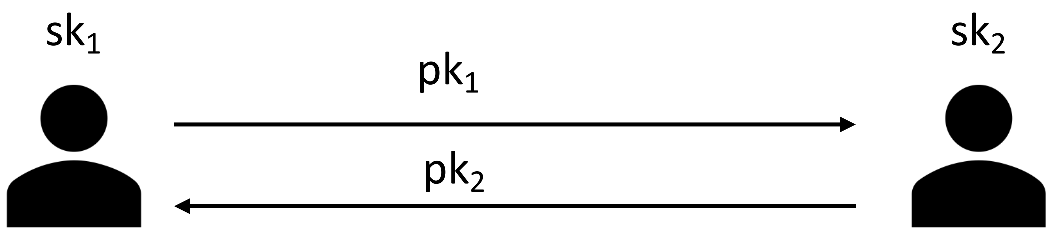
Fine-Grained Non-Interactive Key-Exchange without Idealized Assumptions

Yuyu Wang[1], Chuanjie Su[1], Jiaxin pan[2]

1. University of Electronic Science and Technology of China
2. University of Kassel

Non-interactive key exchange (NIKE)

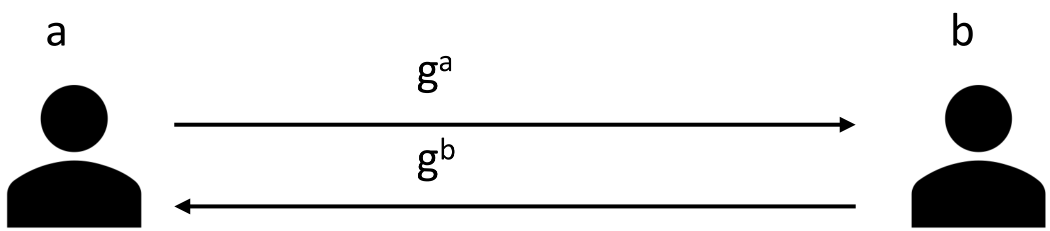
2-party setting



$K = \text{Share}(pk_1, pk_2, sk_1) = \text{Share}(pk_1, pk_2, sk_2)$

Non-interactive key exchange (NIKE)

2-party setting

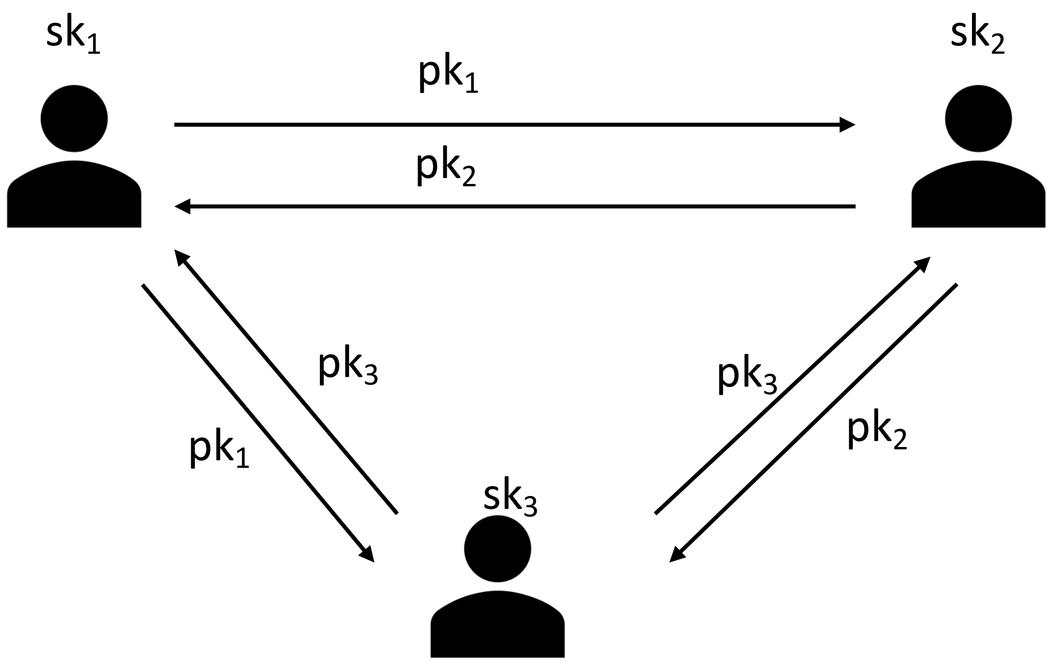


$K=g^{ab}$

Diffie-Hellman NIKE

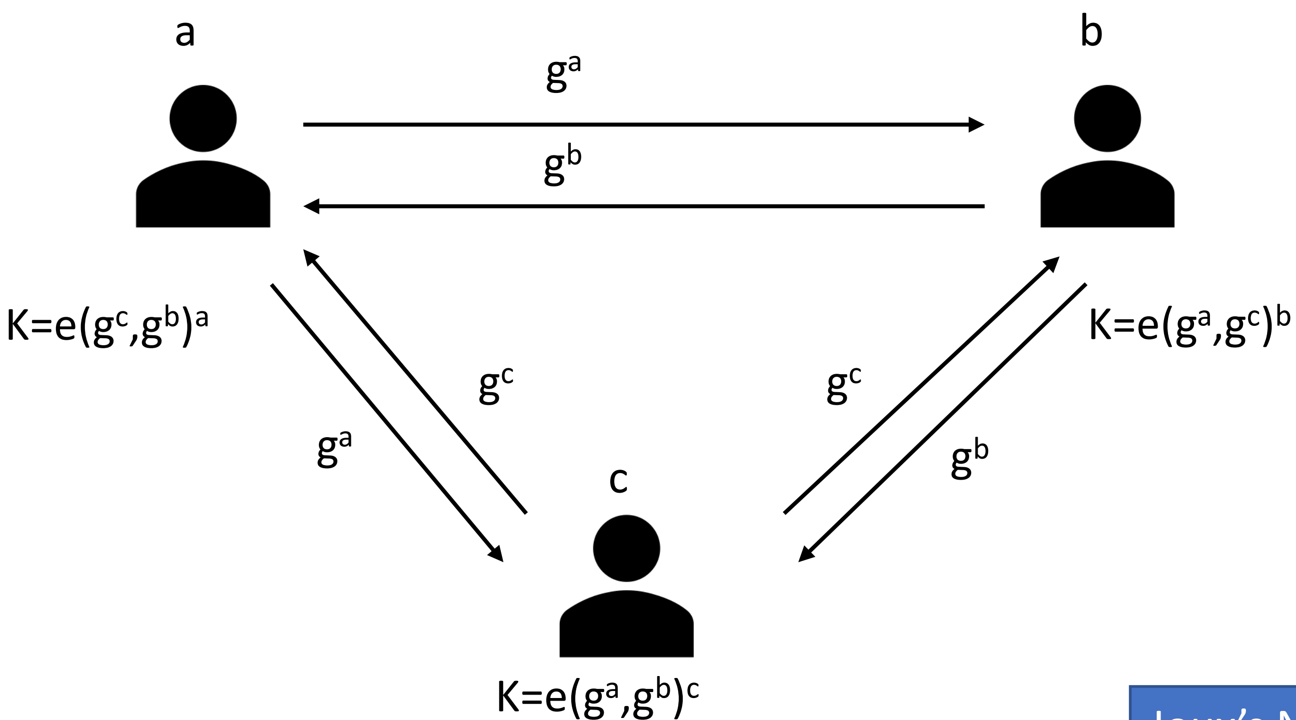
Non-interactive key exchange (NIKE)

3-party setting



Non-interactive key exchange (NIKE)

3-party setting

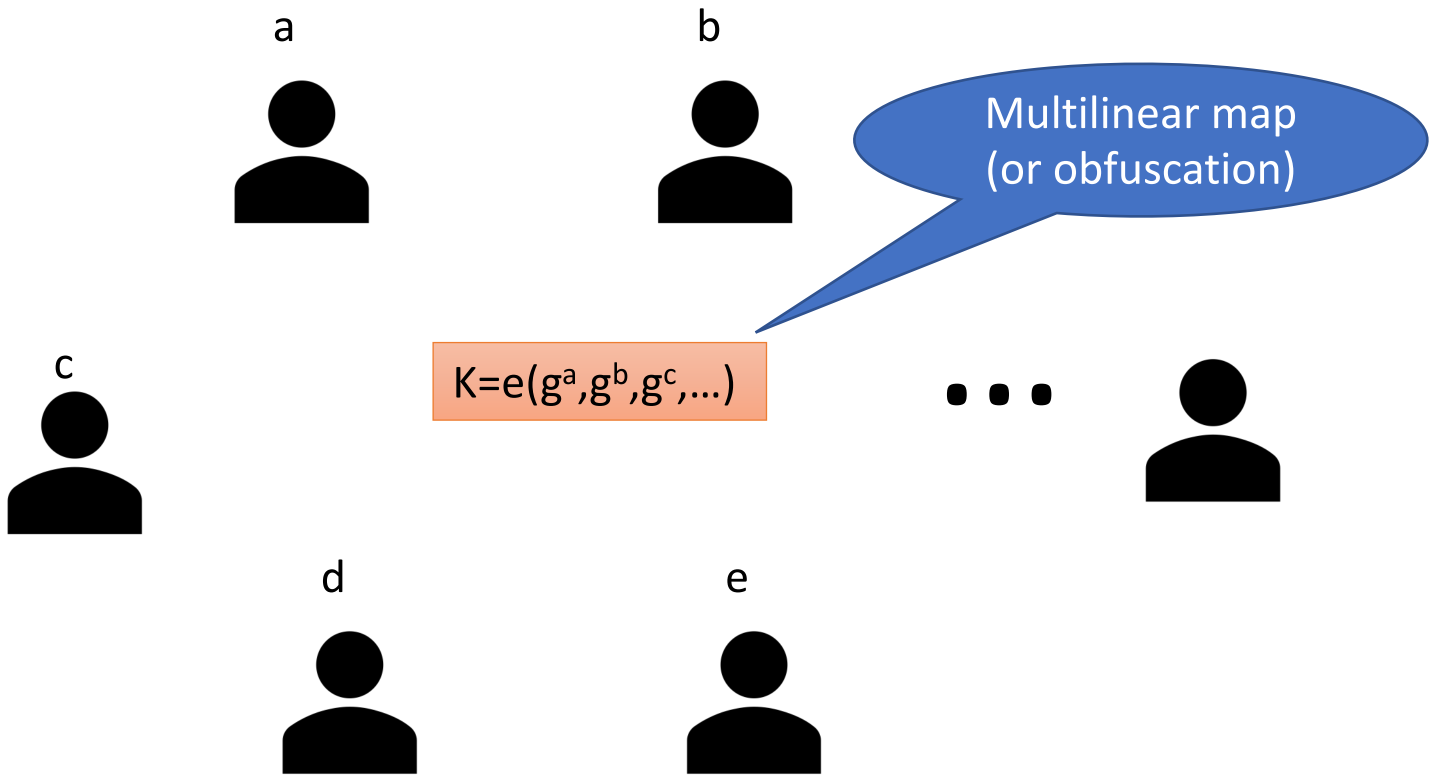


$K=e(g,g)^{abc}$

Joux's NIKE

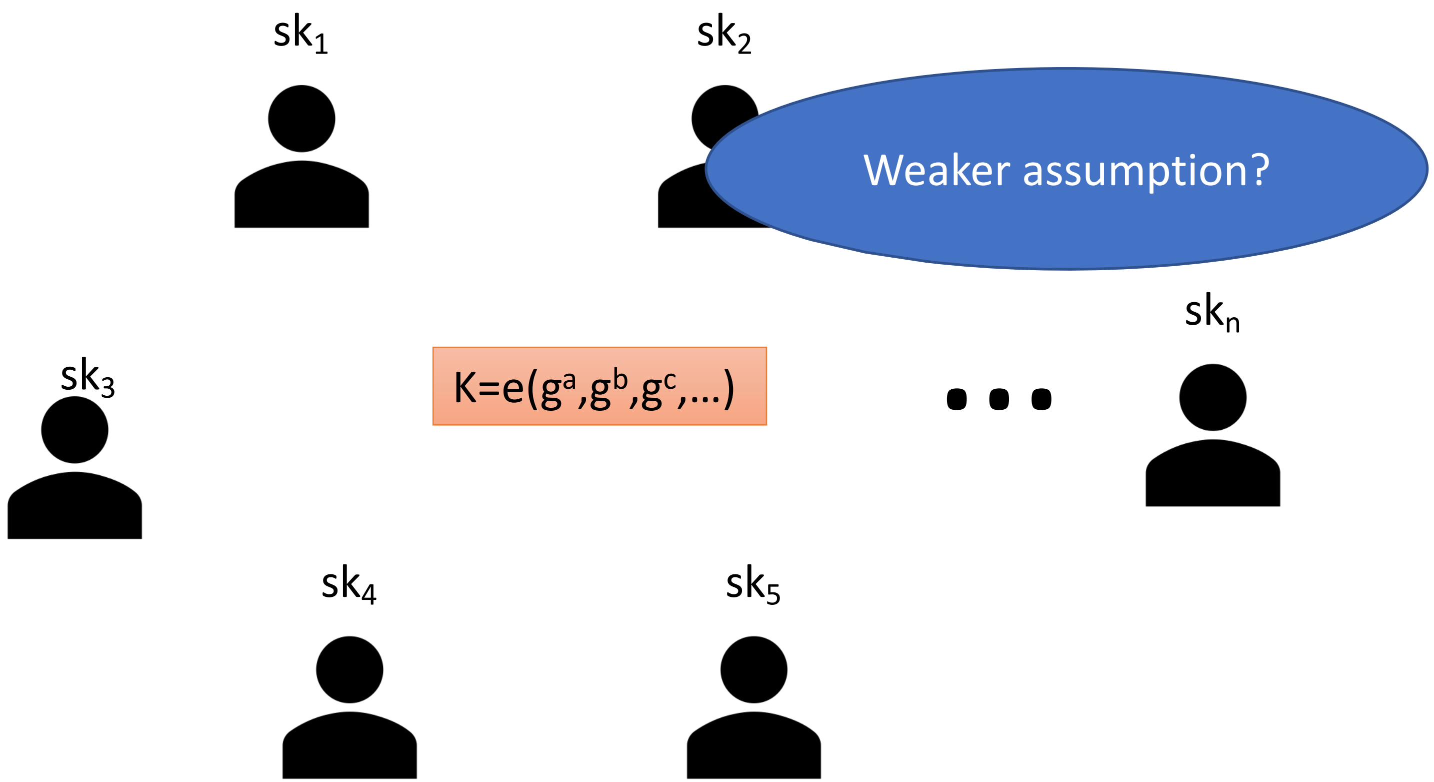
Non-interactive key exchange (NIKE)

n-party setting



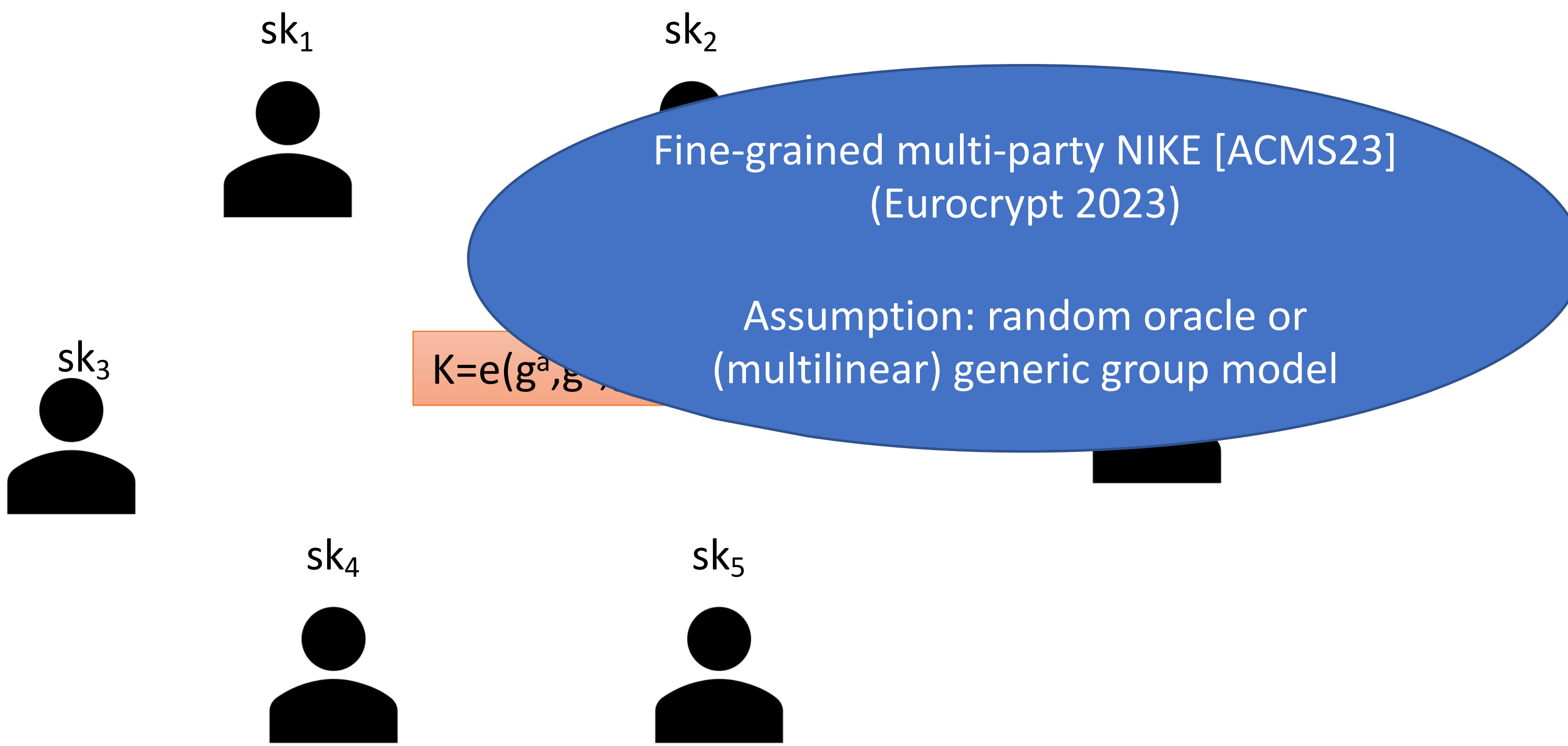
Non-interactive key exchange (NIKE)

n-party setting



Non-interactive key exchange (NIKE)

n-party setting



Fine-grained cryptography

Honest party

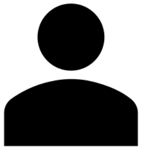
An honest party uses less resources than the adversary

Adversary

The resources of an adversary can be a-prior bounded

Fine-grained cryptography

Honest party



An honest party uses less resources than the adversary

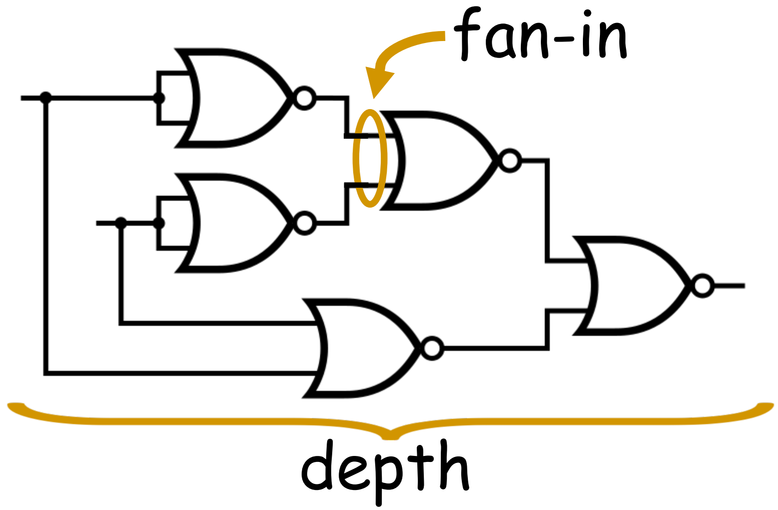
Adversary

The resources of an adversary can be a-prior bounded

- Based only on mild assumption

Existing works

- **Bounded parallel-time setting** [Hås87/DVV16/WP22]
  - Primitive: OWP / PRG, weak-PRF, SKE, CRHF/NIZK for AC0
  - Assumption: None
  - Honest party: $\mathcal{C}_1 = \mathsf{NC}^0/\mathsf{AC}^0$
  - Adversary: $\mathcal{C}_2 = \mathsf{AC}^0$

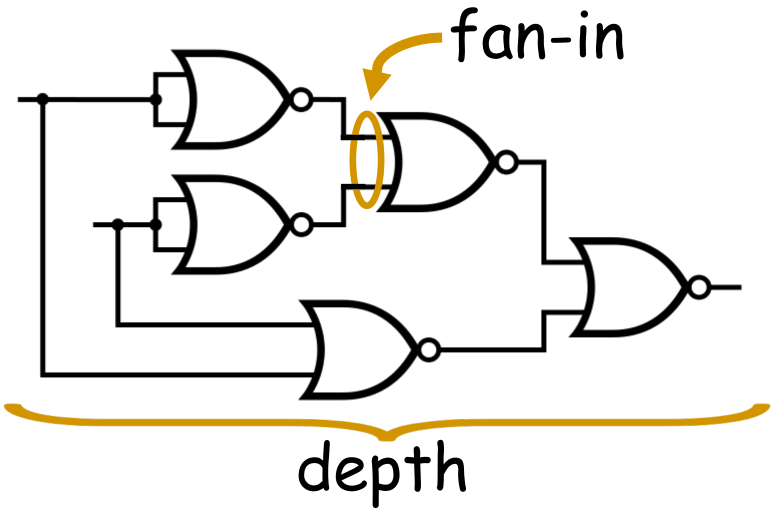


[Hås87] Johan Håstad. One-way permutations in nc0
[DVV16] A. Degwekar, V. Vaikuntanathan, and P. N. Vasudevan. Fine-grained cryptography.
[WP22] Y. Wang, Jiaxin. Pan. Unconditionally secure NIZK in the fine-grained setting.

