Garbled Circuits via Structured Encryption

Seny Kamara – Microsoft Research

Lei Wei – University of North Carolina

Garbled Circuits

- Fundamental cryptographic primitive
- Possess *many* useful properties
  - Homomorphic
  - Functional
  - General-purpose
  - Verifiable
  - Computationally efficient (free XOR, pipelining, garbled row reduction, …)

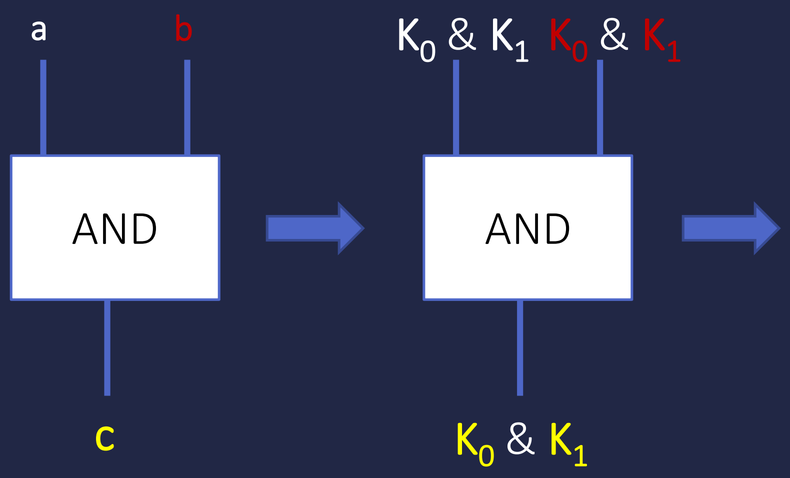
Applications of Garbled Circuits

- Two-party computation [Yao82]
- Server-aided multi-party computation [K.-Mohassel-Raykova12]
- Covert multi-party computation [Chandran-Goyal-Sahai-Ostrovsky07]
- Homomorphic encryption [Gentry-Halevi-Vaikuntanathan10]
- Functional encryption [Seylioglu-Sahai10]
- Single-round oblivious RAMs [Lu-Ostrovsky13]
- Leakage-resilient OT [Jarvinen-Kolesnikov-Sadeghi-Schneider10]
- One-time programs [Goldwasser-Kalai-Rothblum08]
- Verifiable computation [Gennaro-Gentry-Parno10]
- Randomized encodings [Applebaum-Ishai-Kushilevitz06]

