# Efficient Pairing-free Adaptable k-out-of-N Oblivious Transfer Protocols

K. Khosravani, T. Eghlidos, M.R. Aref
Sharif university of Technology
Dept. of Electrical Engineering

Oct. 2025

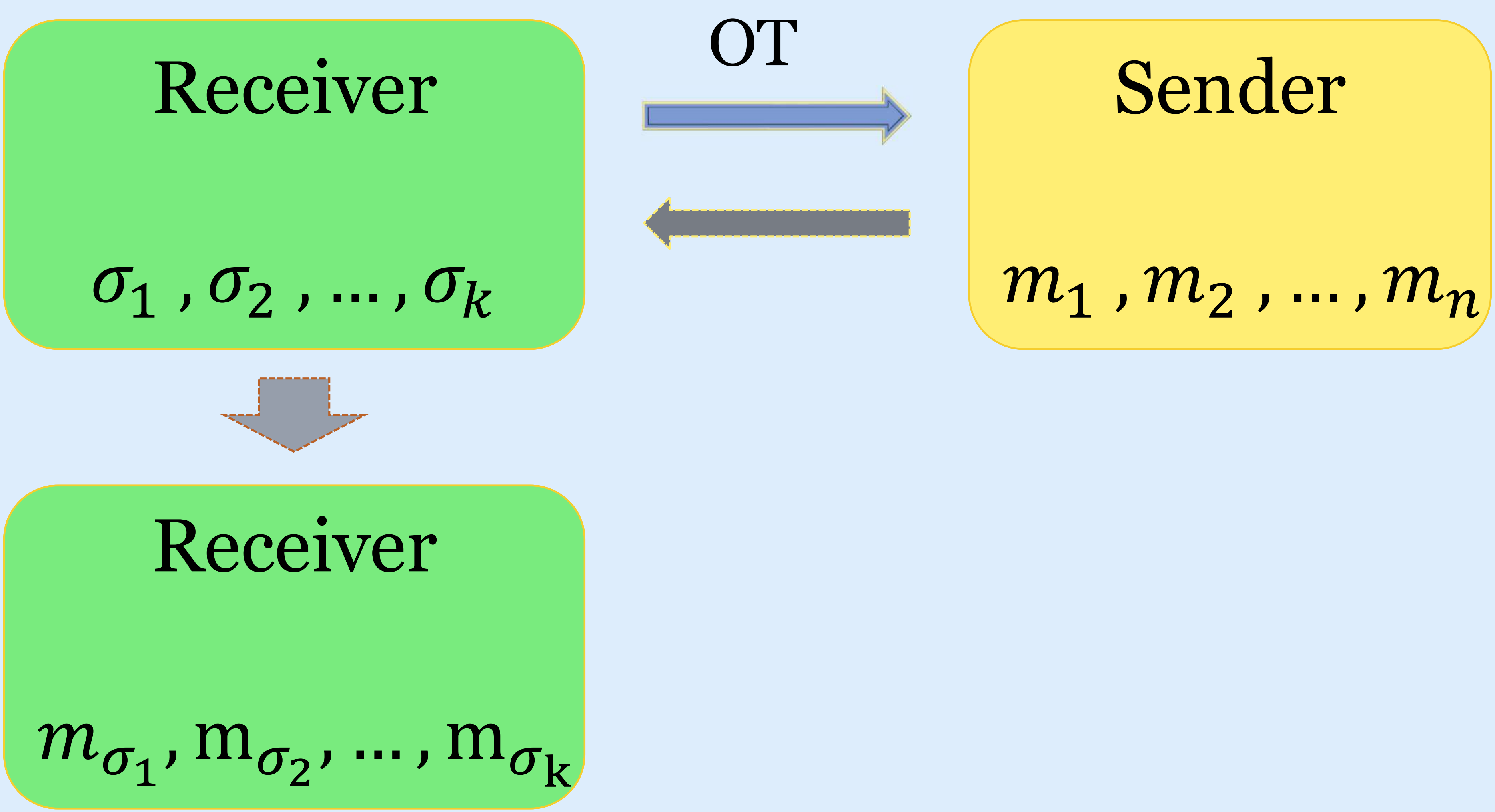
# Outline



- **Introduction**
- **K-out-of-N Oblivious Transfer**
- **Conclusion**

# Definition




OT
Receiver
$\sigma_1 , \sigma_2 , \dots , \sigma_k$
Sender
$m_1 , m_2 , \dots , m_n$
Receiver
$m_{\sigma_1}, \mathrm{m}_{\sigma_2}, \dots , \mathrm{m}_{\sigma_\mathrm{k}}$

