MPTS 2023: NIST Workshop on Multi-Party Threshold Schemes 2023

Gadgets for Threshold AES:

Correlation Robust Hash and Authenticated Garbling

Hongrui Cui Chenkai Weng

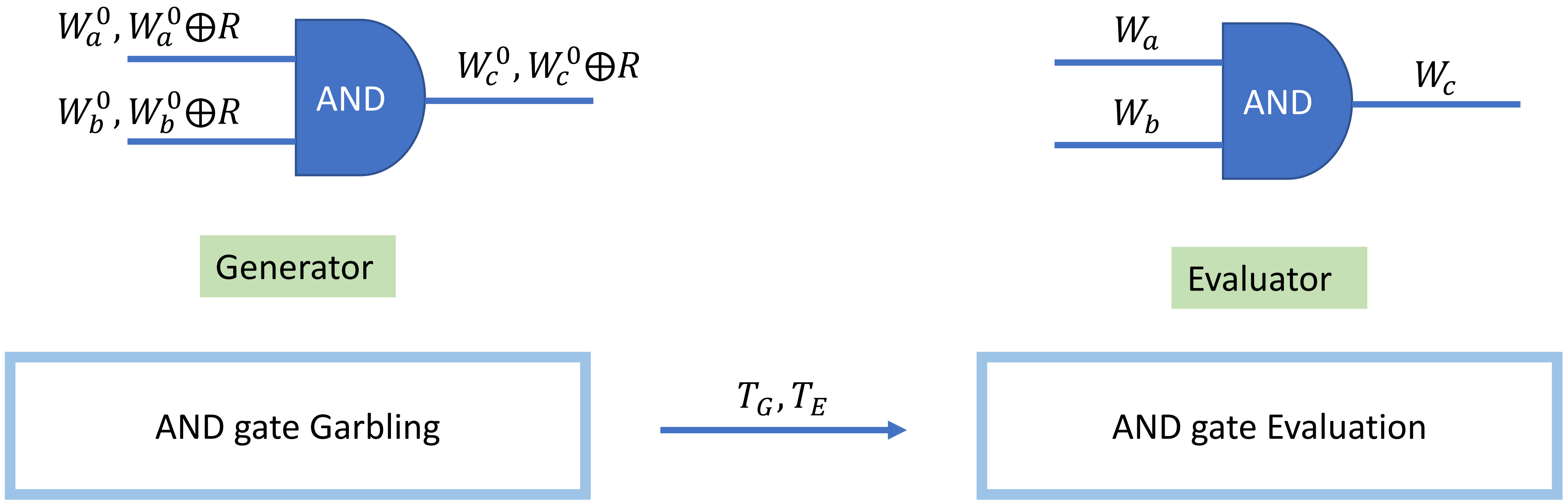
09/28/2023

Correlation Robust Hash Functions

Overview

- Previous half-gates *implementation.*
  - Weakness.
  - Attack.
- A new design of correlation robust hash.
  - Concrete security.
  - Performance.

Half-gates: Garble an AND gate



$$T_G = H(W_a^0, j) \oplus H(W_a^1, j) \oplus p_b R$$

$$T_E = H\left(W_b^0, j'\right) \oplus H\left(W_b^1, j'\right) \oplus W_a^0$$

Attack overview

Designed for better performance (compared to SHA)

- Exploit the weakness when $H()$ is instantiated by fixed-key AES.
  - π modeled as a random permutation.

$$H(x, i) = \pi(2x \oplus i) \oplus 2x \oplus i.$$

Attack overview

- Exploit the weakness when $H()$ is instantiated by fixed-key AES.
  - π modeled as a random permutation.