Yao’s Garbled Circuits

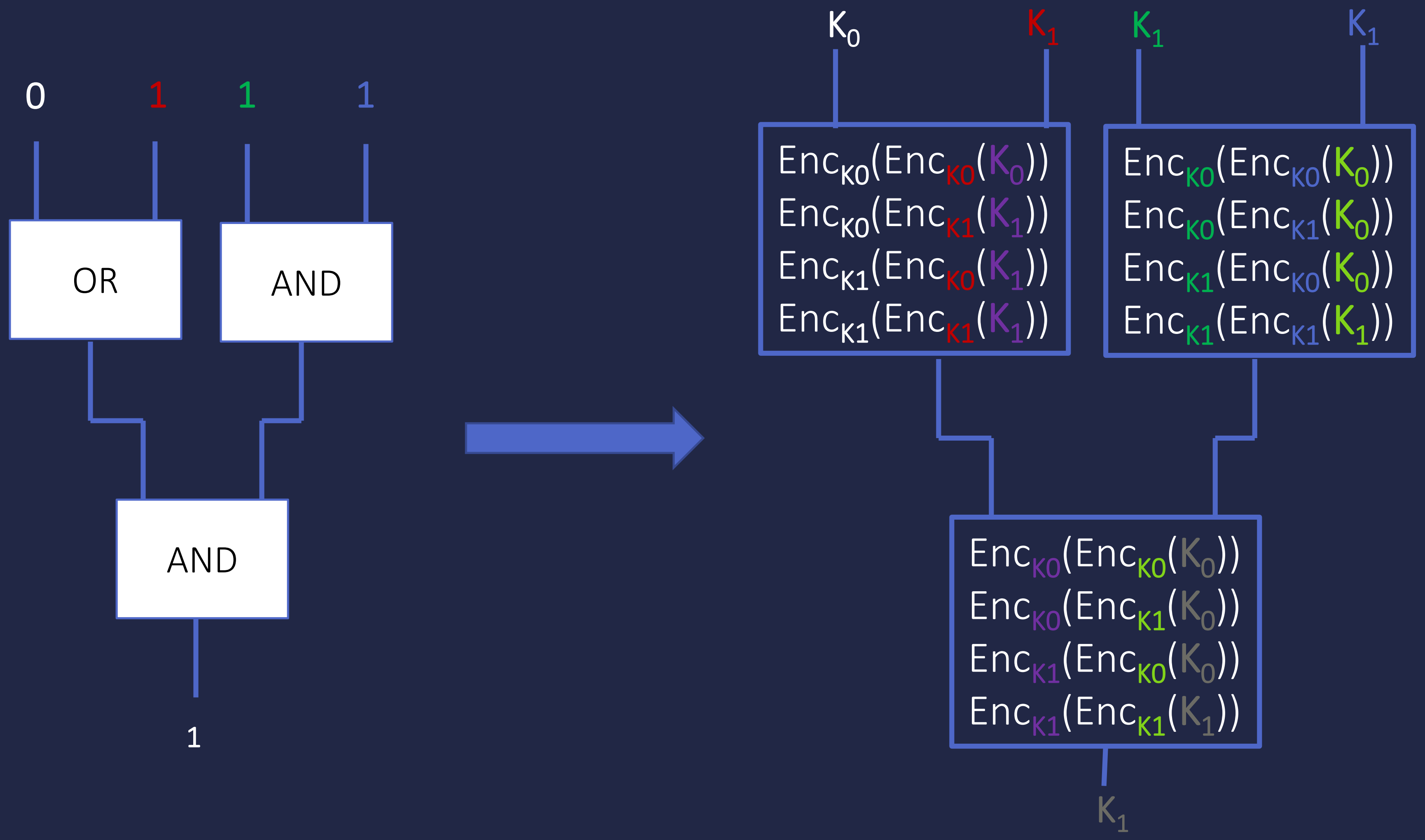
AND:

0	0	0
0	1	0
1	0	0
1	1	1



$Enc_{K0}(Enc_{K0}(K_0))$
$Enc_{K0}(Enc_{K1}(K_0))$
$Enc_{K1}(Enc_{K0}(K_0))$
$Enc_{K1}(Enc_{K1}(K_1))$

Yao's Garbled Circuits



Defining Garbled Circuits

Garbling Scheme

$\text{Grb}(1^k, C) \Longrightarrow (\tilde{C}, \mathbf{dk}, sk)$

$\text{GI}(sk, x) \Longrightarrow \tilde{x}$

$\text{Eval}(\tilde{C}, \tilde{x}) \Longrightarrow \tilde{y}$

$\text{Dec}(\mathbf{dk}_i, \tilde{y}) \Longrightarrow \{\perp, y_i\}$

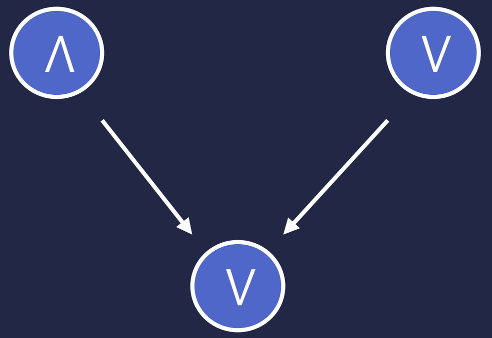
Input Privacy

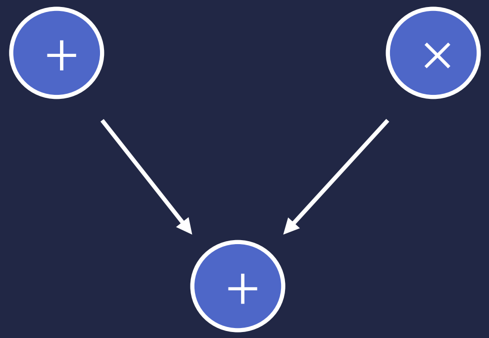
SIM1: "($\tilde{C}$, $\tilde{x}$, $\boldsymbol{dk}$) can be simulated given only $\tilde{C}$ and $f(x)$"