# Applications

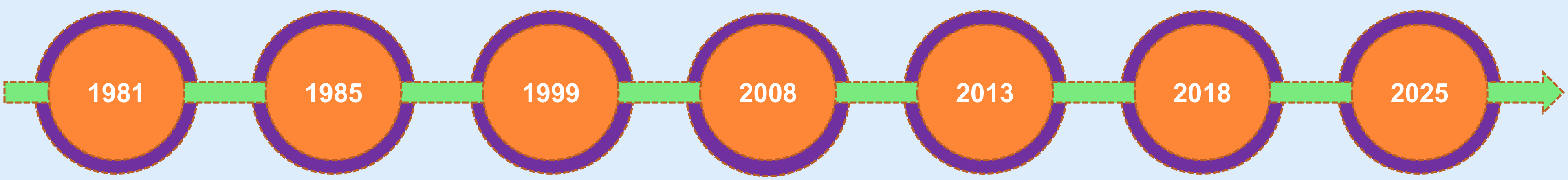


- Building blocks for SMPC protocols
  - Garbled Circuits
  - Private Set Intersection
  - Private Function Evaluation
- E-Voting
- Private Database Query
- E-commerce Transactions
- Location-based services

# History



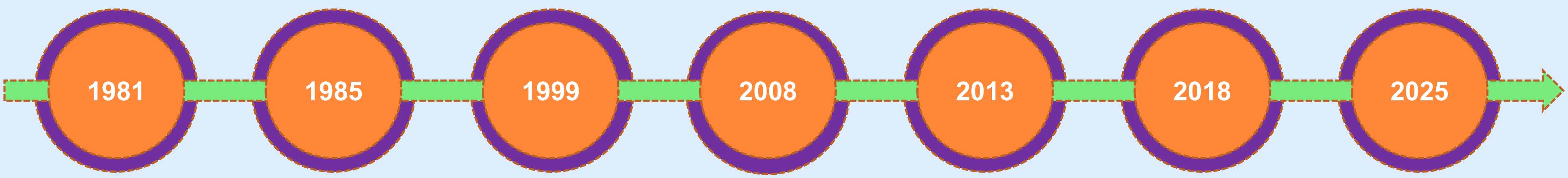
- First Idea of Oblivious Transfer [Robin 1981].
- The sender doesn't know if the receiver got the message.



# History



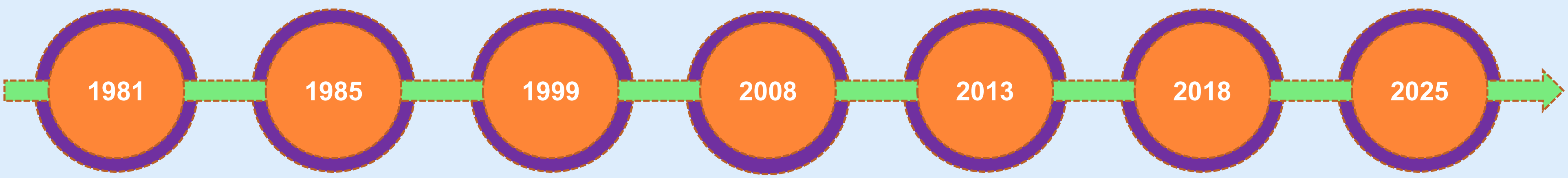
- Introduced 1-out-of-2 OT [Goldreich et al. 1985].
- Receiver can obtain one of two messages.



# History



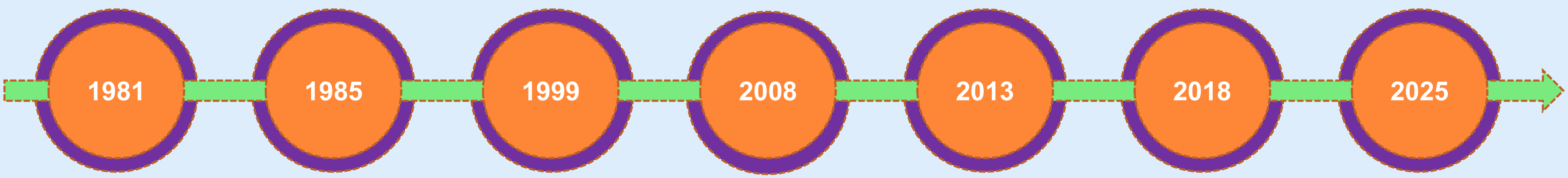
- Introduced efficient K-out-of-N OT protocol [Naor et al. 1999].



