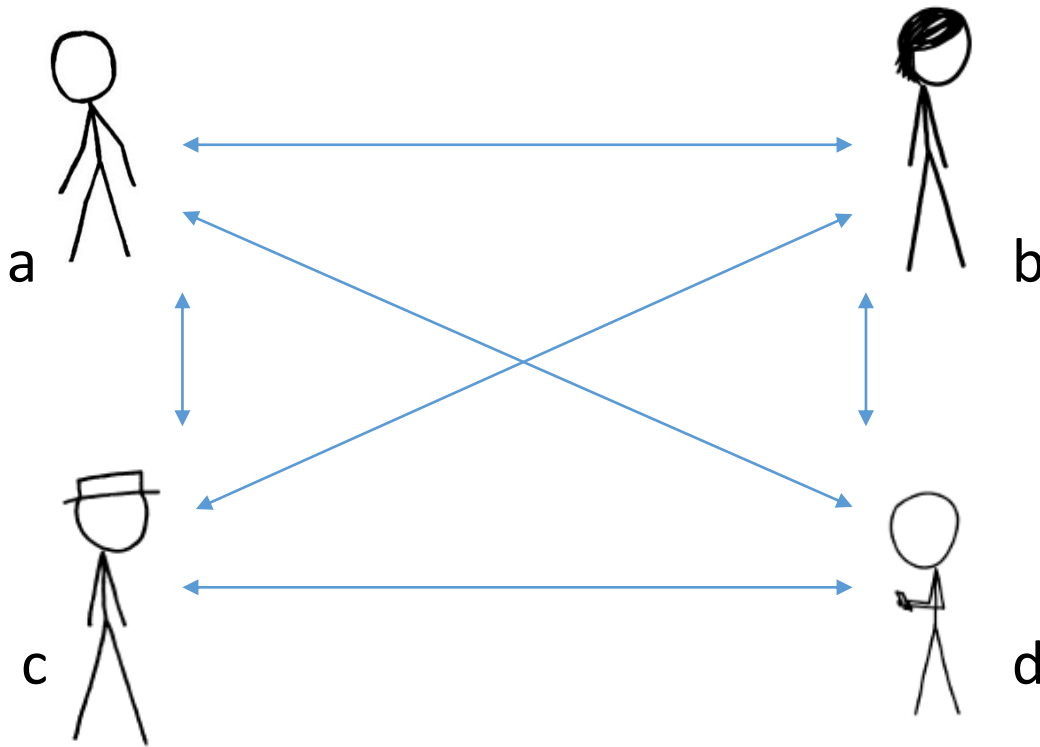


TinyKeys: A New Approach to Efficient Multi-Party Computation

Carmit Hazay, Emmanuela Orsini, Peter Scholl and
Eduardo Soria-Vazquez

Based on slides prepared by Peter Scholl and Eduardo Soria-Vazquez

Secure Multi-Party Computation (MPC)

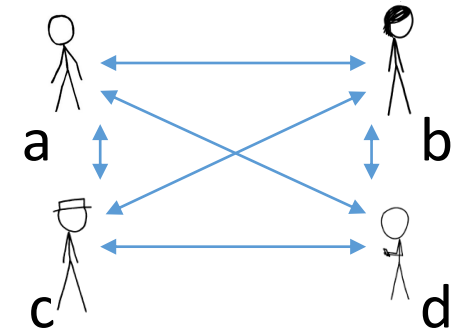


Goal: Compute $f(a,b,c,d)$

Secure computation has many applications

- Auctions with private bids
- Privacy-preserving data mining
- Private health records
- Cryptographic key protection
- Secure statistical analyses
- Smart city research – gender inequity
- ...

MPC - Past and Present



Feasibility results:

Back to the 80's [Yao86,GMW87,BGW88,CCD88,Kilian88,RB89,BMR90]

Broad focus on improving efficiency in past decade:

Two-party setting

[LP07,KS08,NO09,IKOPS11,NNOB12,HKK+14,ZRE15,RR16,GLNP15,WMK17,WRK17,HIV17,KRRW18],

Multi-party setting

[IPS08-09,DPSZ12,DKL+13,LPSY15,WRK17b,HSS17,KPR18,CGHIKLN18]

Properties of MPC Protocols

Computational model: Boolean/arithmetic circuits, RAM

Adversary model:

Passive (semi-honest) or **active** (malicious)

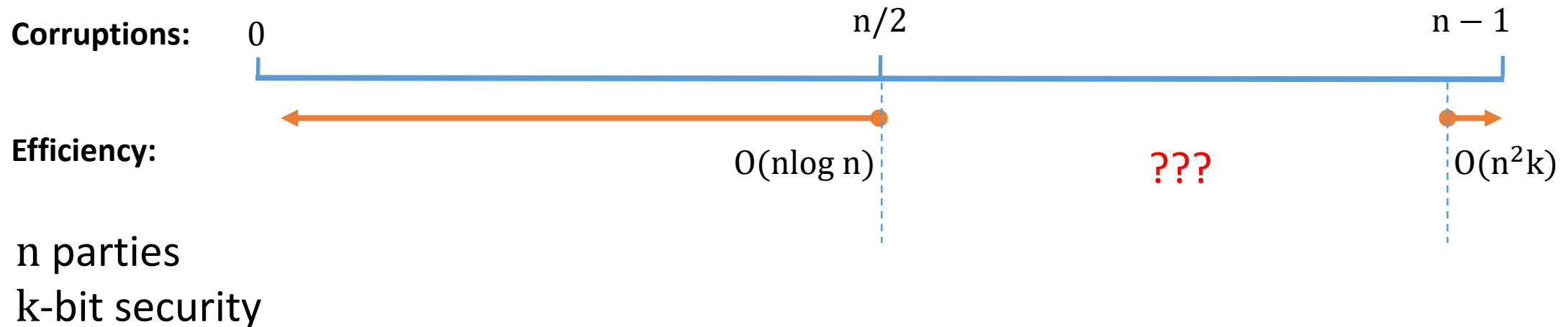
Threshold t (number of corrupted parties)

Efficiency:

Computation/ communication complexity

Round complexity

Corruption Thresholds vs Communication Complexity of Practical MPC



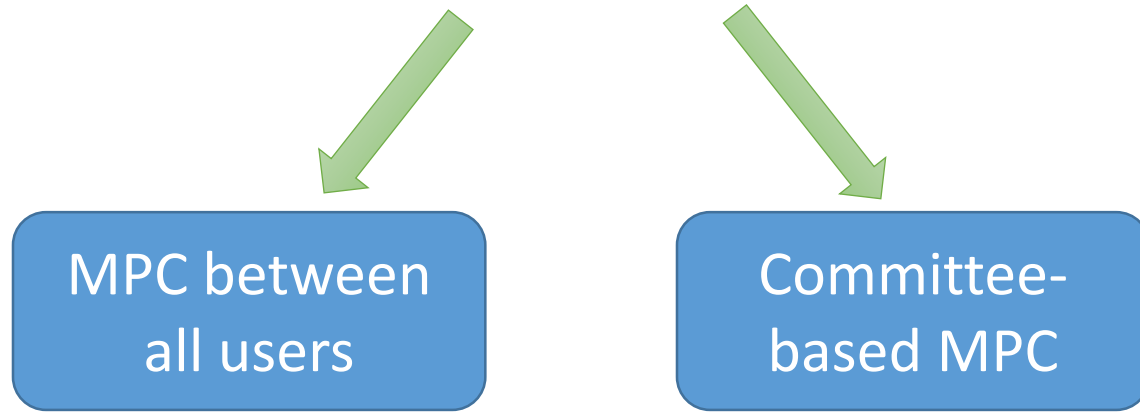
Can we design concretely efficient MPC protocols where each honest party can be leveraged to increase efficiency?

Main Question

*Can we trade off the **number of corrupt parties** for a more efficient, **practical protocol**?*

Motivation: Large Scale, Dishonest Majority

Large number of users want to conduct surveys, auctions, statistical analysis, measure network activity, etc.



Dishonest Majority:
More parties \Rightarrow More trustworthy

The screenshot shows the TorMetrics website interface. At the top, there is a navigation bar with links for News, Sources, Operation, Development, and Research. Below this is a header with the TorMetrics logo and a quote: "Tor metrics are the ammunition that advocates argue for a more private position of data, rather than just...". A secondary navigation bar includes Home, Users, Servers, Traffic, Performance, Onion Services, and Applications. The main content area features a "Welcome!" message followed by the question "What would you like to know about the Tor network?". Below this are six interactive cards: "Users" (describing user origins and connections), "Servers" (describing relay and bridge status), "Traffic" (describing network traffic volume), "Performance" (describing network speed and reliability), "Onion Services" (describing the number and traffic of onion services), and "Applications" (describing the number and updates of Tor applications). A footer note says "Let us know if we're missing anything, or if we should measure something else."