SIM2: "($\tilde{C}$, $\tilde{x}$, $\boldsymbol{dk}$) can be simulated given only C and $f(x)$,
even when x is chosen as a function of $\tilde{C}$ "

Designing Garbled Circuits

General-Purpose Garbling Schemes





BOOLEAN CIRCUITS

[Yao82]: public-key techniques

[Lindell-Pinkas09]: double encryption

[Naor-Pinkas-Sumner99]: hash functions

[Bellare-Hoang-Rogaway12]: dual-key ciphers

ARITHMETIC CIRCUITS

[Applebaum-Ishai-Kushilevitz12]: affine randomized encodings

General-Purpose Garbling Schemes

Boolean circuits

- Efficient: bit-wise operations (e.g., shifts, comparisons, …)
- Inefficient: arithmetic operations

Arithmetic circuits

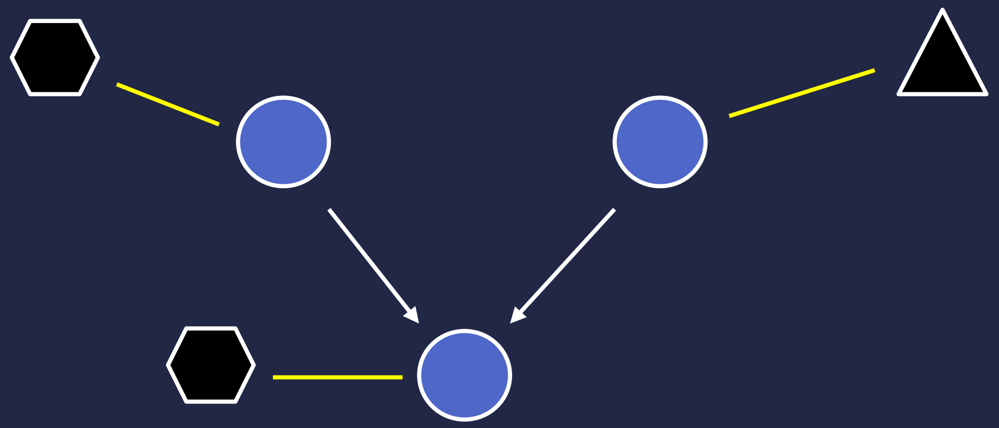
- Efficient: arithmetic operations (e.g., additions, multiplications, polynomials, …)
- Inefficient: bit-wise operations

Many problems are neither

- [Naor-Nissim01]: circuits with lookup tables ≈ RAMs
- [Barkol-Ishai05]: constant-depth circuits
- [Gordon et al.12]: DB lookups

} Not Garbling Schemes

Structured Circuits



Efficient for “structured problems”

- Search, graphs, DFAs, branching programs

Can be garbled

- 2PC, homomorphic encryption, one-time programs, verifiable computation, …

Structured Encryption [Chase-K.10]