Existing works

- **Bounded parallel-time setting** [DVV16/CG18/EWT19/EWT21/WPC21/WP22]
  - Primitive: OWF, PRG, PKE, CRHF / SHE, VC

    / OWP, HPS (imply CCA PKE), TDF / full domain TDF / ABE,QANIZK/NIZK,FHE
  - Assumption: $\mathsf{NC}^1 \neq \oplus \mathsf{L}/poly$
  - Honest party: $\mathcal{C}_1 = \mathsf{NC}^1$
  - Adversary: $\mathcal{C}_2 = \mathsf{NC}^1$



[DVV16] A. Degwekar, V. Vaikuntanathan, and P. N. Vasudevan. Fine-grained cryptography.
[CG18] Matteo Campanelli and Rosario Gennaro. Fine-grained secure computation.
[EWT19, EWT21] S. Egashira, Y. Wang, and K. Tanaka. Fine-grained Cryptography revisited.
[WPC21,WPC23] Y. Wang, Jiaxin. Pan, Y. Chen. Fine-grained secure attribute-based encryption.
[WP22] Y. Wang, Jiaxin. Pan. Non-interactive zero-knowledge proofs with fine-grained security.

Existing works

- **Bounded time setting** [Mer78, BGI08/LLW19/ACMS23]
  - Primitive: (Multi-party) key exchange
  - Assumption: random oracle, exponentially strong OWF / average-case hard zero k-clique/multilinear Shoup's GGM
  - Honest party: $\mathcal{C}_1 = O(t)$
  - Adversary: $\mathcal{C}_2 = o(t^2)$ / $o(t^{1.5})$ / $o(t^{n/n-1})$
- **Bounded storage setting** [CM97]
  - Primitive: Key exchange
  - Assumption: None
  - Honest party: $\mathcal{C}_1 = O(s)$
  - Adversary: $\mathcal{C}_2 = o(s^2)$

[Mer78] Ralph C. Merkle. Secure communications over insecure channels.
[BGI08] Eli Biham, Yaron J. Goren, and Yuval Ishai. Basing weak public-key cryptography on strong one-way functions.
[LLW19] Rio LaVigne, Andrea Lincoln and Virginia Vassilevska Williams. Public-Key Cryptography in the Fine-Grained Setting
[CM97] Christian Cachin and Ueli Maurer. Unconditional security against memory-bounded adversaries.
[ACMS23] Afshar, Couteau, Mahmoody, and Sadeghi. Fine-grained non-interactive key-exchange: Constructions and lower bounds

Existing works

- **Bounded time setting** [Mer78, BGI08/LLW19/ACMS23]
  - Primitive: (Multi-party) key exchange
  - Assumption: random oracle, expo… ro k-clique/multilinear Shoup's GG…
  - Honest party: $\mathcal{C}_1 = O(t)$
  - Adversary: $\mathcal{C}_2 = o(t^2)$ /$o(t$…

Fine-grained multi-party NIKE
[BCS24]
Assumption: exponential secure injective PRGs and sub-exponential hardness of CDH/multilinear Maurer‘s GGM

- **Bounded storage setting** [CM97]
  - Primitive: Key exchange
  - Assumption: None
  - Honest party: $\mathcal{C}_1 = O(s)$
  - Adversary: $\mathcal{C}_2 = o(s^2)$

[Mer78] Ralph C. Merkle. Secure communications over insecure channels.
[BGI08] Eli Biham, Yaron J. Goren, and Yuval Ishai. Basing weak public-key cryptography on strong one-way functions.
[LLW19] Rio LaVigne, Andrea Lincoln and Virginia Vassilevska Williams. Public-Key Cryptography in the Fine-Grained Setting
[CM97] Christian Cachin and Ueli Maurer. Unconditional security against memory-bounded adversaries.
[ACMS23] Afshar, Couteau, Mahmoody, and Sadeghi. Fine-grained non-interactive key-exchange: Constructions and lower bounds

Existing works

- **Bounded time setting** [Mer78, BGI08/LLW19/ACMS23]
  - Primitive: (Multi-party) key exchange
  - Assumption: random oracle, exponentially strong OWF / average-case hard zero k-clique/multilinear Shoup's GGM
  - Honest party: $\mathcal{C}_1 = O(t)$
  - Adversary: $\mathcal{C}_2 = o(t^2)$ /$o(t^{1.5})$/$o(t^{n/n-1})$
- **Bounded storage setting** [CM97]
  - Primitive: Key exchange
  - Assumption: None
  - Honest party: $\mathcal{C}_1 = O(s)$
  - Adversary: $\mathcal{C}_2 = o(s^2)$

[Mer78] Ralph C. Merkle. Secure communications over insecure channels.
[BGI08] Eli Biham, Yaron J. Goren, and Yuval Ishai. Basing weak public-key cryptography on strong one-way functions.
[LLW19] Rio LaVigne, Andrea Lincoln and Virginia Vassilevska Williams. Public-Key Cryptography in the Fine-Grained Setting
[CM97] Christian Cachin and Ueli Maurer. Unconditional security against memory-bounded adversaries.
[ACMS23] Afshar, Couteau, Mahmoody, and Sadeghi. Fine-grained non-interactive key-exchange: Constructions and lower bounds

Existing works

- **Bounded time setting** [Mer78, BGI08/LLW19/ACMS23]
  - Primitive: (Multi-party) key exchange
  - Assumption: random oracle, exponentially strong OWF / average-case hard zero k-clique/multilinear Shoup's GGM
  - Honest party: $\mathcal{C}_1 = O(t)$
  - Adversary: $\mathcal{C}_2 = o(t^2)$ /$o(t^{1.5})$,

Multi-party NIKE without idealized assumptions?

- **Bounded storage setting** [CM97]
  - Primitive: Key exchange
  - Assumption: None
  - Honest party: $\mathcal{C}_1 = O(s)$
  - Adversary: $\mathcal{C}_2 = o(s^2)$

[Mer78] Ralph C. Merkle. Secure communications over insecure channels.
[BGI08] Eli Biham, Yaron J. Goren, and Yuval Ishai. Basing weak public-key cryptography on strong one-way functions.
[LLW19] Rio LaVigne, Andrea Lincoln and Virginia Vassilevska Williams. Public-Key Cryptography in the Fine-Grained Setting
[CM97] Christian Cachin and Ueli Maurer. Unconditional security against memory-bounded adversaries.
[ACMS23] Afshar, Couteau, Mahmoody, and Sadeghi. Fine-grained non-interactive key-exchange: Constructions and lower bounds

Our results

Multi-party NIKE in the bounded parallel-time model

Multi-party NIKE in the bounded time model

Multi-party NIKE in the bounded storage model

Our results

Multi-party NIKE in the bounded parallel-time model

Adversary: NC1
Honest user: AC0[2] (included in NC1)
Assumption: $\mathrm{NC}^1 \neq \oplus \mathrm{L/poly}$

Multi

Multi-party NIKE in the bounded storage model

Our results

Multi-party NIKE in the bounded parallel-time model

Multi

Adversary: NC1
Honest user: AC0[2] (included in NC1)
Assumption: $\mathrm{NC}^1 \neq \oplus \mathrm{L/poly}$

Multi

AC0[2]: circuits with constant depth, polynomial size, and unbounded fan-in using AND, OR, NOT, and PARITY gates

Our results

Multi-party NIKE in the bounded parallel-time model

Multi

Adversary: NC1
Honest user: AC0[2] (included in NC1)
Assumption: $\mathrm{NC}^1 \neq \oplus \mathrm{L/poly}$

Multi-party NIKE in the … torage model

NC1: circuits with logarithm depth, polynomial-size and fan-in 2 gates

Our results

Multi-party NIKE in the bounded parallel-time model

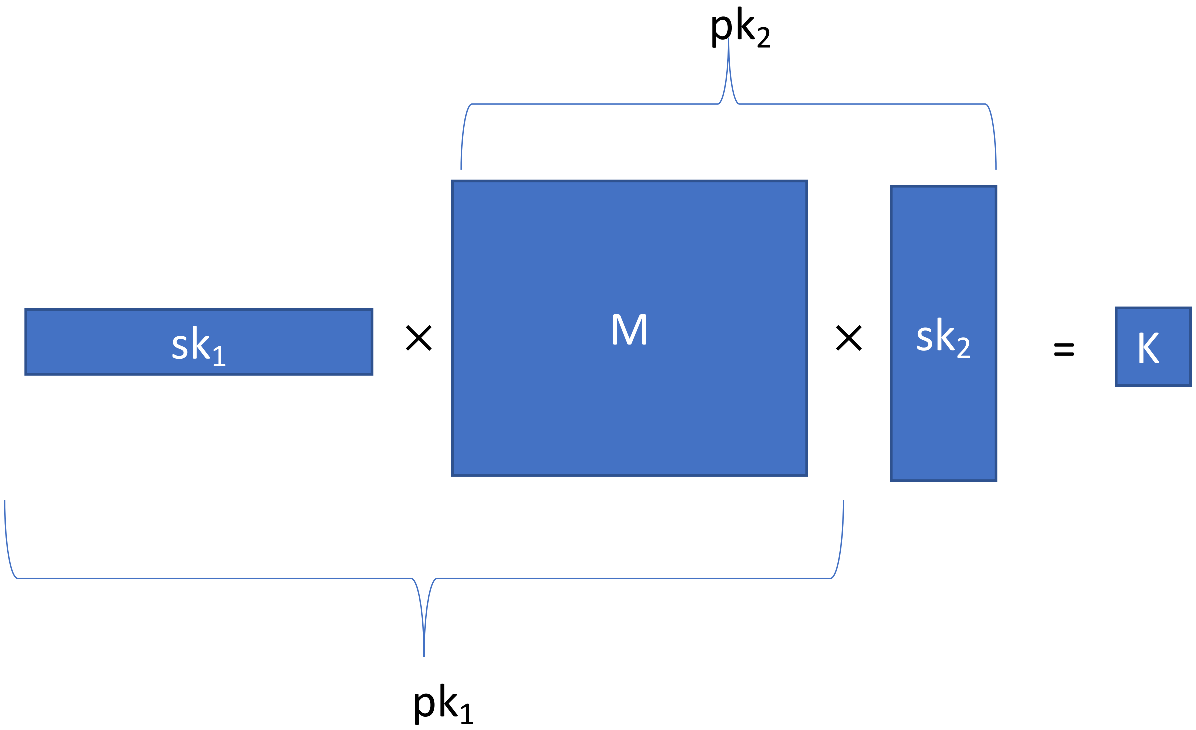
Multi

Adversary: NC1
Honest user: AC0[2] (included in NC1)
Assumption: $\mathrm{NC}^1 \neq \oplus \mathrm{L/poly}$

Multi-party NIKE in th... model

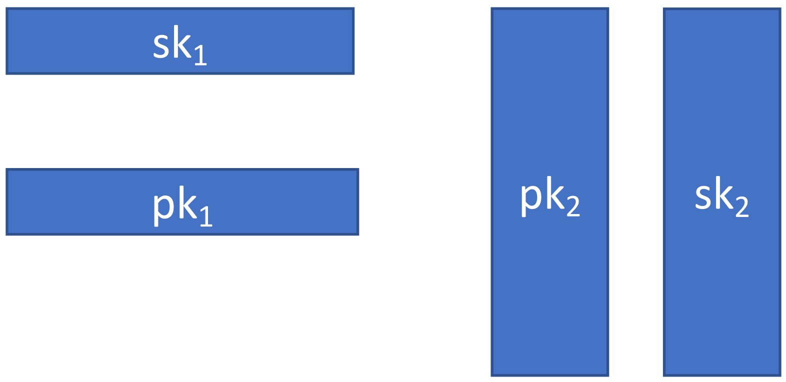
$\oplus$ L/poly : log space turning machine with parity acceptance

Starting point: fine-grained HPS



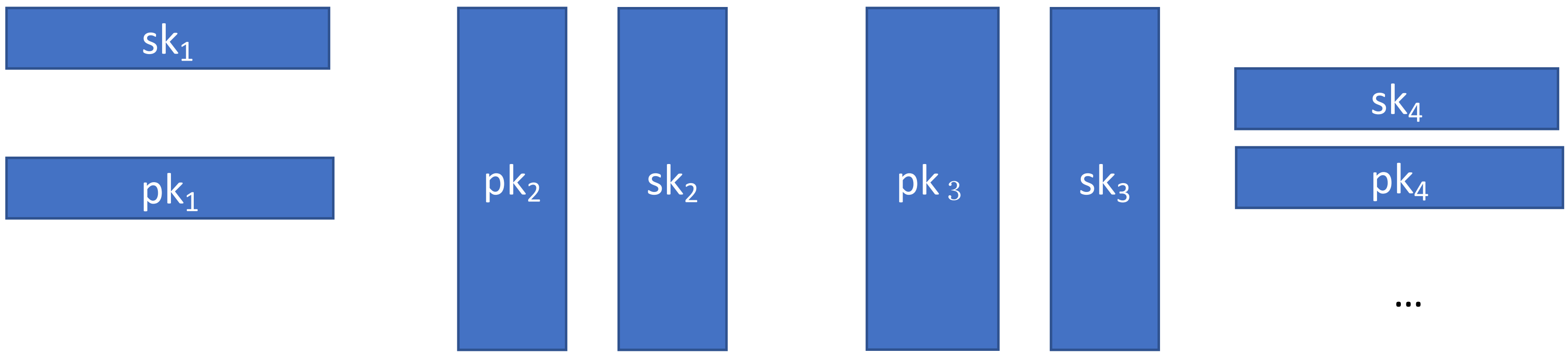
Security: smoothness of HPS based on $\mathrm{NC}^1 \neq \oplus \mathrm{L/poly}$

Starting point: fine-grained HPS



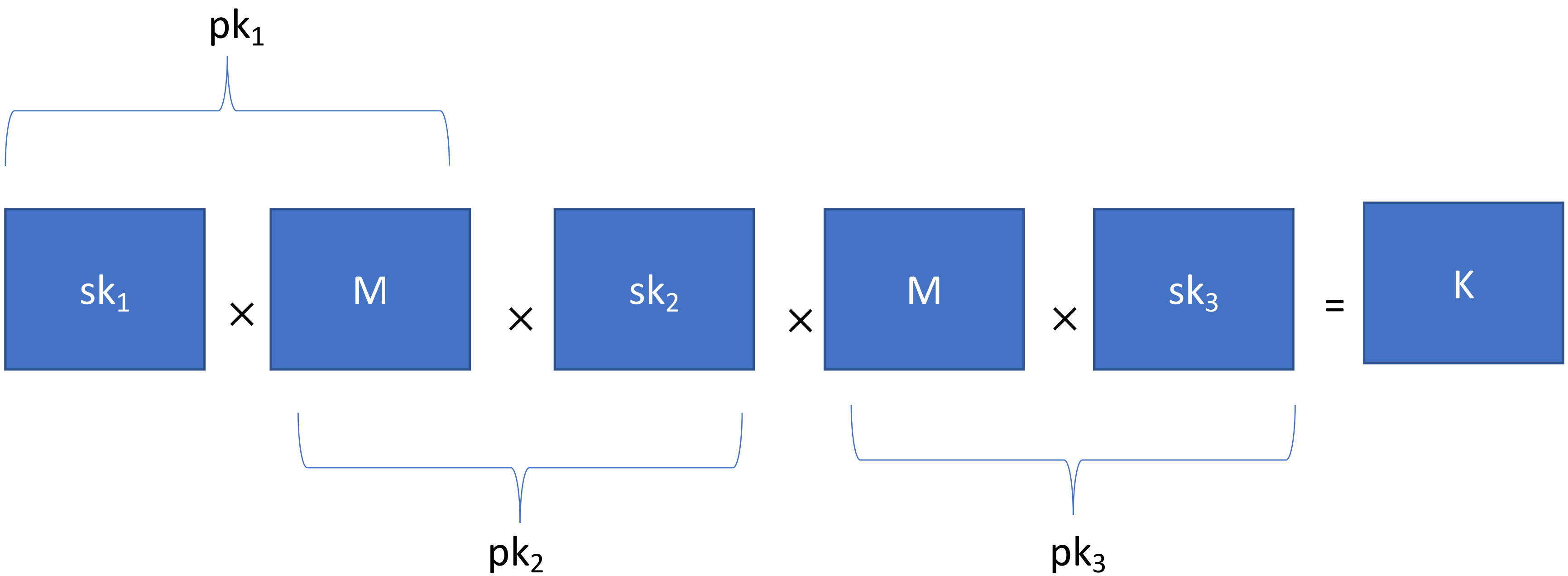
Key pairs are vectors generated by different types of algorithms

Starting point: fine-grained HPS

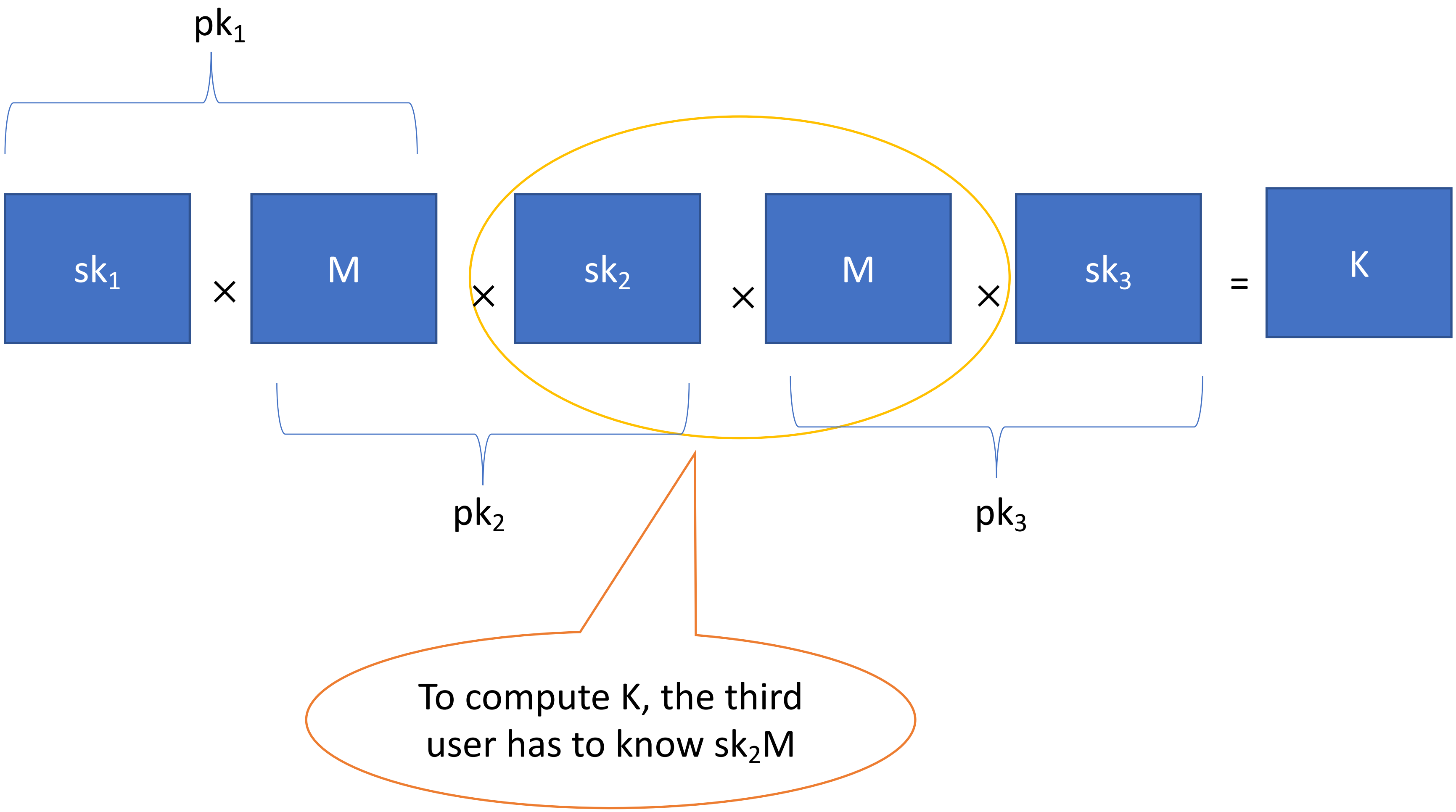


When more parties are involved, it is unclear how to combine a bunch of vectors to generate a session key