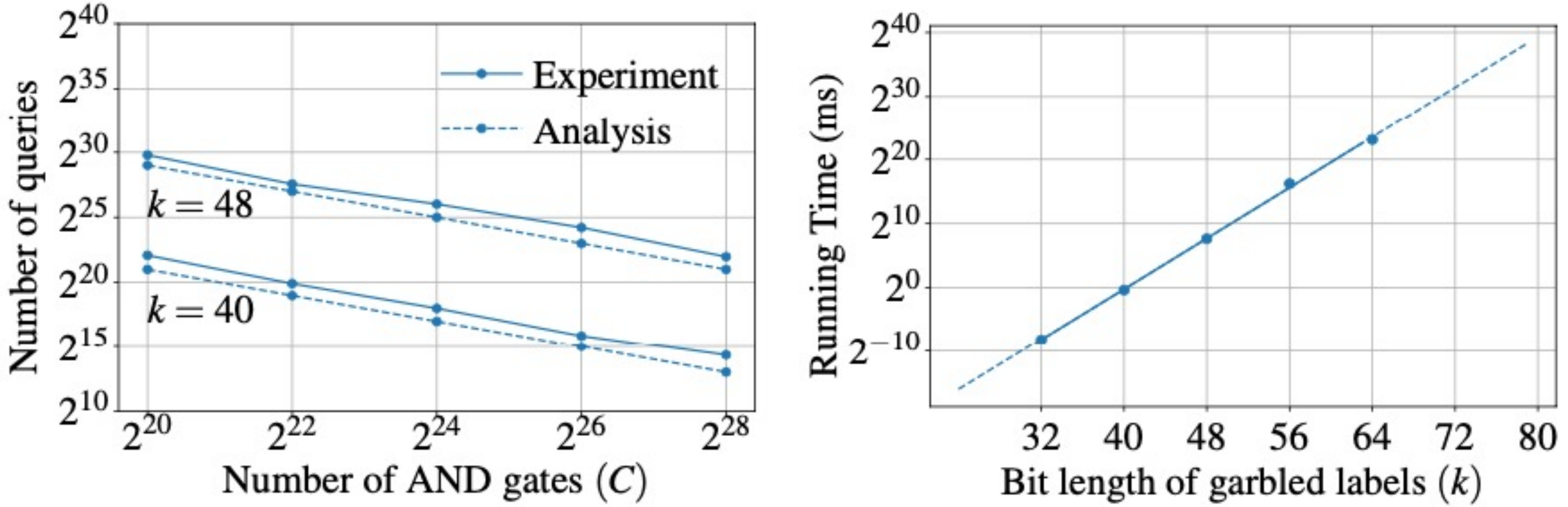
$$H(x, i) = \pi(2x \oplus i) \oplus 2x \oplus i.$$

- Attacker succeed in running time $O\left(2^k / C\right)$.
  - Circuit with $k = 80$ and $C = 2^{40}$ would be completely broken.
  - Circuit with $k = 128$ and $C = 2^{40}$ has only ~80 bit security.
  - Extend to multi-instance case: Attack is effective when *C* is the total size of multiple circuits.

k: bit length of the labels
C: # of AND gates

Attack overview

- Implementation of the attack is consistent with analysis.