$$\mathrm{Gen}(1^k) \Rightarrow \mathrm{K}$$

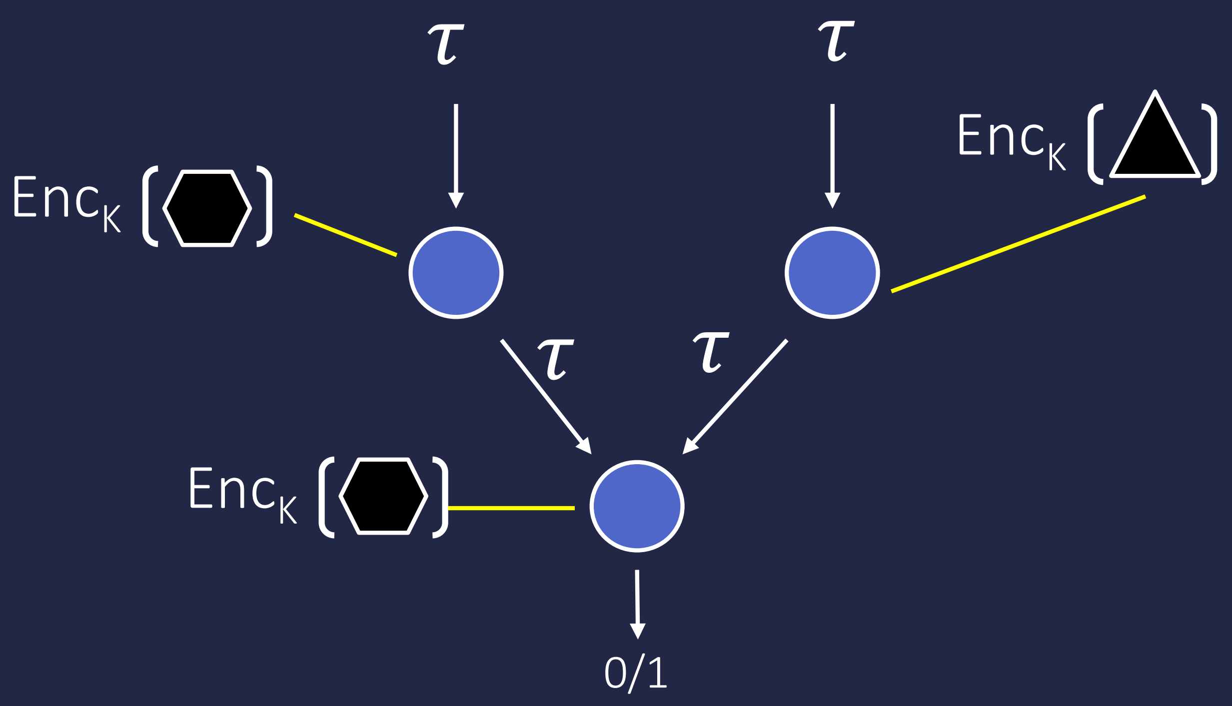
$$\mathrm{Enc}_K(\delta, \overline{m}) \Rightarrow \gamma$$

$$\mathrm{Token}_K(q) \Rightarrow \tau$$

$$\mathrm{Query}(\gamma, \tau) \Rightarrow I$$

$$\mathrm{Dec}_K(c_i) \Rightarrow m_i$$

How to Garble a Structured Circuit



Correctness

- Encrypt data structures
- Associativity (store & release tokens)
- Dimensionality (merge tokens)

