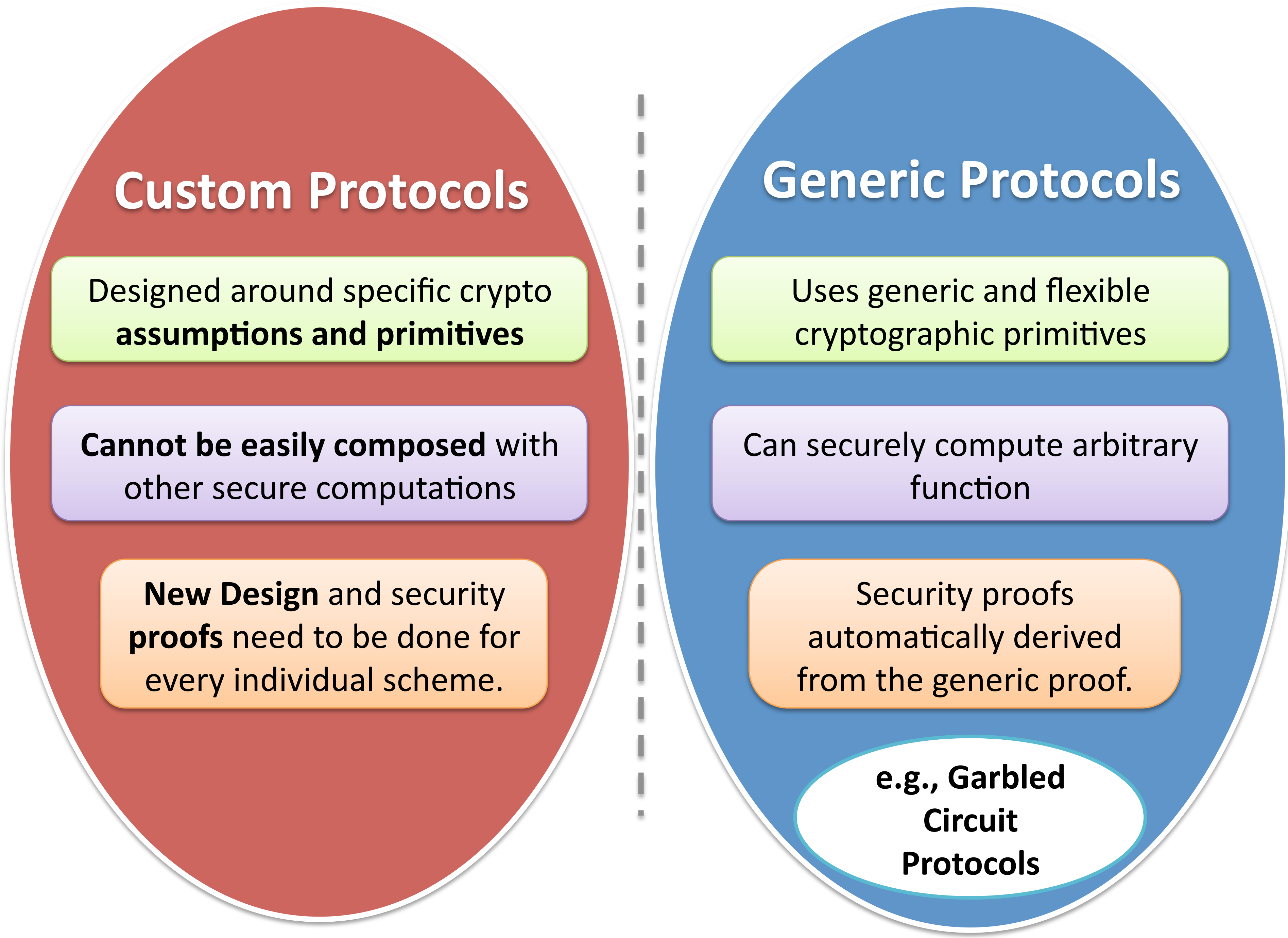
University of Virginia, University of Maryland

www.MightBeEvil.org

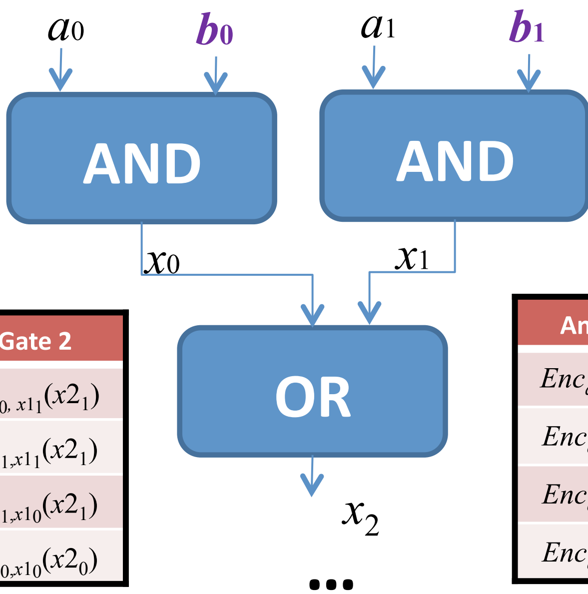
Motivation --- Common Acquaintances

http://www.mightbeevil.com/mobile/


TCC 2008
Efficient
Financial Crypto 2010
Linear-Complexity Private Set Intersection Protocols Secure in Malicious Model*
Emiliano De Cristofaro[1], Jihye Kim[2], Gene Tsudik[1]
[1] Computer Science Department, University of California, Irvine
[2] Department of Mathematical Sciences, Seoul National University
Abstract
Private Set Intersection (PSI) protocols allow one party ("client") to compute an intersection of its input set with that of another party ("server"), such that the client learns nothing other than the set intersection and the server learns nothing beyond client input size. Prior work yielded a range of PSI protocols secure under different cryptographic assumptions. Protocols operating in the semi-honest model offer better (linear) complexity while those in the malicious model are often significantly more costly. In this paper, we construct PSI and Authorized PSI (APSI) protocols secure in the malicious model under standard cryptographic assumptions, with both *linear* communication and computational complexities. To the best of our knowledge, our APSI is the first solution to do so. Finally, we show that our linear PSI is appreciably more efficient than the state-of-the-art.
and one of our
a relaxed definition where one
(formalized through indistinguishability) is
intersection that is fully simulatable in the model of
means that a malicious adversary can cheat, but will then be
composable
union, intersection, and element
techniques to a wide range of practical problems,
cient results than those of previous work.
as considering the problem of


Custom Protocols
Designed around specific crypto **assumptions and primitives**
Cannot be easily composed with other secure computations
New Design and security **proofs** need to be done for every individual scheme.
Generic Protocols
Uses generic and flexible cryptographic primitives
Can securely compute arbitrary function
Security proofs automatically derived from the generic proof.
e.g., Garbled Circuit Protocols

