# Revisiting OKVS-based OPRF and PSI:
## Cryptanalysis and Better Construction

Asiacrypt 2024

Kyoohyung Han[1] **Seongkwang Kim**[1]
Byeonghak Lee[1] Yongha Son[2]

[1]Samsung SDS, Seoul, Korea
[2]Sungshin Women's University, Seoul, Korea

# Overview

- Malicious attack on OKVS-based OPRFs and PSIs
    - We constructed practical overfitting algorithm for oblivious key-value store (OKVS)
    - We attacked VOLE-PSI framework [RS21] using the overfitting algorithm
- New OPRF based on SoftSpokenVOLE
    - We constructed Minicrypt OPRF and PSI based on SoftSpokenVOLE [Roy22]
    - It reduces the performance gap between Minicrypt PSI and LPN-based PSI

# OKVS Overfitting Attack

# Key-Value Store

| Key | Value |
|---|---|
| Bitcoin | 97,127 |
| Ethereum | 3,315 |
| Doge | 0.41 |
| ⋮ | ⋮ |

Key = “Bitcoin”

Value = 97,127

User

# Key-Value Store

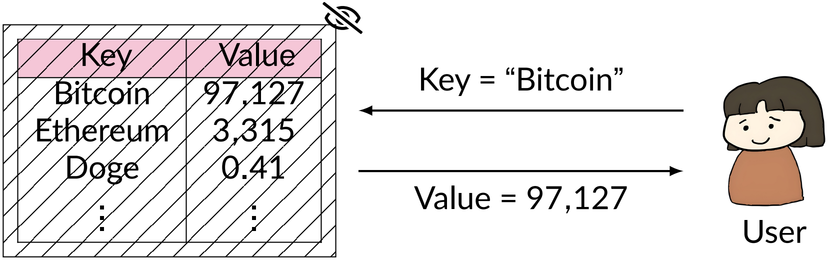
| Key | Value |
|---|---|
| Bitcoin | 97,127 |
| Ethereum | 3,315 |
| Doge | 0.41 |
| ⋮ | ⋮ |

Key = “Asiacrypt”

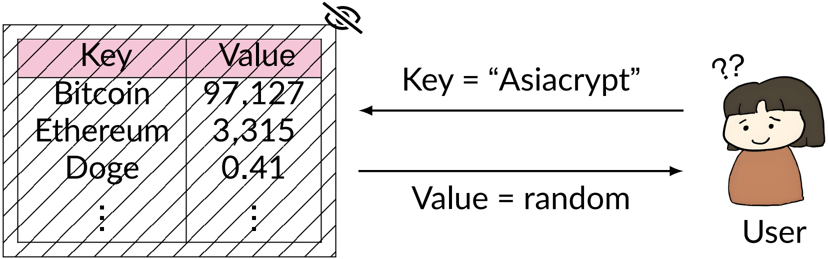
Value = $\perp$

User

# Oblivious Key-Value Store (OKVS)



# Oblivious Key-Value Store (OKVS)



# Applications of OKVS

- OPRF and PSI
- OPPRF and circuit-PSI
- Sparse OT extension

# Applications of OKVS

- OPRF and PSI
- OPPRF and circuit-PSI
- Sparse OT extension

# Recent OKVS Interface

OKVS.Ecd()

OKVS.Dcd()

# Recent OKVS Interface

OKVS.Ecd()

$$\{(x_i, y_i)\}$$

$$\Big\downarrow \text{Build}$$

$$\begin{bmatrix} -\mathsf{row}(x_1)- \\ -\mathsf{row}(x_2)- \\ \vdots \\ -\mathsf{row}(x_n)- \end{bmatrix} \cdot \begin{bmatrix} \\ P \\ \\ \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

$$\Big\downarrow \text{Solve}$$

OKVS $P \in \mathbb{F}_2^{m \times \ell}$

OKVS.Dcd()