# History



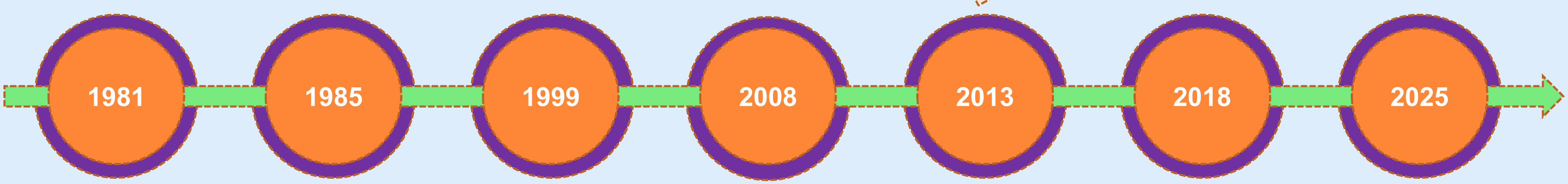
- Efficient K-out-of-N OT [Chu et al. 2008].
- Lowest communication cost in pairing free protocols.
- Based on ROM.



# History



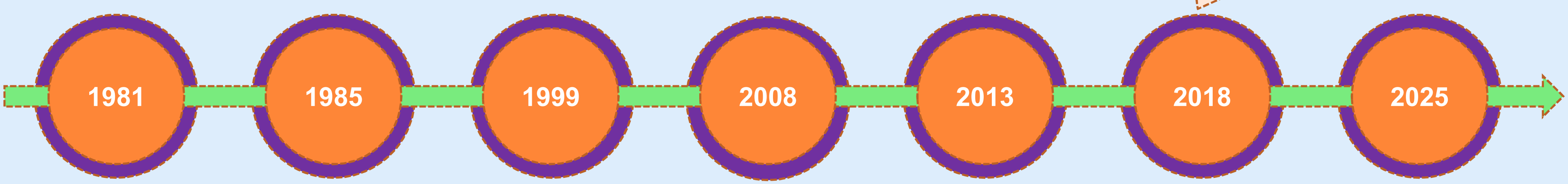
- Introduced first pairing based OT protocol [Guo et al 2013].
- Lower communication cost.
- Higher computation cost.
- Requires TTP.



# History



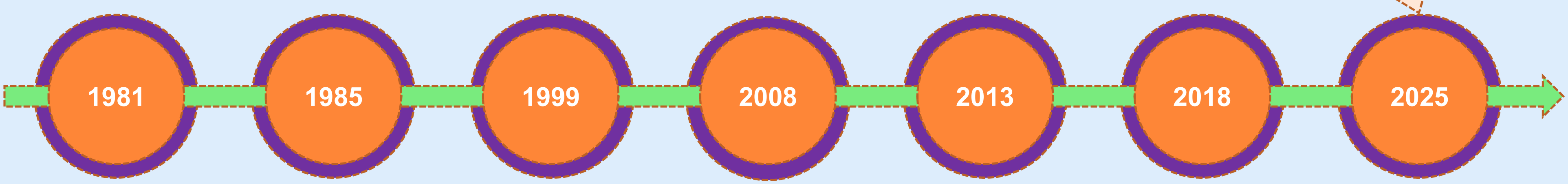
- Introduced lowest communication cost OT.[Lai et al. 2018]
- Based on pairing.
- High computation cost.
- Requires TTP.



# History



- Research on lower communication and computation cost.
- Extra features like adaptability, UC, multi receiver.



# Outline



- Introduction
- **K-out-of-N Oblivious Transfer**
- Conclusion

# Contribution



- Theoretical Contribution
  - Proposed two problems and proved their hardness via reduction:
    - $GBRSA \geqslant RSA$
    - $AGCDH \geqslant CDH$

- Practical Contribution
  - New Oblivious Transfer Protocols
    - Designed two new efficient OT schemes based on the hardness of new problems.
  - New RSA oblivious key exchange mechanism
    - Applied this technique to construct Scheme B.