MPC Setting in This Talk

Main focus:

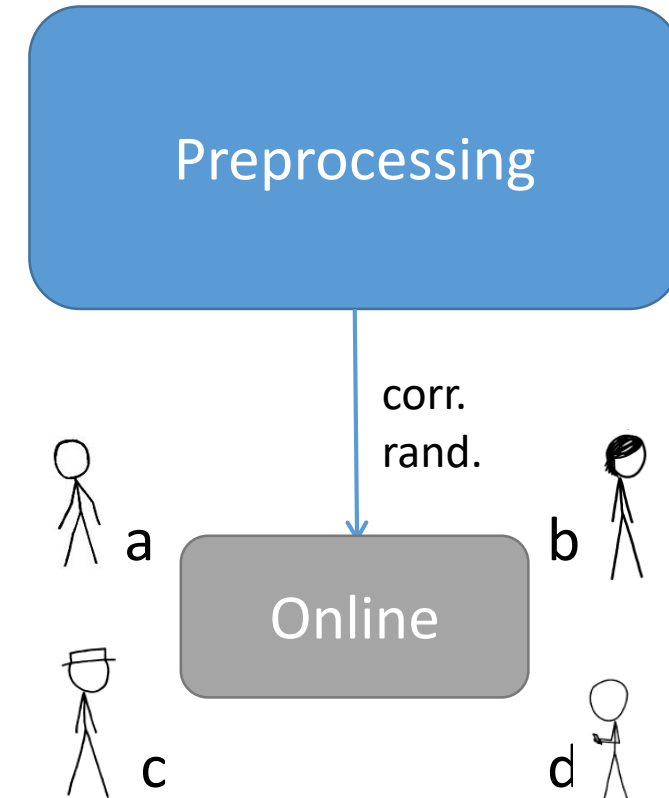
- Concrete efficiency for large numbers of parties (e.g. n in 10s, 100s)

Adversary:

- Static, passive
- Dishonest majority ($t > n/2$)

Model of Computation:

- Boolean circuits
- Preprocessing phase



Our Results

New dishonest majority protocols exploiting **more honest parties**:

1. Passive GMW-style MPC based on OT

Up to **25x less communication** compared with $n - 1$ corruptions

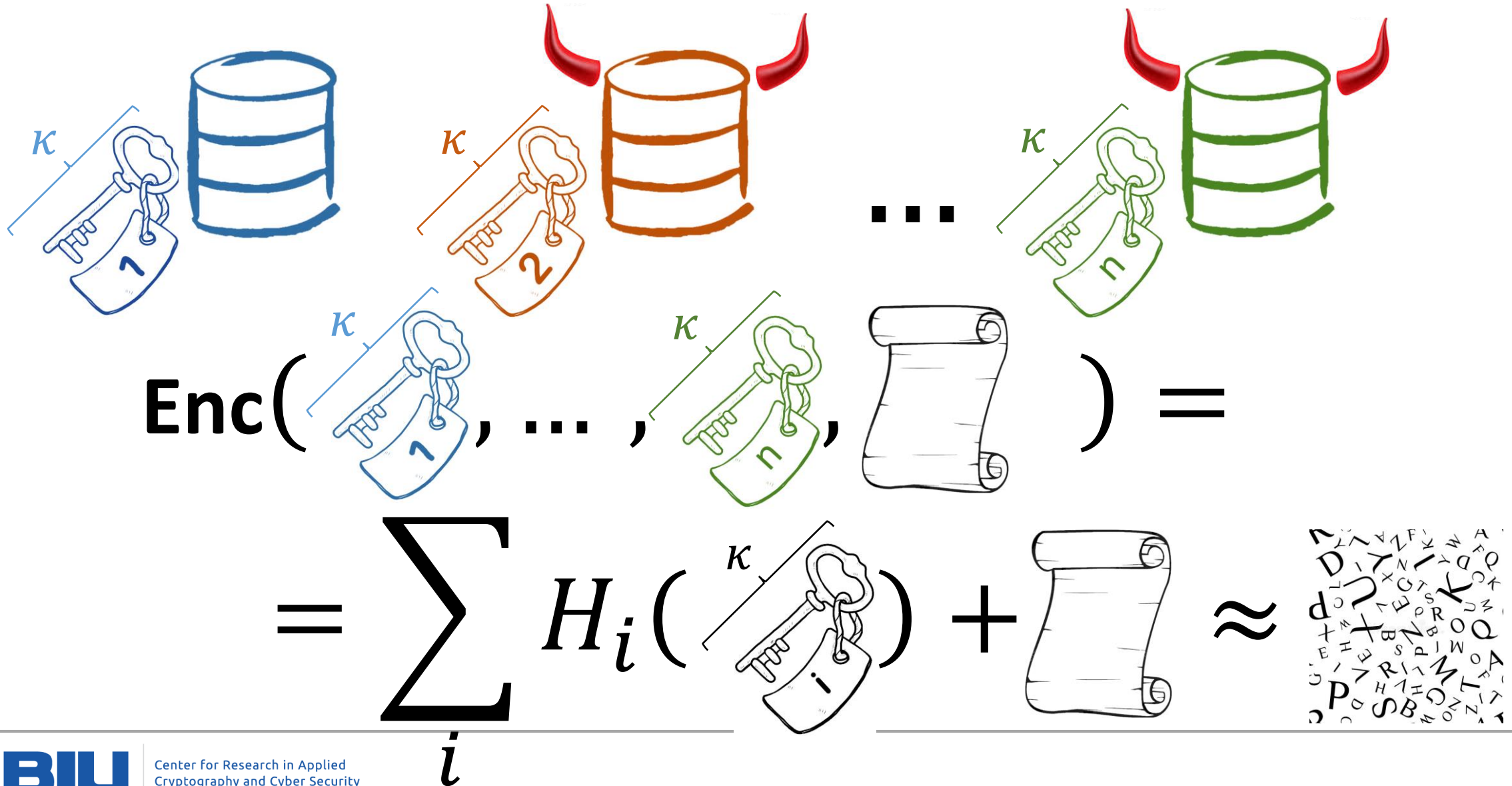
2. Passive constant-round BMR-style MPC based on garbled circuits

Up to **7x reduction** in GC size and communication cost

Best improvements with **20+ parties** when **70-90%** are corrupt

The TinyKeys Technique

Warm-up: Distributed Encryption



Distributed Encryption: Can We Do Better?



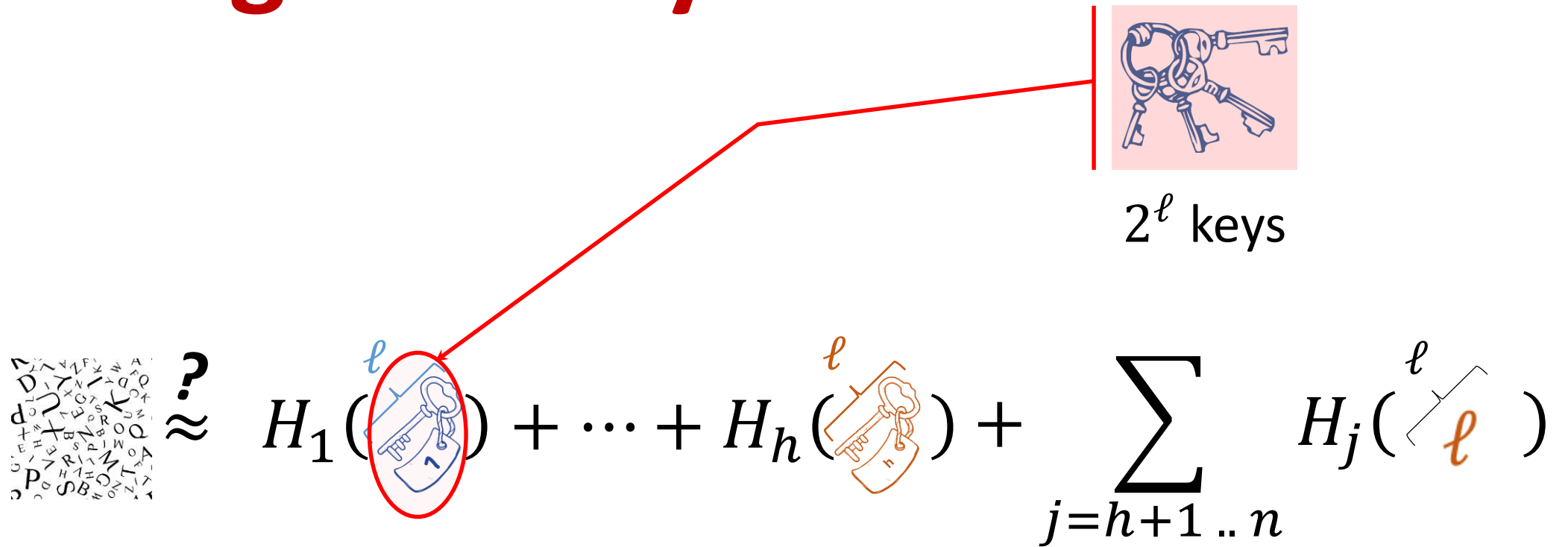
Distributed Encryption with TinyKeys




$$\text{Enc} \left(\begin{array}{c} \ell \\ \text{key } 1 \end{array}, \dots, \begin{array}{c} \ell \\ \text{key } n \end{array}, \text{document} \right) =$$