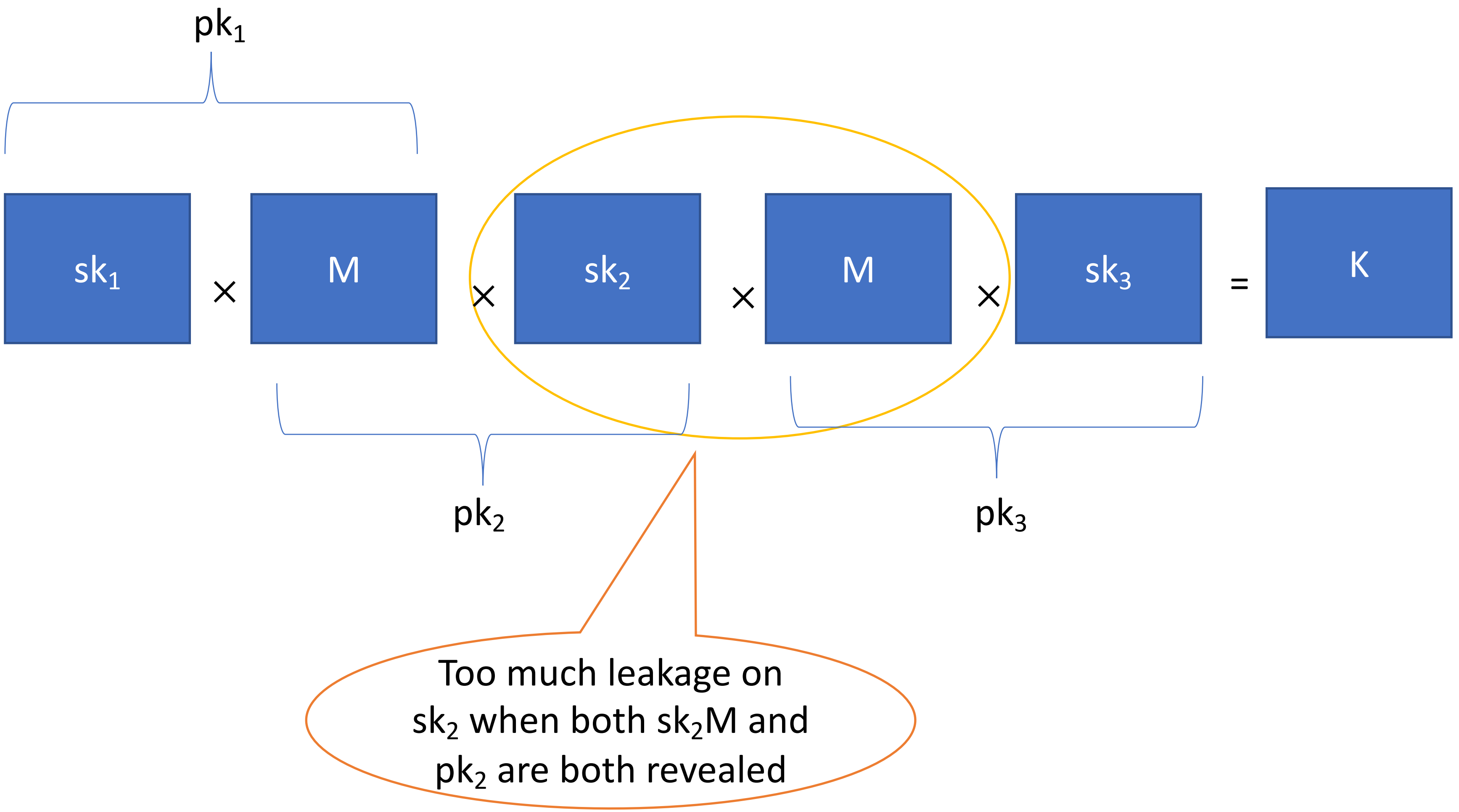
Strawman solution: vectors to matrices



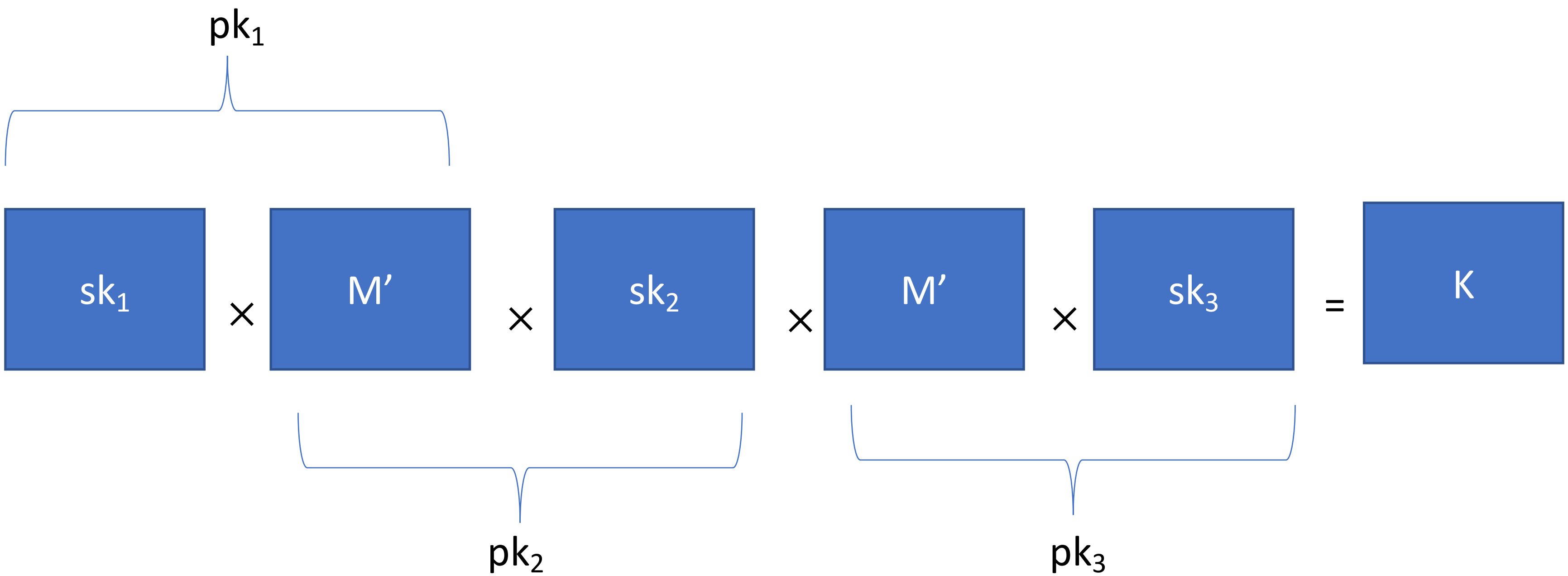
Strawman solution: vectors to matrices



Strawman solution: vectors to matrices

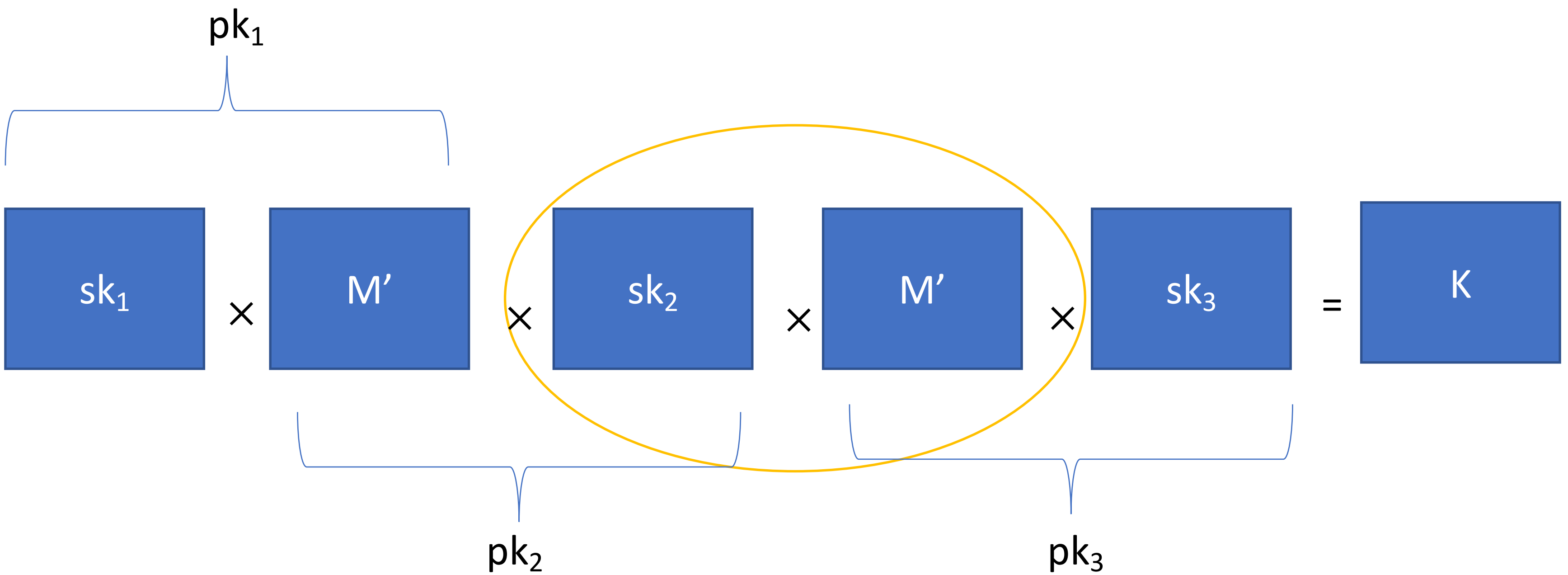


Solution: symmetric matrices



$M' = M^{\top}M$ and $sk_i \leftarrow \mathrm{SymR}$,
where SymR is the uniform
distribution over symmetric matrices

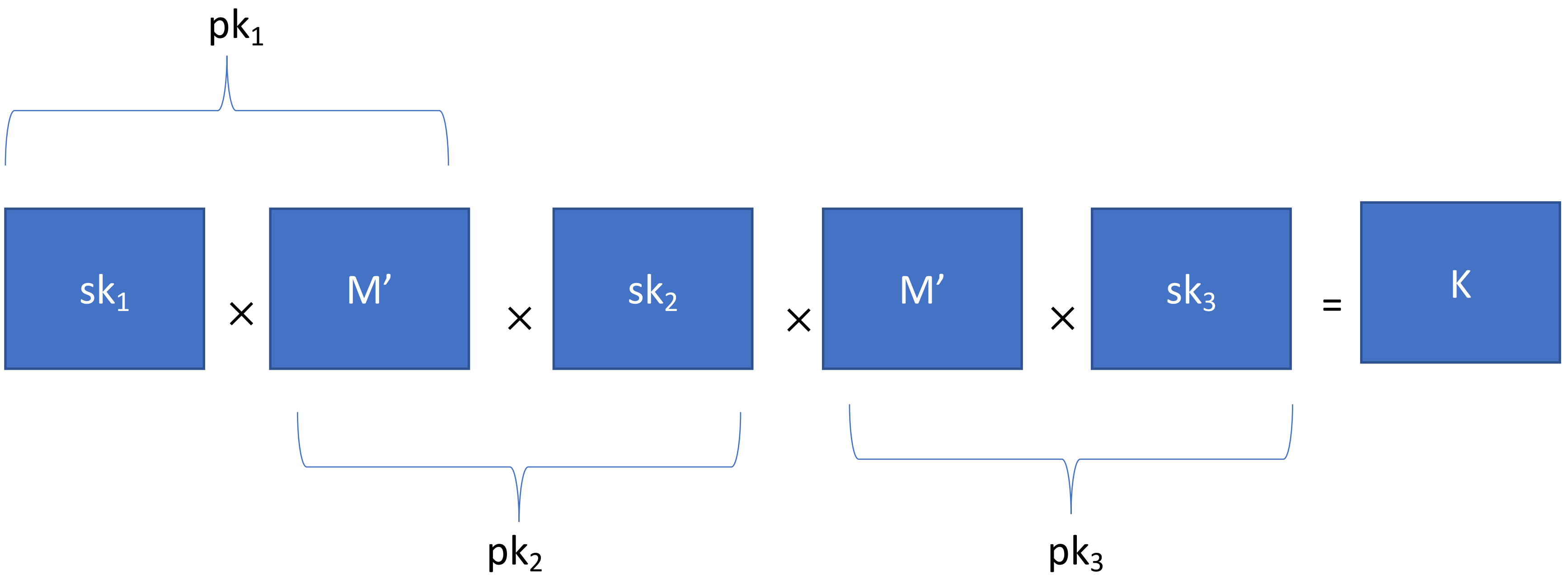
Solution: symmetric matrices



The third party knows $sk_2M' = pk_2^T$ now

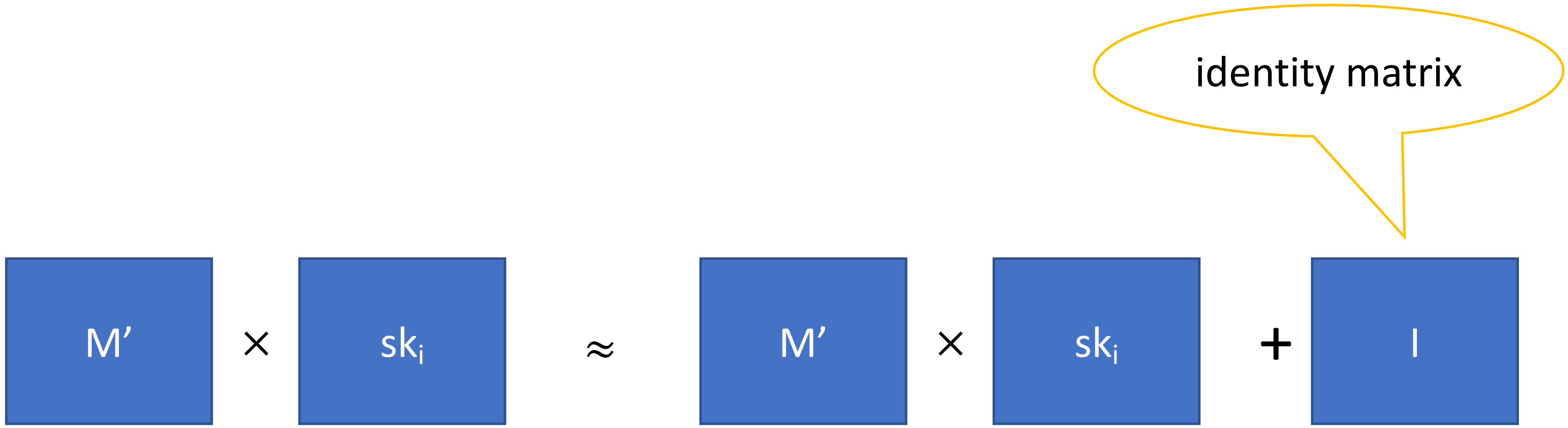
=> Correctness is guaranteed

Solution: symmetric matrices



Smoothness of HPS cannot be used for security proof since sk_i are not uniformly random

Core lemma



Solution: symmetric matrices

$$sk_1 \times M' \times sk_2 \times M' \times sk_3 = K$$

$$\approx$$

$$sk_1 \times \left(M' \times sk_2 + I \right) \times \left(M' \times sk_3 + I \right)$$

$$= sk_1 \times \left(M' \times R + I \right) \quad \text{for some } R$$

Last column is outside the span of M’

Solution: symmetric matrices

$$sk_1 \times M' \times sk_2 \times M' \times sk_3 = K$$

$$\approx$$

$$sk_1 \times \left(M' \times sk_2 + I \right) \times \left(M' \times sk_3 + I \right)$$

$$= sk_1 \times \left(M' \times R + I \right)$$ for some R

The bottom-right bit of the result is a proof with smoothness

Solution: symmetric matrices

$$sk_1 \times M' \times sk_2 \times M' \times sk_3 = K$$

$$\approx$$

$$sk_1 \times \left(M' \times sk_2 + I \right) \times \left(M' \times sk_3 + I \right)$$

$$= sk_1 \times \left(M' \times R + I \right) \quad \text{for some } R$$

Key length can be increased by running the scheme in parallel

Our results

Multi-party NIKE in the bounded parallel-time model

Multi-party NIKE in the bounded time model

Multi-party NIKE in the bounded storage model

Our results

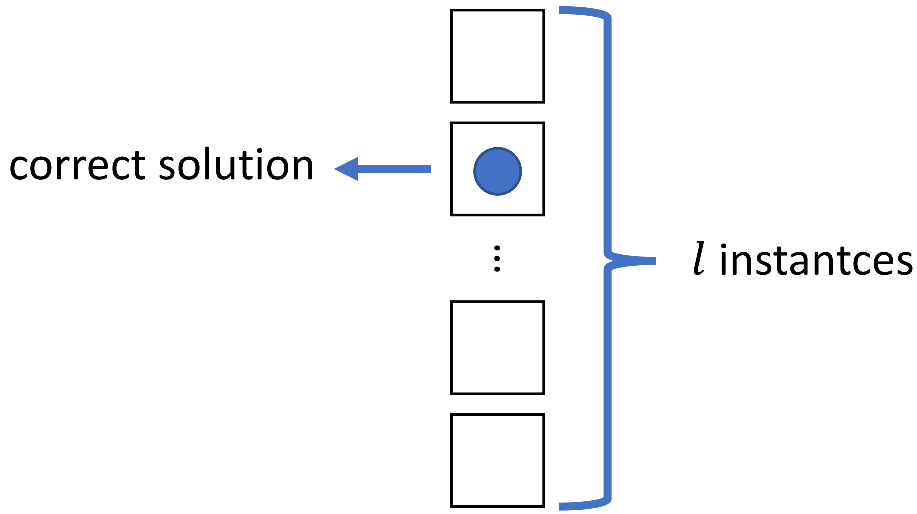
Multi-party NIKE in the bounded parallel-time model

Multi-party NIKE in the bounded time model

Multi-

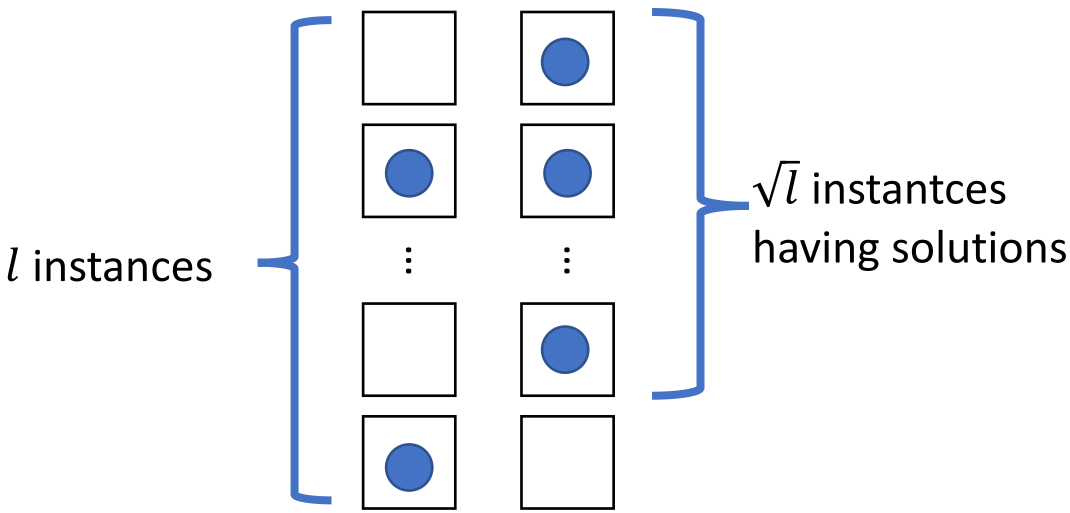
Adversary: $\tilde{O}\left(\lambda^{n_p+k}\right)$
Honest user: $\tilde{O}\left(\lambda^{n_p+k-1}\right)$
Assumption: average-case hard zero k-clique