Garbled Circuits & Oblivious Transfers

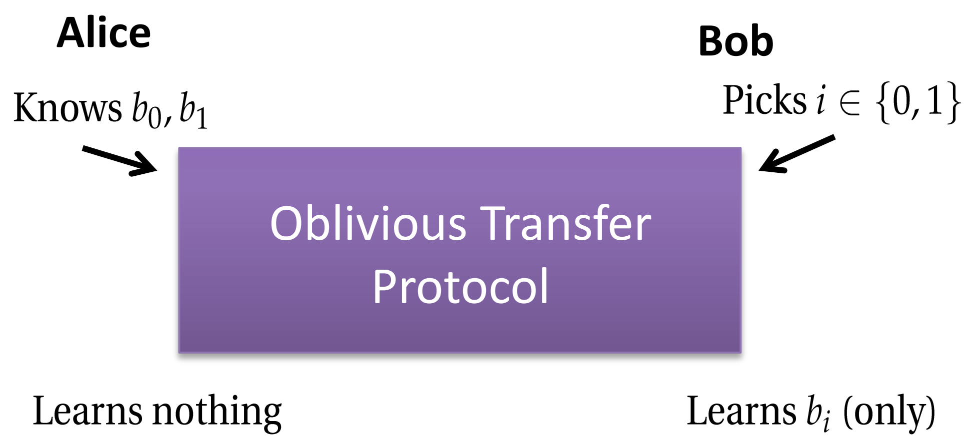


Or Gate 2
$Enc_{x0_0,\, x1_1}(x2_1)$
$Enc_{x0_1, x1_1}(x2_1)$
$Enc_{x0_1, x1_0}(x2_1)$
$Enc_{x0_0, x1_0}(x2_0)$

And Gate 1
$Enc_{a1_0,\, b1_1}(x1_0)$
$Enc_{a1_1, b1_1}(x1_1)$
$Enc_{a1_1, b1_0}(x1_0)$
$Enc_{a1_0, b1_0}(x1_0)$

Andrew Yao, 1982/1986

***Free-XOR** technique*, Kolesnikov and Shneider, 2008



Rabin, 1981; Even, Goldreich, and Lempel, 1985; Naor and Pinkas 2001, Ishai et al., 2003

Y. Huang, D. Evans, J. Katz, L. Malka, Faster Secure Computation Using Garbled Circuits, USENIX Security 2011.

Threat Model

Semi-Honest Adversary: **follows the protocol as specified**, but tries to learn more from the protocol execution transcript

Generic PSI Protocols Overview

Protocols	Cost in non-XOR gates	Best for
Bitwise-AND (BWA)	2^{σ}	Small element space
Pairwise-Comparison (PWC)	$O(\sigma n^2)$	
Sort-Compare-Shuffle-WN (SCS-WN)	$O(\sigma n \log n)$	Large element space

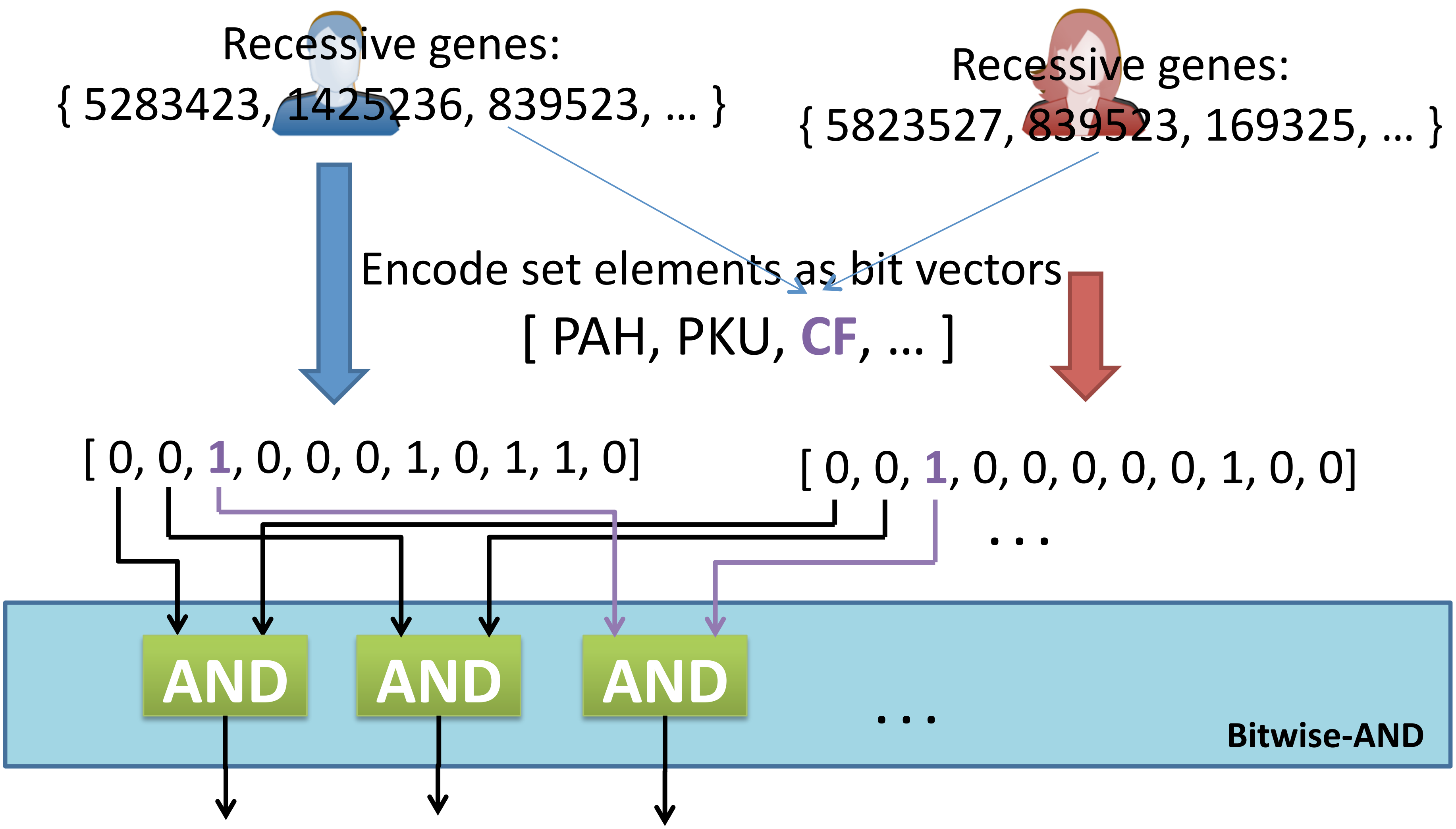
σ – the number of bits used to denote a set element
n – the size of the sets