# Generalized Blinded RSA (GBRSA)

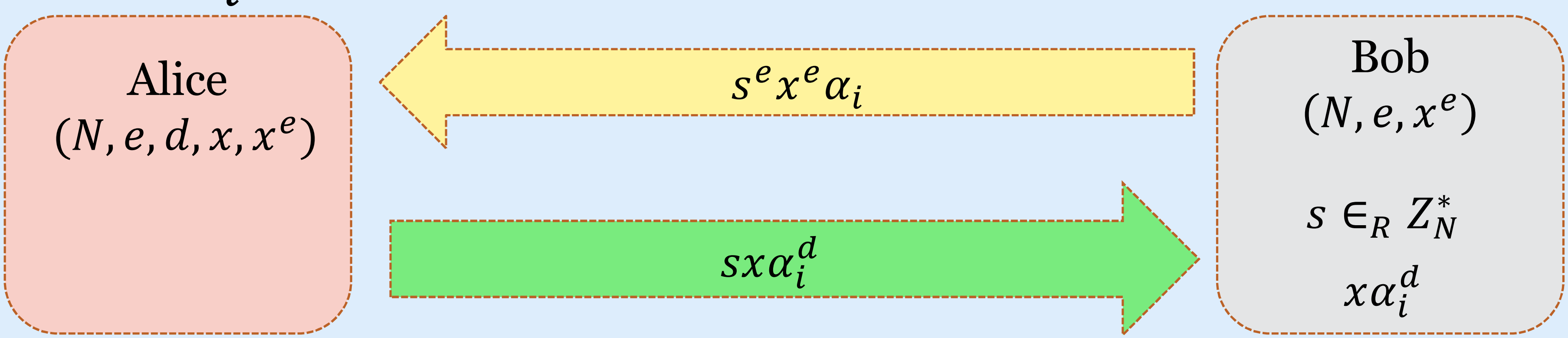


- Setup:
  - RSA with public key $(N, e)$
  - random non-identity elements $x, \beta_1, \beta_2, \dots, \beta_{k+1} \in_R Z_N^*$
- Given: $(N, e, x^e, \beta_1, \dots, \beta_{k+1}, x\beta_1^d, \dots, x\beta_k^d)$
- Goal: compute $x\beta_{k+1}^d$

# Oblivious Key Exchange Mechanism



- Setup:
  - $a, b, p = 2a + 1, q = 2b + 1$ are prime
  - RSA with public key $(N, e)$
  - $x \in Z_N^*$ such that $x^4 \not\equiv 1 \bmod N$
  - $\alpha_1, \dots, \alpha_n \in Z_N^*$
- Public parameters: $(N, e, x^e, \alpha_1, \dots, \alpha_n)$
- Goal: $\alpha_i^d x$



# Scheme B



Receiver

Sender

setup

$m_1, m_2, \ldots, m_n$

$C_i = m_i * x\alpha_i^d$

$\sigma_1, \sigma_2, \ldots, \sigma_k$

$s_1, s_2, \ldots, s_k \in_R \mathbb{Z}_N^*$

$A_i = \alpha_{\sigma_i} x^e s_i^e$

$A_1, A_2, \ldots, A_k$ →

$B_i = A_i^d$

← $B_1, \ldots, B_k, C_1, \ldots, C_n$

$D_i = s_i^{-1} * B_i$

$m_{\sigma_i} = C_{\sigma_i} * D_i^{-1}$

# Scheme A



**Receiver**

$\sigma_1, \sigma_2, \dots, \sigma_k$

$s_1, s_2, \dots, s_k \in_R \mathbb{Z}_q^*$

$A_i = (g^{\alpha_{\sigma_i}})^{s_i}$

**Sender**

setup

$m_1, m_2, \dots, m_n$

$r \in_R \mathbb{Z}_q^*$

$C_i = m_i * g^{r\alpha_i}$

$\xrightarrow{A_1, A_2, \dots, A_k}$

$B_i = A_i^r$

$\xleftarrow{B_1, \dots, B_k, C_1, \dots, C_n}$

$D_i = B_i^{-s_i^{-1}}$

$m_{\sigma_i} = C_{\sigma_i} * D_i$

# Features of Proposed Schemes



- No need for a third party.
- The lowest communication and computational cost among existing protocols.
- Only requires modular arithmetic.
- Supports offline precomputation.
- Supports "one-sender, multiple-receivers" model.
- The minimum possible number of rounds.
- Secure in the standard model.
- Adaptable.