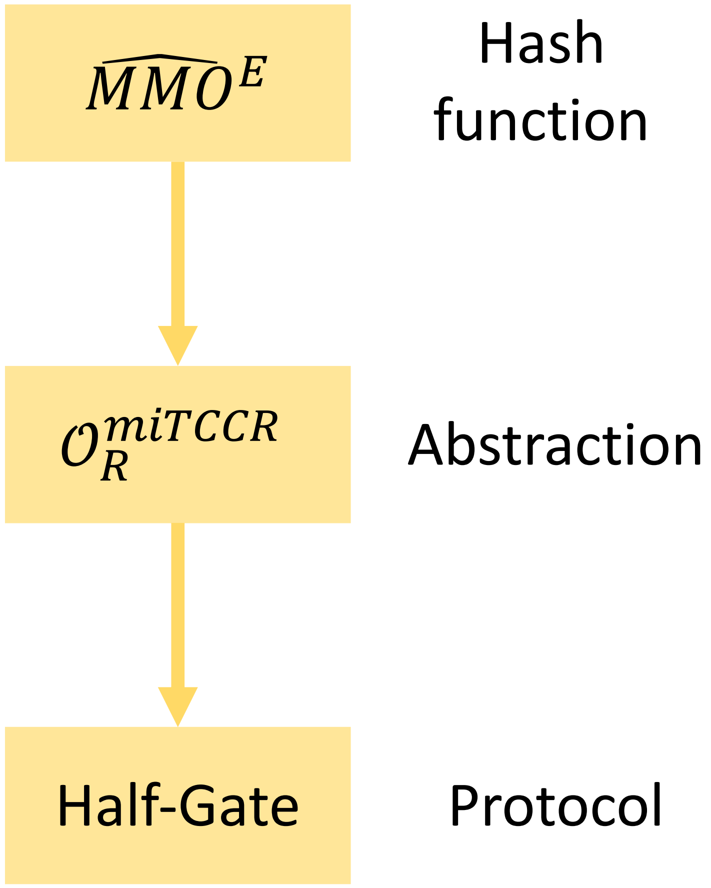


(a) Number of π-queries for the attack to succeed, on a log/log scale.

(b) The running time of our attack with $C = 2^{30}$ and different values of k.

Interpolate:
When k=80, $C = 2^{30}$ attack needs 267 machine-months and $3500.

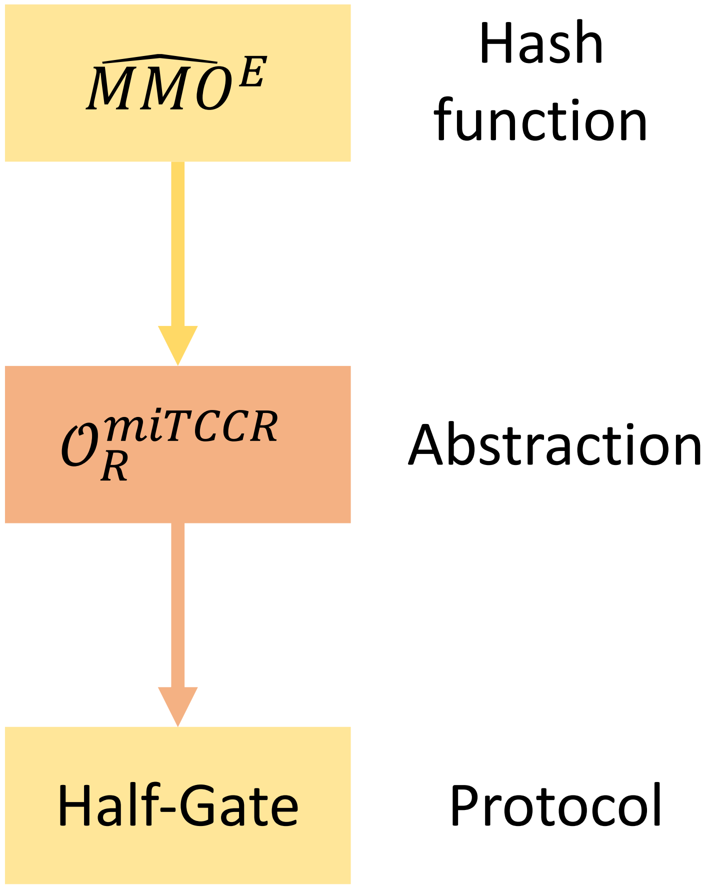
Better concrete security



Abstraction

$$\mathcal{O}_R^{miTCCR}(w, i, b) \stackrel{\text{def}}{=} H(w \oplus R, i) \oplus b \cdot R$$

