Making malicious security orders of magnitude more efficient than previous semi-honest

N FOR
SEM ERSARIES

Tore Frederik en[3], Yehuda Lindell[1,2], Valery Osheter[2], Benny Pinkas[1]

15 min vs. 41 sec

1: BIU Center for Research in Applied Cryptography and Cyber Security

2: UNBOUND

3: ALEXANDRA INSTITUTE

ישראל

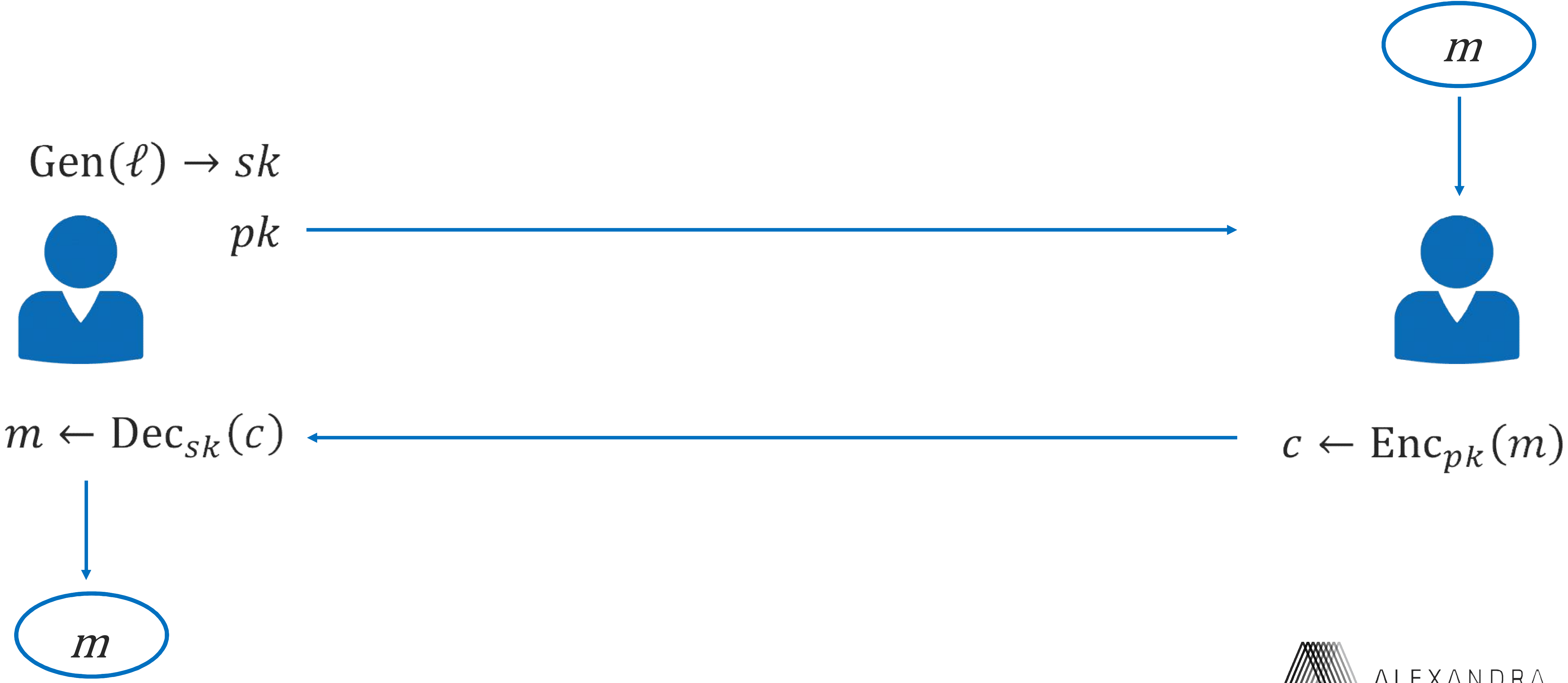
הקרן הלאומית למדע
ISRAEL SCIENCE FOUNDATION

SODA
Scalable Oblivious Data Analytics

OUTLINE

- **Introduction**
- Semi-honest construction
- Malicious construction
- Efficiency
- Conclusion