| Scheme | Security Model / Assumption | Adv. Model | # rounds | Need TTP | Sender's comp | Receiver's comp. | Comm. |
|---|---|---|---|---|---|---|---|
| **[Chu 08]** | CDH-ROM | Malicious | 2 | ✗ | $(k+n).\mathrm{M_e} + 2n.\mathrm{H} + n.\mathrm{Enc}$ | $2k.\mathrm{M_e} + k.\mathrm{Inv} + k.\mathrm{Dec} + 2k.\mathrm{H}$ | $n+2k$ |
| **[Hsu 18]** | RSA-standard | Semi-honest | 3 | ✗ | $(n+k).\mathrm{M_e}$ | $2k.\mathrm{M_e}$ | $n+2k$ |
| **[Wu 03]** | DDH-standard | Semi-honest | 3 | ✗ | $(n+k).\mathrm{M_e} + 1.\mathrm{Inv}$ | $2k.\mathrm{M_e} + k.\mathrm{Inv}$ | $n+2k$ |
| **[Yang 21]** | CDH-ROM | Malicious | 3 | ✓ | $(2k+1).M_e + 2k.M_m + 2k.H + k.XOR + k.Inv$ | $(\mathrm{n}+2)M_e + \mathrm{n}.M_m + \mathrm{n.H} + \mathrm{n.XOR}$ | $\mathrm{n}+2\mathrm{k}+3$ |
| **[Wang 20]** | GFP-standard | Semi-honest | 2 | ✗ | $nH + nXOR + (20n + 20k + 20\log(s_S))A_A + (40n + 40k + 40\log(s_S))A_M$ | $kH + kXOR + (36 + 40k + 20k + 20\log(s_R))A_A + (70 + 80k + 40k + 40\log(s_R))A_M$ | $\mathrm{n}+20\mathrm{k}$ |
| **Scheme A** | CDH-standard | Semi-honest | 2 | ✗ | $(k+n).\mathrm{M_e} + n.\mathrm{M_m}$ | $2k.\mathrm{M_e} + k.\mathrm{Inv} + k.\mathrm{M_m}$ | $n+2k$ |
| **Scheme B** | RSA-standard | Semi-honest | 2 | ✗ | $(k+n).\mathrm{M_e} + n.\mathrm{M_m}$ | $k.\mathrm{M_e} + k.\mathrm{Inv} + 2k.\mathrm{M_m}$ | $n+2k$ |

# Implementation

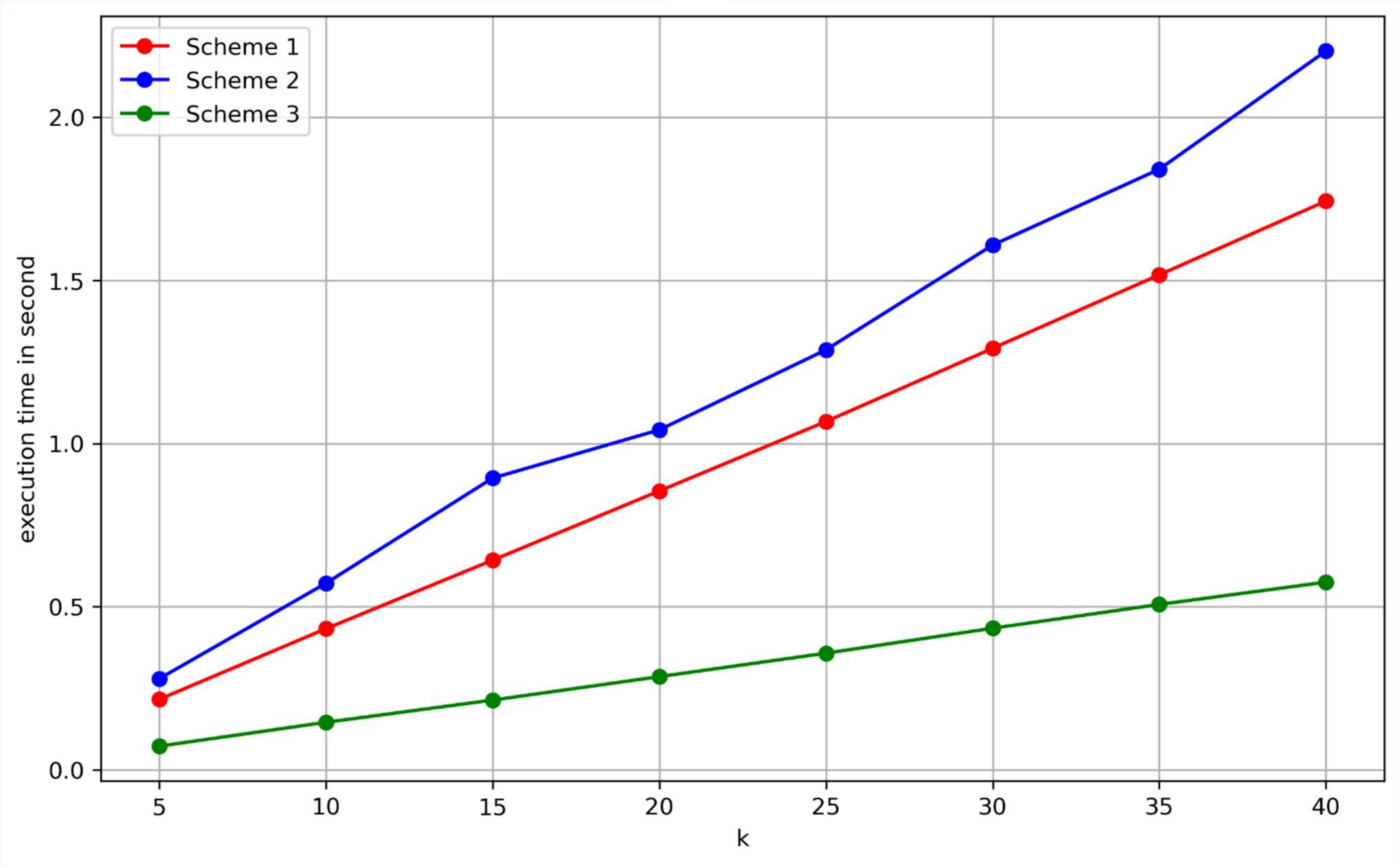


- Implementation of the proposed protocols using Python, leveraging the gmpy2 and SageMath libraries.
- Codes are available on GitHub.