$\mathsf{row} : \{0,1\}^* \to \mathbb{F}_2^m \quad (m = (1+\varepsilon)n)$

# Recent OKVS Interface

OKVS.Ecd()

$$\{(x_i, y_i)\}$$

$$\Big\downarrow \text{Build}$$

$$\begin{bmatrix} -\mathsf{row}(x_1)- \\ -\mathsf{row}(x_2)- \\ \vdots \\ -\mathsf{row}(x_n)- \end{bmatrix} \cdot \begin{bmatrix} \\ P \\ \\ \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

$$\Big\downarrow \text{Solve}$$

OKVS $P \in \mathbb{F}_2^{m \times \ell}$

OKVS.Dcd()

$$\begin{aligned} &\mathsf{Decode}(P, x) \\ &= \langle \mathsf{row}(x), P \rangle \\ &= \begin{cases} y_i & \text{if } x = x_i \text{ for some } i \\ \$ & \text{otherwise} \end{cases} \end{aligned}$$

$$\mathsf{row} : \{0,1\}^* \to \mathbb{F}_2^m \quad (m = (1+\varepsilon)n)$$

# OKVS Overfitting Problem

- $(n, n')$-OKVS overfitting game [GPRTY21]
  - If a PPT adversary $\mathcal{A}$ with random oracle $H : \{0,1\}^* \to \{0,1\}^\ell$ can make an OKVS $P$ such that the size of

    $$X = \{x | x \text{ is queried to } H\text{, and Decode}(P, x) = H(x)\}$$

    is larger than $n'$, $\mathcal{A}$ wins the game

# OKVS Overfitting Problem

- $(n, n')$-OKVS overfitting game [GPRTY21]
  - If a PPT adversary $\mathcal{A}$ with random oracle $H : \{0,1\}^* \to \{0,1\}^\ell$ can make an OKVS $P$ such that the size of

    $$X = \{x | x \text{ is queried to } H, \text{ and Decode}(P, x) = H(x)\}$$

    is larger than $n'$, $\mathcal{A}$ wins the game
- Information-theoretic bound [PRTY20]
  - For $n' = 2m$, $\ell$ is roughly $2\lambda - \log m$
  - $n' \leq m$ cannot be accomplished

# OKVS Overfitting Problem

- $(n, n')$-OKVS overfitting game [GPRTY21]
  - If a PPT adversary $\mathcal{A}$ with random oracle $H : \{0,1\}^* \to \{0,1\}^\ell$ can make an OKVS $P$ such that the size of

    $$X = \{x|x \text{ is queried to } H\text{, and Decode}(P, x) = H(x)\}$$

    is larger than $n'$, $\mathcal{A}$ wins the game
- Information-theoretic bound [PRTY20]
  - For $n' = 2m$, $\ell$ is roughly $2\lambda - \log m$
  - $n' \leq m$ cannot be accomplished
- Computational hardness? Unknown [GPRTY21]

# OKVS Overfitting Problem

- $(n, n')$-OKVS overfitting game [GPRTY21]
  - If a PPT adversary $\mathcal{A}$ with random oracle $H : \{0,1\}^* \to \{0,1\}^\ell$ can make an OKVS $P$ such that the size of

$$X = \{x | x \text{ is queried to } H, \text{ and Decode}(P, x) = H(x)\}$$

    is larger than $n'$, $\mathcal{A}$ wins the game
- Information-theoretic bound [PRTY20]
  - For $n' = 2m$, $\ell$ is roughly $2\lambda - \log m$
  - $n' \leq m$ cannot be accomplished
- Practical algorithm? This work!