Base



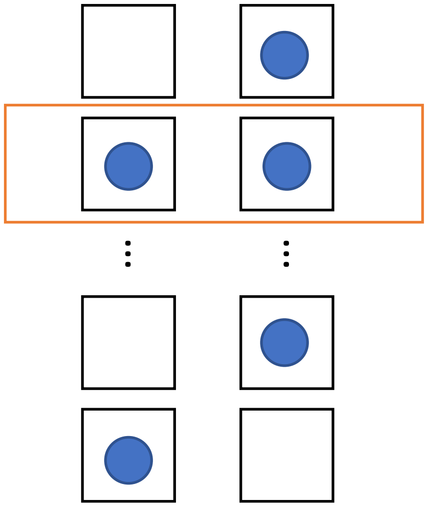
Finding the correct solution requires essentially solving all instances [LLW19]

Starting point: two party key exchange



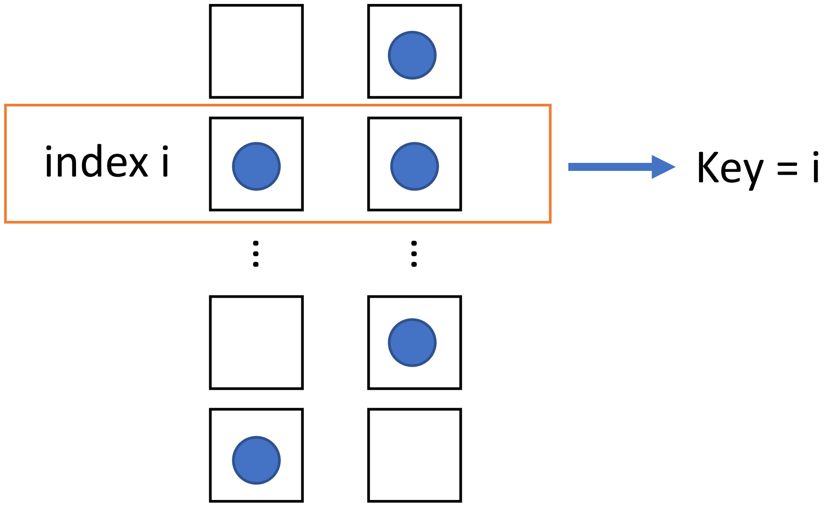
Two parties exchange lists

Starting point: two party key exchange

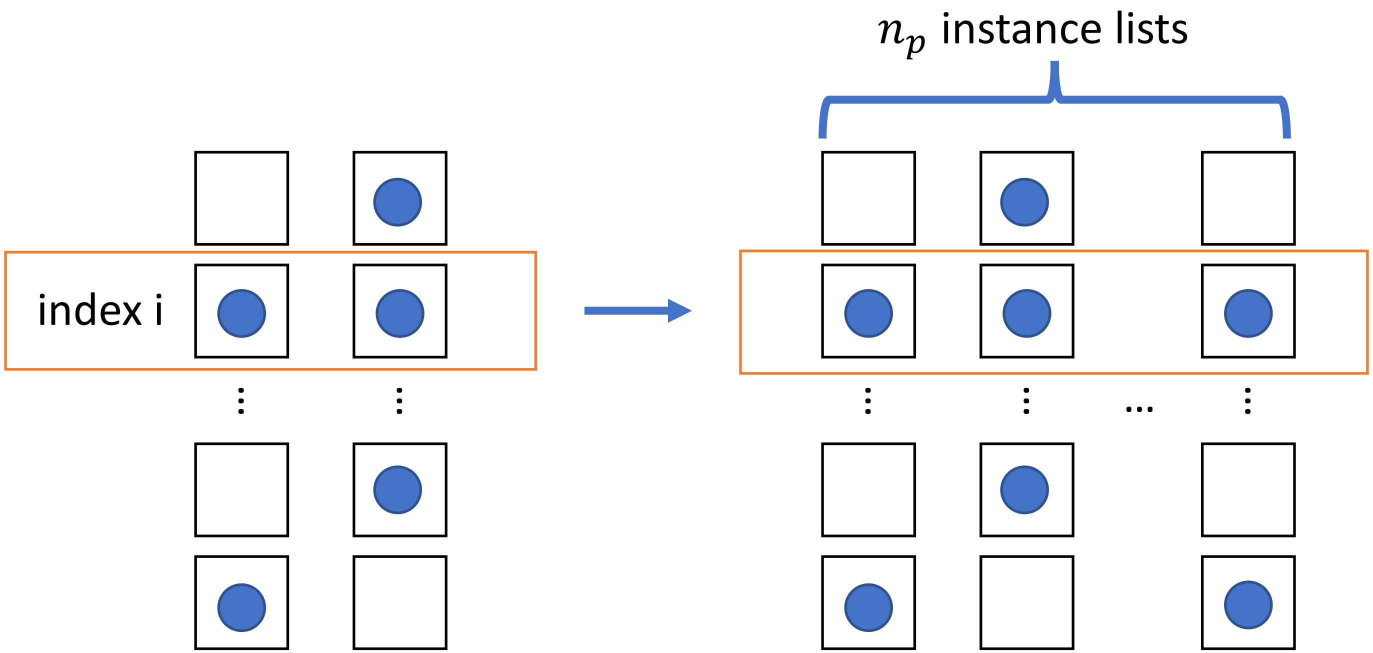


There would be a single instance in common having a solution

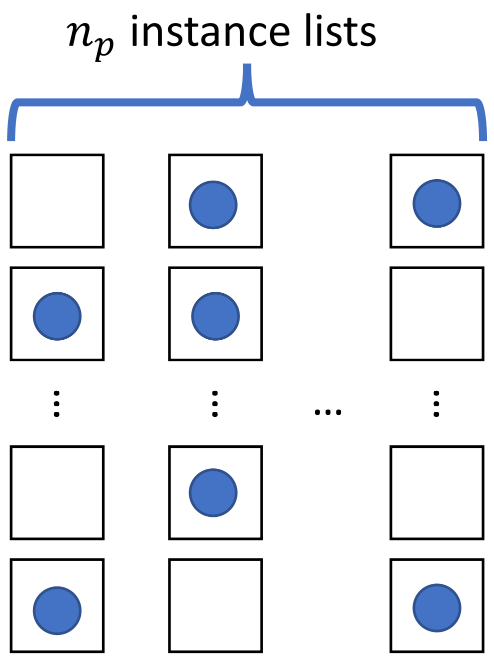
Starting point: two party key exchange



Extension to multi-party setting

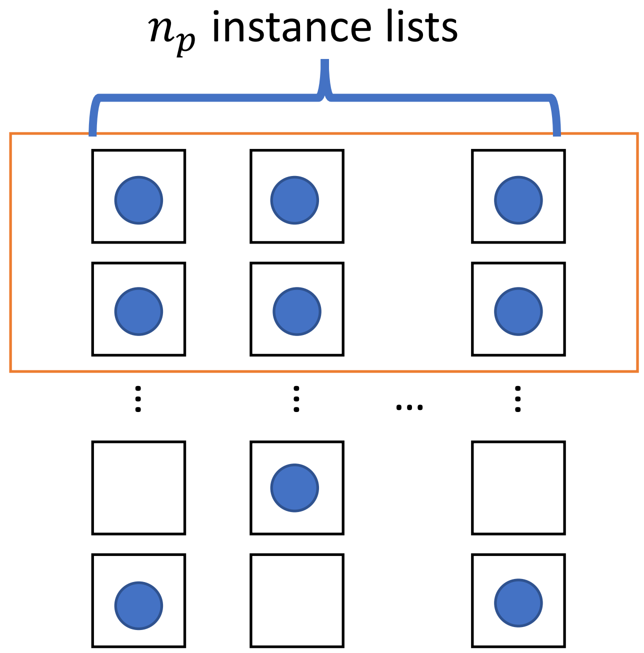


Extension to multi-party setting



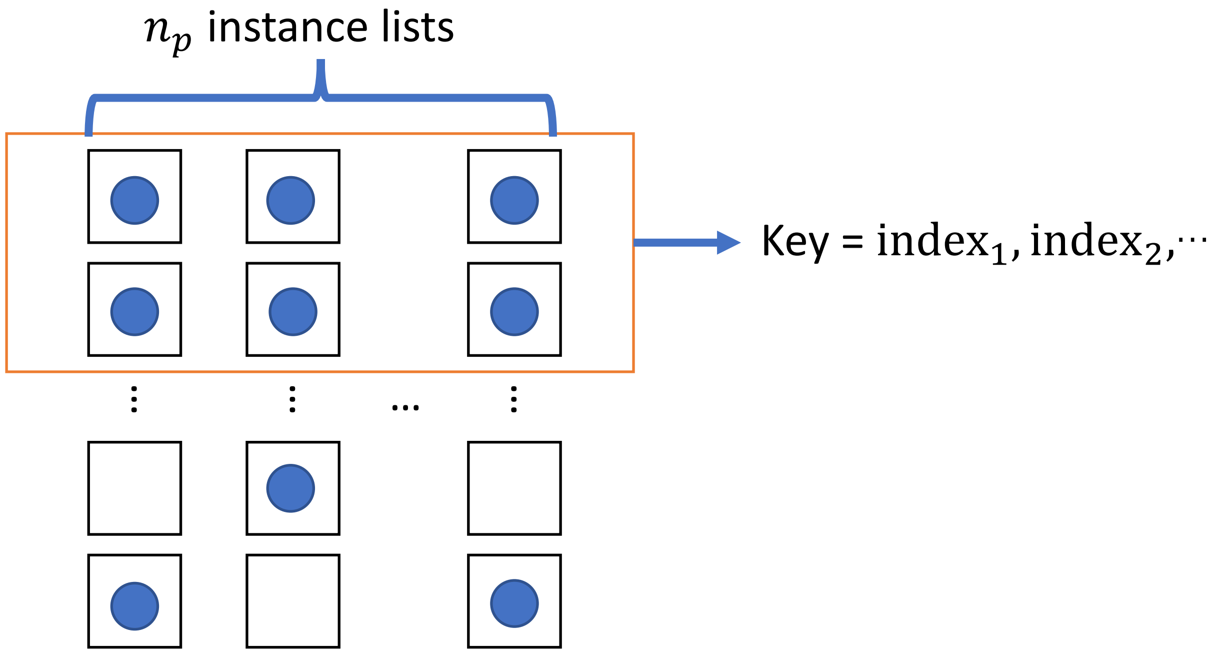
Challenge: the probability of users sharing the same indices decreases rapidly

Extension to multi-party setting



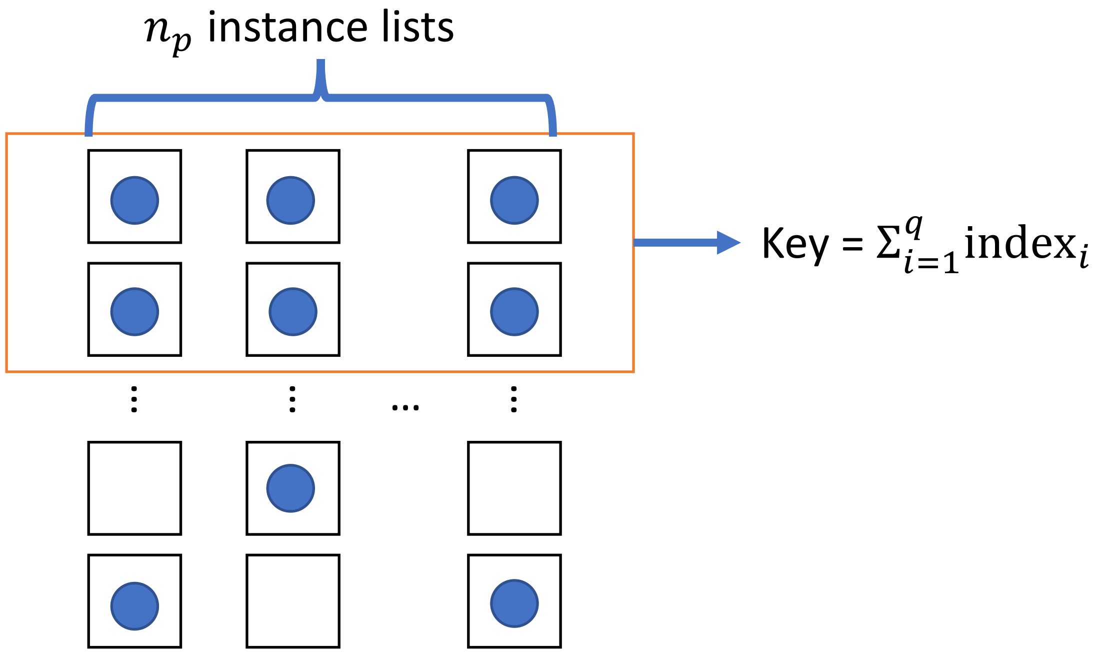
Solution: increase the number of instances with solutions

Extension to multi-party setting



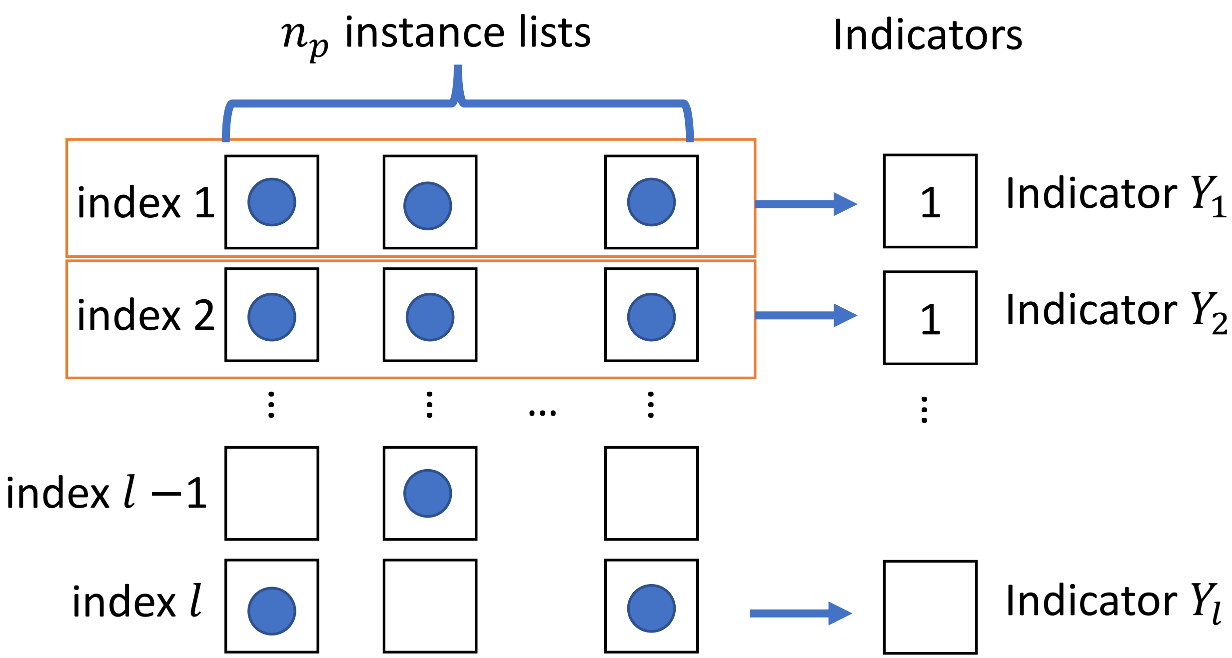
As the number of instances increases, the number of the common indices also increases

Extension to multi-party setting



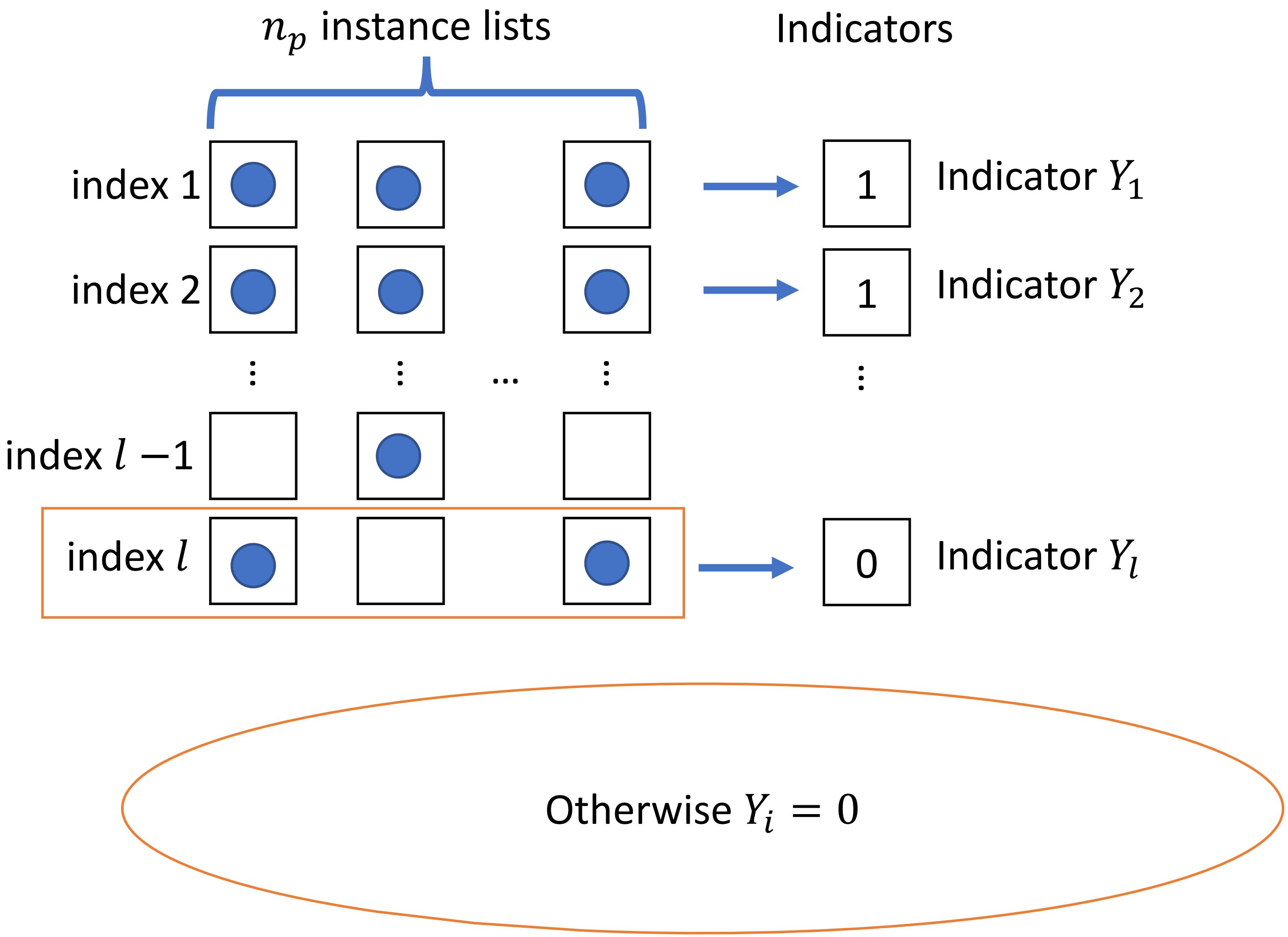
Share the sum of the overlapping part

Correctness

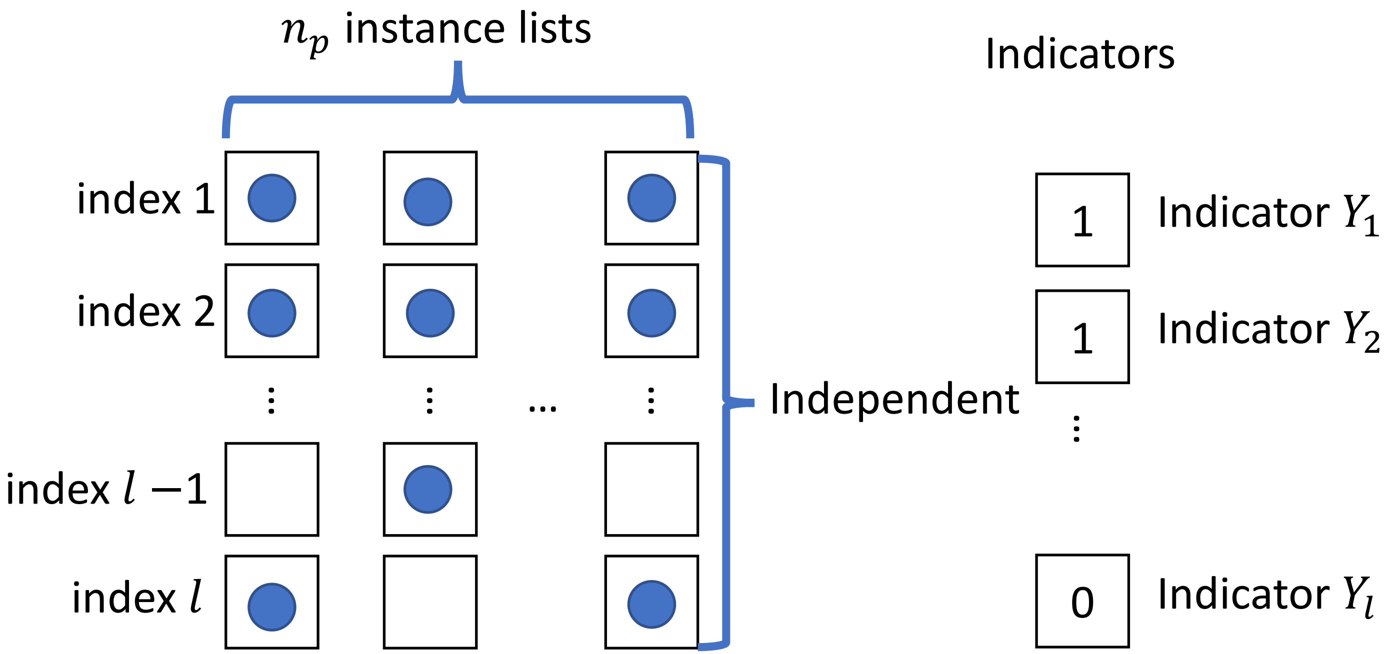


$Y_i = 1$ if the i'th index chosen by the last party is also chosen by all parties