Generic PSI Protocols Overview

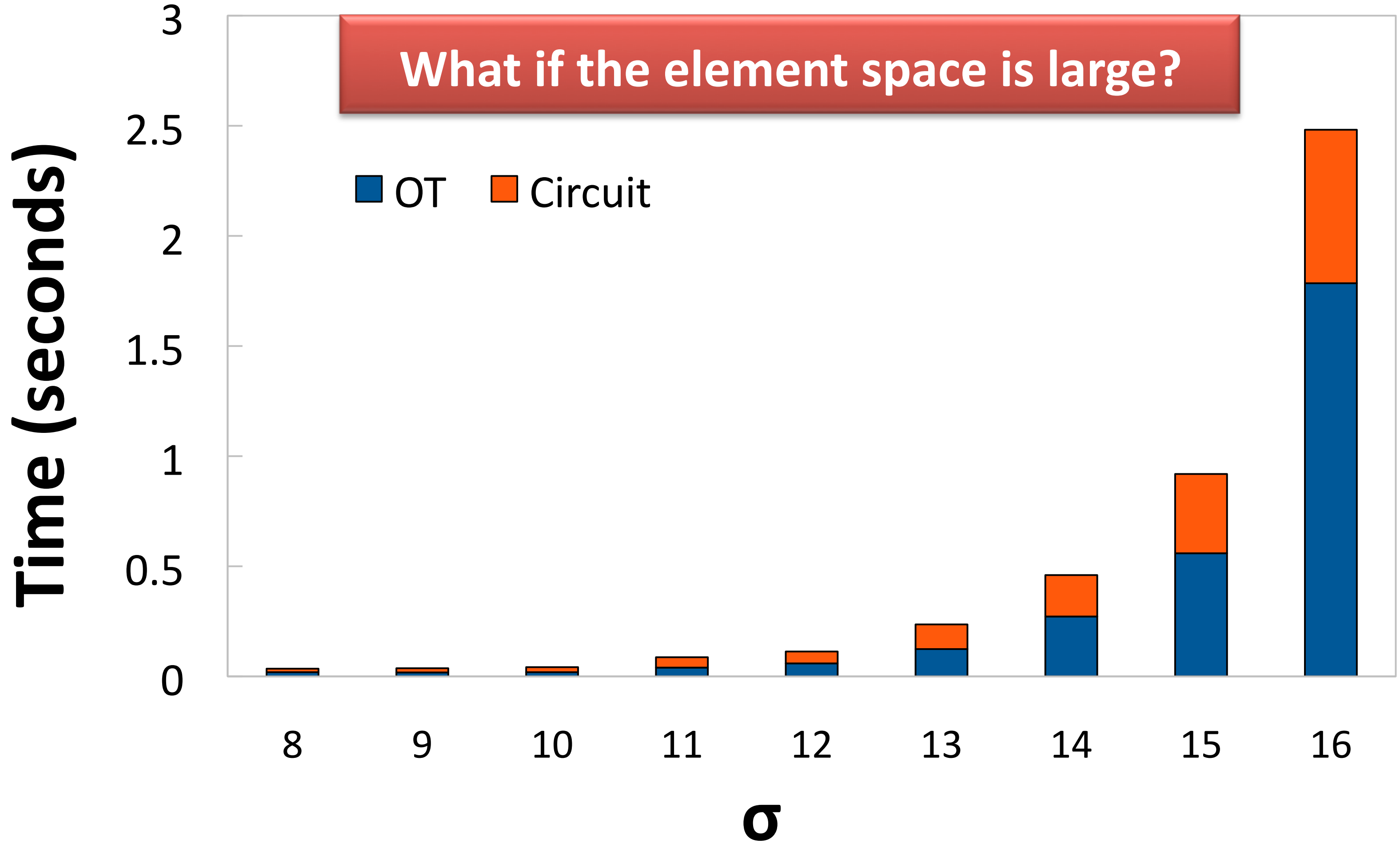
Protocols	Cost in non-XOR gates	Best for
Bitwise-AND (BWA)	2^{σ}	Small element space
Pairwise-Comparison (PWC)	$O(\sigma n^2)$	
Sort-Compare-Shuffle-WN (SCS-WN)	$O(\sigma n \log n)$	Large element space