| Scheme | Basis | Modulus Size |
| --- | --- | --- |
| Scheme A.1 | Based on a multiplicative group | 2048-bit modulus |
| Scheme A.2 | Based on elliptic curve cryptography | 224-bit modulus |
| Scheme B | Based on RSA | 2048-bit modulus |

# Effect of $k$ on the Execution Time of $(k, 45)$-OT



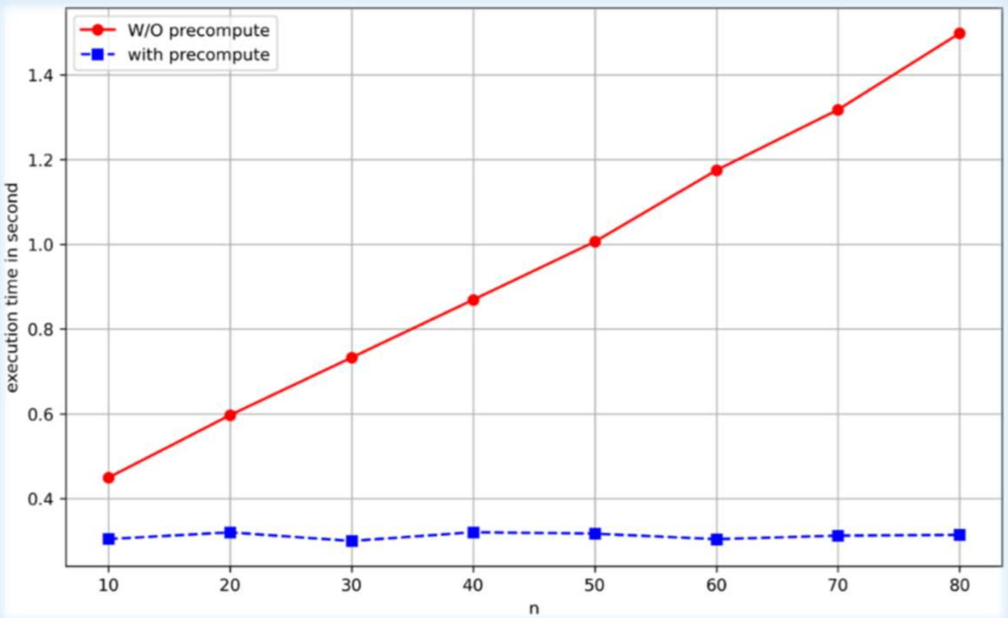


# Online vs. Offline Computation

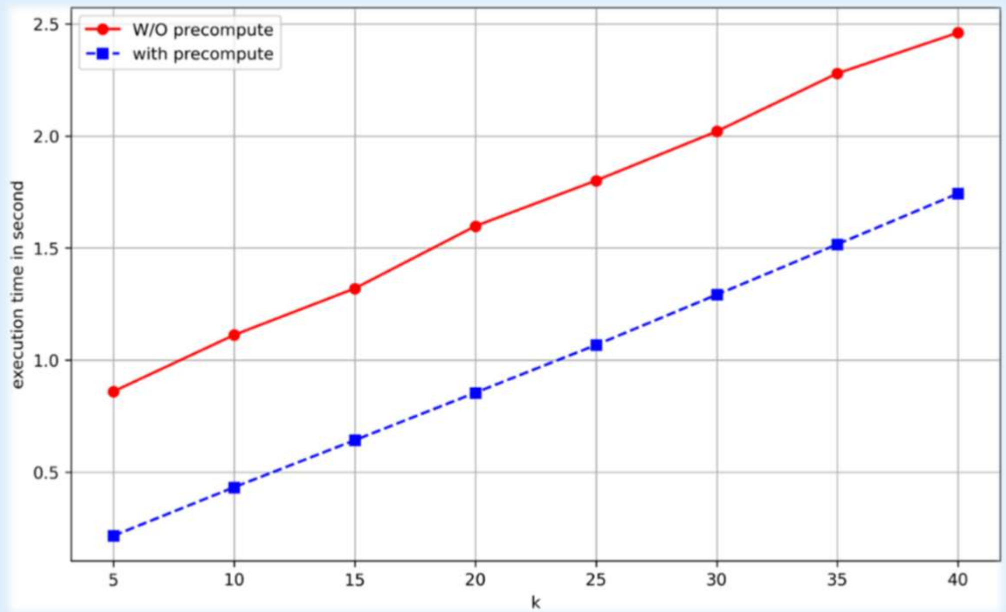


Scheme A.1

Effect of $n$ in $(7, n)$-OT




Effect of $k$ in $(k, 45)$-OT



# Outline



- Introduction
- K-out-of-N Oblivious Transfer
- Conclusion

# Conclusion



- Analysis of Prior Work & Limitations
  - Third-Party Requirement.
  - Performance Issues.
  - Security in the ROM.
  - Not suitable for resource constrained devices.
- Our Contribution