INTRODUCTION – PUBLIC KEY ENCRYPTION



m

$\mathrm{Gen}(\ell) \to sk$

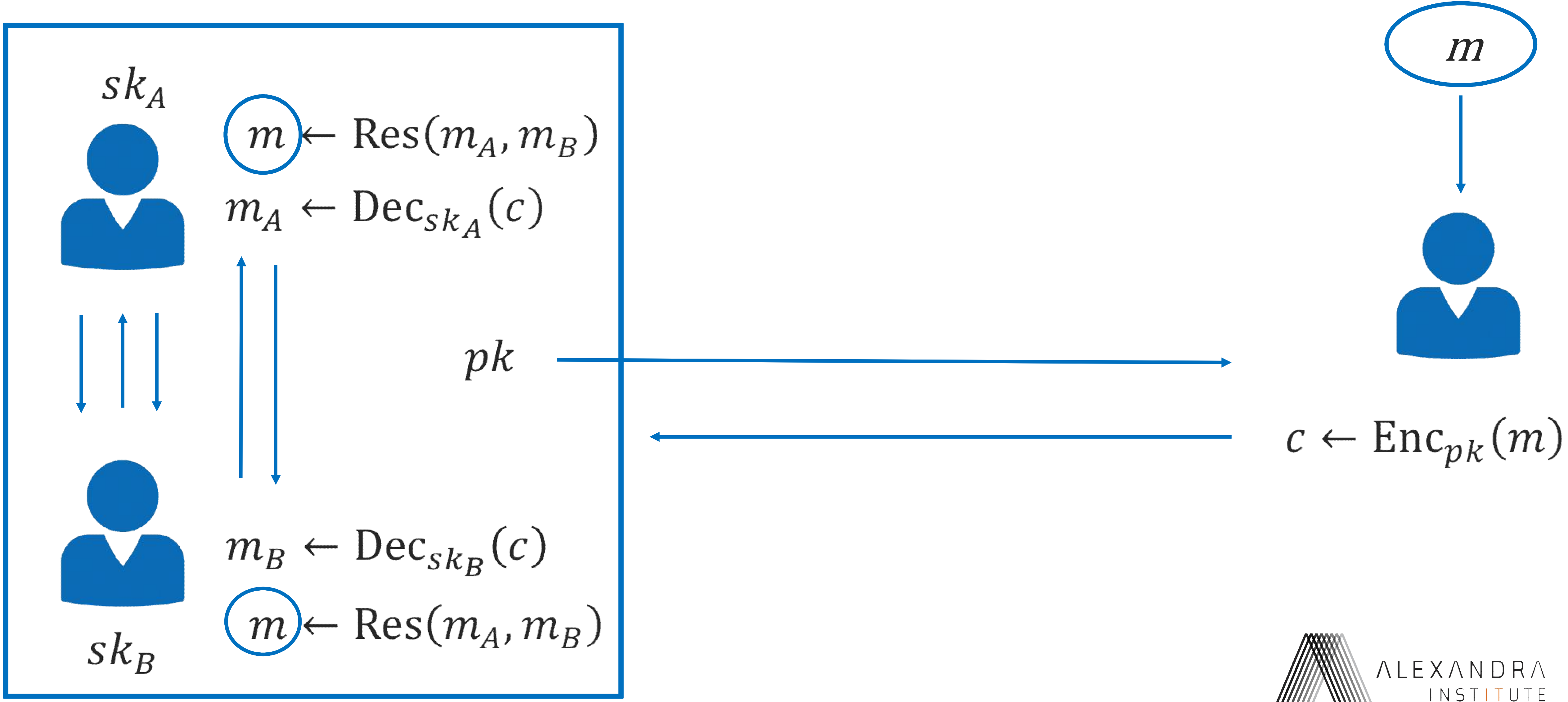
pk

$m \leftarrow \mathrm{Dec}_{sk}(c)$

$c \leftarrow \mathrm{Enc}_{pk}(m)$

m

INTRODUCTION – DISTRIBUTED PKE



sk_A

$m \leftarrow \mathrm{Res}(m_A, m_B)$

$m_A \leftarrow \mathrm{Dec}_{sk_A}(c)$

pk

$m_B \leftarrow \mathrm{Dec}_{sk_B}(c)$

$m \leftarrow \mathrm{Res}(m_A, m_B)$

sk_B

m

$c \leftarrow \mathrm{Enc}_{pk}(m)$

INTRODUCTION – MOTIVATION

- *Sometimes* it can also be used for distributed signature schemes
  - Which is an end in itself
- Relevant for MPC protocols
  - CDN01, semi-homomorphic PKE
  - DPSZ12, somewhat-homomorphic PKE
- Cloud based key management
  - SEPIOR
  - UNBOUND