σ – the number of bits used to denote a set element
n – the size of the sets

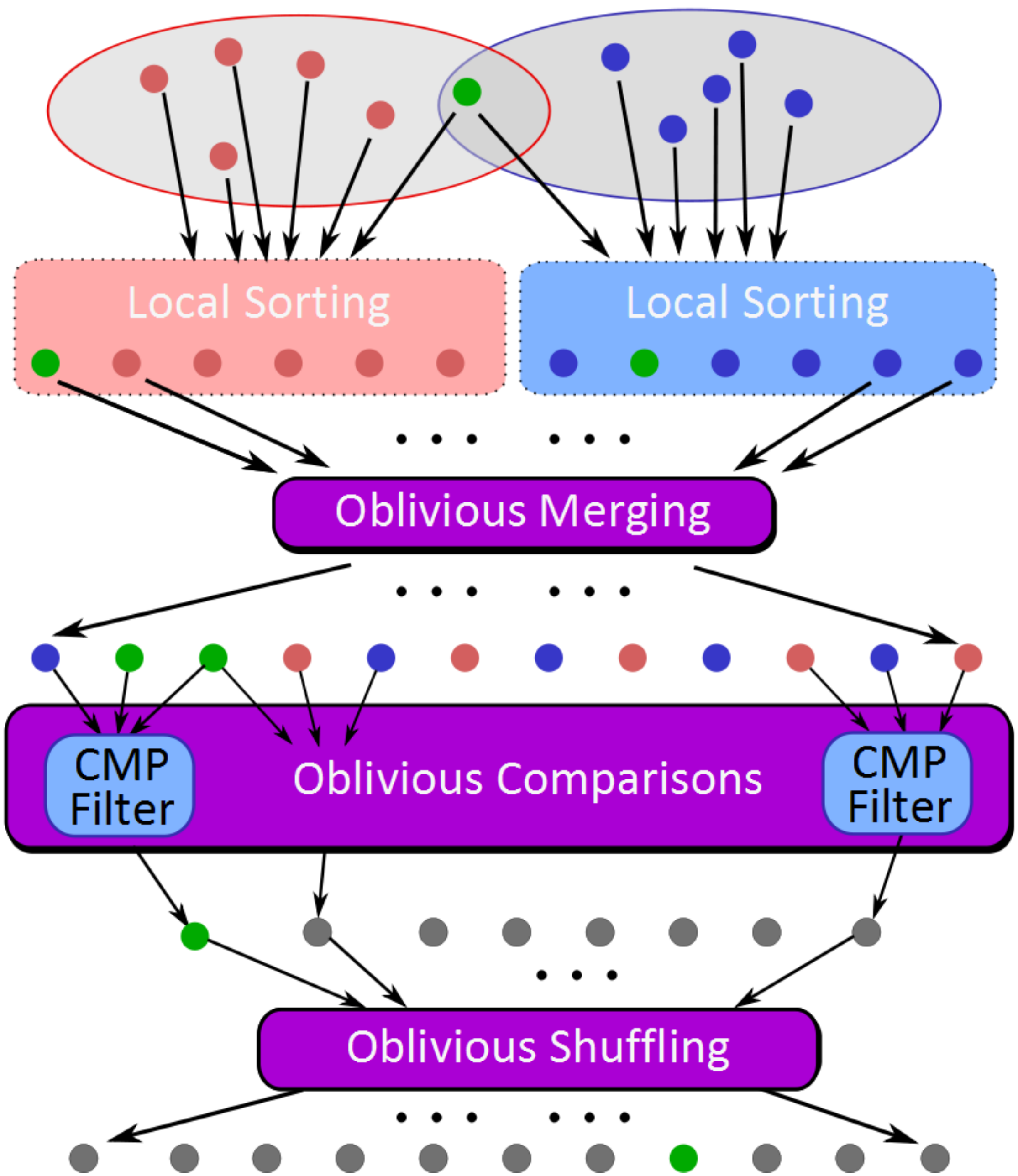
PSI: Needn't be Complex



BWA Performance



Sort-Compare-Shuffle

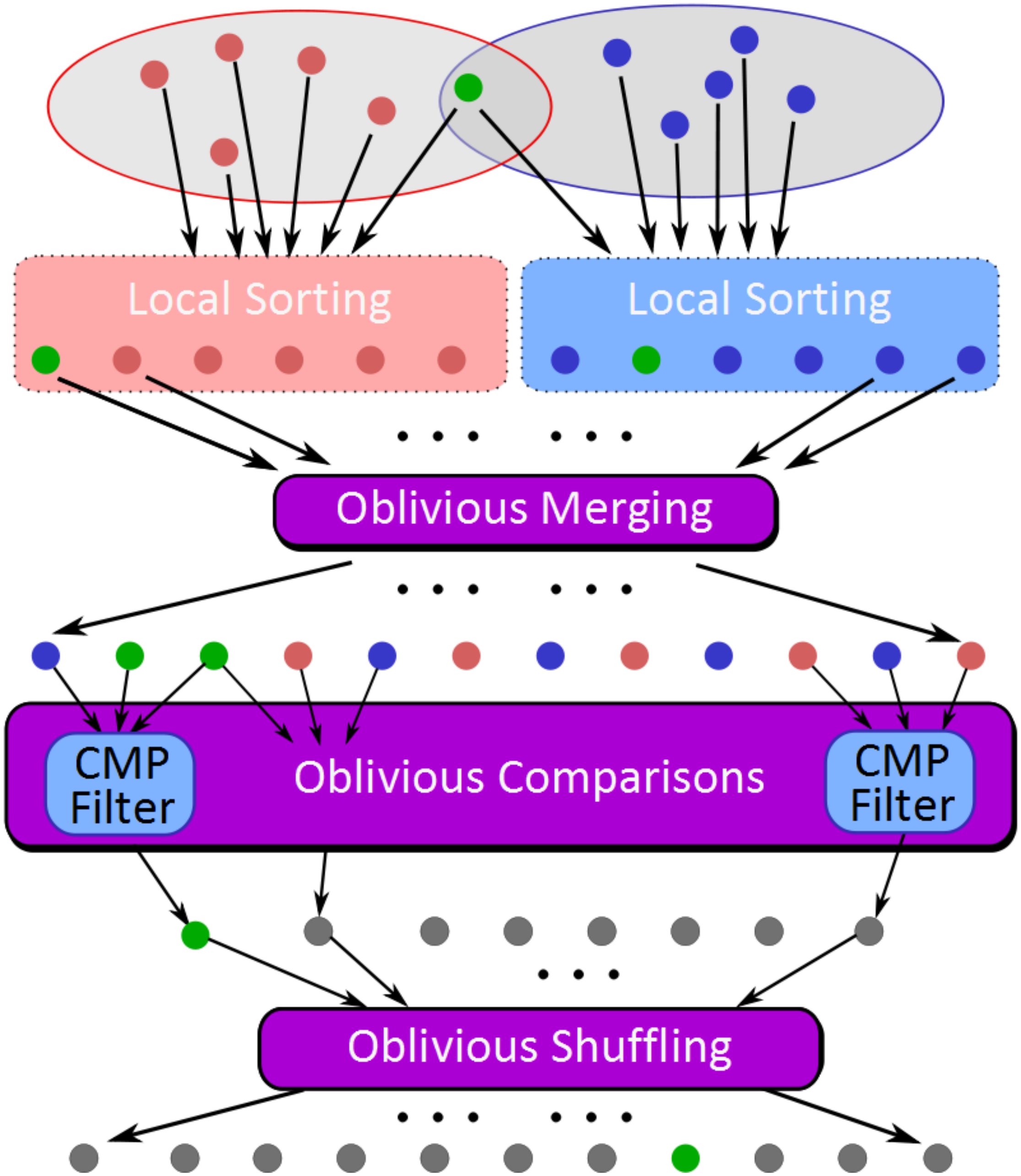


Sort: Take advantage of total order of elements

Compare adjacent elements

Shuffle to hide positions