# OKVS Overfitting Attack

- We want to find $\{x_1, \ldots, x_{n'}\}$ and $P \in \mathbb{F}_2^{m \times \ell}$ such that

$$\begin{bmatrix} -\text{row}(x_1)- \\ -\text{row}(x_2)- \\ \vdots \\ -\text{row}(x_{n'})- \end{bmatrix} \cdot \begin{bmatrix} \\ P \\ \\ \end{bmatrix} = \begin{bmatrix} H(x_1) \\ H(x_2) \\ \vdots \\ H(x_{n'}) \end{bmatrix}$$

# OKVS Overfitting Attack

- We want to find $\{x_1, \ldots, x_{n'}\}$ and $P \in \mathbb{F}_2^{m \times \ell}$ such that

$$\begin{bmatrix} -\text{row}(x_1)- \\ -\text{row}(x_2)- \\ \vdots \\ -\text{row}(x_{n'})- \end{bmatrix} \cdot \begin{bmatrix} \\ P \\ \\ \end{bmatrix} = \begin{bmatrix} H(x_1) \\ H(x_2) \\ \vdots \\ H(x_{n'}) \end{bmatrix}$$

- row$()$ looks like (e.g., [RS21]):

$$[\underbrace{0 \ldots 010 \ldots 010 \ldots 0}_{\text{two 1s (dim } m)} \quad \underbrace{1011 \ldots 0101}_{\text{dense (dim } d \approx 40)}]$$

# Our Attack

# Our Attack

1. Bucketize $Q$ items with respect to the sparse part

$$x_1 \xrightarrow{\text{row()}} [100100 \quad \ldots] \longrightarrow \text{Bucket } B_{1,4}$$

$$x_2 \xrightarrow{\text{row()}} [010001 \quad \ldots] \longrightarrow \text{Bucket } B_{2,6}$$

# Our Attack

1. Bucketize $Q$ items with respect to the sparse part

$$x_1 \xrightarrow{\text{row()}} [100100 \quad \ldots] \longrightarrow \text{Bucket } B_{1,4}$$

$$x_2 \xrightarrow{\text{row()}} [010001 \quad \ldots] \longrightarrow \text{Bucket } B_{2,6}$$

2. Build a singular $k \times k$-binary matrix with row weight 2

$$\begin{array}{cc} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} & \begin{array}{l} B_{1,2} \\ B_{2,3} \\ B_{3,4} \\ B_{1,4} \end{array} \end{array}$$

# Our Attack

1. Bucketize $Q$ items with respect to the sparse part

$$x_1 \xrightarrow{\text{row()}} [100100 \quad \ldots] \longrightarrow \text{Bucket } B_{1,4}$$

$$x_2 \xrightarrow{\text{row()}} [010001 \quad \ldots] \longrightarrow \text{Bucket } B_{2,6}$$

2. Build a singular $k \times k$-binary matrix with row weight 2

$$\begin{array}{cc} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} & \begin{array}{c} B_{1,2} \\ B_{2,3} \\ B_{3,4} \\ B_{1,4} \end{array} \end{array}$$

3. Solve $k$-XOR problem (next slides)

# Our Attack

1. Choose buckets

$$\left[\begin{array}{c:c} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} \begin{array}{l} B_{1,2} \\ B_{2,3} \\ B_{3,4} \\ B_{1,4} \end{array} & \\ \text{row}(X) & \end{array}\right] \cdot \begin{bmatrix} P \end{bmatrix} = \begin{bmatrix} H(X) \end{bmatrix}$$

# Our Attack

## 2. Solve $k$-XOR

$\sum_i (\text{row}(x_i) \| H(x_i)) = 0$

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} \begin{array}{l} B_{1,2} \ni x_1 \\ B_{2,3} \ni x_2 \\ B_{3,4} \ni x_3 \\ B_{1,4} \ni x_4 \end{array}$$

$$\begin{bmatrix} \text{row}(X) \end{bmatrix} \cdot \begin{bmatrix} P \end{bmatrix} = \begin{bmatrix} H(X) \end{bmatrix}$$