INTRODUCTION – RSA

- RSA:
  - Find ℓ bit primes p and q
  - **Public key:** $pq = N, e\ (= 3, 2^{16} + 1)$
  - **Private key:** $d \equiv e^{-1} \bmod (p-1)(q-1)$
- RSA is widely in use
  - TLS, PGP, …
- Lots of previous work on the distributed setting
  - …, [Gil99], [BF01], [ACS02], [DM10], [HMR+12]
- Challenging to solve efficiently

INTRODUCTION – DISTRIBUTED RSA

- Distributed RSA:
  - Find ℓ bit primes $p = p_A + p_B$ and $q = q_A + q_B$
  - **Public key:** $(p_A + p_B) \cdot (q_A + q_B) = N, e\ (= 3, 2^{16} + 1)$
  - **Private key:** $d_A + d_B \equiv e^{-1} \bmod (p-1)(q-1)$

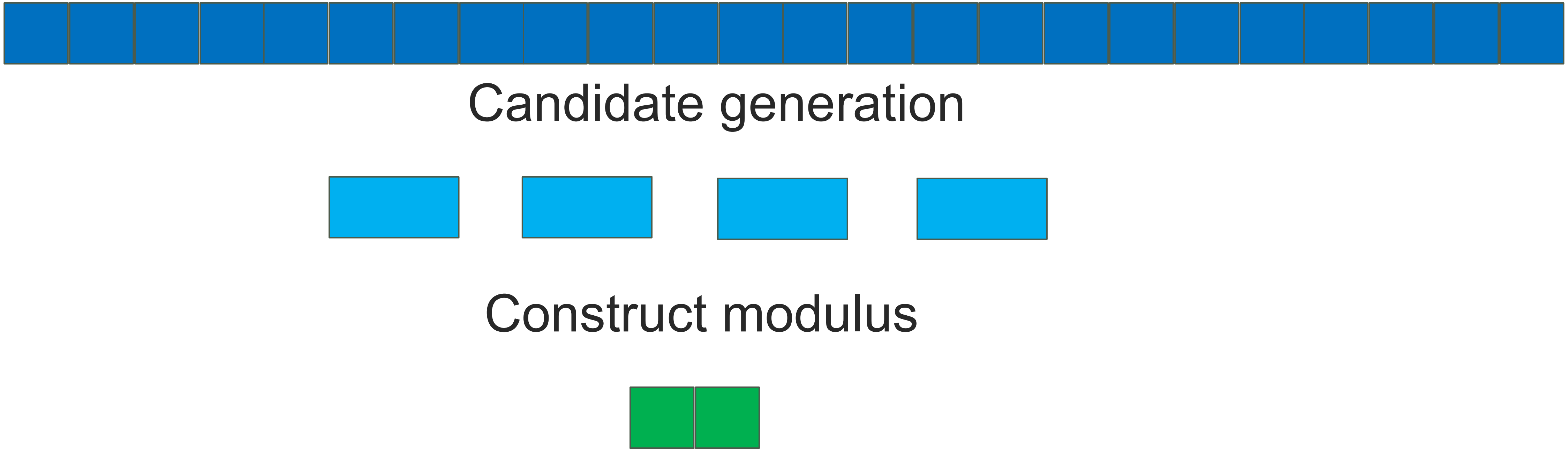
    - Pick random p_A, q_A, p_B, q_B
    - Do Rabin-Miller
    - Repeat

MPC

INTRODUCTION – DISTRIBUTED RSA

- Candidate generation
  - Sampling random p_A, q_A, p_B, q_B s.t. $p = p_A + p_B$ and $q = q_A + q_B$
- Construct modulus
  - Compute $N = (p_A + p_B) \cdot (q_A + q_B)$
- Verify modulus
  - Check that N is the product of two primes
- Construct keys
  - Construct shares d_A and d_B s.t. $d \equiv e^{-1} \bmod (p-1) \cdot (q-1)$