Correctness

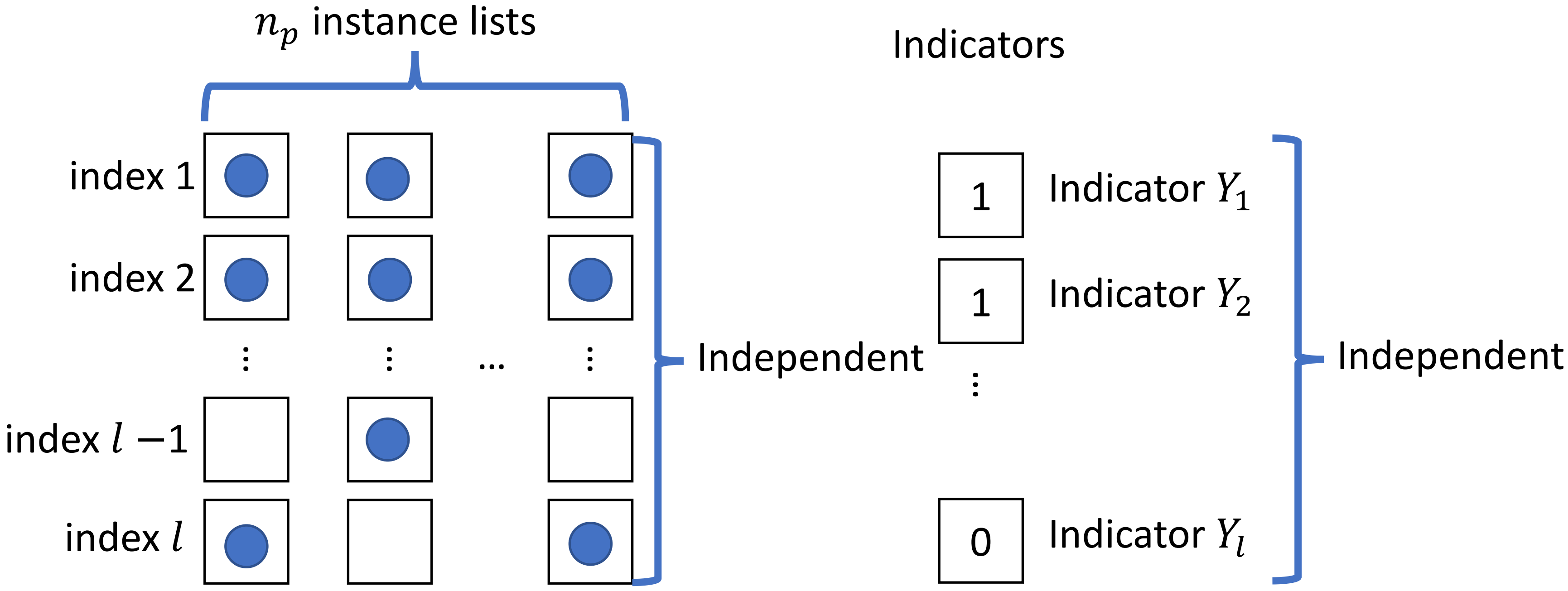


Correctness



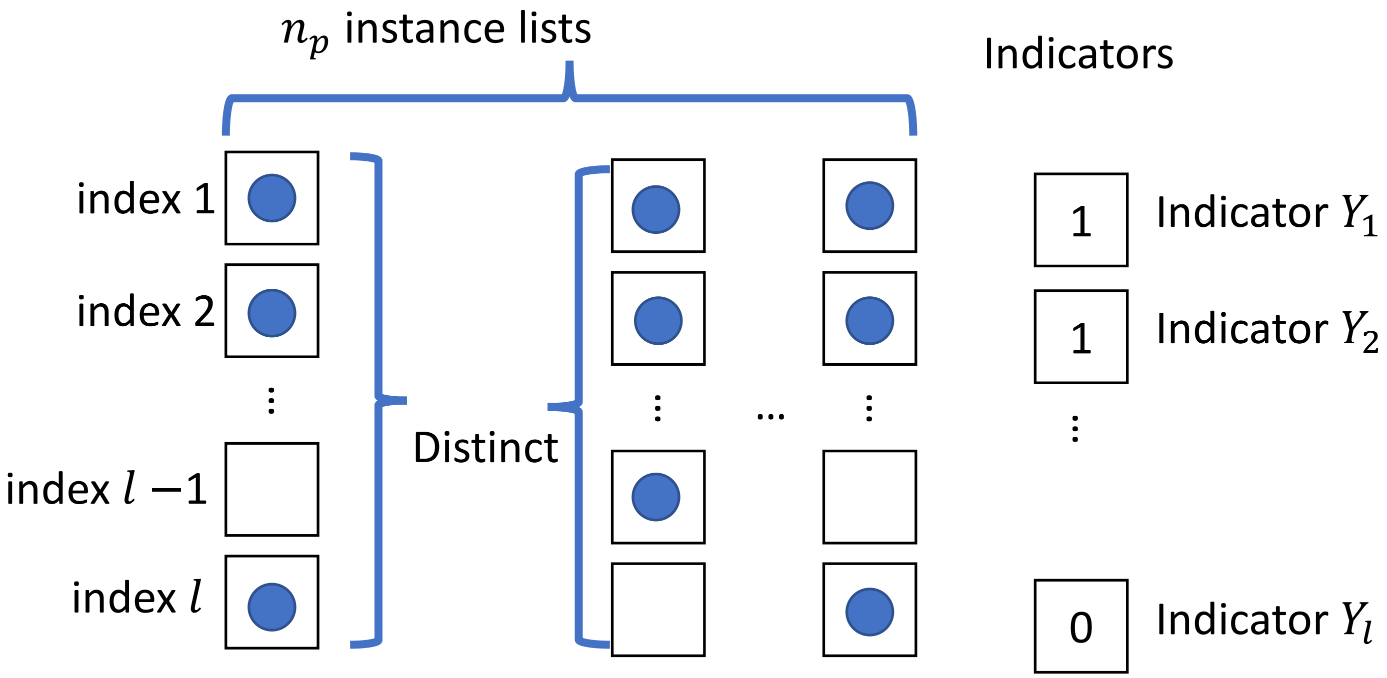
w.o.l.g., we assume the indices chosen by the last party are genuinely independent

Correctness



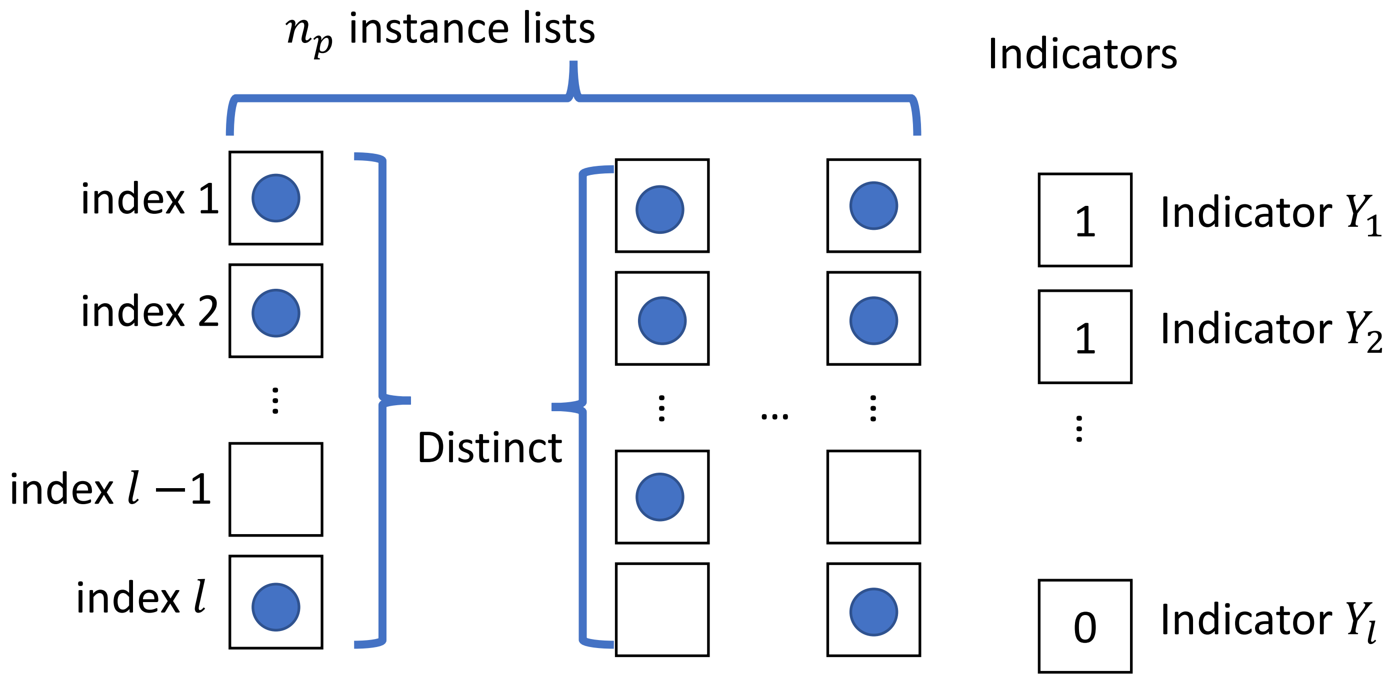
This guarantees the independence of these indicators

Correctness



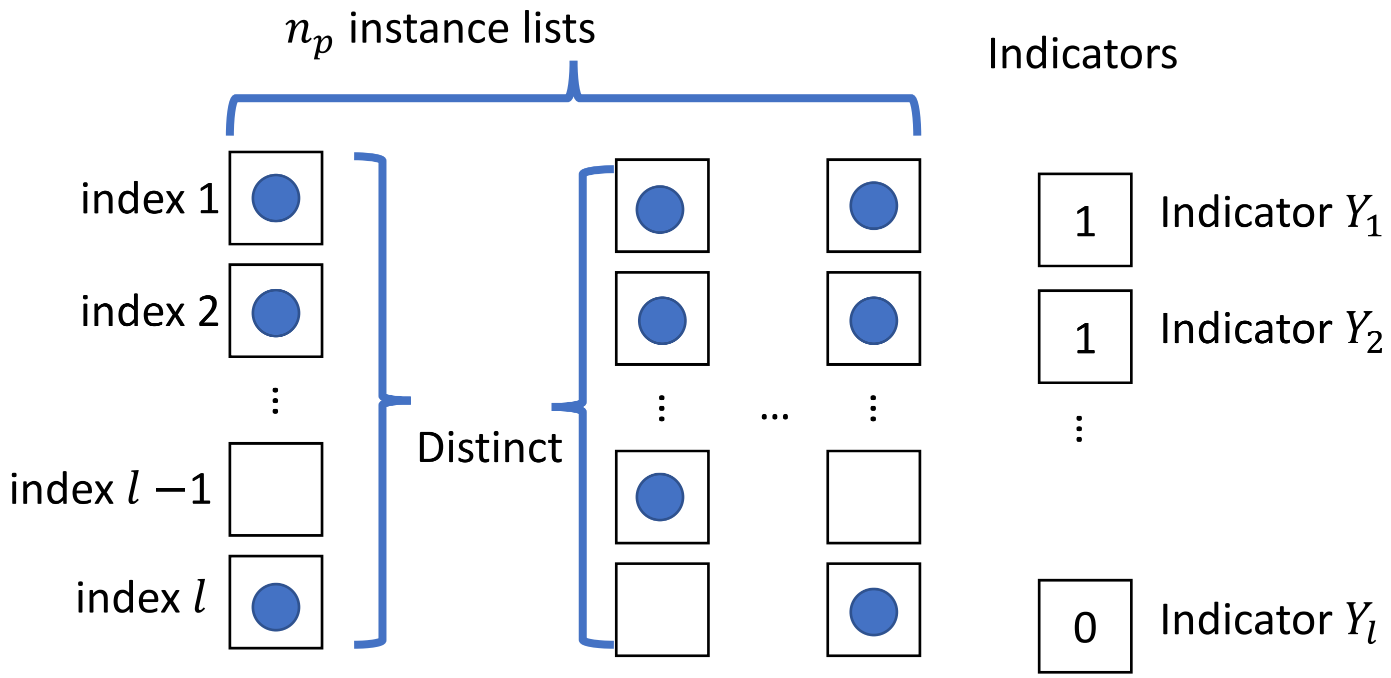
Indices chosen by all other parties are independent but all distinct

Correctness



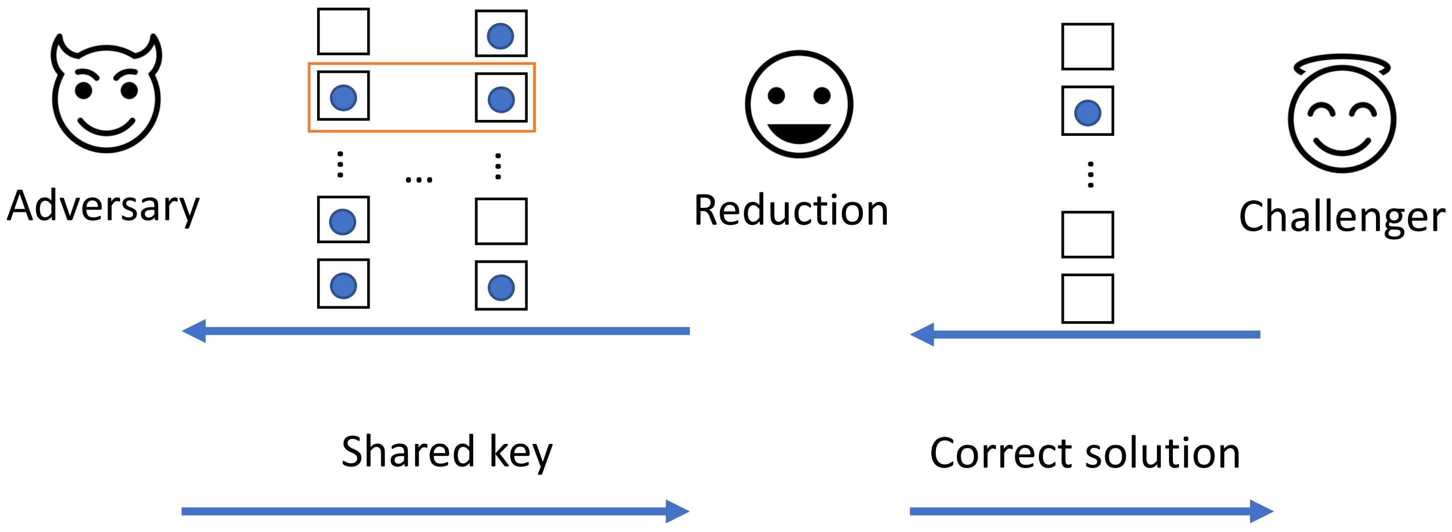
$\Pr[Y_i = 0]$ is small

Correctness



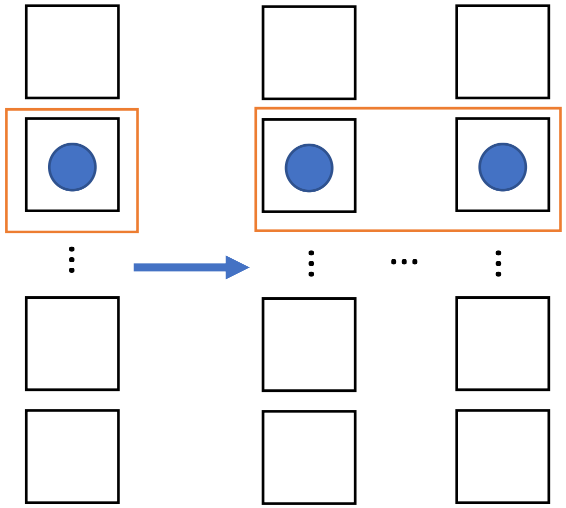
$1 - \prod \Pr[Y_i = 0]$ is large

Security



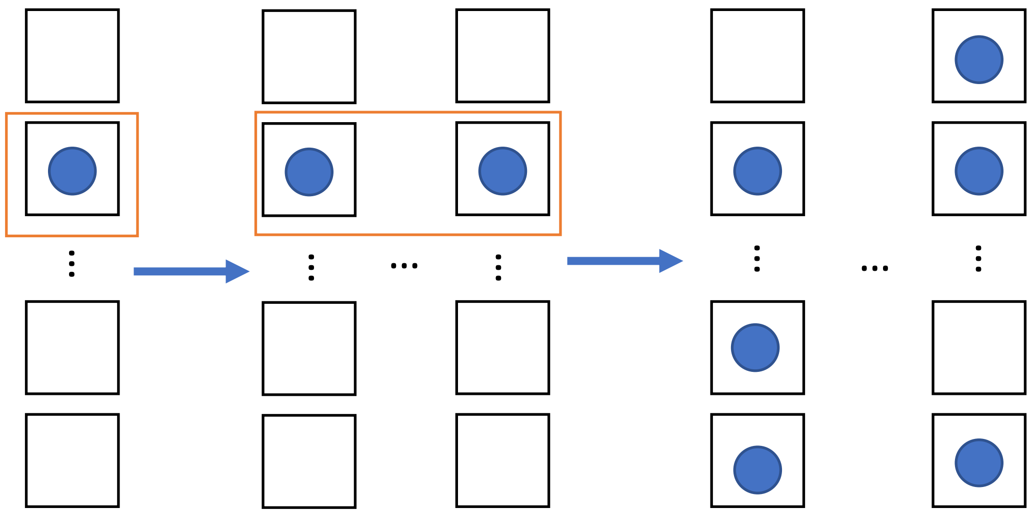
The reduction aims to find the correct solution by making use of an adversary

Security



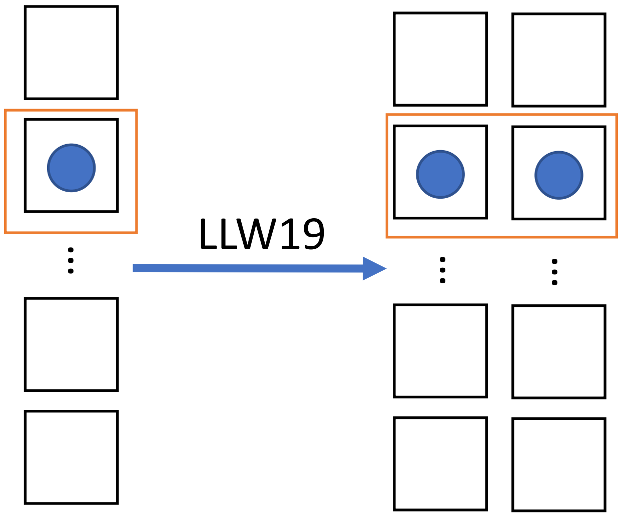
To simulate the view of the adversary, the reduction splits the list into multiple ones