$$= \sum_i H_i \left(\begin{array}{c} \ell \\ \text{key } i \end{array} \right) + \text{document} \approx \text{scrambled document}$$

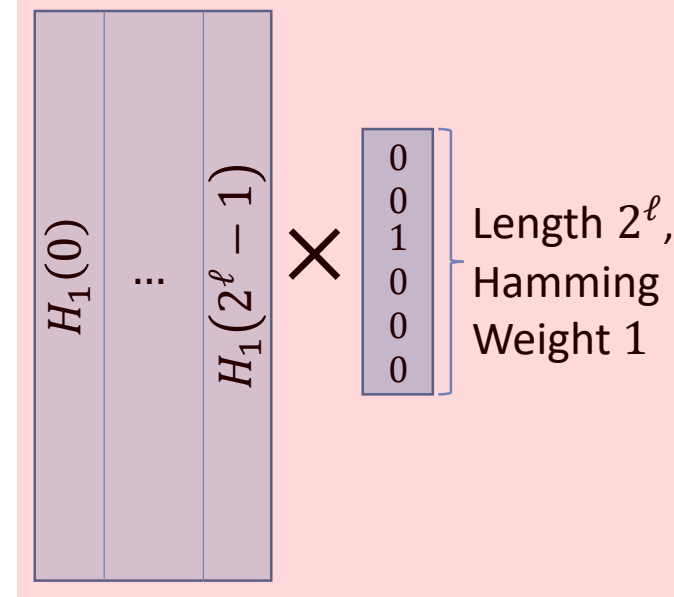
Breaking Security



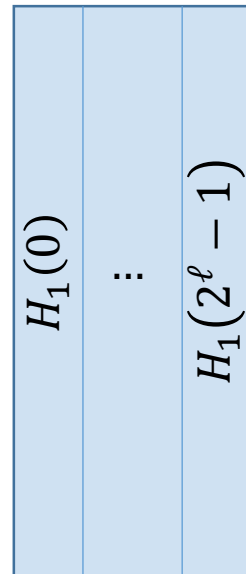
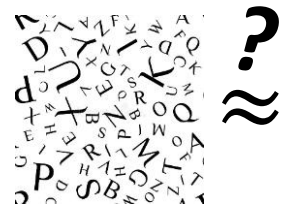
Breaking Security

 $\approx ?$

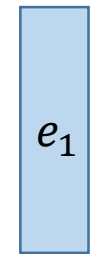
$$H_1(\overset{\ell}{\text{key}}) + \dots + H_h(\overset{\ell}{\text{key}})$$



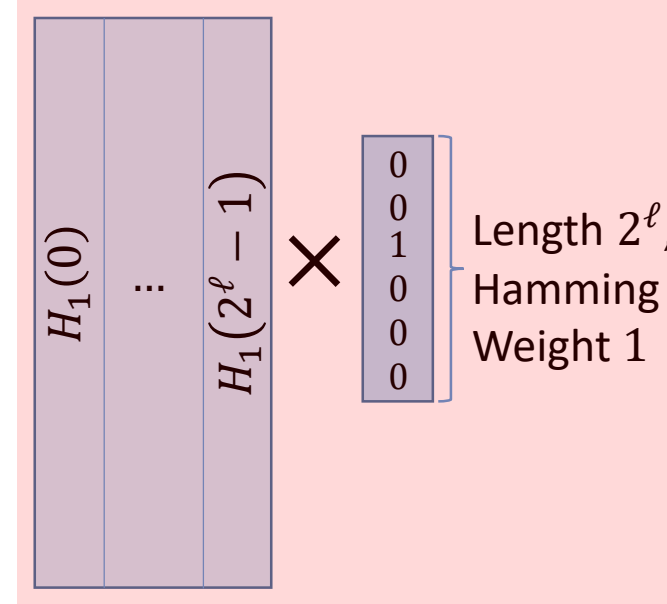
Breaking Security



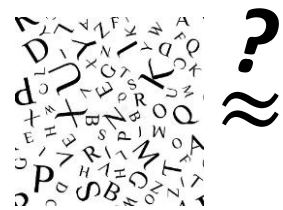
$$+ \dots + H_h \left(\begin{array}{c} \ell \\ \text{key} \end{array} \right)$$



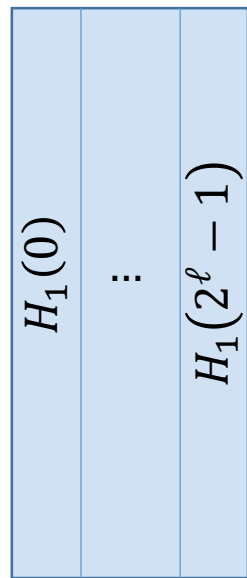
Length 2^ℓ ,
Hamming
Weight 1



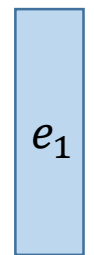
Breaking Security



~?

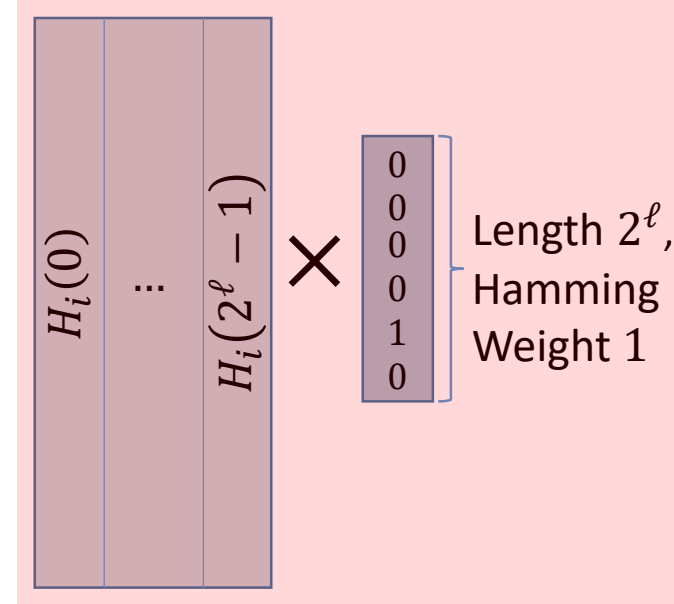


$$+ \dots + H_h \left(\begin{array}{c} \ell \\ \text{key} \end{array} \right)$$

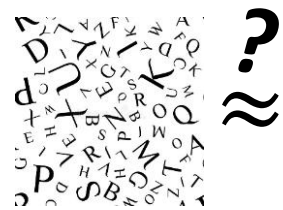


Length 2^ℓ ,
Hamming
Weight 1

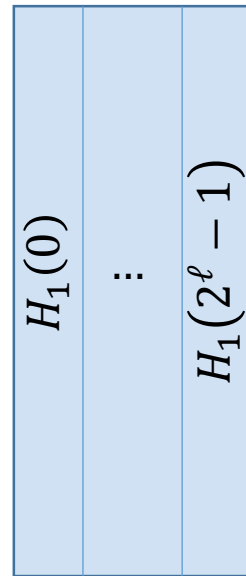
⋮



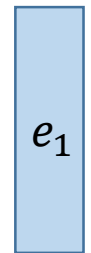
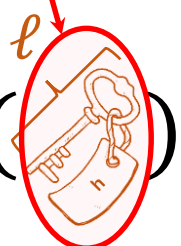
Breaking Security