INTRODUCTION – INTUITION



Candidate generation

Construct modulus

Verify modulus

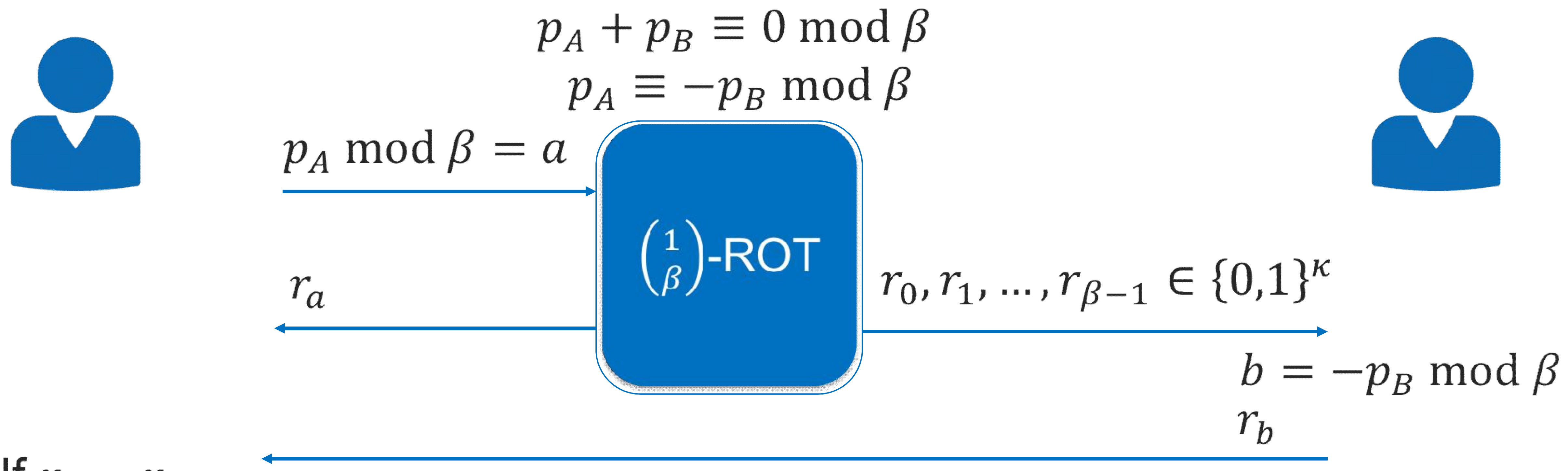
Construct keys

OUTLINE

- Introduction
- **Semi-honest construction**
- Malicious construction
- Efficiency
- Conclusion

SEMI-HONEST – CANDIDATE GENERATION

- $p_A, p_B \in \mathbb{Z}_{2^{1024}}$ s.t. $p = p_A + p_B \equiv 3 \bmod 4$
- Trial division by small prime β [PS98]