Security



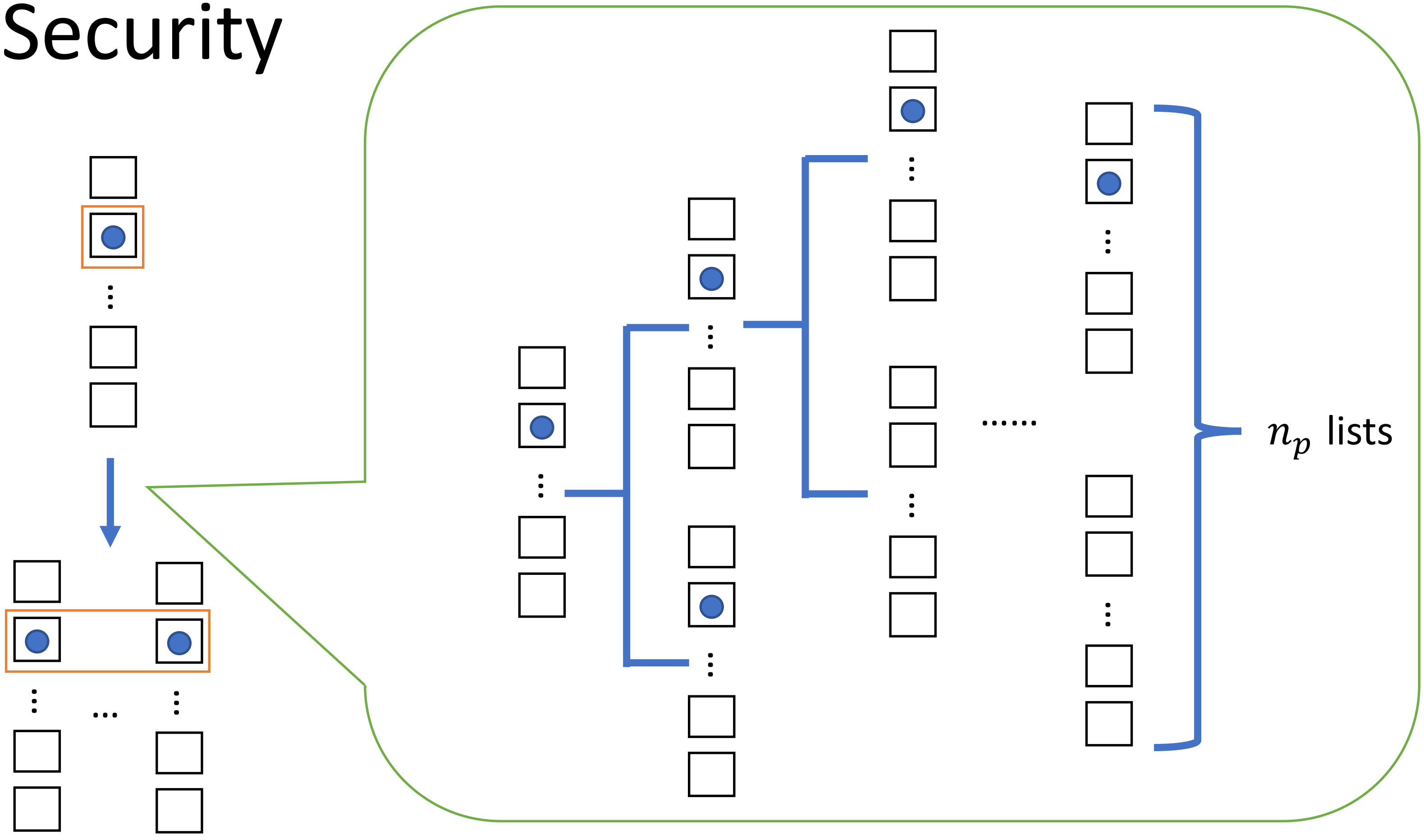
Then plants other solutions into the list

Security



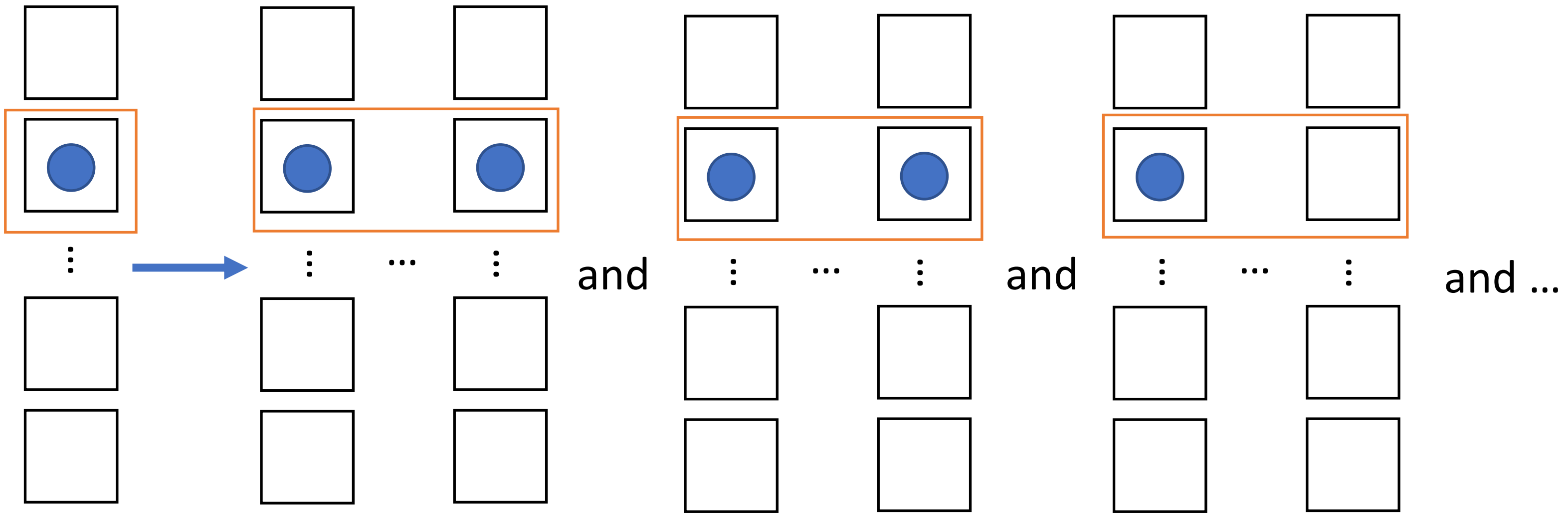
The existing splitting algorithm splits the list into list pairs [LLW19]

Security



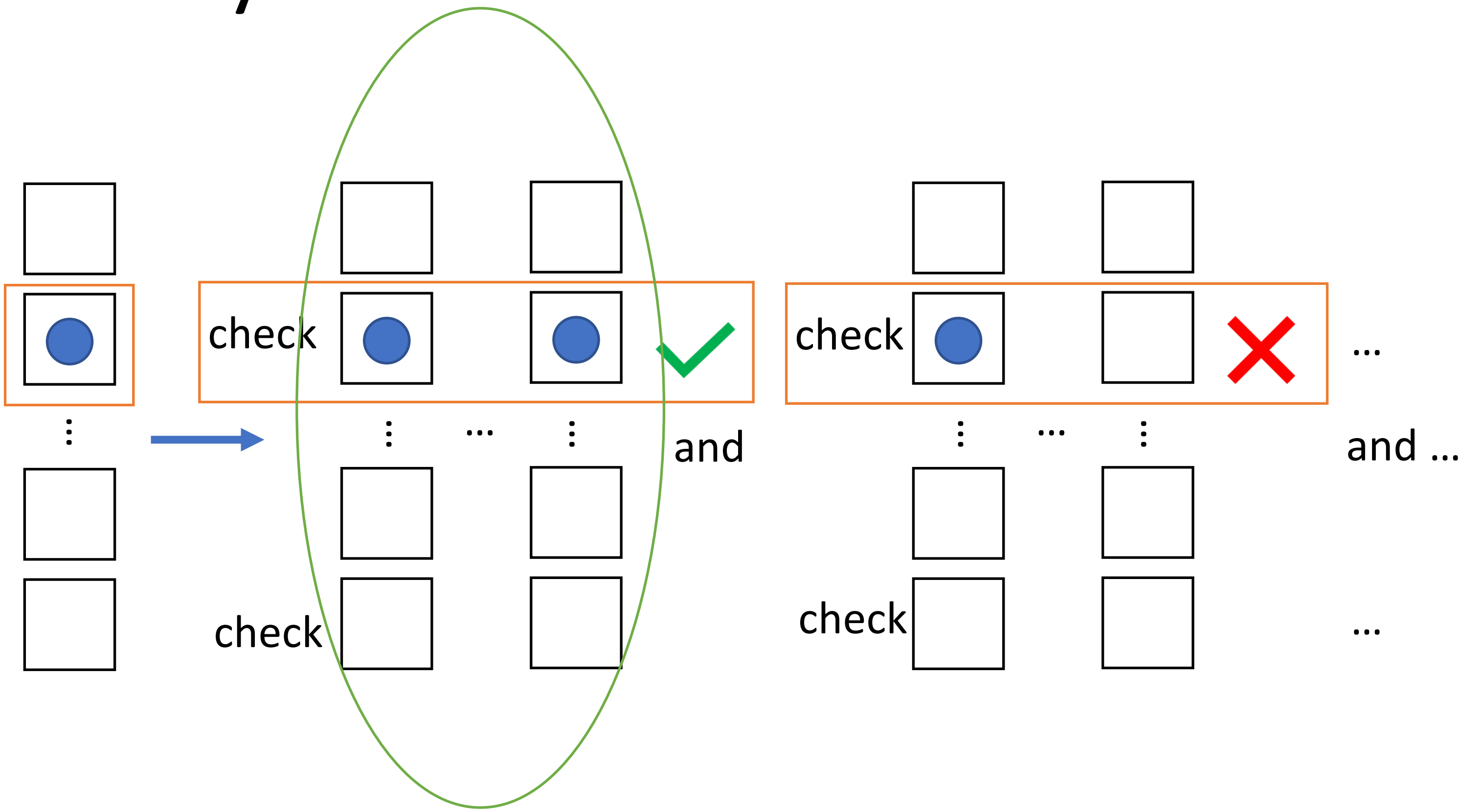
To split one instance into multiple ones,
we introduce a binary tree structure

Security



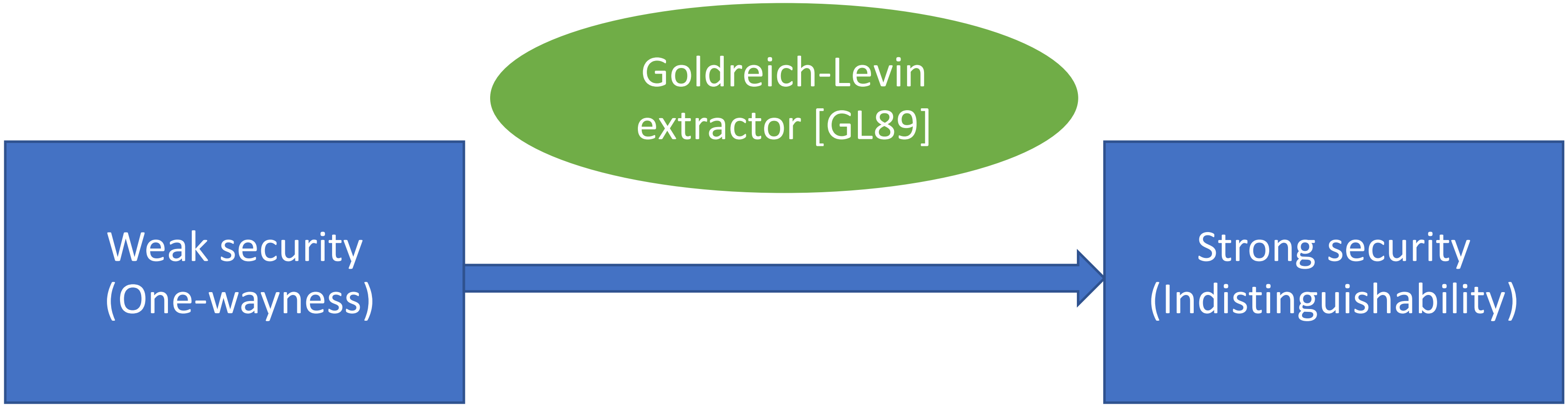
The generalized splitting algorithm also produce incorrect instance lists with different solutions.

Security



By introducing additional efficient checking procedures, we can eliminate incorrect lists.

Privacy amplification



Our results

Multi-party NIKE in the bounded parallel-time model

Multi-party NIKE in the bounded time model

Multi-party NIKE in the bounded storage model

Our results

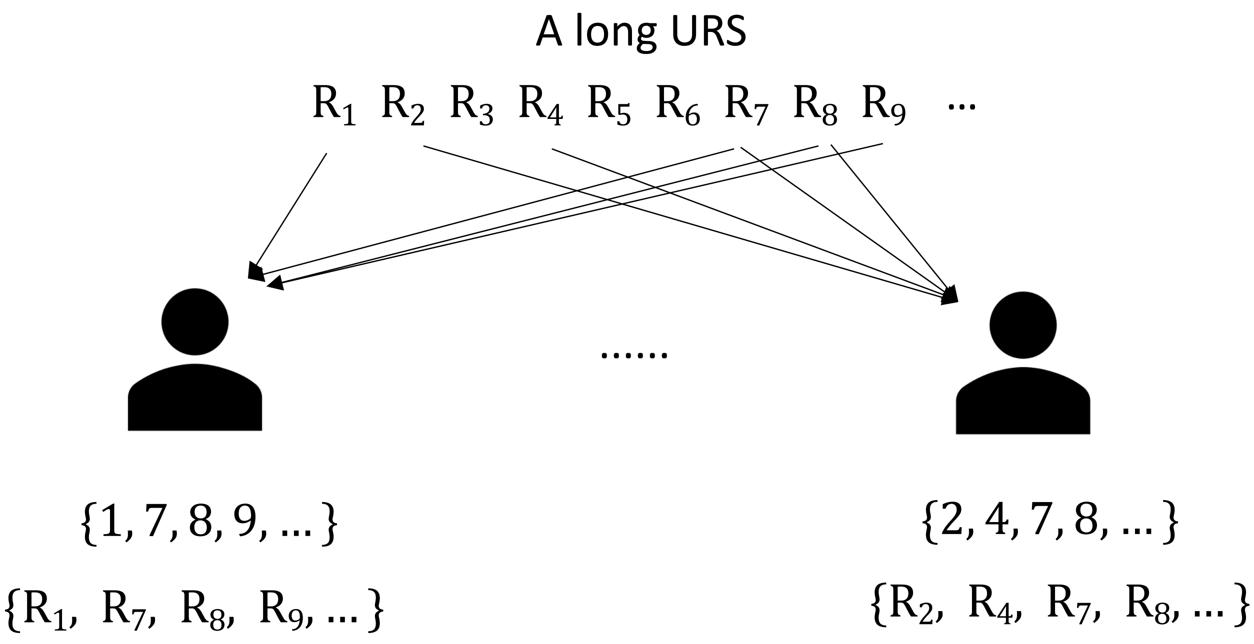
Multi-party NIKE in the bounded parallel-time model

Multi-party NIKE in the bounded time model

Multi-party NIKE in the bounded storage model

Adversary: $O(\lambda^{n_p+1})$
Honest user: $O(\lambda^{n_p})$

NIKE in the BSM



Each party stores a set of bits in the URS

NIKE in the BSM

....

$\mathrm{pk}_1 = \{1, 7, 8, 9, \ldots\}$

$\mathrm{sk}_1 = \{\mathrm{R}_1, \mathrm{R}_7, \mathrm{R}_8, \mathrm{R}_9, \ldots\}$

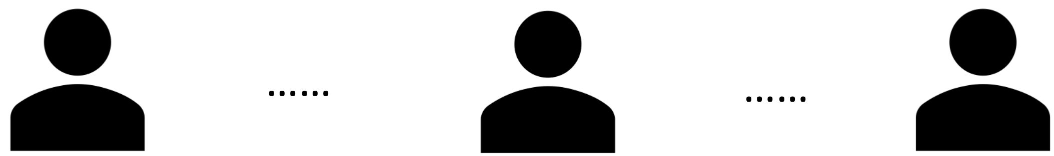
$\mathrm{pk}_{n_p} = \{2, 4, 7, 8, \ldots\}$

$\mathrm{sk}_{n_p} = \{\mathrm{R}_2, \mathrm{R}_4, \mathrm{R}_7, \mathrm{R}_8, \ldots\}$

$\mathrm{pk}_1, \ldots, \mathrm{pk}_{n_p}$

Once the URS disappears, the parties exchange the indices

NIKE in the BSM



$\mathrm{pk}_1 = \{1, 7, 8, 9, \ldots\}$

$\mathrm{sk}_1 = \{\mathrm{R}_1,\ \mathrm{R}_7,\ \mathrm{R}_8,\ \mathrm{R}_9, \ldots\}$

$\mathrm{pk}_{n_p} = \{2, 4, 7, 8, \ldots\}$

$\mathrm{sk}_{n_p} = \{\mathrm{R}_2,\ \mathrm{R}_4,\ \mathrm{R}_7,\ \mathrm{R}_8, \ldots\}$

$\mathrm{pk}_1, \ldots, \mathrm{pk}_{n_p}$ ⟷

$\{7, 8, \ldots\}$

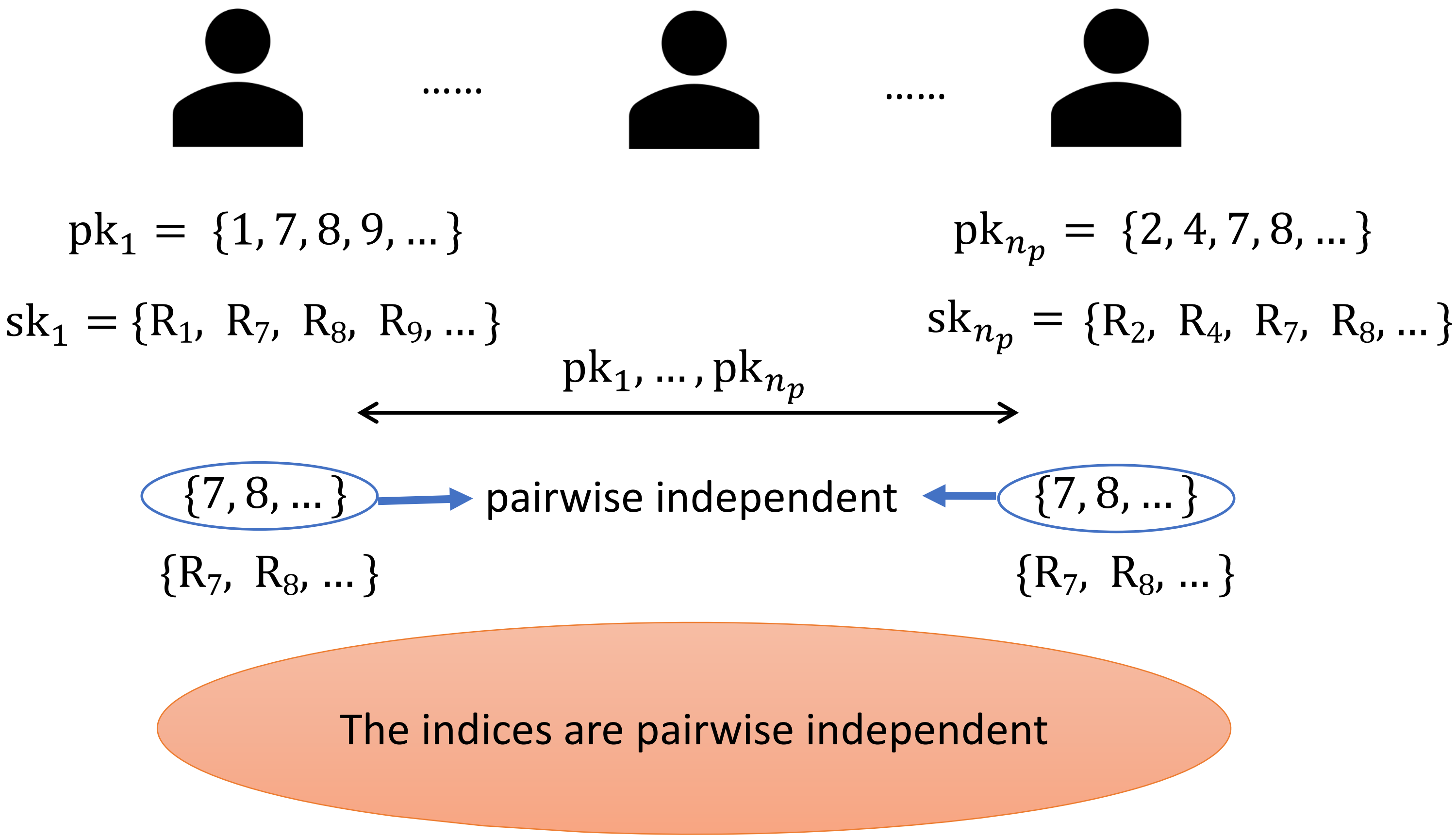
$\{\mathrm{R}_7,\ \mathrm{R}_8, \ldots\}$