Complexity

$O\left(2^{\frac{d+\ell}{1+\lfloor \log k \rfloor}}\right)$

# Our Attack

3. Collect such inputs

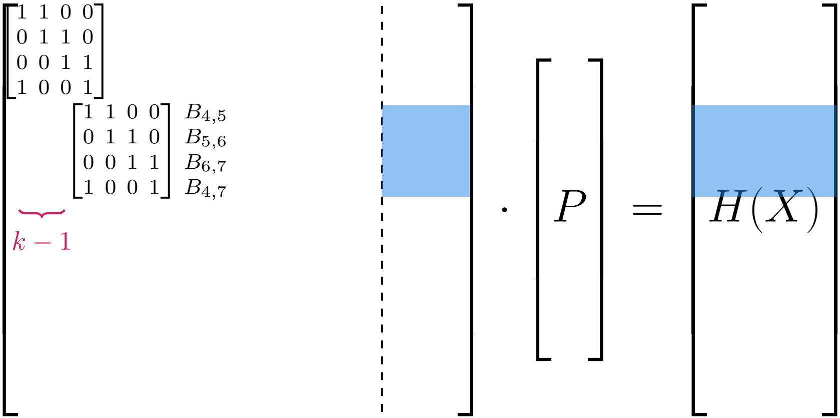
$$L \leftarrow L \cup \{x_1, x_2, x_3, x_4\}$$

$$\begin{array}{cc} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} & \begin{array}{c} B_{1,2} \\ B_{2,3} \\ B_{3,4} \\ B_{1,4} \end{array} \end{array}$$

$$\begin{bmatrix} \text{row}(X) \end{bmatrix} \cdot \begin{bmatrix} P \end{bmatrix} = \begin{bmatrix} H(X) \end{bmatrix}$$

# Our Attack

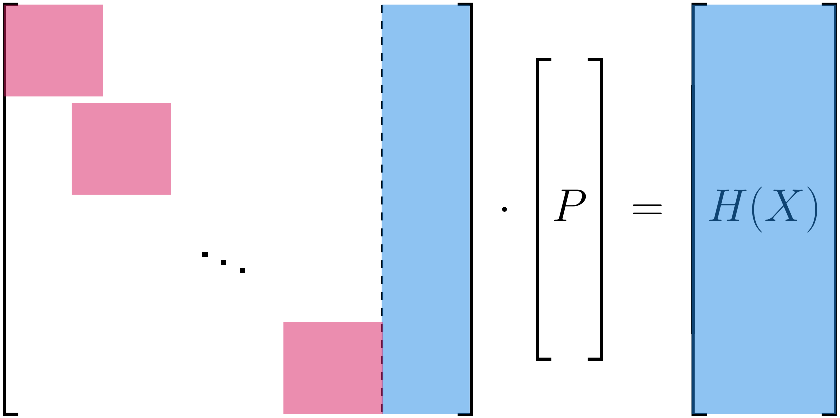
4. Repeat



# Our Attack

5. Solve w.r.t. $P$

$\text{rank}(\text{row}(X)) = \text{rank}(\text{row}(X) \| H(X))$



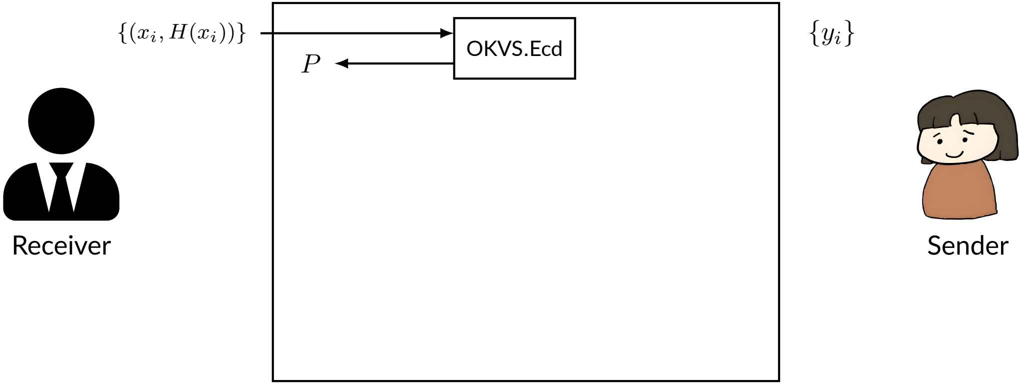
# Efficacy of the Attack

- Malicious user can encode more than permitted
  - Encoding $\frac{km}{k-1}$ items requires $O(m^2 2^{\frac{d+\ell}{1+\lfloor \log k \rfloor}})$ time
  - (PaXoS with $n = 2^{20}$, $\ell = 128$) Encoding $3.2n$ items in $2^{99}$ time