Security

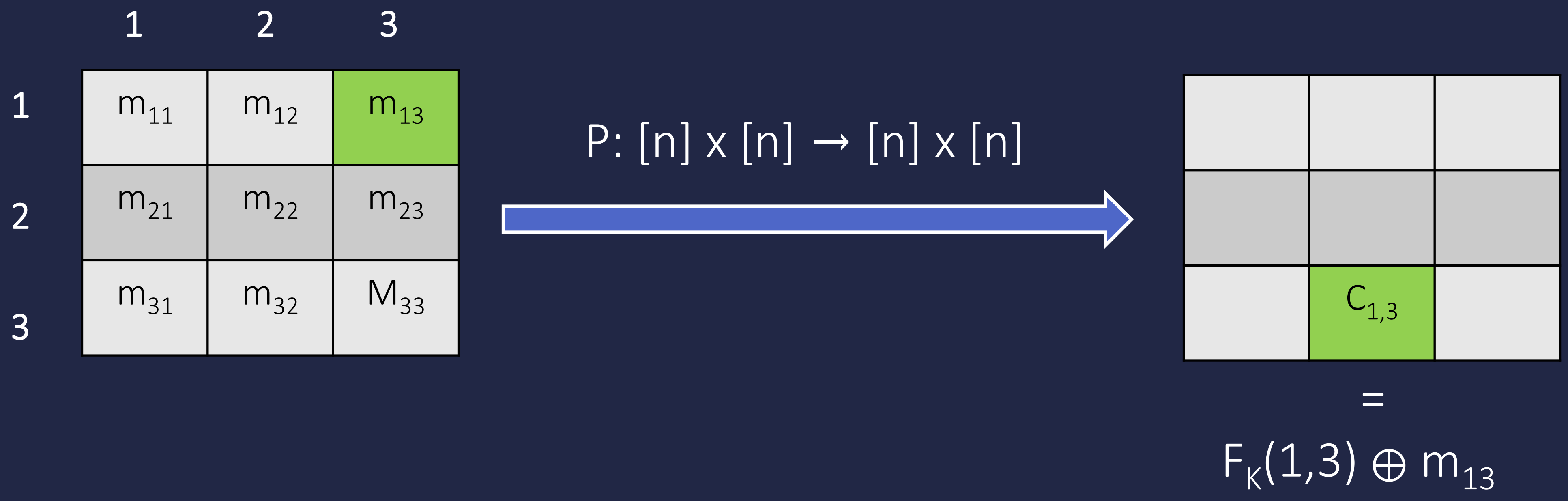
- CQA1 enc ⇒ SIM1 & UNF1 garbling
- CQA2 enc ⇒ SIM2 & UNF2 garbling

Previous Structured Encryption

- Associativity
  - [Curtmola-Garay-K.-Ostrovsky06]: CQA1 & CQA2 inverted index encryption
  - [Chase-K.10]: CQA2 matrix, graph & web graph encryption
- Dimensionality
  - All previously-known constructions are 1-D

2-D Matrix Encryption

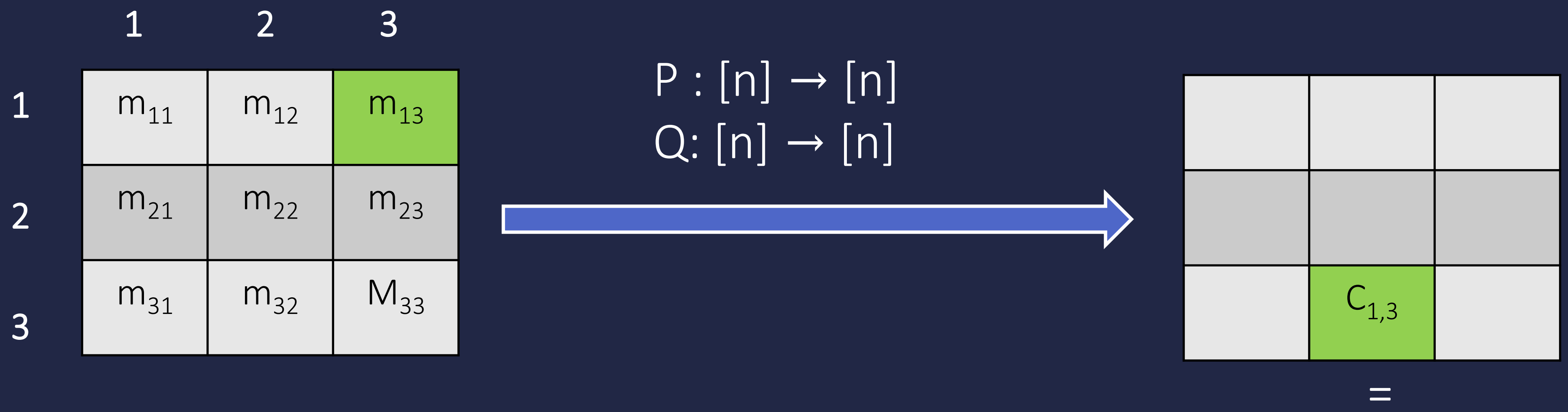
1-D Matrix Encryption [Chase-K.10]