$\{7, 8, \ldots\}$

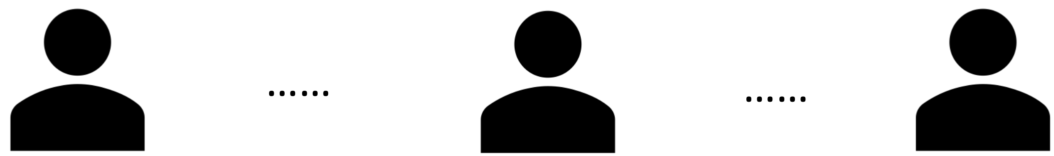
$\{\mathrm{R}_7,\ \mathrm{R}_8, \ldots\}$

Then save the bits in common

NIKE in the BSM



NIKE in the BSM



$\mathrm{pk}_1 = \{1, 7, 8, 9, \ldots\}$

$\mathrm{sk}_1 = \{R_1,\ R_7,\ R_8,\ R_9, \ldots\}$

$\mathrm{pk}_{n_p} = \{2, 4, 7, 8, \ldots\}$

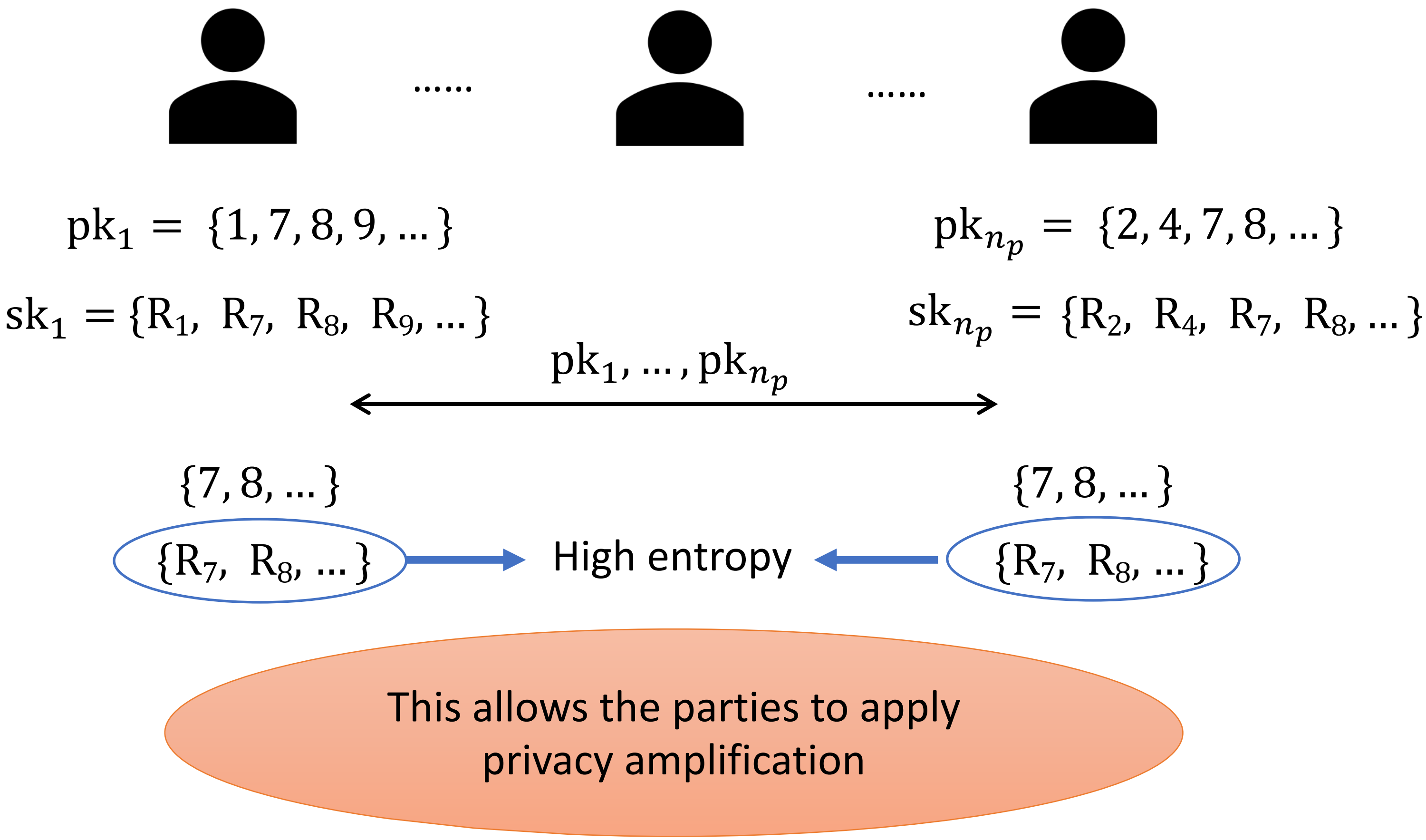
$\mathrm{sk}_{n_p} = \{R_2,\ R_4,\ R_7,\ R_8, \ldots\}$

$\mathrm{pk}_1, \ldots, \mathrm{pk}_{n_p}$

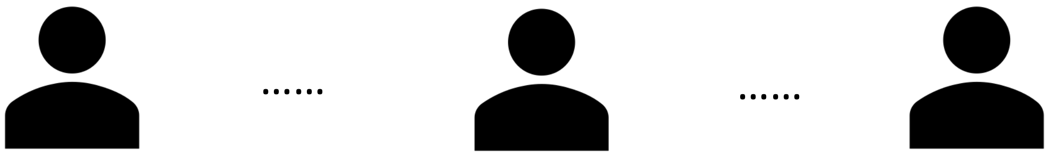
$\{7, 8, \ldots\}$ $\{R_7,\ R_8, \ldots\}$ → High entropy ← $\{R_7,\ R_8, \ldots\}$ $\{7, 8, \ldots\}$

So the shared bits have high entropy

NIKE in the BSM



NIKE in the BSM



$\mathrm{pk}_1 : h_1 \leftarrow H_n$

$\mathrm{pk}_{n_p} : h_{n_p} \leftarrow H_n$

$\mathrm{sk}_1 = \{\mathrm{R}_{h_1(1)}, \mathrm{R}_{h_1(2)}, \mathrm{R}_{h_1(3)}, \dots, \mathrm{R}_{h_1(q)}\}$

$\mathrm{sk}_{n_p} = \{\mathrm{R}_{h_{n_p}(1)}, \mathrm{R}_{h_{n_p}(2)}, \mathrm{R}_{h_{n_p}(3)}, \dots, \mathrm{R}_{h_{n_p}(q)}\}$

$\mathrm{pk}_1, \dots, \mathrm{pk}_{n_p}$

To ensure efficiency, the parties utilize strongly 2-universal hash functions

Security in the multi-party setting

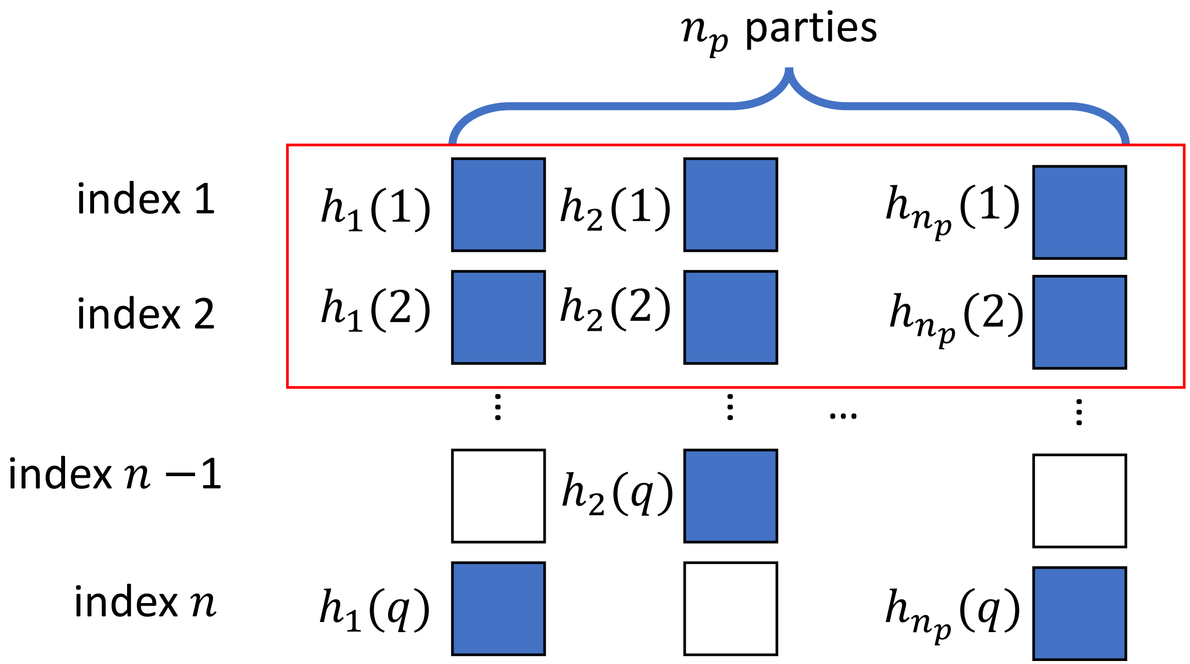
$\mathrm{pk}_1 : h_1 \leftarrow H_n$ $\qquad$ $pk_{n_p} : h_{n_p} \leftarrow H_n$

$$\bigcap_{i=1}^{n_p} \left(h_i(j)\right)_{j\in[q]}$$

← pairwise independent

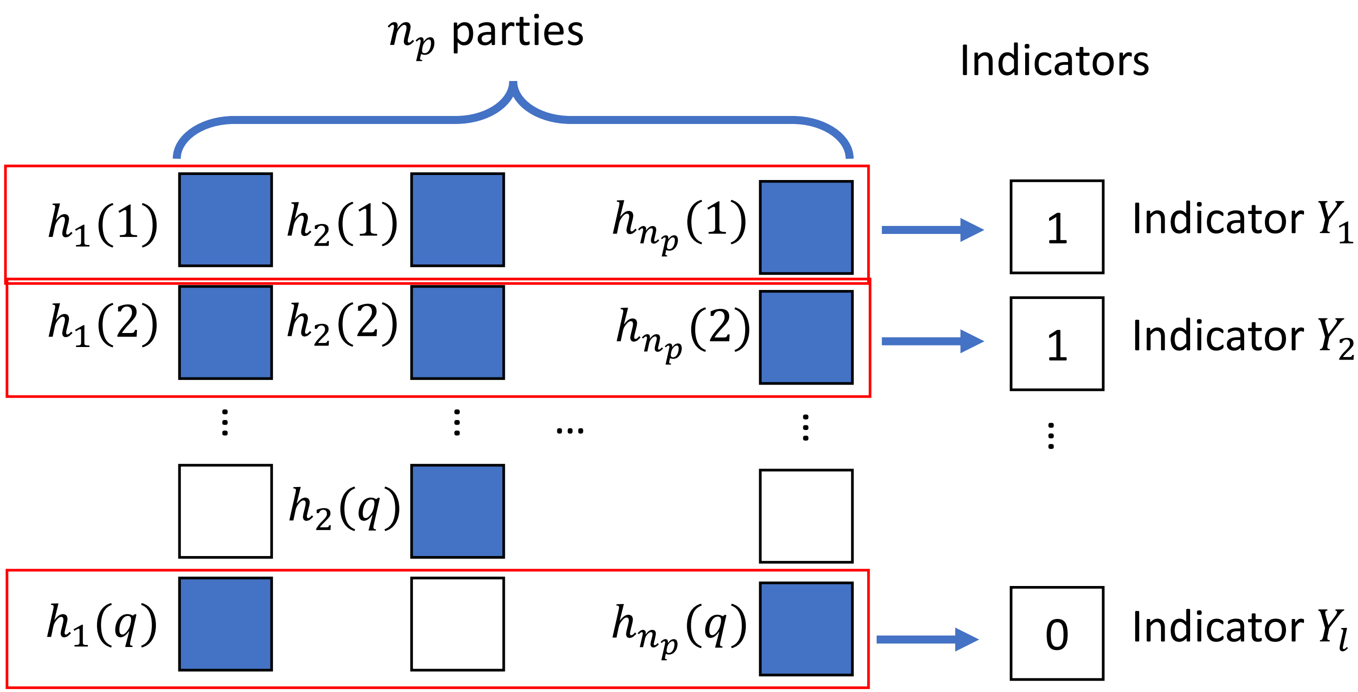
The security is guaranteed by the pairwise independence of the indices from the intersection

Correctness in the multi-party setting

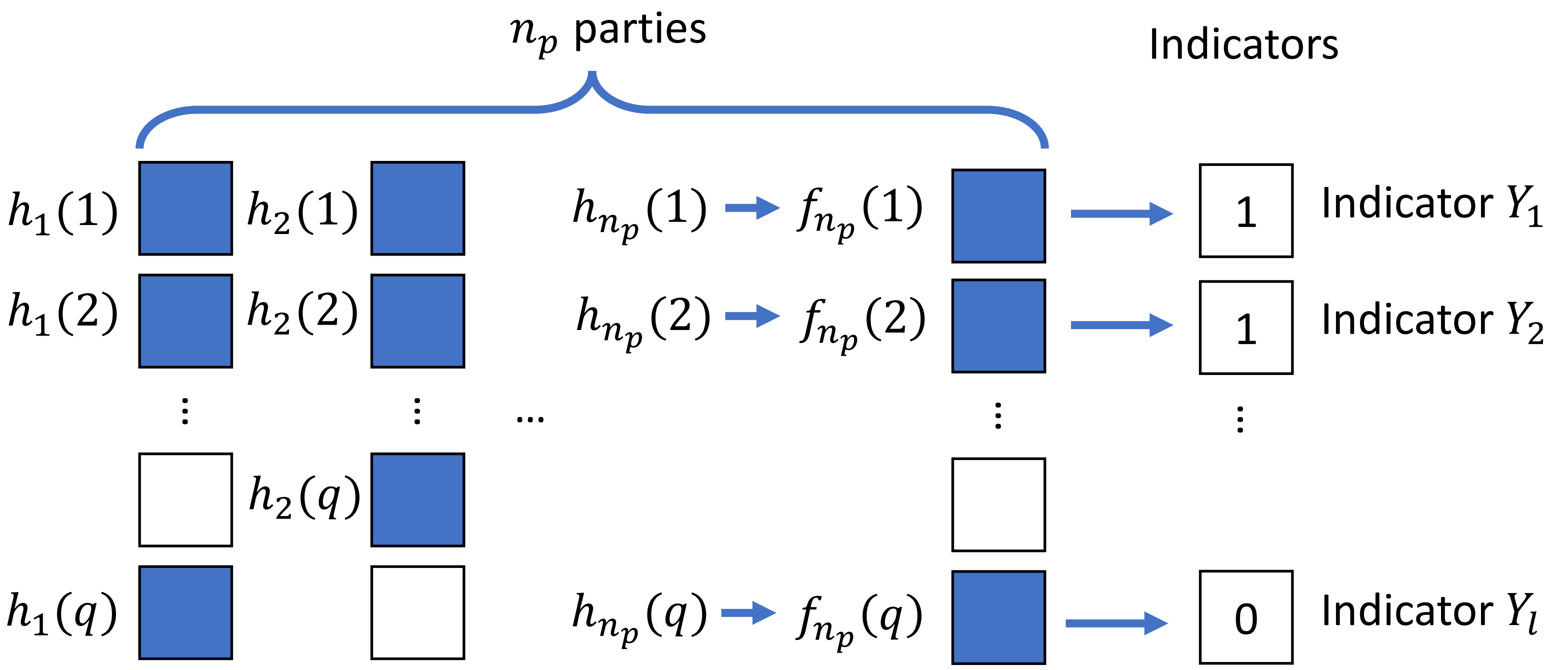


The correctness is guaranteed by that the size of the intersection is sufficiently large

Correctness in the multi-party setting

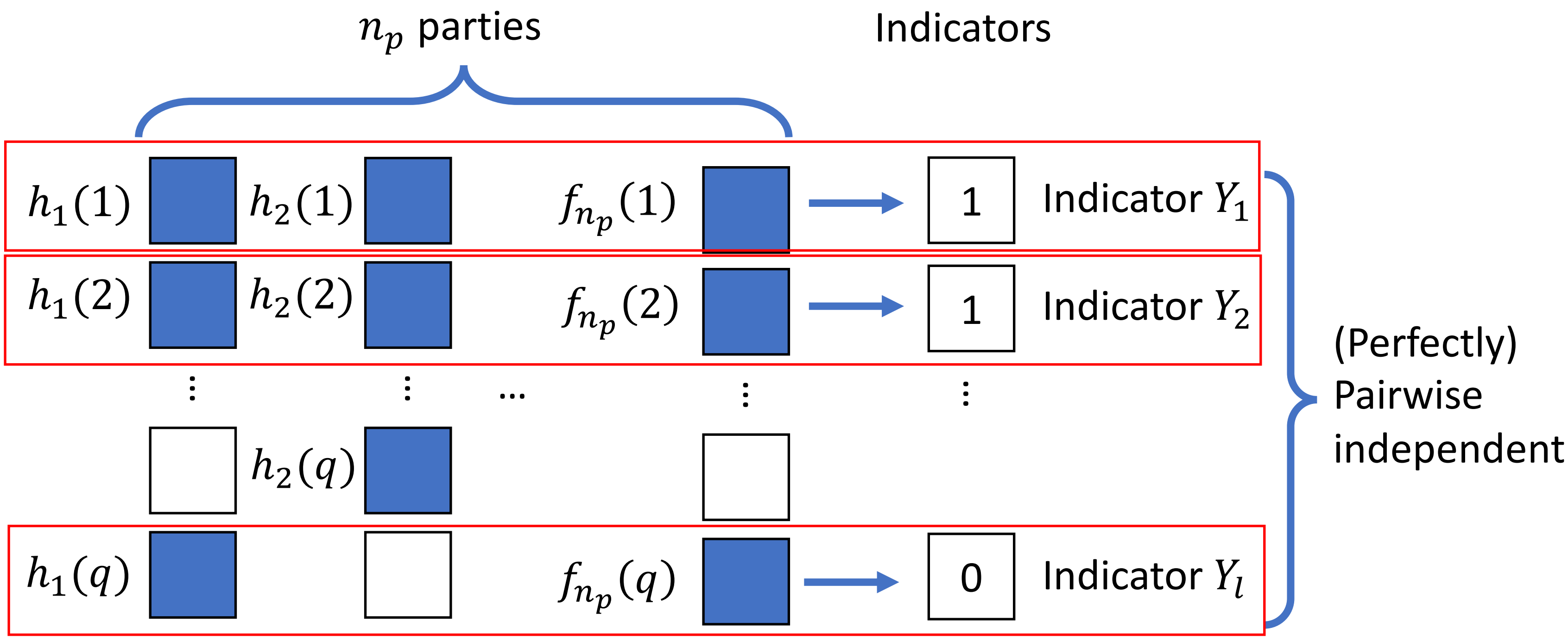


Correctness in the multi-party setting



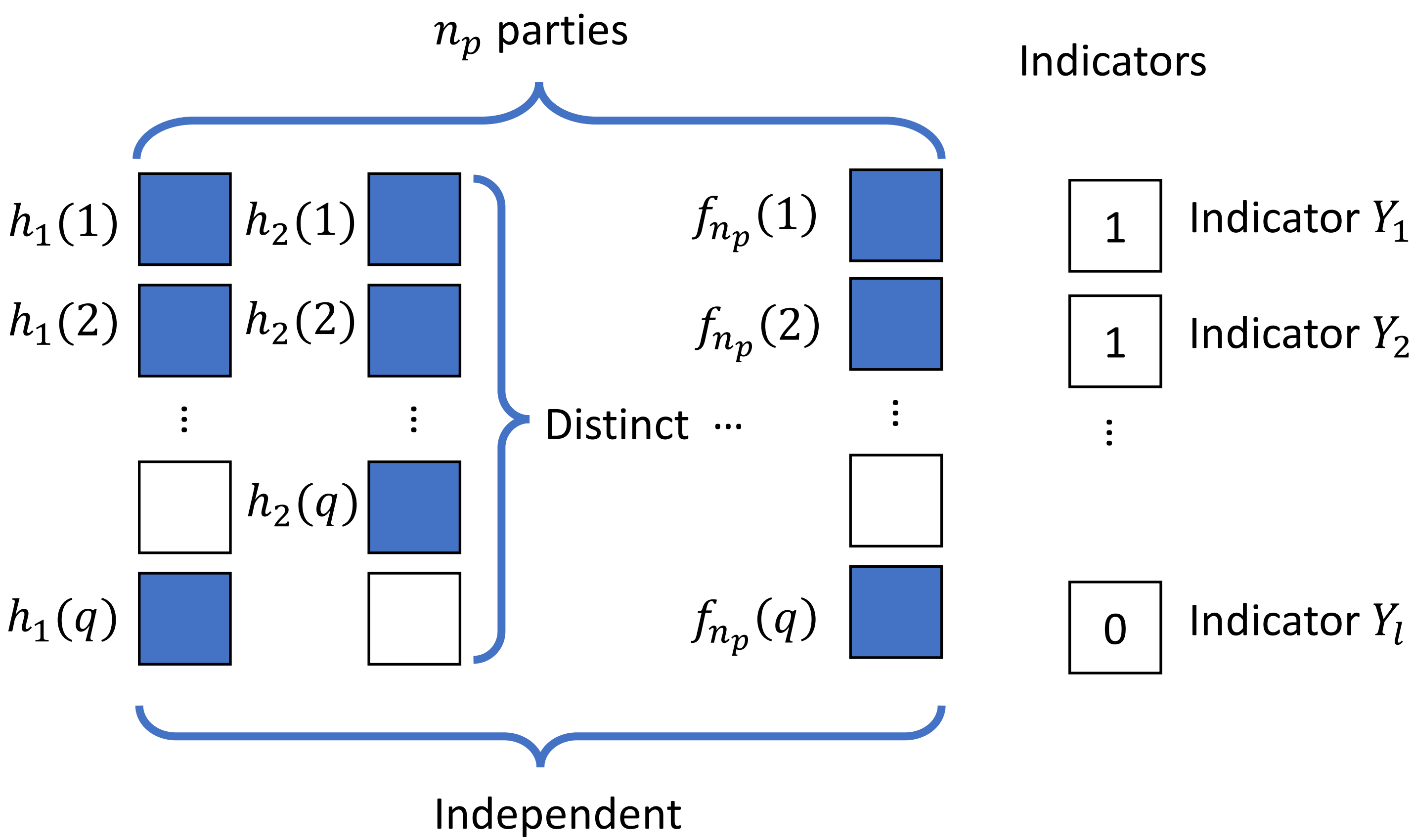
Firstly, we change the approximately pairwise indices into the perfectly pairwise indices