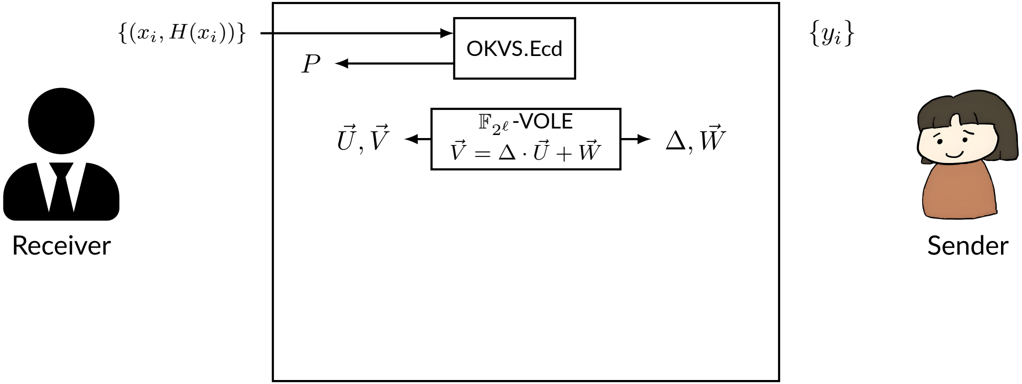
# Efficacy of the Attack

- Malicious user can encode more than permitted
  - Encoding $\frac{km}{k-1}$ items requires $O(m^2 2^{\frac{d+\ell}{1+\lfloor \log k \rfloor}})$ time
  - (PaXoS with $n = 2^{20}$, $\ell = 128$) Encoding $3.2n$ items in $2^{99}$ time
- This attack can be utilized to OPRF and PSI
  - VOLE-PSI [RS21] = OKVS + VOLE
  - [RS21] originally claimed $\ell = 128$ achieves $n' = m$
  - Overfitting OKVS reveals PRF values of overly many items

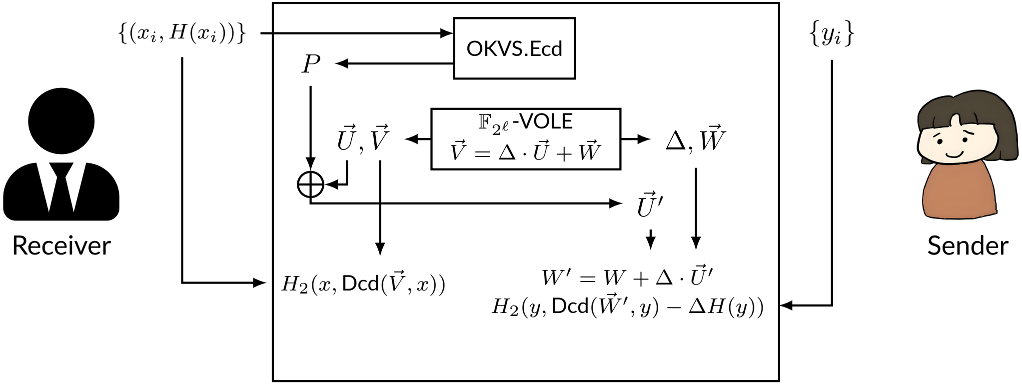
# VOLE-PSI [RS21]