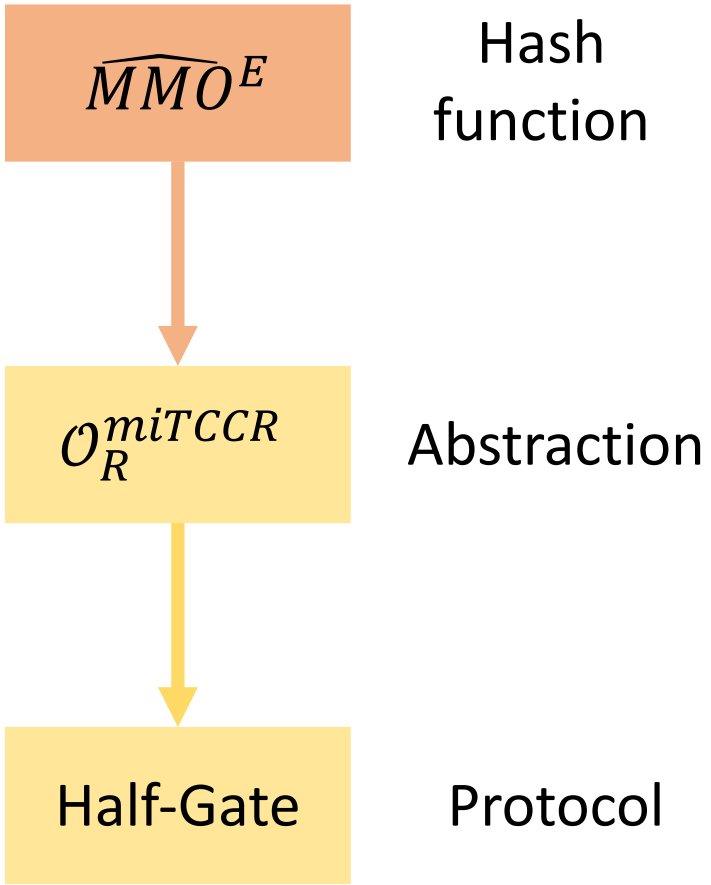
- Adversary is given u oracle instances.
- Never queries both $(w, i, 0), (w, i, 1)$.
- Same i is used at most μ times.



The Hash Function

$$\widehat{MMO}^E(x,i) \stackrel{\text{def}}{=} E\big(i,\sigma(x)\big)\oplus\sigma(x)$$

- $\sigma(x)$: a linear orthomorphism.
  - $\sigma(x_L \parallel x_R) = x_R \oplus x_L \parallel x_L$
- E: modeled as an ideal cipher.
  - Key scheduling for each i.
  - i starts at a random value.



Concrete Security

- Concrete security of Half-Gates.

$$\varepsilon = \underbrace{\frac{\mu p}{2^{k-2}}}_{\text{Computational security}} + \underbrace{\frac{(\mu - 1)C}{2^{k-2}} + \frac{(2C)^{\mu+1}}{(\mu + 1)! \cdot 2^{\mu L}}}_{\text{Statistical security}}$$

μ: reuse of tweak i.
p: #queries to E.
L: in/output length of E.

- Examples.

k (bit)	C	Comp. sec. (bit)	Sta. sec. (bit)
80	$\leq 2^{43.5}$	78	40
128	$\leq 2^{61}$	125	64

Implementation & optimization

- Performance with different hash functions

Hash function	NI support?	k	Comp. sec. (bits)	100 Mbps	2 Gbps	localhost
Zahur et al.	Y	128	89	0.4	7.8	23
SHA-3	N	128	125	0.27	0.27	0.28
SHA-256	N	128	125	0.4	1.1	1.2
SHA-256	Y	128	125	0.4	2.1	2.45
$\widehat{\text{MMO}}^E$	Y	128	125	0.4	7.8	15
$\widehat{\text{MMO}}^E$	Y	88	86	0.63	12	15

We optimized it to 20 since then

Extra Note

- Correlation robust hash function is also important to other MPC protocols, e.g. oblivious transfers.

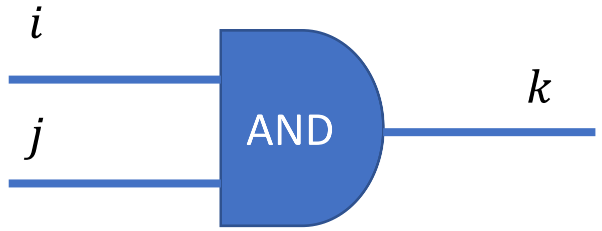
C. Guo, J. Katz, X. Wang and Y. Yu, "Efficient and Secure Multiparty Computation from Fixed-Key Block Ciphers," 2020 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2020, pp. 825-841, doi: 10.1109/SP40000.2020.00016.