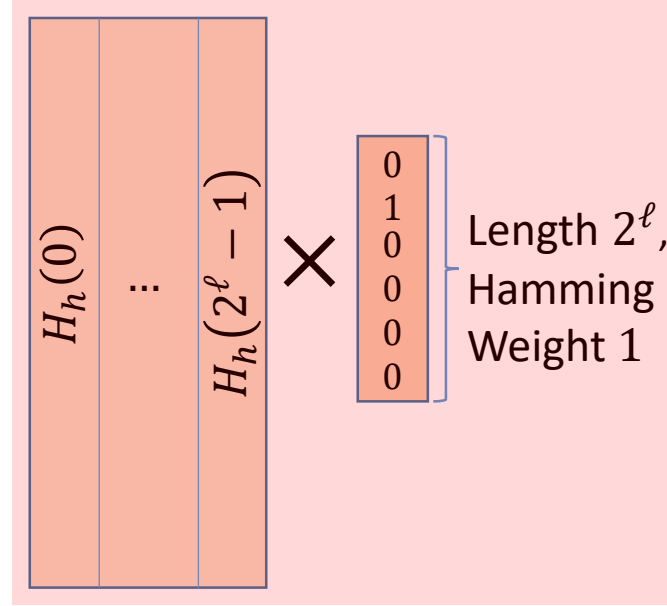
?



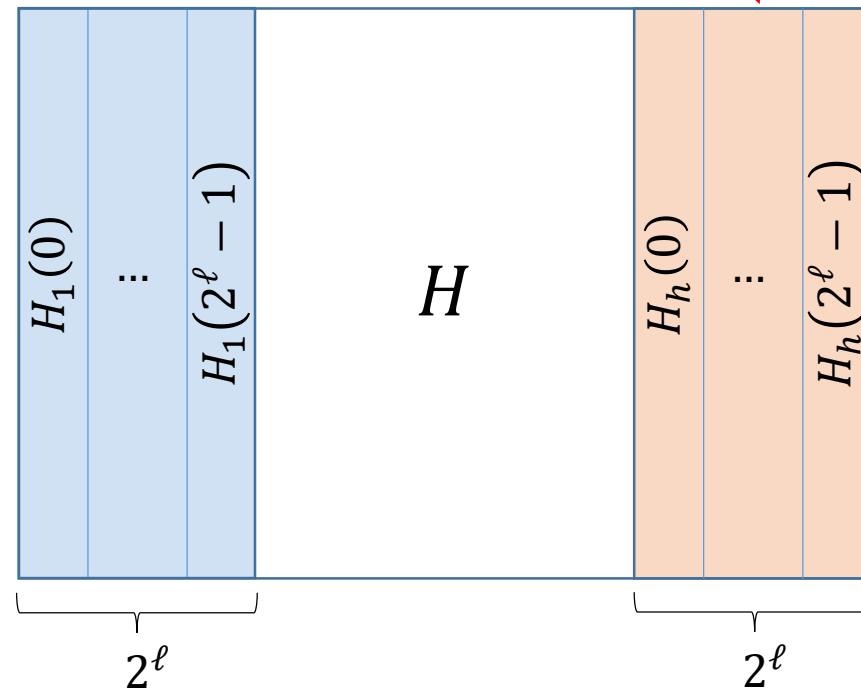
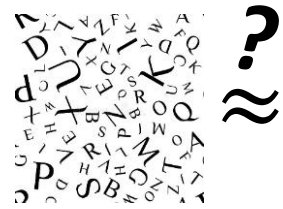
$$+ \dots + H_h(\ell)$$