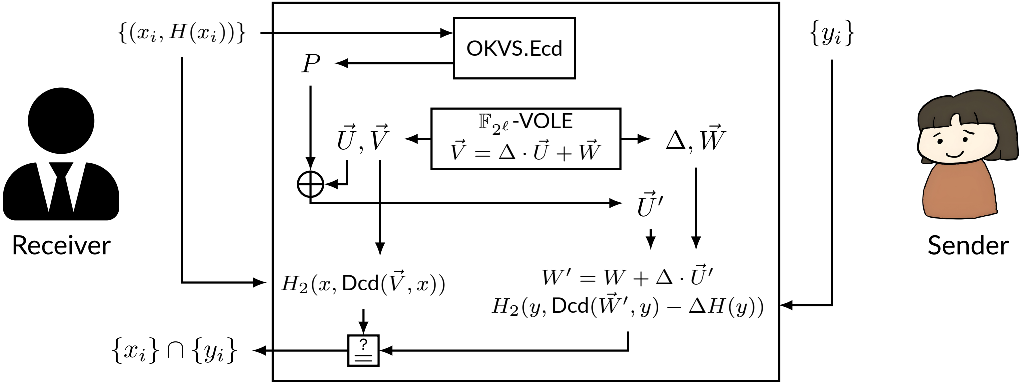
# VOLE-PSI [RS21]



# VOLE-PSI [RS21]



# VOLE-PSI [RS21]



# Effect on OPRF and PSI

In $2^{128}$ time,

# Effect on OPRF and PSI

In $2^{128}$ time,

- OPRF: a corrupt receiver can know
  - (RS21) $4.8n$ random PRF evaluations
  - (RR22) $1.26n$ random PRF evaluations
  - (BPSY23) $1.23n$ random PRF evaluations

# Effect on OPRF and PSI

In $2^{128}$ time,

- OPRF: a corrupt receiver can know
  - (RS21) $4.8n$ random PRF evaluations
  - (RR22) $1.26n$ random PRF evaluations
  - (BPSY23) $1.23n$ random PRF evaluations
- PSI: a corrupt receiver can know membership of
  - (RS21) $2.1n$ random items + $n$ chosen items
  - (RR22) $0.237n$ random items + $n$ chosen items

# Effect on OPRF and PSI

In $2^{128}$ time,

- OPRF: a corrupt receiver can know
  - (RS21) $4.8n$ random PRF evaluations
  - (RR22) $1.26n$ random PRF evaluations
  - (BPSY23) $1.23n$ random PRF evaluations
- PSI: a corrupt receiver can know membership of
  - (RS21) $2.1n$ random items + $n$ chosen items
  - (RR22) $0.237n$ random items + $n$ chosen items
- Even non-membership information can be a leakage!

# Effect on OPRF and PSI

In $2^{128}$ time,

- OPRF: a corrupt receiver can know
  - (RS21) $4.8n$ random PRF evaluations
  - (RR22) $1.26n$ random PRF evaluations
  - (BPSY23) $1.23n$ random PRF evaluations
- PSI: a corrupt receiver can know membership of
  - (RS21) $2.1n$ random items + $n$ chosen items
  - (RR22) $0.237n$ random items + $n$ chosen items
- Even non-membership information can be a leakage!
- Find our mitigations in the paper!

# New Minicrypt OPRF

# VOLE Types

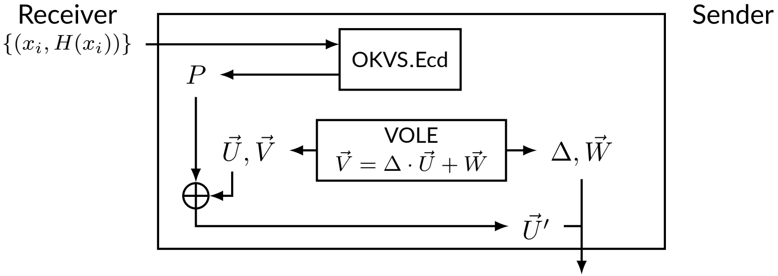
# VOLE Types

- Silent VOLE
  - Used in recent PSI protocols
  - Efficient even for large fields
  - Structured (dual) LPN assumption