Authenticated Garbling

Overview

- Semi-honest GC flaws against active adversary
  - Selective failure attack against privacy
  - Inconsistent circuit attack against correctness
- How to use authenticated garbling to fix those attacks
  - Selective Failure -> Distributed Garbling
  - Inconsistent Circuit -> IT-MAC Authentication
- Further improvements

Semi-honest Garbled Circuit



i	j	k
0	0	0
0	1	0
1	0	0
1	1	1

Permute →

i	j	k
λ_i	λ_j	λ_k
λ_i	$\bar{\lambda}_j$	λ_k
$\bar{\lambda}_i$	λ_j	λ_k
$\bar{\lambda}_i$	$\bar{\lambda}_j$	$\bar{\lambda}_k$

Encrypt →

Λ_i	Λ_j	ciphertext
λ_i	λ_j	$H(L_{i,\lambda_i}, L_{j,\lambda_j}) \oplus L_{k,\lambda_k}$
λ_i	$\bar{\lambda}_j$	$H(L_{i,\lambda_i}, L_{j,\bar{\lambda}_j}) \oplus L_{k,\lambda_k}$
$\bar{\lambda}_i$	λ_j	$H(L_{i,\bar{\lambda}_i}, L_{j,\lambda_j}) \oplus L_{k,\lambda_k}$
$\bar{\lambda}_i$	$\bar{\lambda}_j$	$H(L_{i,\bar{\lambda}_i}, L_{j,\bar{\lambda}_j}) \oplus L_{k,\bar{\lambda}_k}$

Sort ↓

Λ_i	Λ_j	ciphertext
0	0	$H(L_{i,0}, L_{j,0}) \oplus L_{k,0} \oplus (\lambda_i \cdot \lambda_j \oplus \lambda_k)\Delta_A$
0	1	$H(L_{i,0}, L_{j,1}) \oplus L_{k,0} \oplus (\lambda_i \cdot \bar{\lambda}_j \oplus \lambda_k)\Delta_A$
1	0	$H(L_{i,1}, L_{j,0}) \oplus L_{k,0} \oplus (\bar{\lambda}_i \cdot \lambda_j \oplus \lambda_k)\Delta_A$
1	1	$H(L_{i,1}, L_{j,1}) \oplus L_{k,0} \oplus (\bar{\lambda}_i \cdot \bar{\lambda}_j \oplus \lambda_k)\Delta_A$

Wire Index	False Label	Truth Label	Permutation Bit
i	$L_{i,0}$	$L_{i,1} = L_{i,0} \oplus \Delta_A$	λ_i
j	$L_{j,0}$	$L_{j,1} = L_{j,0} \oplus \Delta_A$	λ_j
k	$L_{k,0}$	$L_{k,1} = L_{k,0} \oplus \Delta_A$	λ_k

- $\Lambda_i = \lambda_i \oplus z_i$ -> Masked wire value
- Δ_A -> Garbler's key in Free-XOR

Security Issues against Active Adversaries

- Attack 1: Selective Failure

- Suppose P_B decrypts \$\$\$ and failed
- P_A learns $z_i = \Lambda_i, z_j = \overline{\Lambda_j}$

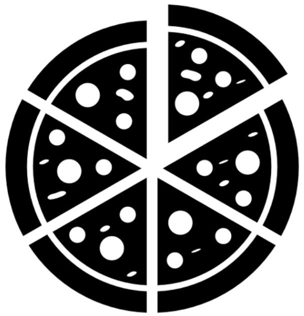
Λ_i	Λ_j	ciphertext
0	0	$H(L_{i,0}, L_{j,0}) \oplus L_{k,0} \oplus (\lambda_i \cdot \lambda_j \oplus \lambda_k)\Delta_A$
0	1	\$\$\$
1	0	$H(L_{i,1}, L_{j,0}) \oplus L_{k,0} \oplus (\bar{\lambda}_i \cdot \lambda_j \oplus \lambda_k)\Delta_A$
1	1	$H(L_{i,1}, L_{j,1}) \oplus L_{k,0} \oplus (\bar{\lambda}_i \cdot \bar{\lambda}_j \oplus \lambda_k)\Delta_A$