$$p_A + p_B \equiv 0 \bmod \beta$$

$$p_A \equiv -p_B \bmod \beta$$

$p_A \bmod \beta = a$

$\binom{1}{\beta}$-ROT

r_a

$r_0, r_1, \dots, r_{\beta-1} \in \{0,1\}^\kappa$

$b = -p_B \bmod \beta$

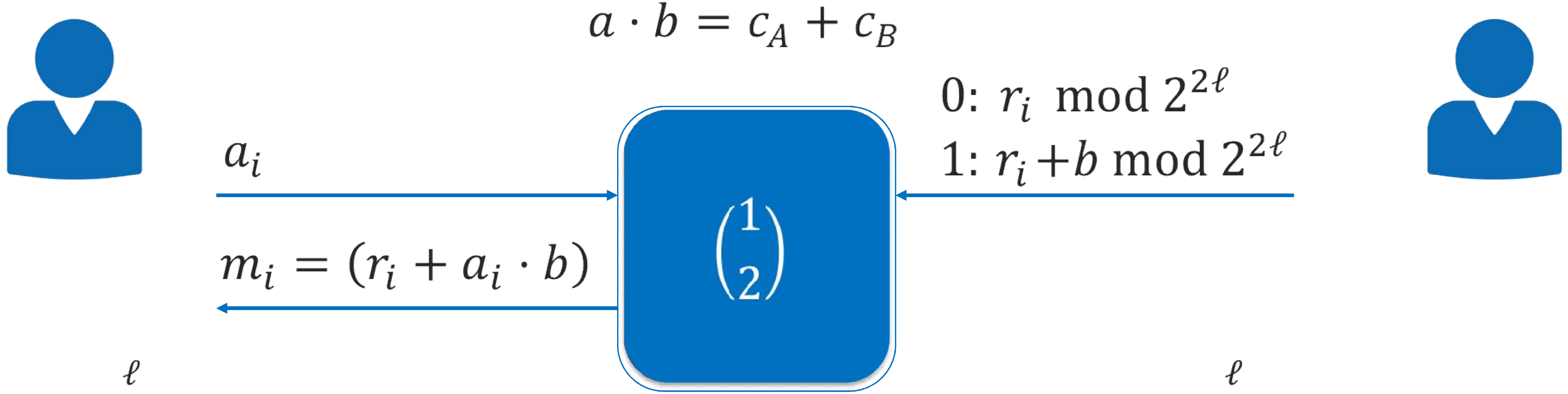
r_b

If $r_a = r_b$
then p not prime

SEMI-HONEST – CONSTRUCT MODULUS

- $(p_A + p_B) \cdot (q_A + q_B) = p_A \cdot q_A + p_B \cdot q_B + \underline{p_A \cdot q_B} + \underline{p_B \cdot q_A}$
- Compute multiplication using OT [Gil99]

$$a \cdot b = c_A + c_B$$



a_i

$m_i = (r_i + a_i \cdot b)$

$\binom{1}{2}$

$0\colon\ r_i \bmod 2^{2\ell}$

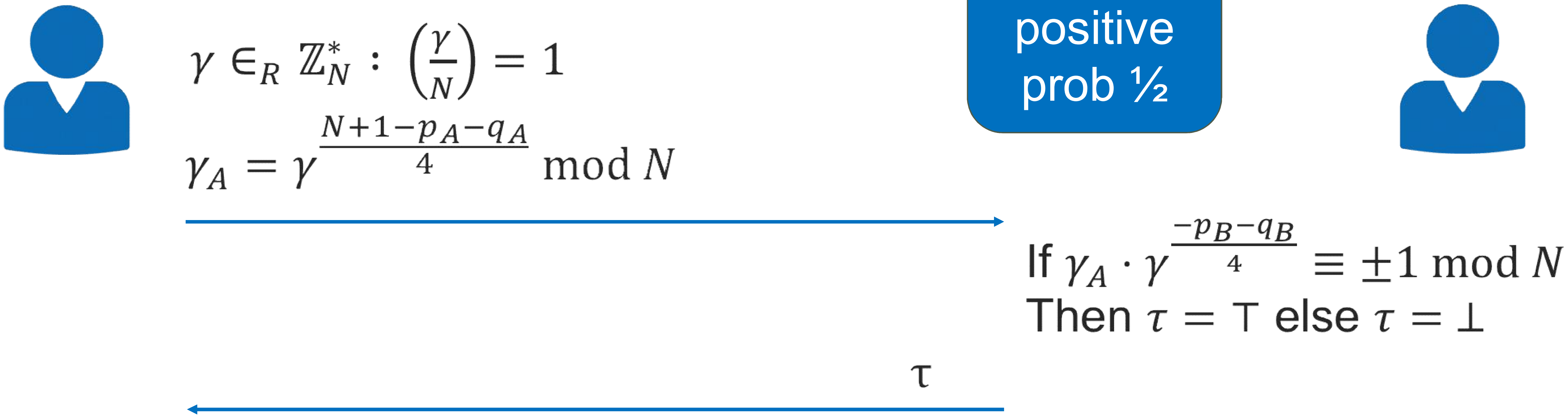
$1\colon\ r_i + b \bmod 2^{2\ell}$

$$c_A = \sum_{i=1}^{\ell} m_i \cdot 2^i \bmod 2^{2}$$

$$c_B = -\sum_{i=1}^{\ell} r_i \bmod 2^{2\ell}$$

SEMI-HONEST – VERIFY MODULUS

- Biprimality test [BF01]