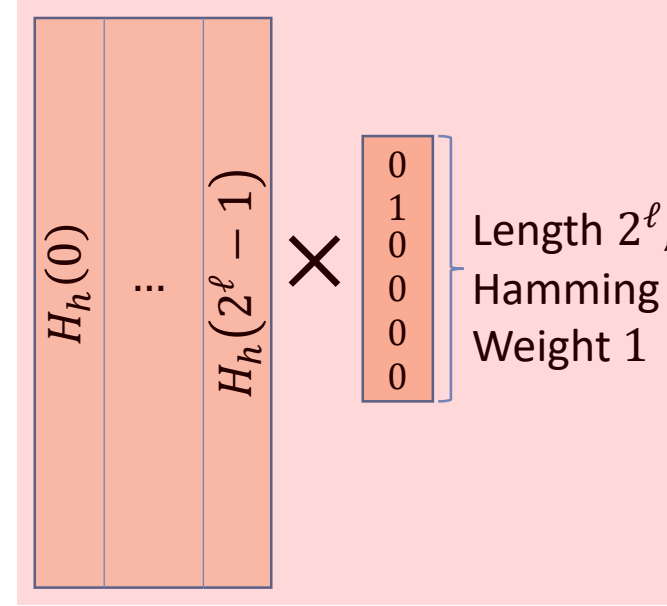
Length 2^ℓ ,
Hamming
Weight 1



Breaking Security

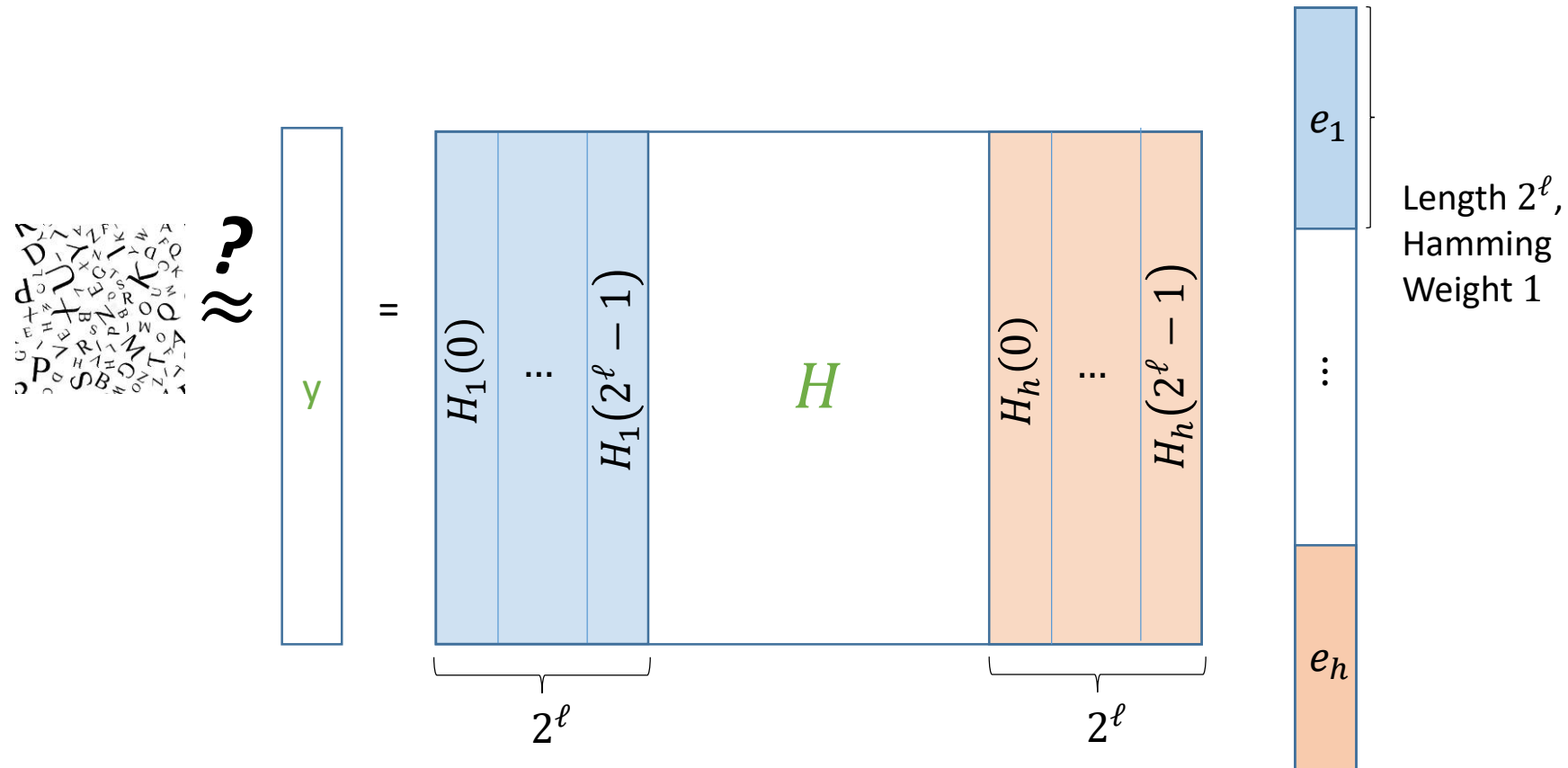


Length 2^ℓ ,
Hamming
Weight 1



Breaking Security

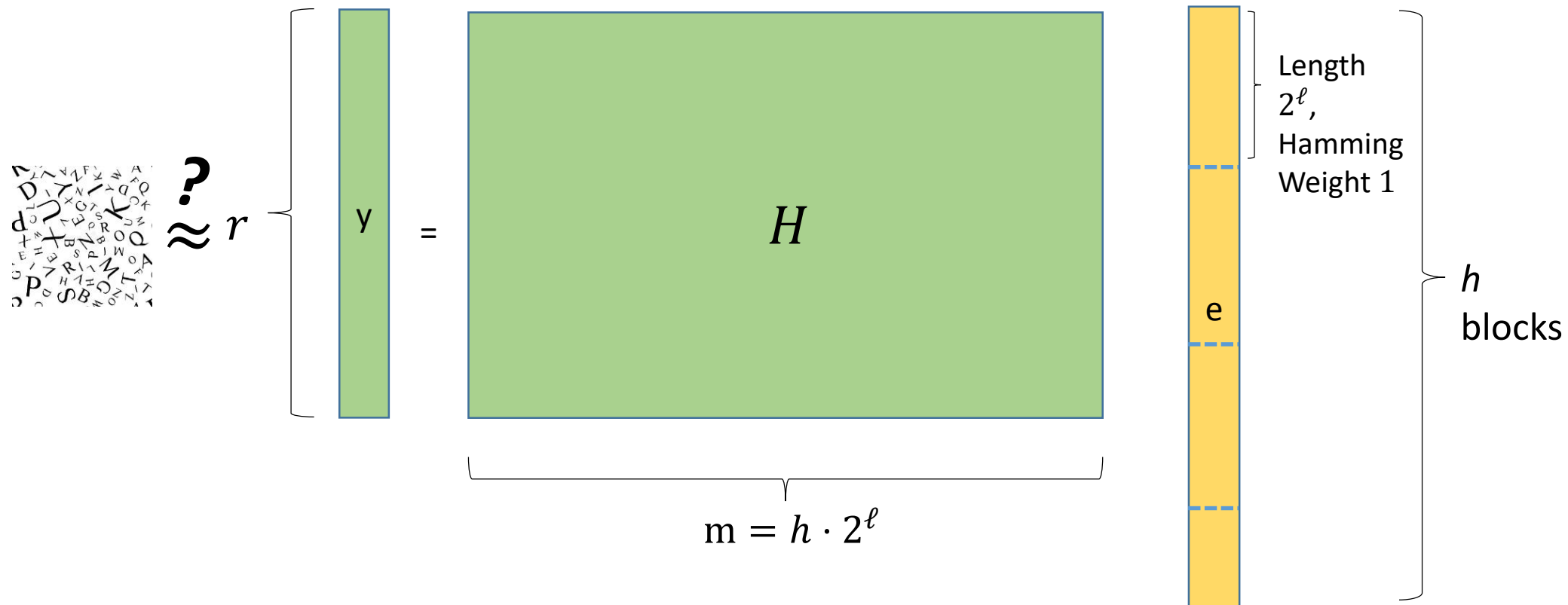
Adv wins: Given H and $y = He$, distinguish y from random



Breaking Security: Regular Syndrome Decoding

Sample random $H \in \{0,1\}^{r \times m}$, and regular $e \in \{0,1\}^m$ of weight h

Adv wins: Given H and $y = He$, find $e \iff$ distinguish y from random



Hardness of Regular Syndrome Decoding

- Used for SHA-3 candidate FSB [Augot Finiasz Sendrier 03]
 - Not much easier than syndrome decoding \Leftrightarrow LPN
- Params: Message length r , key length ℓ , #honest h
- **Statistically hard** for small r /large h

[FS09]

[Saa07]

[MO15]

[NCB11]

[Kir11]

[BM17]

[BJMM12]

[BLN+09]

[CJ04]

[BLP08]

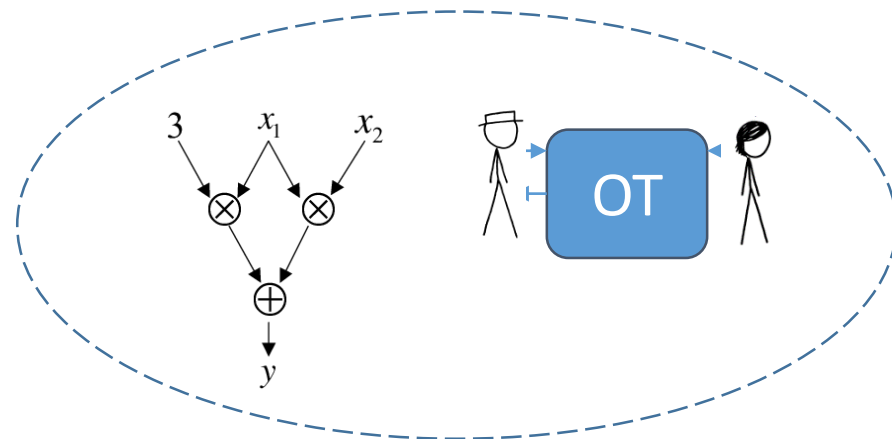
[MS09]

[MMT11]

[BLP11]

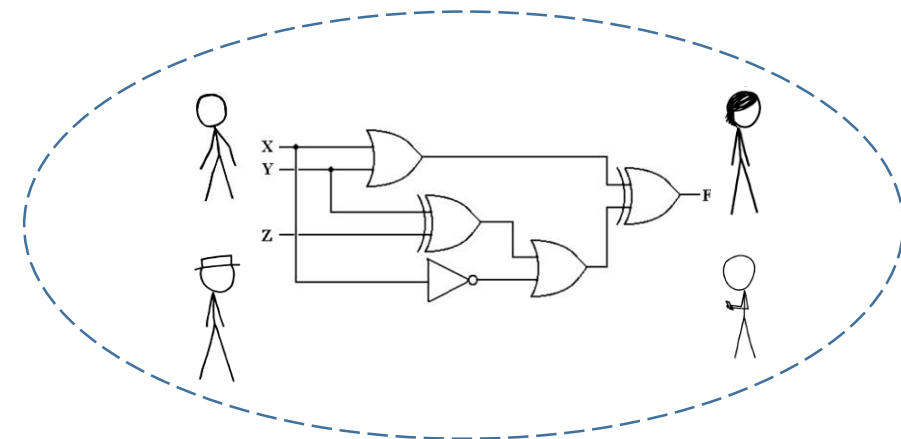
TinyKeys: A Little Honesty Goes a Long Way