- Attack 2: Circuit Logic Inconsistency

- P_A flips each AND gate output
- AND -> NAND

Λ_i	Λ_j	ciphertext
0	0	$H(L_{i,0}, L_{j,0}) \oplus L_{k,0} \oplus (\lambda_i \cdot \lambda_j \oplus \bar{\lambda}_k)\Delta_A$
0	1	$H(L_{i,0}, L_{j,1}) \oplus L_{k,0} \oplus (\lambda_i \cdot \bar{\lambda}_j \oplus \bar{\lambda}_k)\Delta_A$
1	0	$H(L_{i,1}, L_{j,0}) \oplus L_{k,0} \oplus (\bar{\lambda}_i \cdot \lambda_j \oplus \bar{\lambda}_k)\Delta_A$
1	1	$H(L_{i,1}, L_{j,1}) \oplus L_{k,0} \oplus (\bar{\lambda}_i \cdot \bar{\lambda}_j \oplus \bar{\lambda}_k)\Delta_A$

Previous Solutions

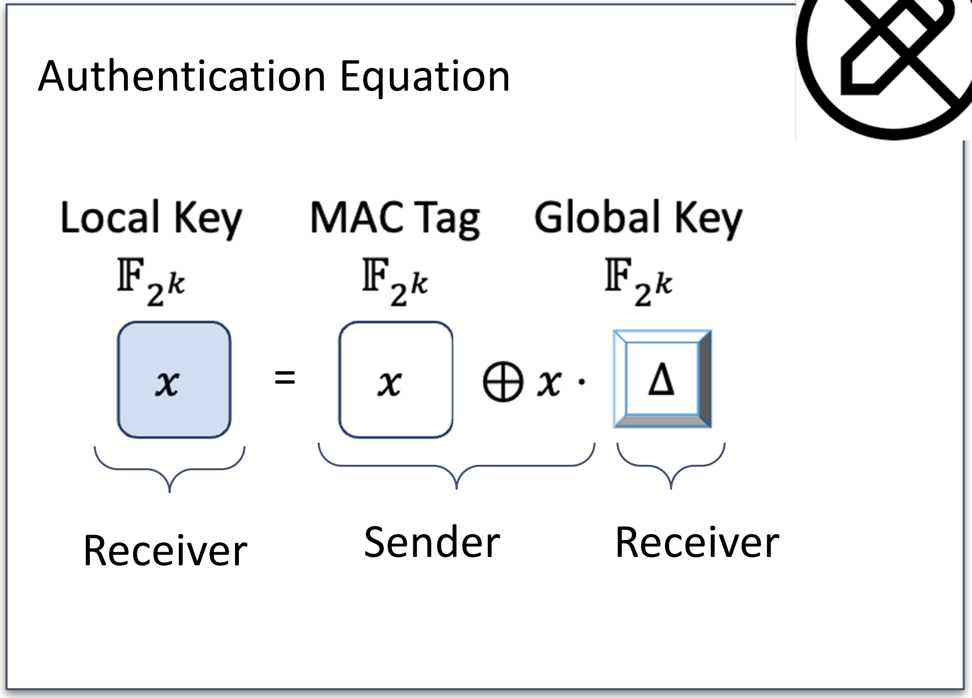


- Cut-and-choose [LP07, NO09, HKE13, NST17…]
- P_A prepares ρ different garbled circuits/gates
- P_B checks $\frac{\rho}{2}$ of them (by requesting random seeds)