  - Proposed GBRSA & AGCDH, with hardness proofs.
  - New RSA-based oblivious key exchange mechanism used in Scheme B.
  - Two efficient k-out-of-N OT schemes.

# References



[Chu 08] Cheng-Kang Chu, Wen-Guey Tzeng, et al. Ef ficient k-out-of-n oblivious transfer schemes. J. Univers. Comput. Sci., 14(3):397–415, 2008.

[Hsu 18] Jen-Chieh Hsu, Raylin Tso, Yu-Chi Chen, and Mu-En Wu. Oblivious transfer protocols based on commutative encryption. In 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), pages 1–5. IEEE, 2018.

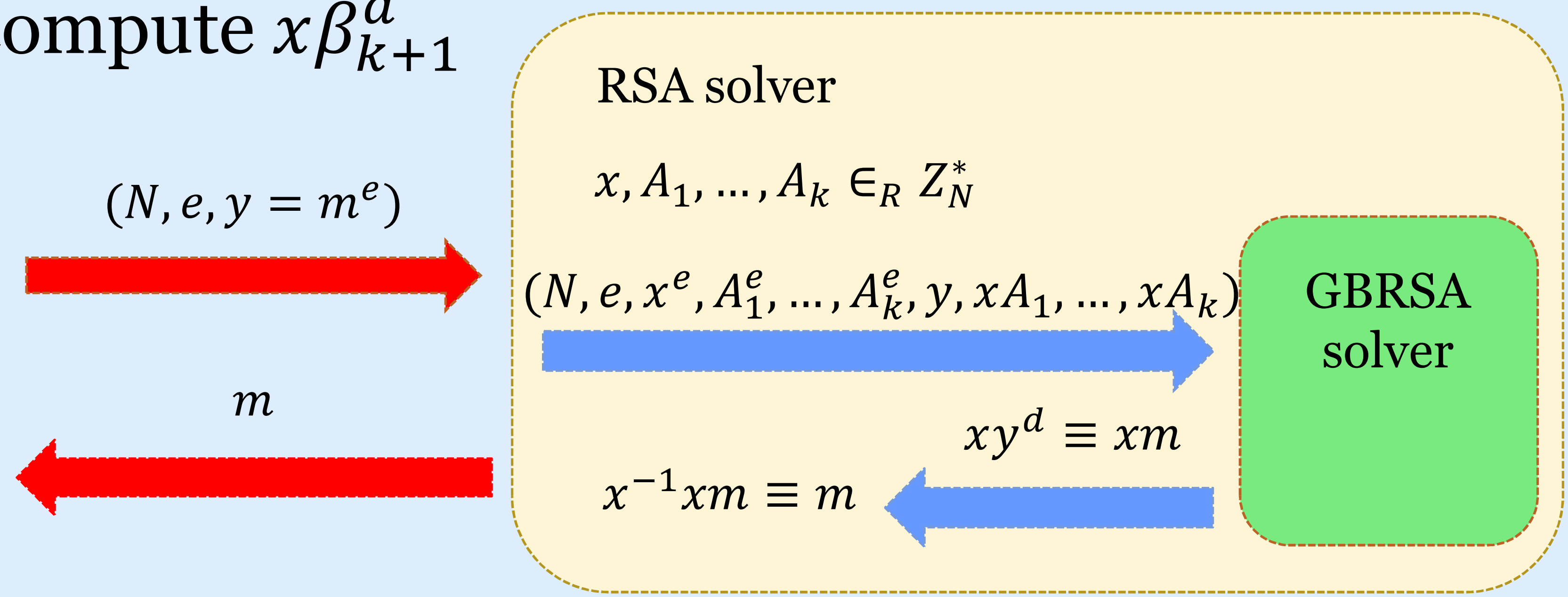
[Wang 20] Xianmin Wang, Xiaohui Kuang, Jin Li, Jing Li, Xiaofeng Chen, and Zheli Liu. Oblivious transfer for privacy-preserving in vanet's feature matching. IEEE transactions on intelligent transportation systems, 22(7):4359–4366, 2020.

