Concretely-Efficient Multi-Key Homomorphic Secret Sharing And Applications

Kaiwen (Kevin) He, Sacha Servan-Schreiber, Geoffroy Couteau, Srini Devadas

ia.cr/2025/1803

github.com/kevin-he-01/mkhss

Summary of our contributions

Couteau et al., Eurocrypt 2025

Multi-Key Homomorphic Secret Sharing (**MKHSS**) was a **theoretical primitive**.

Key exchange take **minutes** if implemented.

Our work

Practical, **open-source** implementation of **MKHSS**.
Useful key exchange applications take **1–8 seconds**.

Our crucial simplification for efficiency could be of **independent interest**.

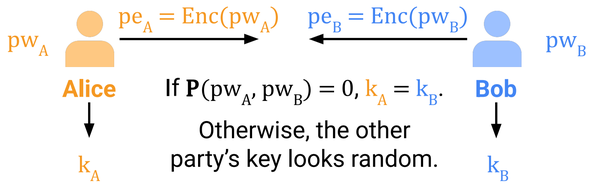
Application of MKHSS:

Attribute-based **non-interactive key exchange** (**ANIKE**)

Policy $\mathbf{P}$

$\mathbf{P}(\mathrm{pw_A}, \mathrm{pw_B}) = \mathbf{0}$ if $\mathrm{pw_A}$ and $\mathrm{pw_B}$ are **similar**

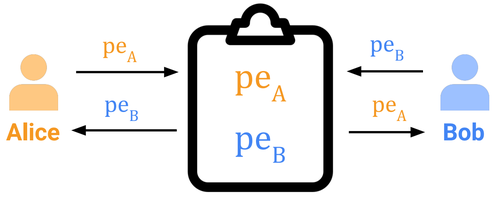
$\mathbf{P}(\mathrm{pw_A}, \mathrm{pw_B}) = \mathbf{1}$ otherwise



Attribute: $\mathrm{pw_A}$, $\mathrm{pw_B}$ (password or shared experience)

Non-interactive key exchange: à la Diffie-Hellman

Why non-interactivity?



Bulletin board

Alice and Bob can complete the key exchange **asynchronously**!

Non-interactivity rules out MPC-based solutions, which requires parties to be **simultaneously online**.

Benchmarking ANIKE

#1. pw is an L-word passphrase, tolerating:

Up to **T** incorrect words

AND

At most **Q** character substitutions per each of the remaining **L − T** words

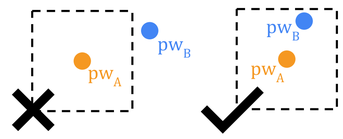
Matching example with **L = 4**, **T = 1**, **Q = 2**:

`correct horse battery staple`

`corrupt hose  buttery stable`

#2. pw is a geolocation coordinate

Tolerate small Chebyshev distance between geolocation coordinates



Evaluation: L-word passphrase

8-word passphrase tolerating:

- **T = 2** incorrect words
- **Q = 2** substitutions per correct word
- At most 9 characters per word
- Each character in 5 bits (latin alphabet)

	Couteau et. al	**Ours**	Our saving
Runtime (sec)	252	**7.56**	33×
Comm. (MB)	3.3	**1.1**	3×

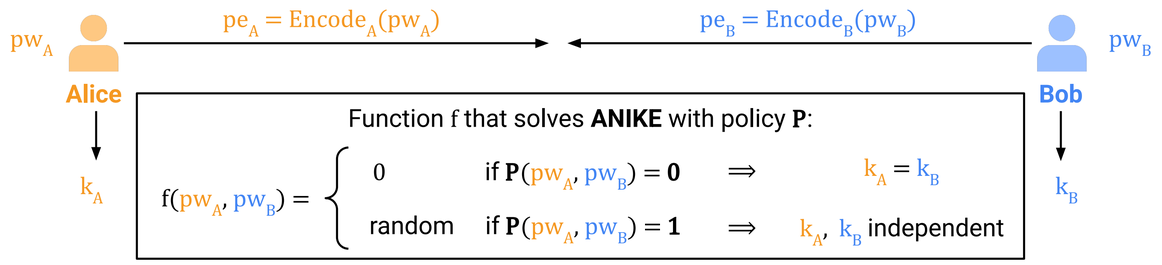
Evaluation: geolocation

2D coordinates with 32 bits of precision: under 1 cm error for latitude-longitude