False positive prob ½

$$\gamma \in_R \mathbb{Z}_N^* : \left(\frac{\gamma}{N}\right) = 1$$

$$\gamma_A = \gamma^{\frac{N+1-p_A-q_A}{4}} \bmod N$$

If $\gamma_A \cdot \gamma^{\frac{-p_B-q_B}{4}} \equiv \pm 1 \bmod N$

Then $\tau = \top$ else $\tau = \bot$

τ

Repeat

SEMI-HONEST – CONSTRUCT KEYS

- **Easy local computation [BF01]**

- Compute
  - $w = N + 1 - p_A - q_A - p_B - q_B \bmod e$
  - $v = w^{-1} \bmod e$
- Alice outputs $d_A = \left\lfloor \frac{-v\cdot(N+1-p_A-q_A)+1}{e} \right\rfloor$
- Bob outputs $d_B = \left\lfloor \frac{-v\cdot(-p_B-q_B)}{e} \right\rfloor$

OUTLINE

- Introduction
- Semi-honest construction
- **Malicious construction**
- Efficiency
- Conclusion

MALICIOUS – IDEA

- Allow adversary to fail good candidates
- Accepted key must be “good” without leakage

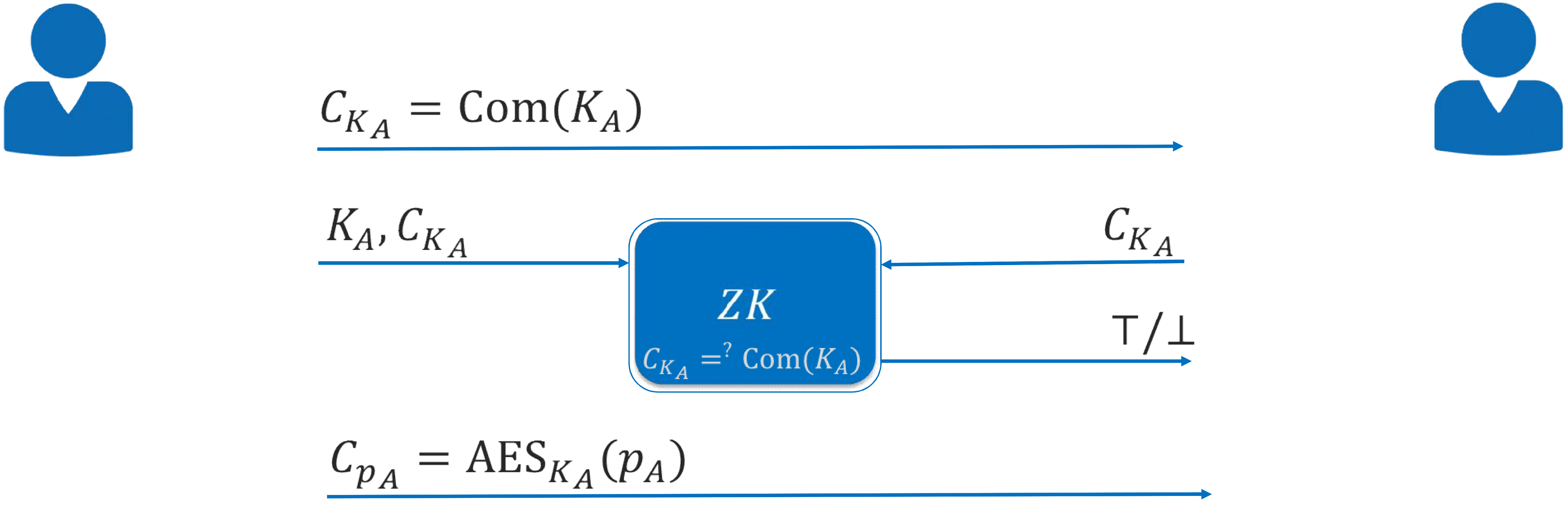
- Selective failure prevention
- Input consistency
- Correctness of biprimality

MALICIOUS – STEPS

- Selective failure prevention
  - Do OT on random, linear encoding
  - Use linearity to obtain correct product
  - Randomness ensures leakage on encoding does not leak on input
- Input consistency
  - Commitments based on AES encryption
  - Zero-knowledge of correct encryption
  - Very efficient commit-many-open-few
- Correctness of biprimality (zero-knowledge)
  - Almost standard proof-of-knowledge of discrete log
  - Few “commitments” on top to ensure composability