[Yang 21] Huijie Yang, Jian Shen, Junqing Lu, Tianqi Zhou, Xueya Xia, and Sai Ji. A privacy-preserving data transmission scheme based on oblivious transfer and blockchain technology in the smart health care. Security and Communication Networks, 2021(1):5781354, 2021.

# Generalized Blinded RSA (GBRSA)



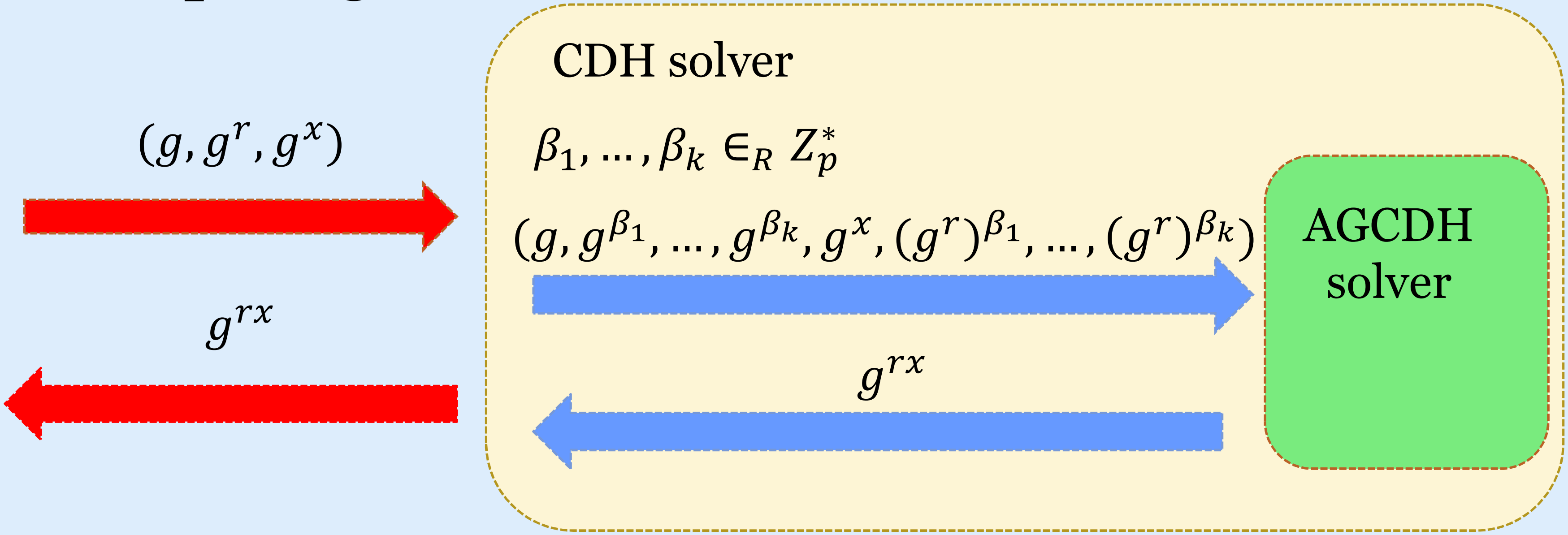
- Setup:
  - RSA with public key $(N, e)$
  - random non-identity elements $x, \beta_1, \beta_2, \ldots, \beta_{k+1} \in_R Z_N^*$
- Given: $(N, e, x^e, \beta_1, \ldots, \beta_{k+1}, x\beta_1^d, \ldots, x\beta_k^d)$
- Goal: compute $x\beta_{k+1}^d$



# Alternative Generalized Computational Diffie-Hellman (AGCDH)



- Setup:
  - Cyclic group $G$ of prime order $p$ with random generator $g$
  - $r, \alpha_1, \alpha_2, \ldots, \alpha_{k+1} \in_R \{0,1,\ldots,p-1\}$
- Given: $(g, g^{\alpha_1}, \ldots, g^{\alpha_{k+1}}, g^{r\alpha_1}, \ldots, g^{r\alpha_k})$
- Goal: Compute $g^{r\alpha_{k+1}}$



# Oblivious Key Exchange Mechanism



- Setup:
  - $a, b, p = 2a + 1, q = 2b + 1$ are prime
  - RSA with public key $(N, e)$
  - $x \in Z_N^*$ such that $x^4 \not\equiv 1 \bmod N$
  - $\alpha_1, \ldots, \alpha_n \in Z_N^*$
- Public parameters: $(N, e, x^e, \alpha_1, \ldots, \alpha_n)$
- Goal: $\alpha_i^d x$