(Tiny)GMW



- Key length: $\ell \geq 1$

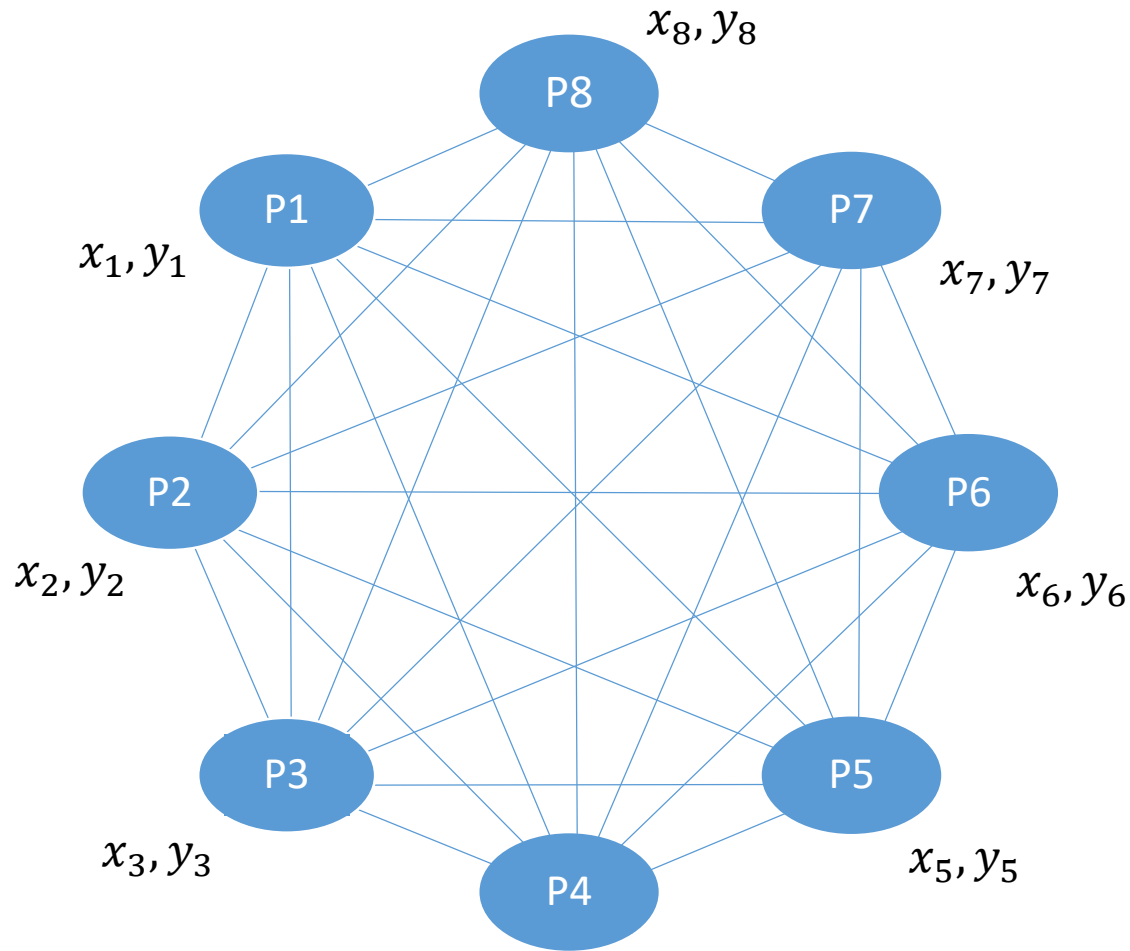
(Tiny)BMR



- Key length: $\ell \geq 5$
- Many challenges:
 - High Fan-Out
 - Enabling FreeXOR

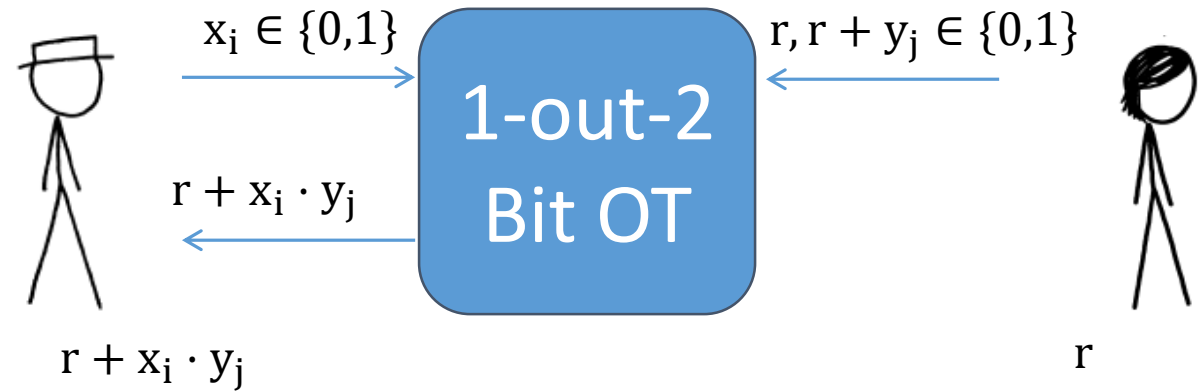
(Tiny)GMW

Quick Recap of GMW

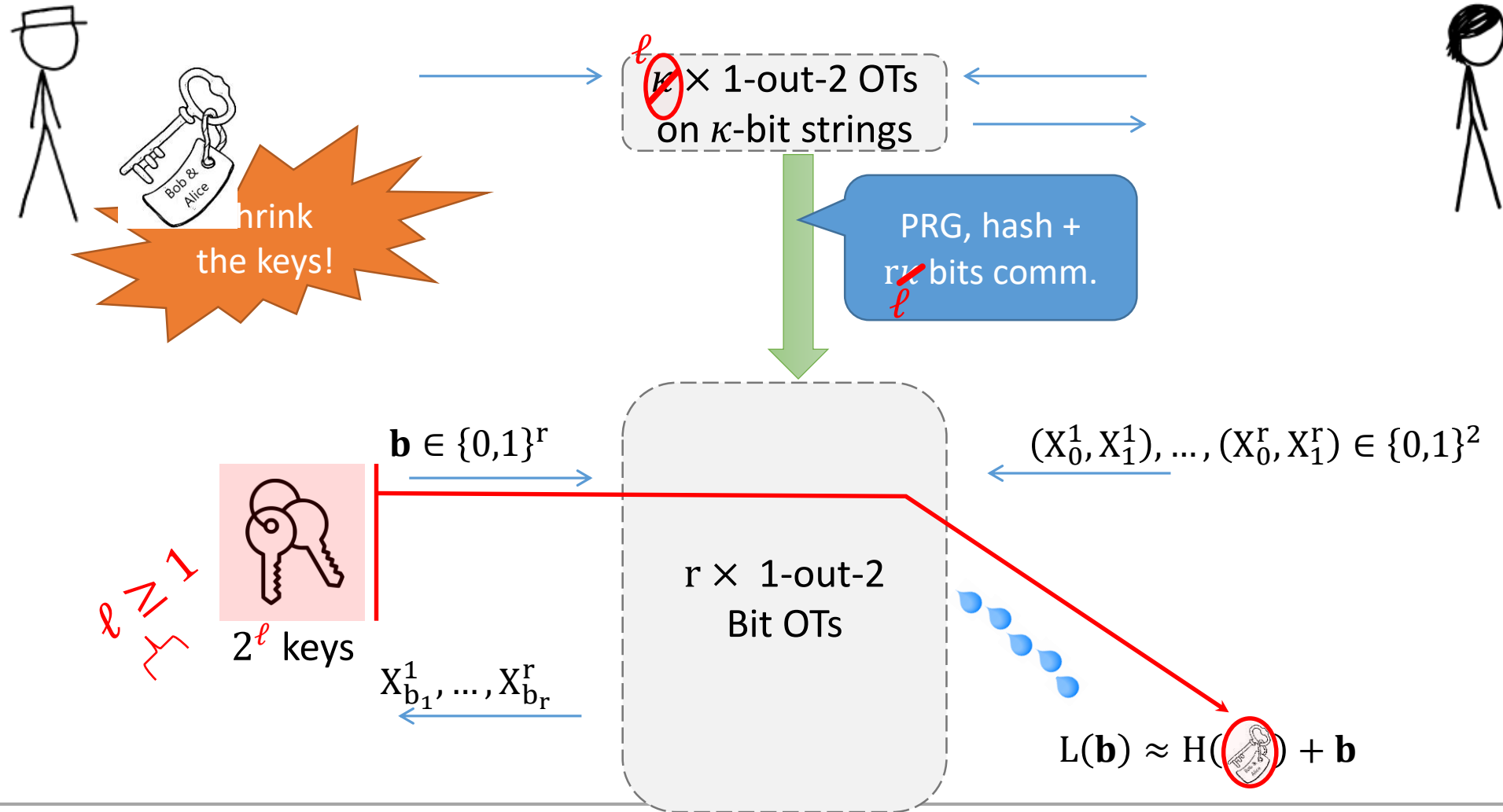


$$\begin{array}{r}
 x = x_1 + \dots + x_n \in \{0,1\} \\
 + y = y_1 + \dots + y_n \in \{0,1\} \\
 \hline
 x + y = (x_1 + y_1) + \dots + (x_n + y_n)
 \end{array}$$