Sort-Compare-Shuffle

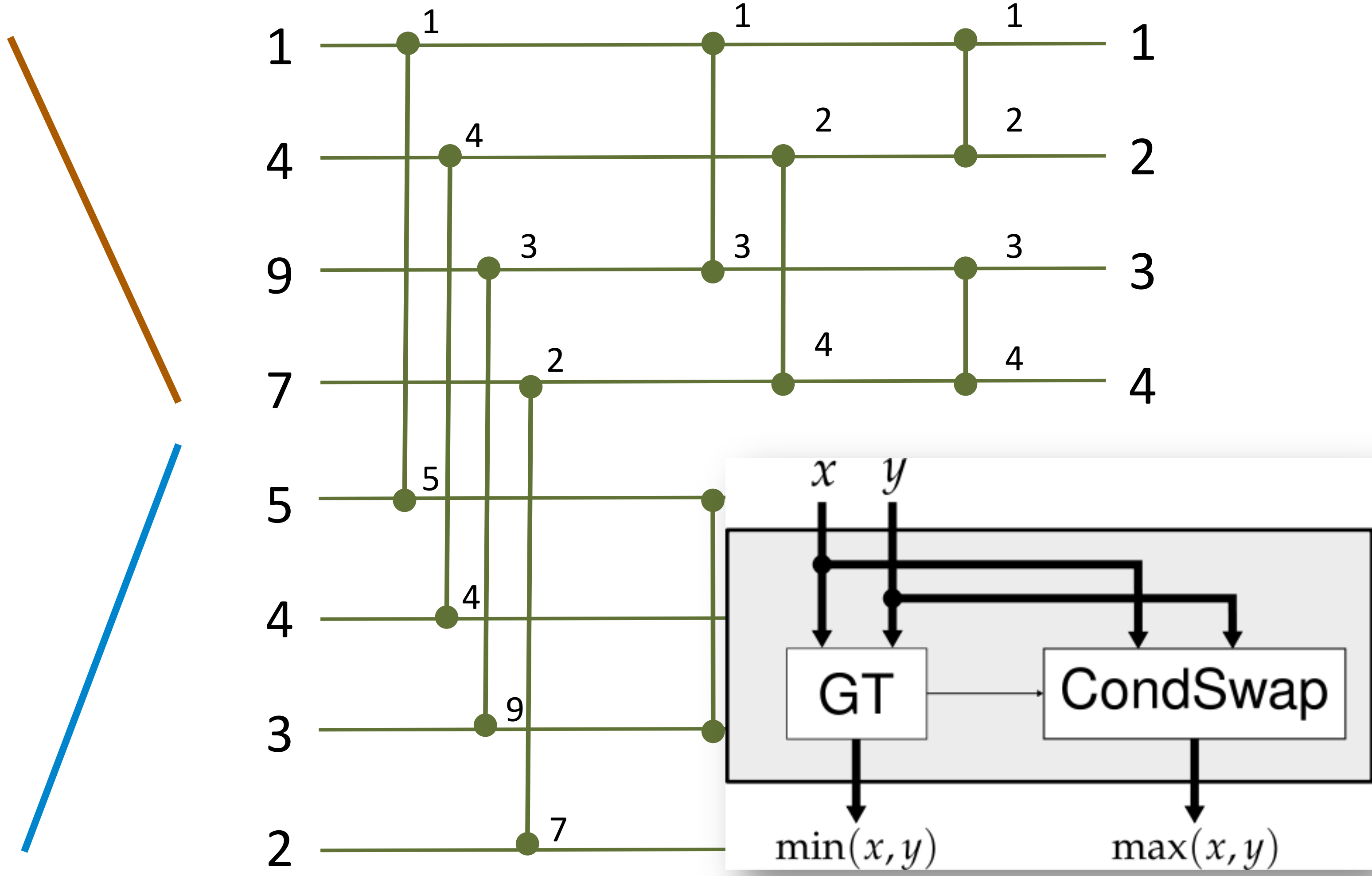


Sort: Take advantage of total order of elements

Compare adjacent elements

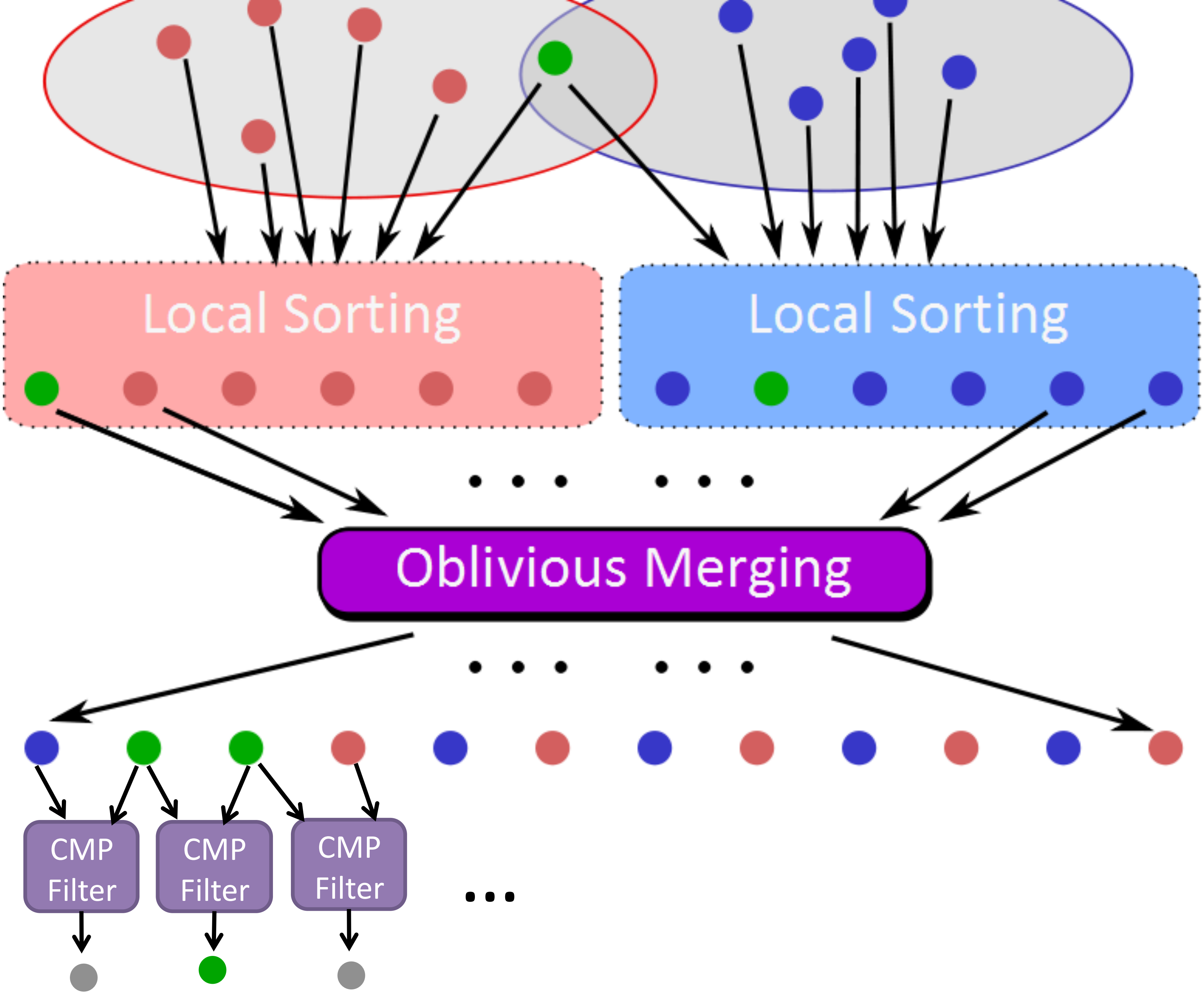
Shuffle to hide positions

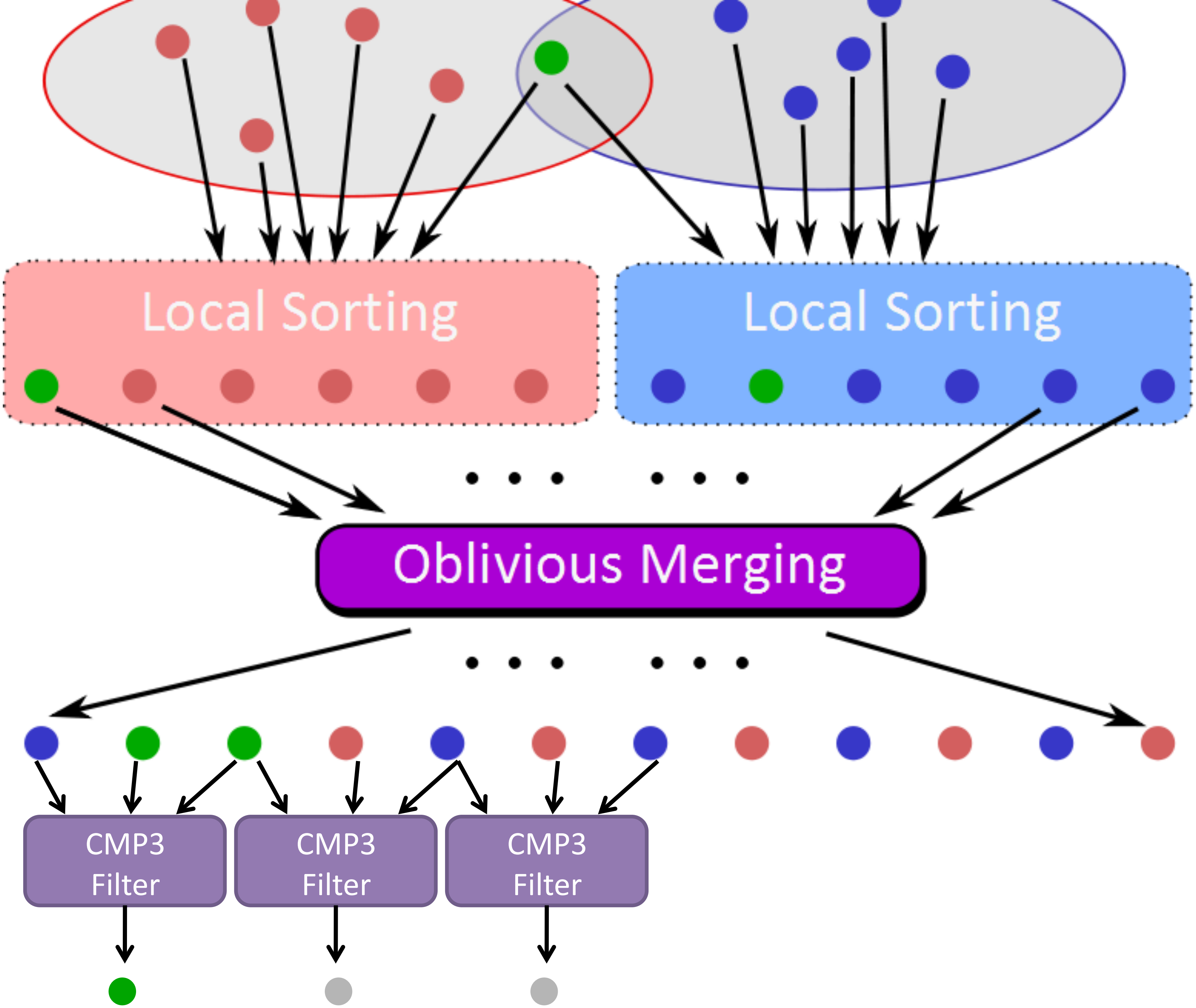
Bitonic Sorting



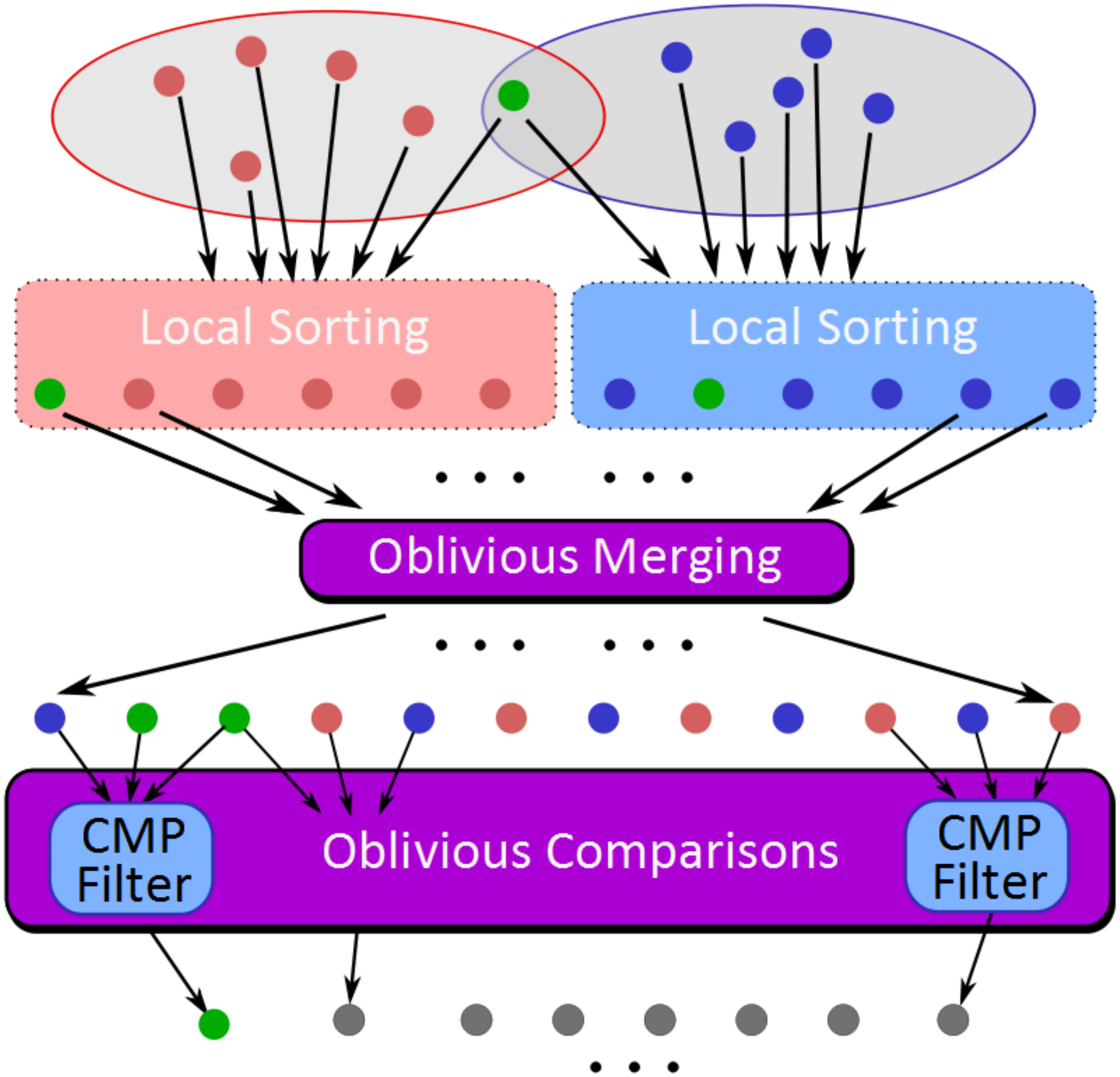
Sort $2n$ bitonic inputs with $n\log(2n)$ CompareSwap circuits.

Sorting Networks and their Applications, Ken Batcher, 1968


Local Sorting
Local Sorting
Oblivious Merging
CMP Filter
CMP Filter
CMP Filter



Local Sorting
Local Sorting
Oblivious Merging
CMP3 Filter
CMP3 Filter
CMP3 Filter



Can’t reveal results yet! Position leaks information.