Encrypt: permute & XOR with PRF-based pad

Search: $\tau(1,3)$ = $F_K(1,3)$, $P(1,3)$

2-D Matrix Encryption



$\text{Synth}[\ F_K(\text{row}|P(1))\ ,\ F_K(\text{col}|Q(3)\ ] \oplus m_{13}$

Encrypt: permute & XOR with synthesizer-based pad

Search: $\tau(1) = F_K(\text{row}|P(1))$ $\quad$ $\tau(3) = F_K(\text{col}|Q(3))$

Matrix Garbling Schemes

[Chase-K.10] + synthesizers ⇒ SIM1-secure Garb schemes for matrices

[Chase-K.10] + synthesizers + SIM1-to-SIM2 ⇒ SIM2-secure schemes for matrices

Observation: Yao garbled gate ⟺ 2-D associative CQA1 matrix encryption scheme

Applications

New Special-Purpose Garbling Schemes!

DFAs

Branching programs

Boolean circuits w/ cheaper gate evaluation than Yao

Adjacency queries on graphs

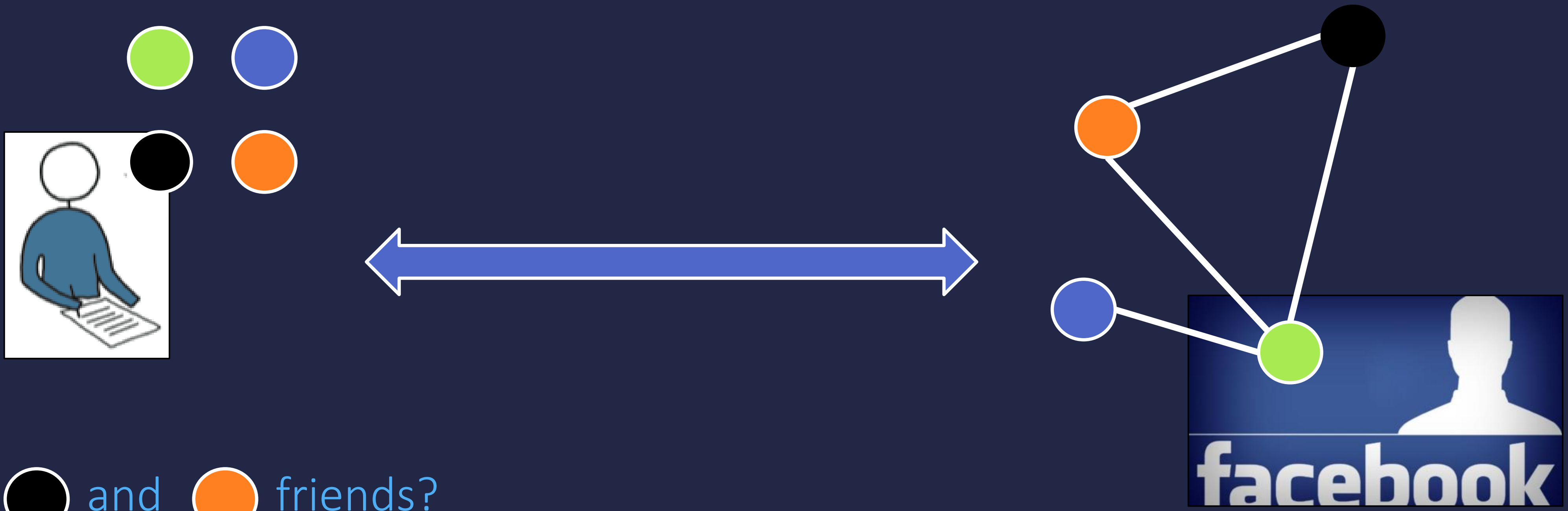
Neighbor queries on graphs

Focused subgraph queries on web graphs

} Our transform + [Chase-K.10]

More efficient: Two-party computation , server-aided multi-party computation, covert multi-party computation, homomorphic encryption, functional encryption, single-round oblivious RAMs, leakage-resilient OT, one-time programs, verifiable computation, randomized encodings, …

Secure Two-Party Graph Computation



Are and friends?

Who are ‘s friends?

Find the friends of anyone who likes my product

Find the friends of anyone with disease X

Thanks