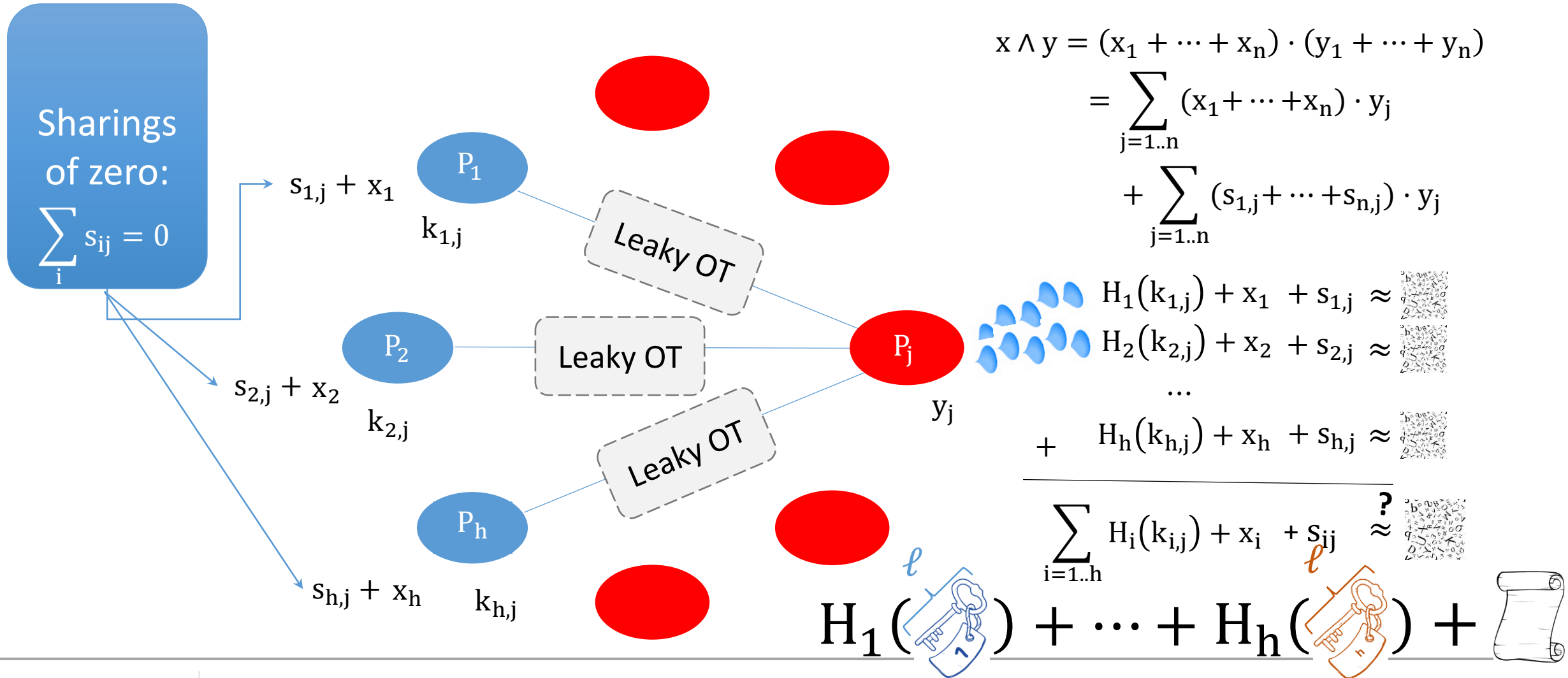
$$x \wedge y = (x_1 + \dots + x_n) \cdot (y_1 + \dots + y_n)$$



“IKNP” OT Extension with Short Keys!



Using leaky OT for GMW-Style MPC



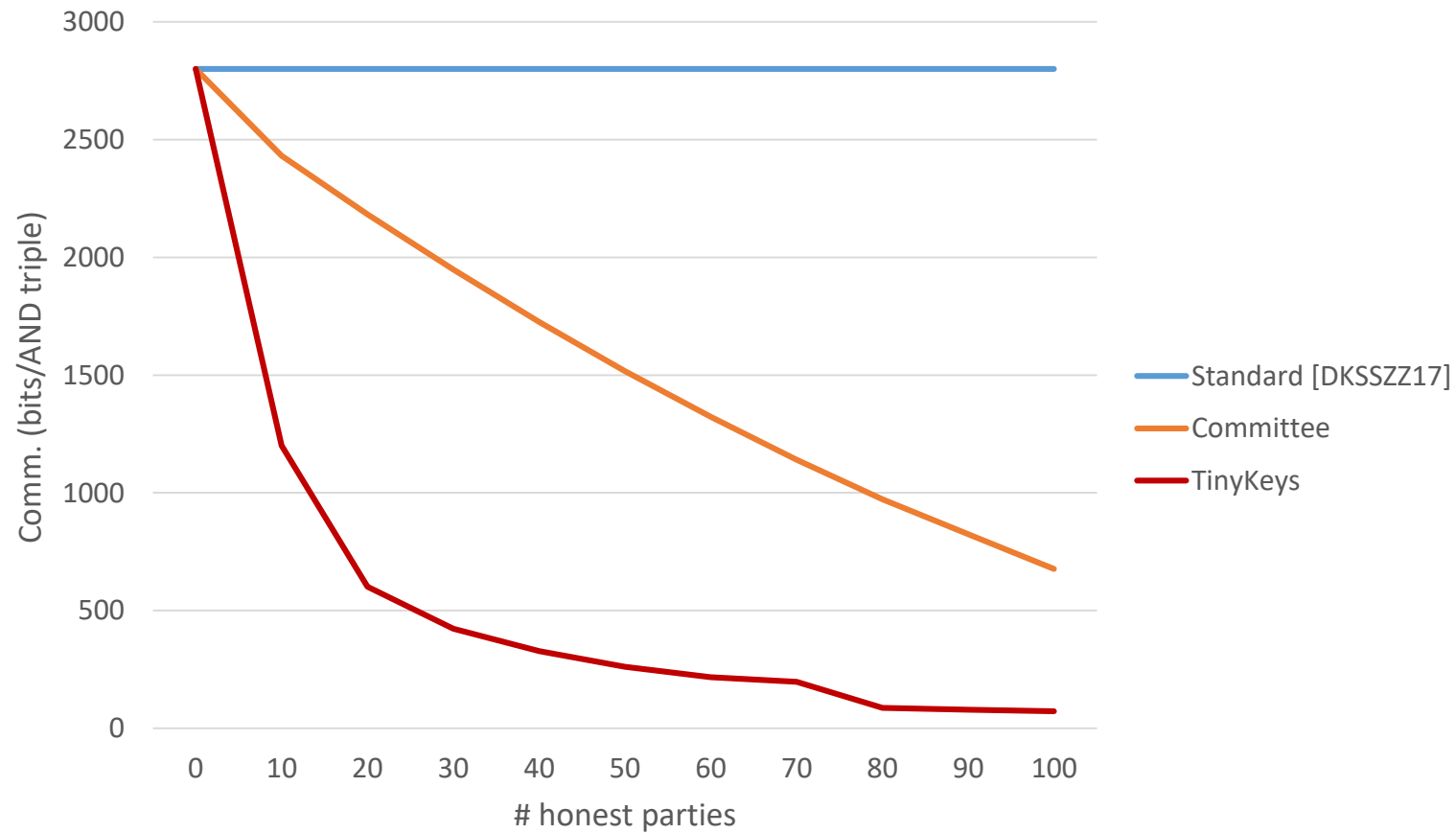
$$\begin{aligned}
 x \wedge y &= (x_1 + \dots + x_n) \cdot (y_1 + \dots + y_n) \\
 &= \sum_{j=1..n} (x_1 + \dots + x_n) \cdot y_j \\
 &\quad + \sum_{j=1..n} (s_{1,j} + \dots + s_{n,j}) \cdot y_j
 \end{aligned}$$

$$\begin{aligned}
 H_1(k_{1,j}) + x_1 + s_{1,j} &\approx \\
 H_2(k_{2,j}) + x_2 + s_{2,j} &\approx \\
 \dots & \\
 + H_h(k_{h,j}) + x_h + s_{h,j} &\approx
 \end{aligned}$$

$$\sum_{i=1..h} H_i(k_{i,j}) + x_i + s_{ij} \approx$$

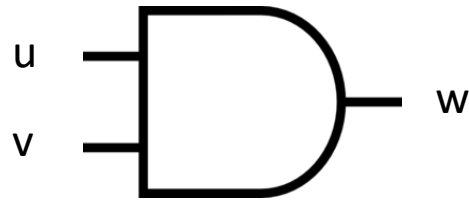
$H_1(\ell) + \dots + H_h(\ell) +$

GMW: Communication Cost of Producing a Single Triple (200 Parties)