To achieve statistical soundness of $2^{-\rho}$
P_A needs to garble ρ circuits ☹

IT-MAC

- TinyOT-style bit authentication
- Open(x) -> Sending [x, x]
- Opening to $\bar{x}$ <-> Sending x $\oplus$ Δ
  <-> Guessing Δ
- Efficient Instantiation:
  - Base OT + Extension [IKNP03, KOS15, Roy22, …]
  - COT PCG [BCGI18, BCGIKS19, YWLZW20, CRR21,RRT23…]



Distributed Garbling

- P_A needs to know λ_i, λ_j to launch selective failure attack
- The attack fails if we share
  - $\lambda_i = a_i \oplus b_i$
  - $\lambda_j = a_j \oplus b_j$
  - $\lambda_i \cdot \lambda_j = \hat{a}_k \oplus \hat{b}_k$

Λ_i	Λ_j	ciphertext
0	0	$H(L_{i,0}, L_{j,0}) \oplus L_{k,0} \oplus (\lambda_i \cdot \lambda_j \oplus \lambda_k)\Delta_A$
0	1	\$\$\$
1	0	$H(L_{i,1}, L_{j,0}) \oplus L_{k,0} \oplus (\bar{\lambda}_i \cdot \lambda_j \oplus \lambda_k)\Delta_A$
1	1	$H(L_{i,1}, L_{j,1}) \oplus L_{k,0} \oplus (\bar{\lambda}_i \cdot \bar{\lambda}_j \oplus \lambda_k)\Delta_A$

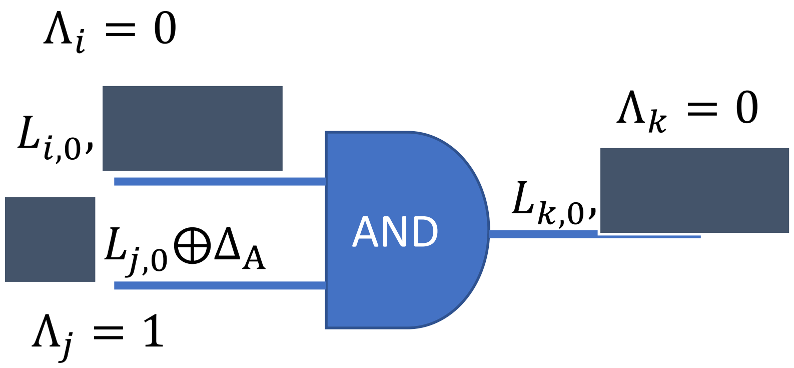
- $(\lambda_i \cdot \lambda_j \oplus \lambda_k) \cdot \Delta_A = (\hat{a}_k \oplus \hat{b}_k \oplus a_k \oplus b_k) \cdot \Delta_A$

P_A can still garble if $b_i, b_j, b_k, \hat{b}_k$ are authenticated by Δ_A

$$b = b \oplus b \cdot \Delta_A$$

$b \in \{b_i, b_j, b_k, \hat{b}_k\}$

Consistency Checking



Λ_i	Λ_j	ciphertext
0	0	[redacted]
0	1	$H(L_{i,0}, L_{i,1}) \oplus L_{k,0} \oplus (\lambda_i \cdot \bar{\lambda}_j \oplus \lambda_k)\Delta_A$
1	0	[redacted]
1	1	[redacted]