	Couteau et. al	**Ours**	Our saving
Runtime (sec)	54.1	**1.65**	33×
Comm. (kB)	897.2	**301.1**	3×

Constructing ANIKE from MKHSS

MKHSS definition: compute **secret shares** of $\mathrm{f}(\mathrm{pw_A}, \mathrm{pw_B})$, i.e., $\mathrm{f}(\mathrm{pw_A}, \mathrm{pw_B}) = \mathrm{k_B} - \mathrm{k_A}$



Our crucial simplification for efficiency

Background

RMS Mult is invoked **repeatedly** to compute f

Subtractive secret share:

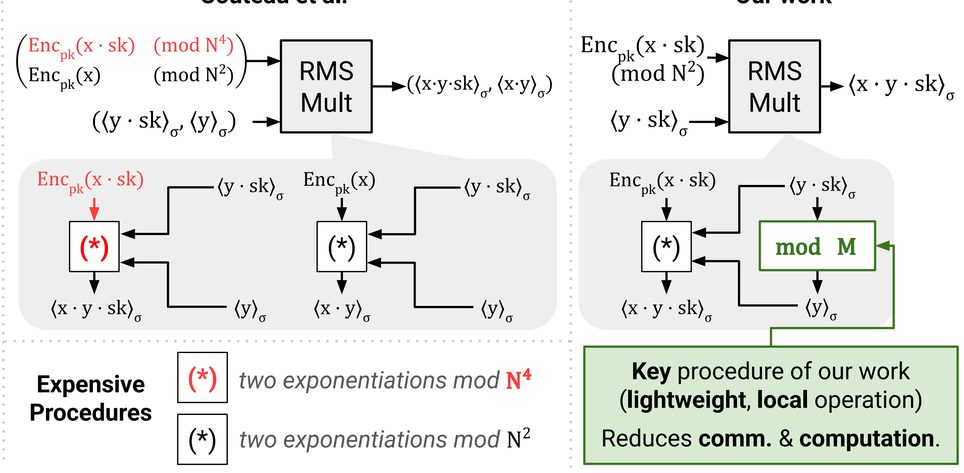
$\mathrm{x} = \langle \mathrm{x} \rangle_\mathrm{B} - \langle \mathrm{x} \rangle_\mathrm{A}$

Party identifier:

$\sigma \in \{\mathrm{A}, \mathrm{B}\}$

Protocol is symmetric

RSA modulus: $\mathrm{N} = \mathrm{pq}$



Main technical trick: locally truncating secret shares

Core observation

$\langle \mathrm{x} \rangle \bmod \mathrm{M}$ remains a valid subtractive secret share of x for sufficiently large M

Advantages of a shorter secret share

1. RMS multiplications are faster, since $\langle \mathrm{x} \rangle$ is the exponent in modular exponentiations.
2. Enables an additional trick:

$$\langle \mathrm{x} + \mathrm{z} \cdot \mathrm{M} \rangle \bmod \mathrm{M} \to \langle \mathrm{x} \rangle$$

This is used by the key procedure of our work to go from $\langle \mathrm{y} \cdot \mathrm{sk} \rangle$ to $\langle \mathrm{y} \rangle$

Example

$$\begin{array}{rr} & \langle \mathrm{x} \rangle_\mathrm{B} = 2748520001 \\ - & \langle \mathrm{x} \rangle_\mathrm{A} = 2748519999 \\ \hline & \mathrm{x} = 2 \end{array}$$

Each party locally truncates their share

$$\begin{array}{rr} & \langle \mathrm{x} \rangle_\mathrm{B} \bmod 100000 = 20001 \\ - & \langle \mathrm{x} \rangle_\mathrm{A} \bmod 100000 = 19999 \\ \hline & \mathrm{x} = 2 \end{array}$$

Legend

- Identical between parties with high probability
- Security parameter (128) bits of slack
- Upper bound on the bit length of x