A Permutation Network

ABRAHAM WAKSMAN

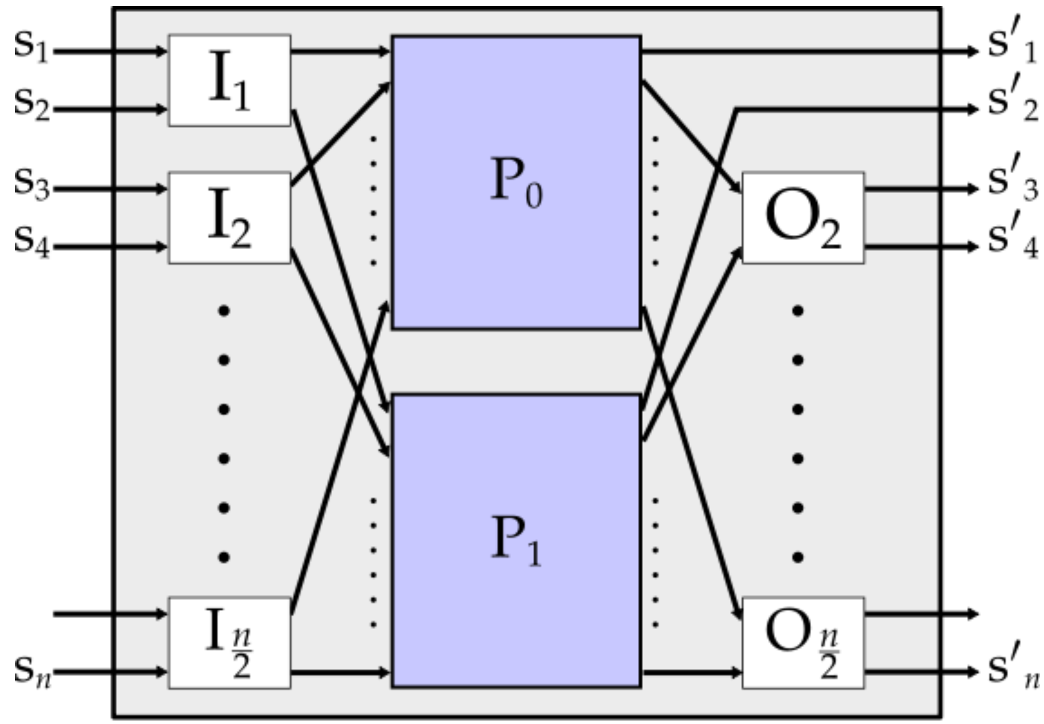
Stanford Research Institute, Menlo Park, California

ABSTRACT. In this paper the construction of a switching network capable of $n!$-permutation of its n input terminals to its n output terminals is described. The building blocks for this network are binary cells capable of permuting their two input terminals to their two output terminals.

The number of cells used by the network is $\langle n \cdot \log_2 n - n + 1 \rangle = \sum_{k=1}^{n} \langle \log_2 k \rangle$. It could be argued that for such a network this number of cells is a lower bound, by noting that binary decision trees in the network can resolve individual terminal assignments only and not the rtitioning of the permutation set itself which requires only $\langle \log_2 n! \rangle = \langle \sum_{k=1}^{n} \log_2 k \rangle$ binary ions.

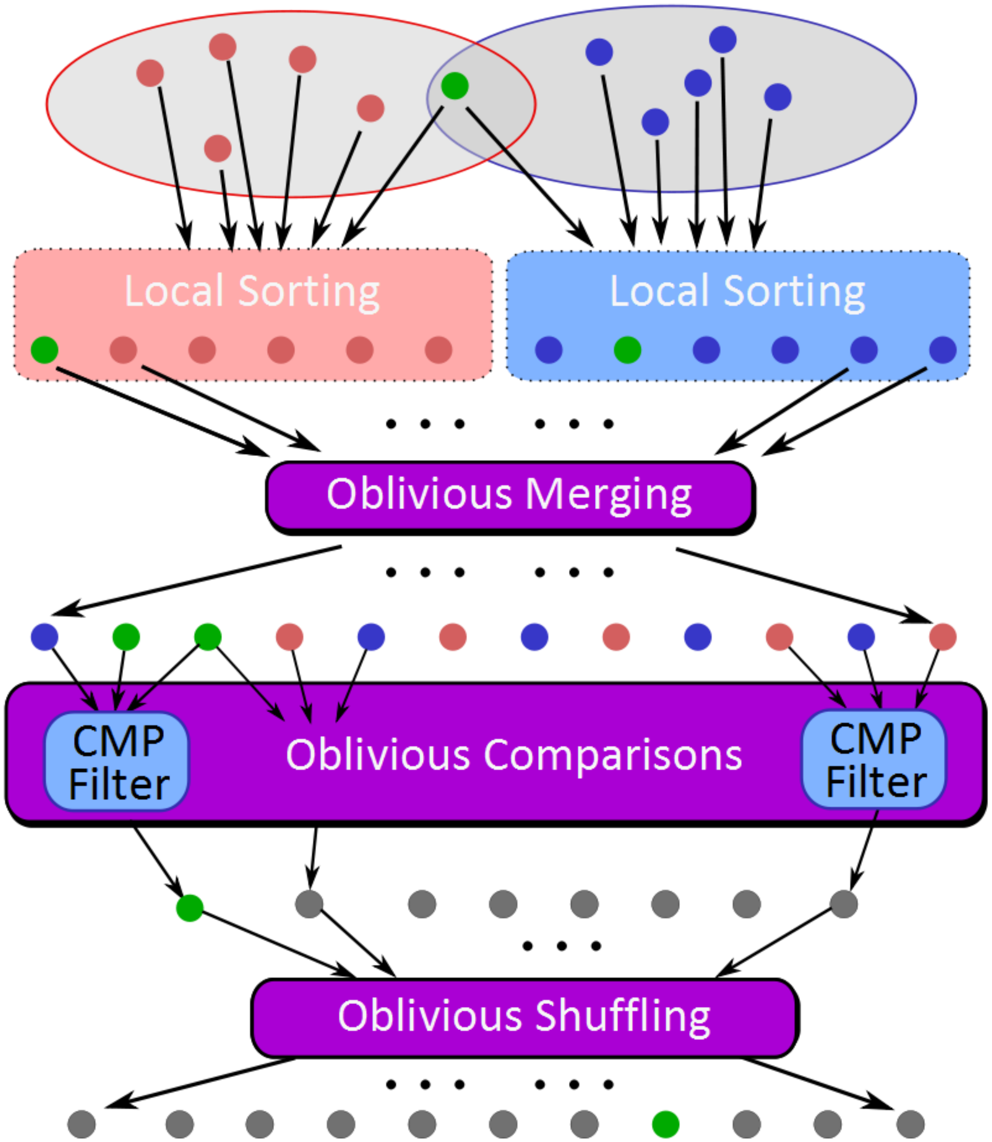
Journal of the ACM, January 1968

Waksman Network



Same circuit can generate any permutation: select a random permutation, and pick swaps