Correctness in the multi-party setting



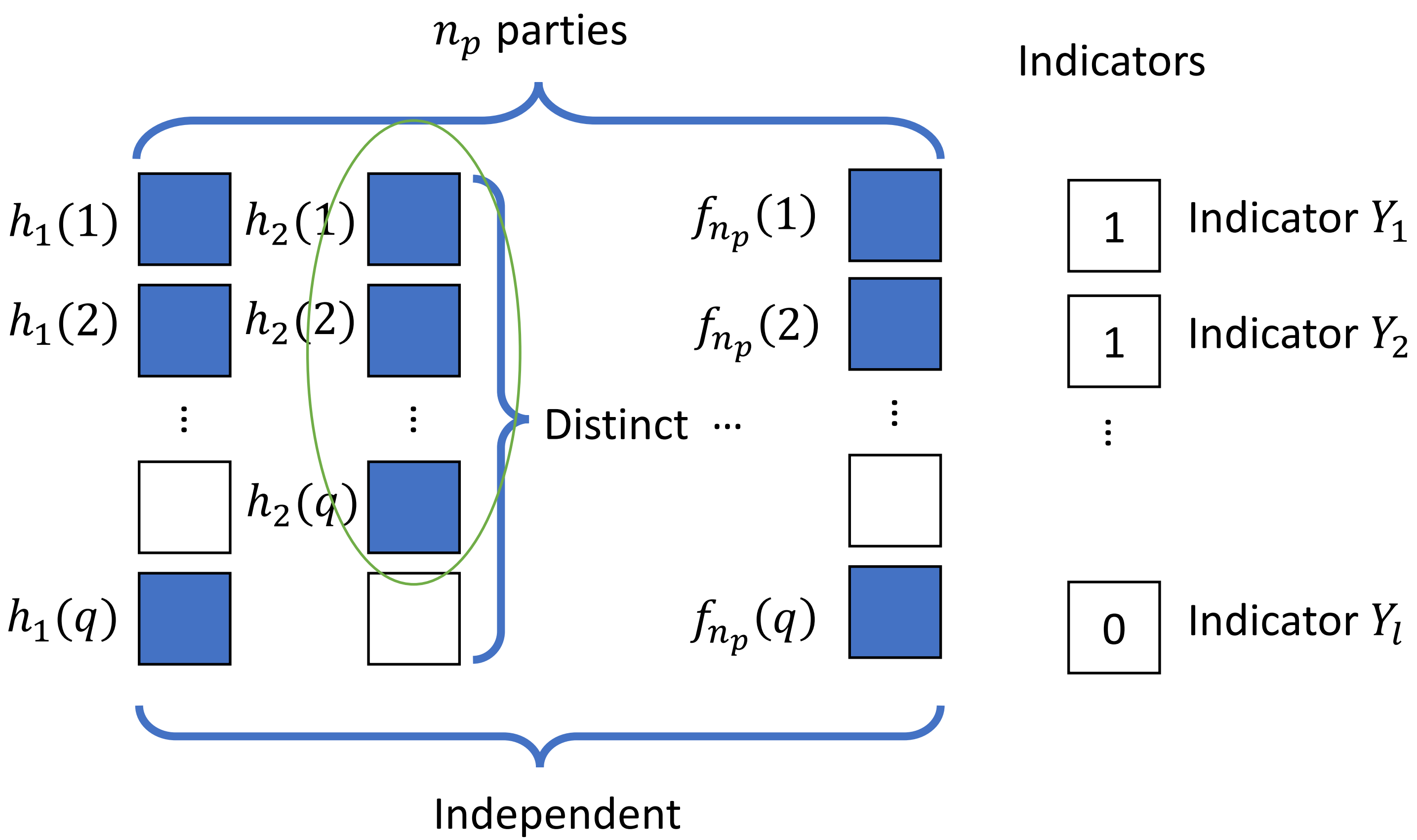
This ensures the pairwise independence of the indicators

Correctness in the multi-party setting



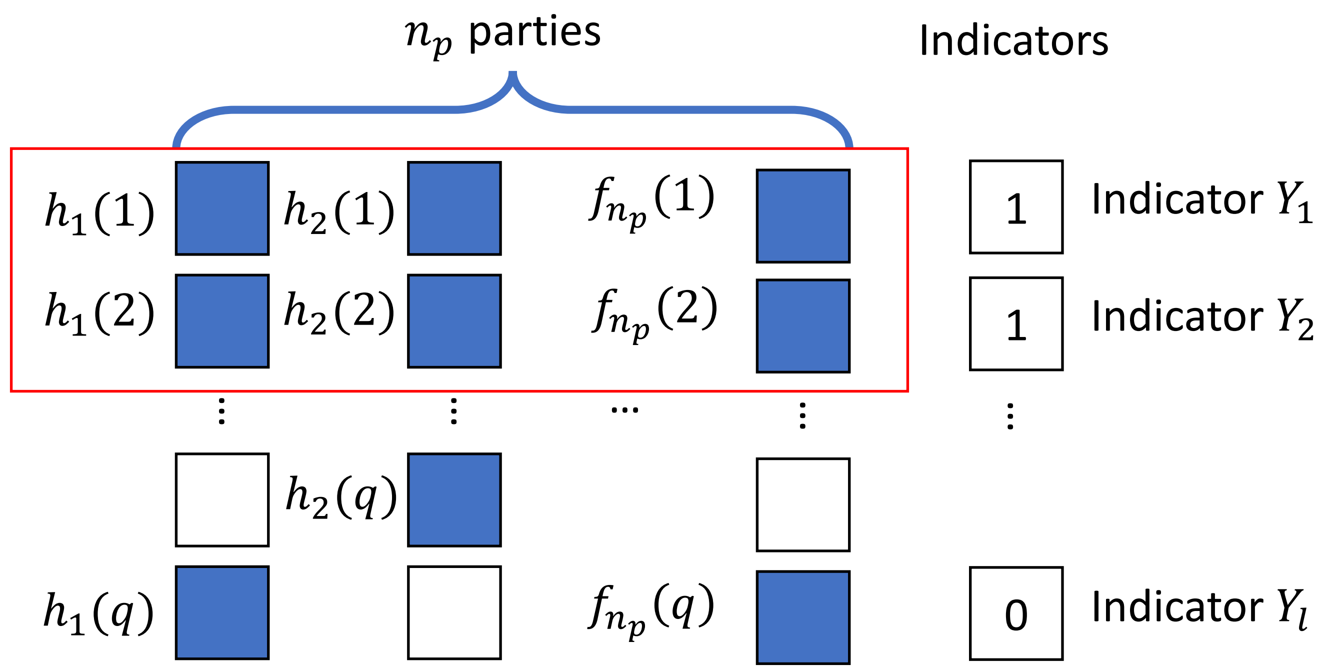
Other indices are approximately pairwise independent

Correctness in the multi-party setting



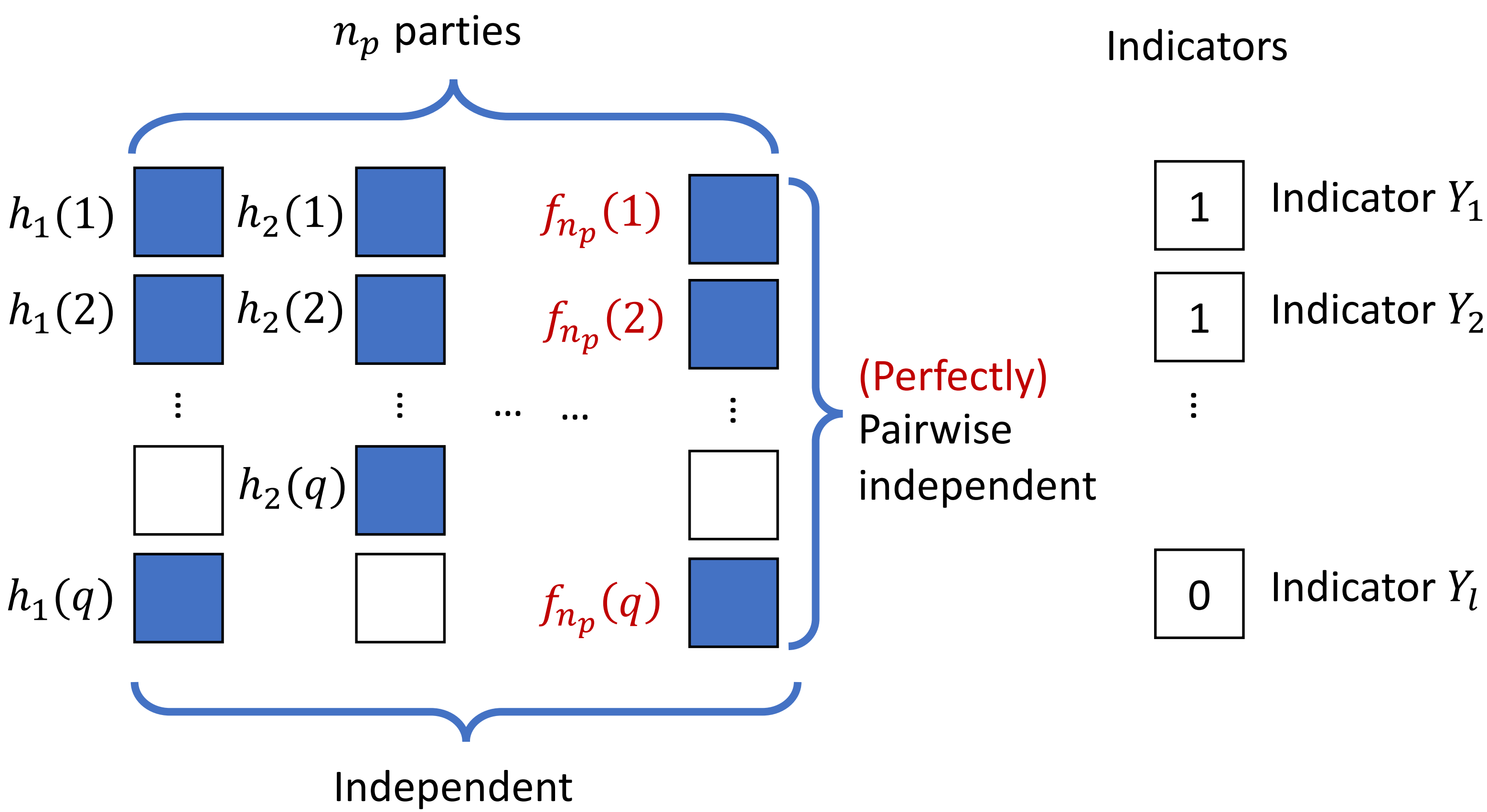
This ensures each indicator equals 1 with high probability

Correctness in the multi-party setting



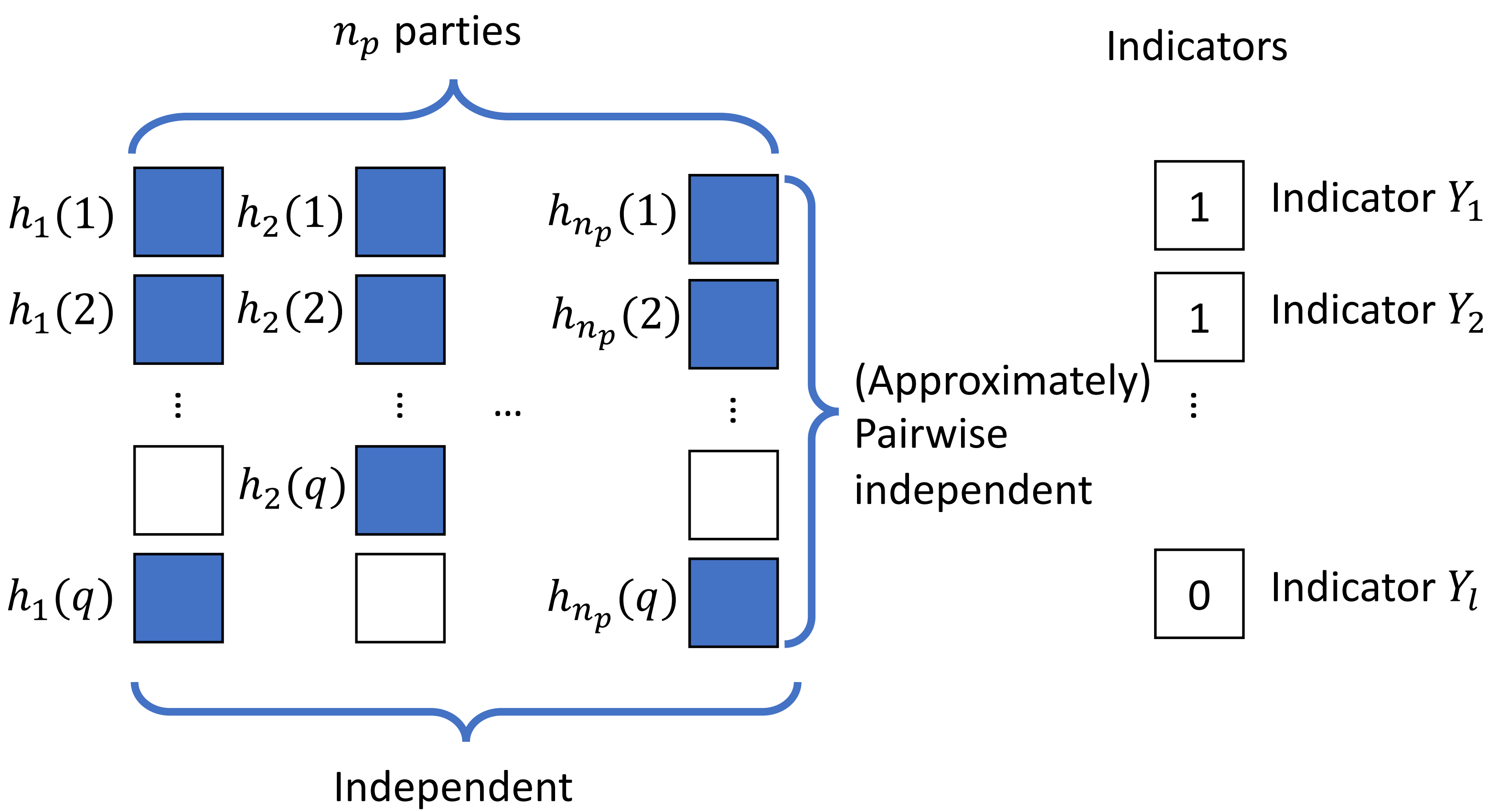
Then we can apply Chebyshev's Inequality and Markov's Inequality to show that the size of intersection is sufficient large with high probability

Correctness in the multi-party setting



The result holds when the indices are perfectly pairwise independent

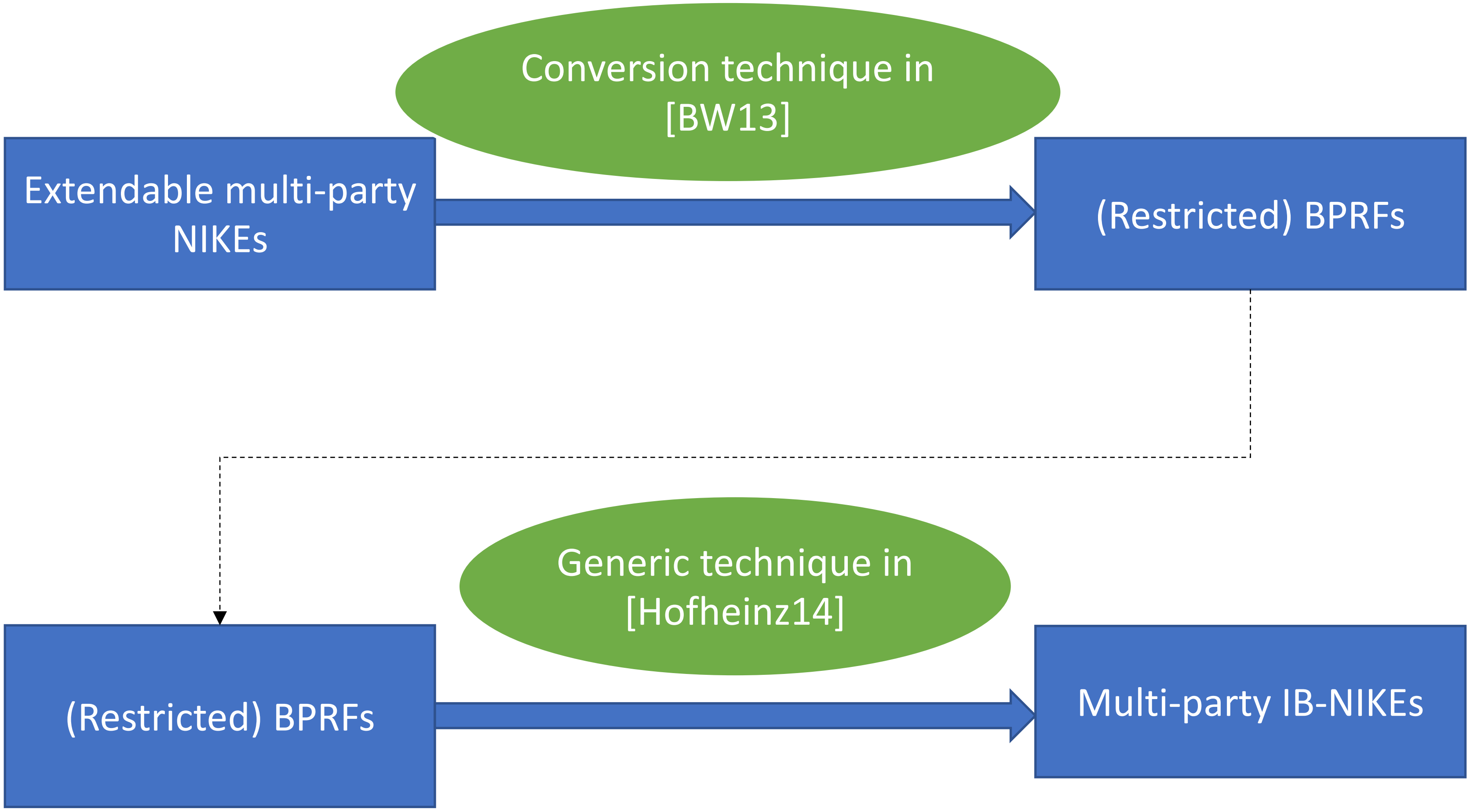
Correctness in the multi-party setting



This only makes the sum smaller and hence does not affect our result.

Extension to IB-NIKE

❖ Multi-party IB-NIKE from multi-party NIKE with extendability



Thank you!