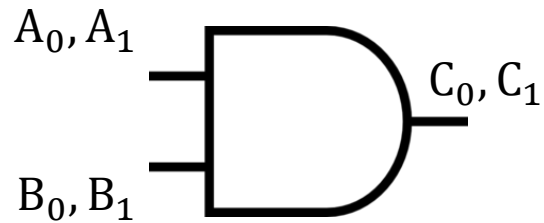
(Tiny) **BMR**

Garbling an AND Gate with Yao



u	v	w
0	0	0
0	1	0
1	0	0
1	1	1

Garbling an AND Gate with Yao



$$E_{A_0, B_0}(C_0)$$

$$E_{A_0, B_1}(C_0)$$

$$E_{A_1, B_0}(C_0)$$

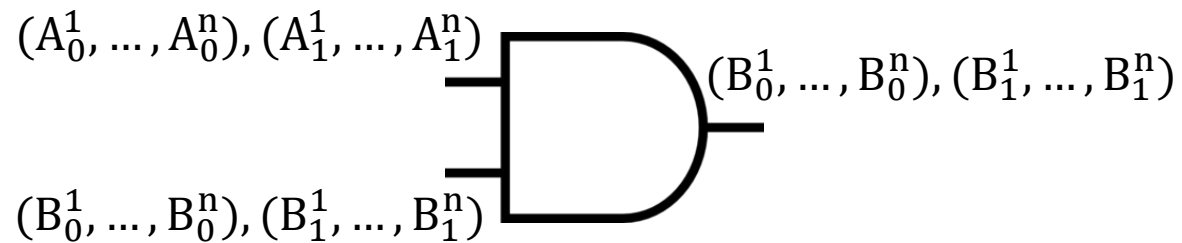
$$E_{A_1, B_1}(C_1)$$

- Pick two random keys for each wire
- Encrypt the truth table of each gate

Randomly **permute** entries

Invariant: evaluator learns **one** key per wire throughout the circuit

Distributed Garbling [BMR90]



Each P_i gets $A_0^i, A_1^i \in \{0,1\}^{\ell}$ etc

Use distributed encryption: $E_{A,B}(C) =$

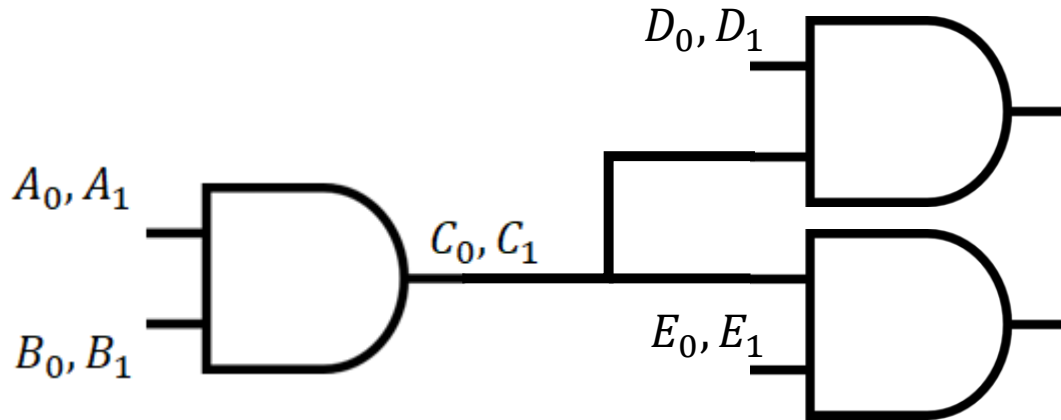
$$\begin{aligned} & H(1 \parallel A^1 \parallel B^1) \\ & \oplus \\ & \dots \\ & \oplus \\ & H(n \parallel A^n \parallel B^n) \\ & \oplus \\ & (C^1, \dots, C^n) \end{aligned}$$

- $E_{A_0, B_0}(C_0)$
- $E_{A_0, B_1}(C_0)$
- $E_{A_1, B_0}(C_0)$
- $E_{A_1, B_1}(C_1)$

Shrink the keys!

For hash function $H : \{0,1\}^* \rightarrow \{0,1\}^{nk}$

BMR with Short Keys



Reusing keys reduces security in regular syndrome decoding problem for:

High fan-out

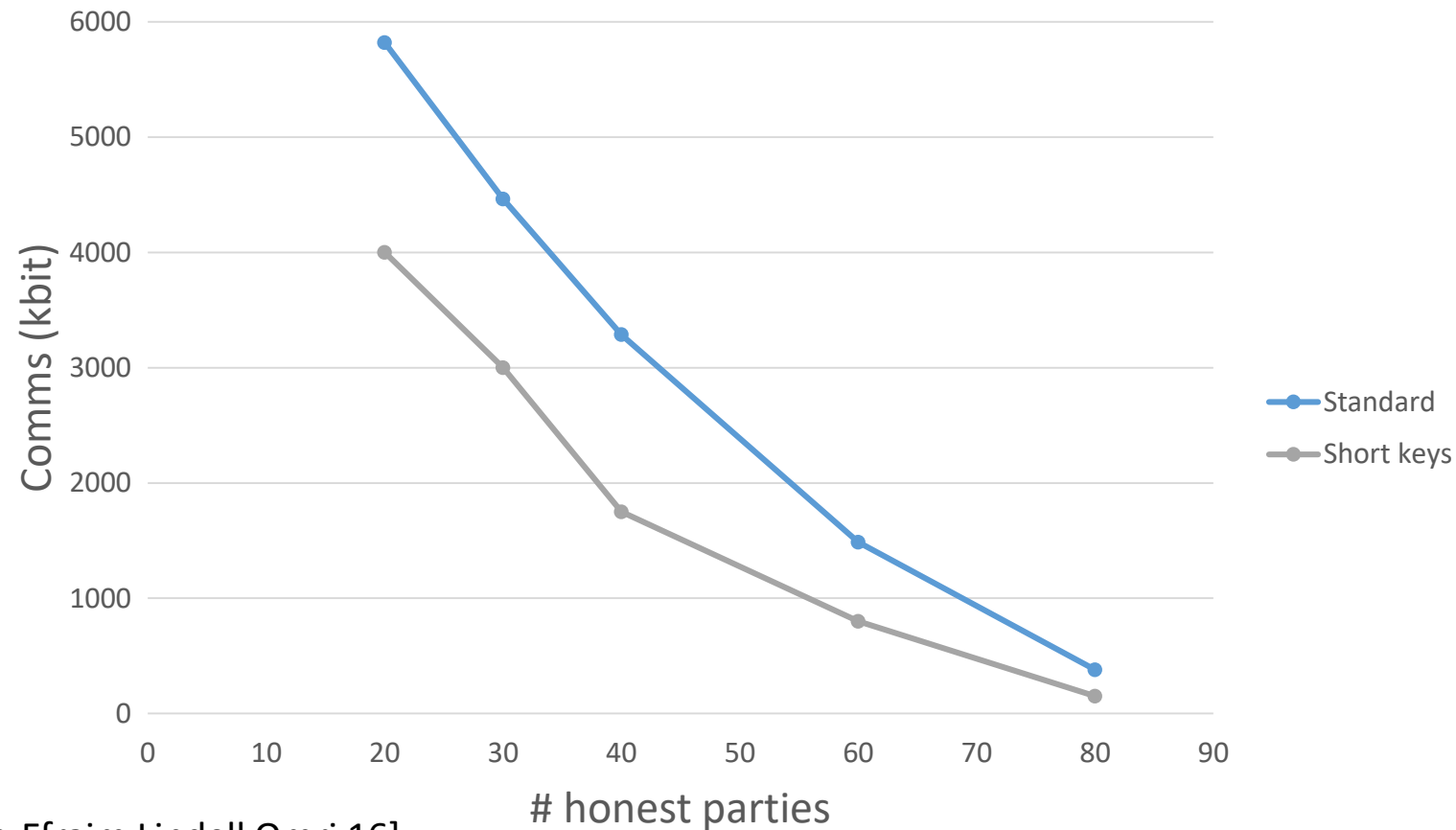
Free-xor

Solution:

Splitter gates [Tate Xu 03] – can be garbled **for free**

Local free-XOR offsets

BMR: Communication Cost of Garbling an AND Gate (100 Parties)



Comparison with [Ben-Efraim Lindell Omri 16]

Conclusion and Future Directions

New technique for **distributing trust** (more honesty \Rightarrow shorter keys)

Improved protocols with 20+ parties

GMW: Up to 25x in communication (vs multi-party [DKSSZZ17])

BMR: Up to 7x in communication (vs [BLO16])

Follow-up work: Active Security – TinyKeys for TinyOT (Asiacrypt '18)

Future challenges:

Optimizations, more **cryptanalysis** (conservative parameters atm)

Thank you! Questions?

Paper: <https://ia.cr/2017/214> [Full version]

TinyKeys: A New Approach to Efficient Multi-Party Computation

Carmit Hazay, Emmanuela Orsini, Peter Scholl and Eduardo Soria-Vázquez