$$\boxed{a} = \boxed{a} \oplus a \cdot \boxed{\Delta_B}$$

$a \in \{a_i, a_j, a_k, \hat{a}_k\}$

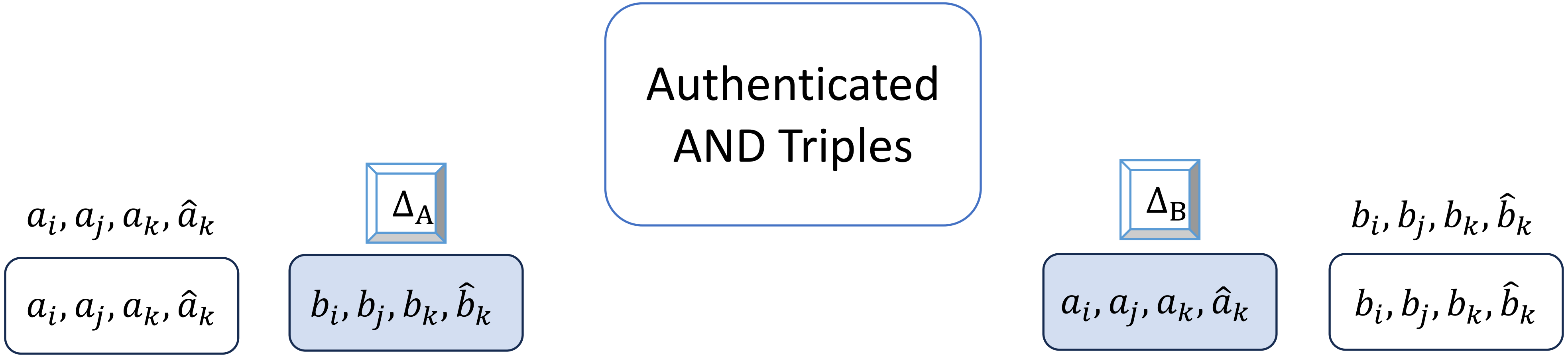
- P_B wants to ensure that $(\lambda_i \oplus \Lambda_i) \cdot (\lambda_j \oplus \Lambda_j) = \lambda_k \oplus \Lambda_k$
  - Use an additional AuthGC to let P_B learn the correct Λ_k [WRK17, DILO22]
  - Add an additional round and let P_B publish $\Lambda_i, \Lambda_j, \Lambda_k$

Λ_i	Λ_j	Alice's AuthGC	Bob's AuthGC
0	0	$L_{k,0} \oplus M[\Lambda_{00}]$	$K[\Lambda_{00}]$
0	1	$L_{k,0} \oplus M[\Lambda_{01}]$	$K[\Lambda_{01}]$
1	0	$L_{k,0} \oplus M[\Lambda_{10}]$	$K[\Lambda_{10}]$
1	1	$L_{k,0} \oplus M[\Lambda_{11}]$	$K[\Lambda_{11}]$

Linear relation on Δ_B-authenticated values

$$\hat{a}_k \oplus \hat{b}_k \oplus \Lambda_j(a_i \oplus b_i) \oplus \Lambda_i(a_j \oplus b_j) \oplus \Lambda_i\Lambda_j = a_k \oplus b_k \oplus \Lambda_k$$

Preprocessing



- TinyOT-style protocol [NNOB12, WRK17, KRRW18]
- Ring-LPN based PCG [BCGIKS20]

$$\overset{A}{\begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}} \cdot \overset{x}{\begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}} + \overset{e}{\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}} = \overset{y}{\begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}} \pmod 2$$

Compressed Preprocessing

- Actually, $\mathrm{H}_\infty(\boldsymbol{b})$ only needs to be $\tilde{O}(\rho)$-bit [DILO22]
- $\boldsymbol{b}$ only prevents selective failure-resilience
- Together with efficient COTs, this brings constant amortized communication in preprocessing [**C**WXY23]

2PC	Rounds		Communication per AND gate	
	Prep.	Online	one-way (bits)	two-way (bits)
Half-gates	1	2	2κ	2κ
HSS-PCG [28]	8	2	$8\kappa + 11$ (4.04×)	$16\kappa + 22$ (8.09×)
KRRW-PCG [32]	4	4	$5\kappa + 7$ (2.53×)	$8\kappa + 14$ (4.05×)
DILO [18]	7	2	$2\kappa + 8\rho + 1$ (2.25×)	$2\kappa + 8\rho + 5$ (2.27×)
DILOv2 [18]	3	4	$2\kappa + 2\rho + 2$ (1.32×)	$2\kappa + 4\rho + 2$ (1.63×)
This work, v.1	8	3	$2\kappa + 5$ (**1.02**×)	$4\kappa + 10$ (2.04×)
This work, v.2	8	2	$2\kappa + \rho + 3$ (1.17×)	$2\kappa + \rho + 4$ (**1.17**×)

Q & A