MALICIOUS – CONSISTENCY

- “Commitment” by encrypting using AES
- Efficient commit-many-open-few



$C_{K_A} = \mathrm{Com}(K_A)$

K_A, C_{K_A}

C_{K_A}

ZK

$C_{K_A} =^{?} \mathrm{Com}(K_A)$

$\top / \bot$

$C_{p_A} = \mathrm{AES}_{K_A}(p_A)$

MALICIOUS – VERIFY MODULUS

$\gamma \in_R \mathbb{Z}_N^* : \left(\frac{\gamma}{N}\right) = 1$

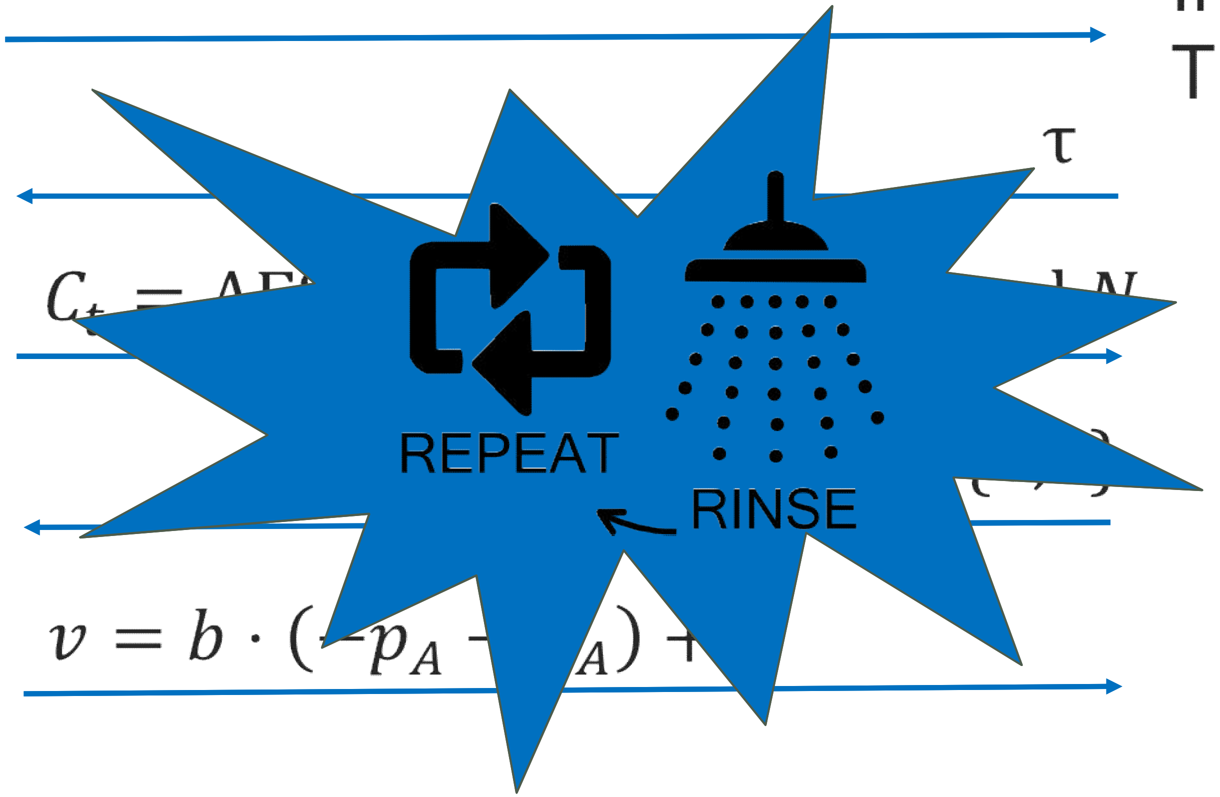
$\gamma_A = \gamma^{\frac{N+1-p_A-q_A}{4}} \bmod N$

If $\gamma_A \cdot \gamma^{\frac{-p_B-q_B}{4}} \equiv \pm 1 \bmod N$
Then $\tau = \perp$ else $\tau = \perp$