$$\frac{\sigma(n \log n - n + 1)}{3} \text{ gates}$$



Private Set Intersection Protocol

Gates to generate and evaluate

Free

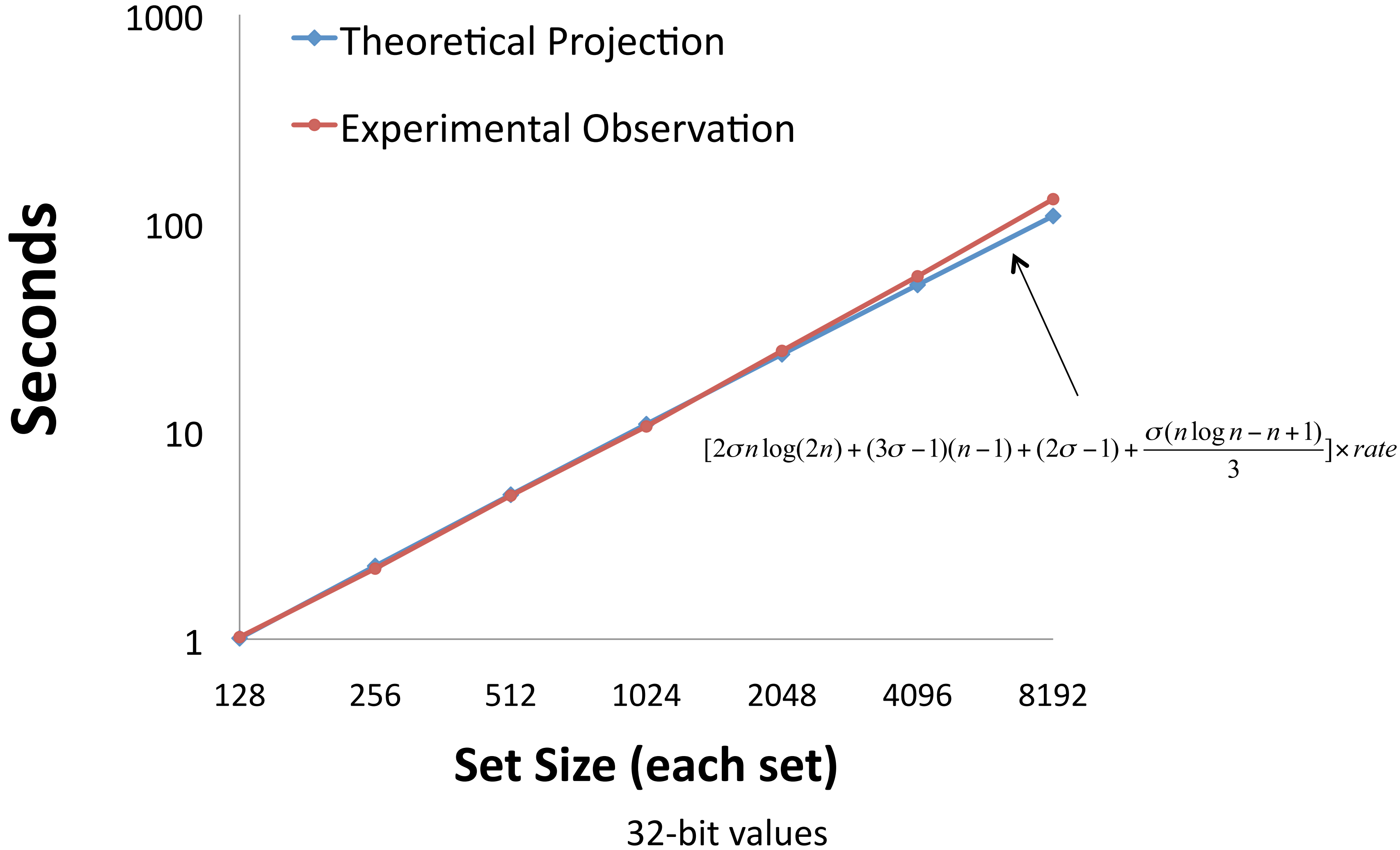
$$n\log(2n) \times 2\sigma$$

$$(3\sigma-1)(n-1)+(2\sigma-1)$$

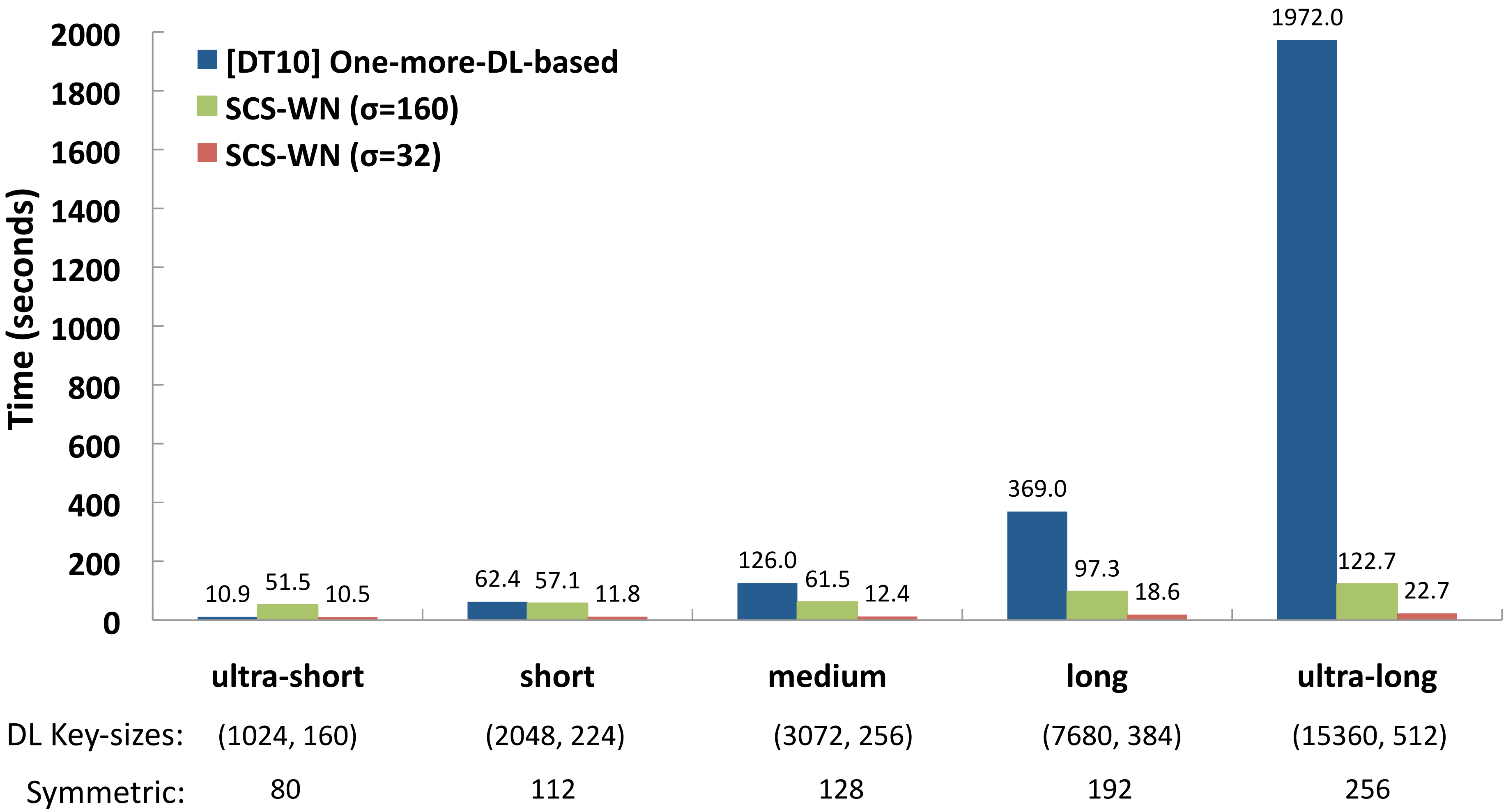
$$\frac{\sigma(n\log n - n + 1)}{3}$$

σ – the number of bits used to denote a set element
n – the size of the sets

SCS-WN Protocol Results



Relating Performance to Security



Conclusion

Generic protocols offer many advantages

- Composability
- Flexibility on hardness assumptions
- Design cost
- Performance

Q & A?