τ

$t \in_R \mathbb{Z}_{N+2^s}$

$C_t =$



$v = b \cdot (-p_A -$

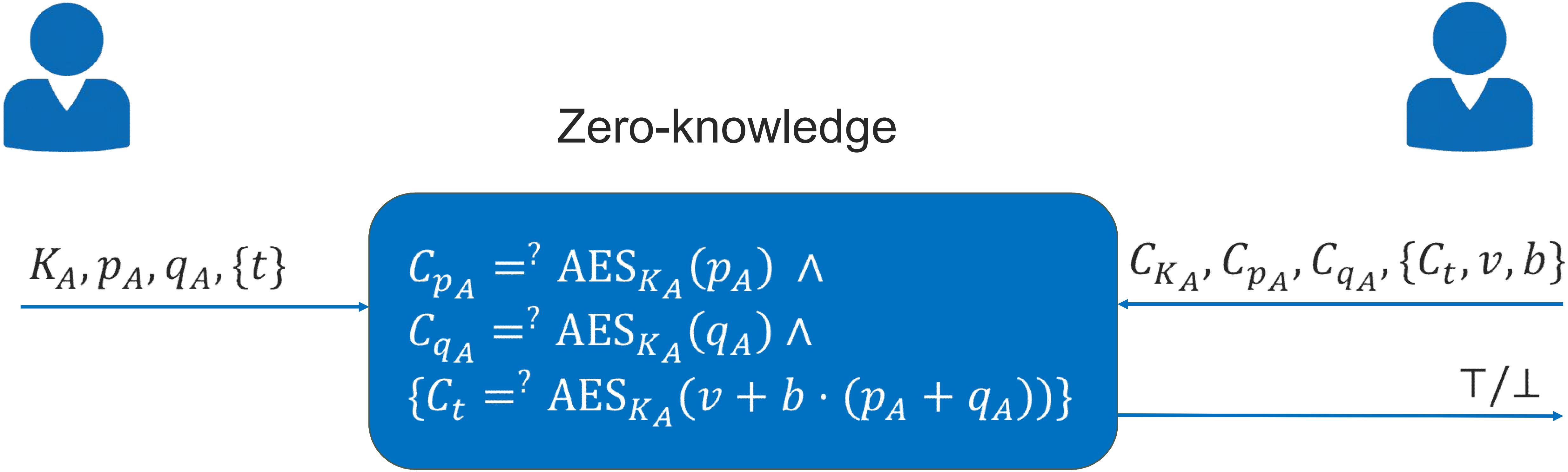
$_A) +$

$\gamma^v \bmod N$

$\overset{?}{=}$

$\overline{\gamma_A} \cdot \gamma_A^b \cdot \gamma^{\frac{-b\cdot(N+1)}{4}} \bmod N$

MALICIOUS – VERIFY MODULUS



Zero-knowledge

$K_A, p_A, q_A, \{t\}$

$$C_{p_A} =^? \mathrm{AES}_{K_A}(p_A) \ \wedge$$
$$C_{q_A} =^? \mathrm{AES}_{K_A}(q_A) \wedge$$
$$\{C_t =^? \mathrm{AES}_{K_A}(v + b \cdot (p_A + q_A))\}$$

$C_{K_A}, C_{p_A}, C_{q_A}, \{C_t, v, b\}$

$\top/\bot$

OUTLINE

- Introduction
- Semi-honest construction
- Malicious construction
- **Efficiency**
- Conclusion

ALEXANDRA INSTITUTE

EFFICIENCY – IMPLEMENTATION 2048 RSA

- AES-NI for AES and PRG
- [KOS15] for OTs (seed OTs using [PVW08])
- [NP99] for 1-out-of-β OTs
- ZK using garbled circuits using [JKO13]
- Primitives based on OpenSSL

IMPLEMENTATION – EXPERIMENTS

Malicious!

- Azure using multi-threaded Xeon machine
- Single-thread min 56, max 598, average 182 seconds
- 8-thread, average 41 seconds
- Best previous 15 minutes for *semi-honest* [HMR+12]

Phase	Percentage
Candidate generation	10
Construct modulus	55
Verify modulus	6
Zero-knowledge	16*
Other	13

OUTLINE

- Introduction
- Semi-honest construction
- Malicious construction
- Efficiency
- **Conclusion**

CONCLUSION

- New protocol for malicious distributed RSA generation
  - Malicious security almost for free
  - No specific number theoretic assumptions
  - Implementation
- New efficient commit-many-open-few protocol
- Effective selective failure prevention for multiplication using OT

Thank you for your attention!

Tore Frederiksen
Cryptography Engineer
tore.frederiksen@alexandra.dk

ישראל
הקרן הלאומית למדע
ISRAEL SCIENCE FOUNDATION
SODA
Scalable Oblivious Data Analytics

ALEXANDRA INSTITUTE