# VOLE Types

- Silent VOLE
  - Used in recent PSI protocols
  - Efficient even for large fields
  - Structured (dual) LPN assumption
- SoftSpokenVOLE [Roy22]
  - Minicrypt assumption
  - Only efficient for small fields
  - This work: SoftSpokenVOLE + VOLE-PSI $\rightarrow$ Minicrypt OPRF

# OPRF in VOLE-PSI



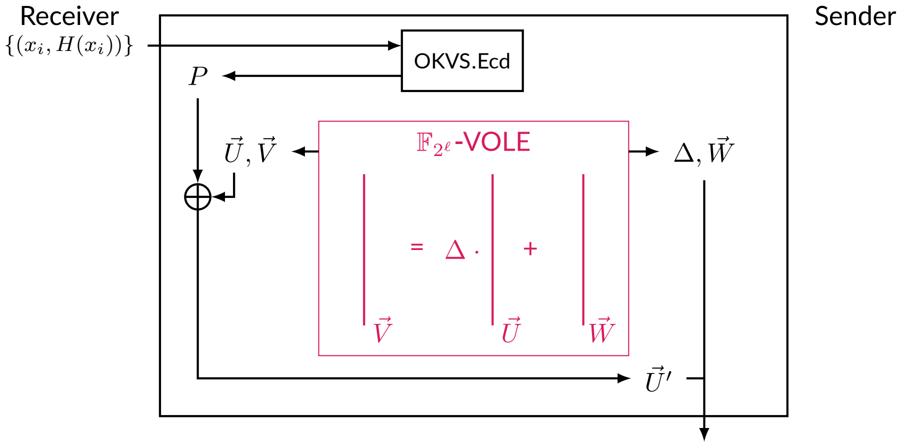

(Locally) compute

$H_2(x_i, \mathsf{Dcd}(\vec{V}, x_i))$

Define $\vec{W}' := \vec{W} + \Delta \cdot \vec{U}'$

and $F_\Delta(x) := H_2(x, \mathsf{Dcd}(\vec{W}', x) - \Delta \cdot H(x))$

# OPRF in VOLE-PSI



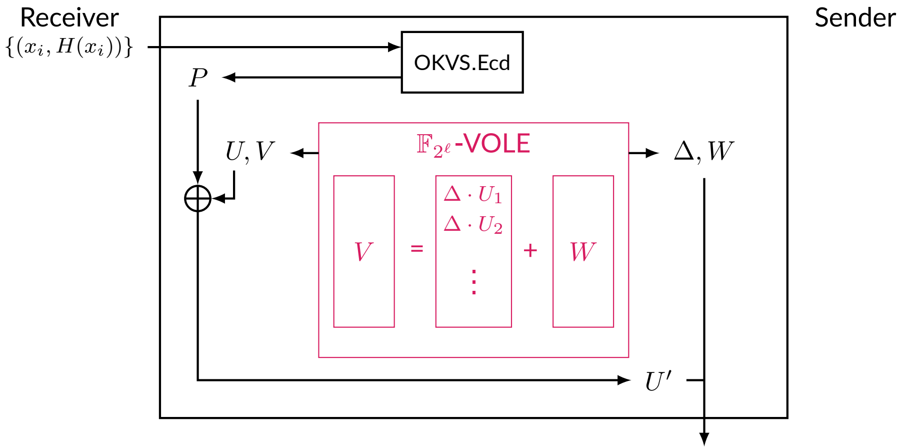

(Locally) compute

$H_2(x_i, \mathsf{Dcd}(\vec{V}, x_i))$

Define $\vec{W}' := \vec{W} + \Delta \cdot \vec{U}'$

and $F_\Delta(x) := H_2(x, \mathsf{Dcd}(\vec{W}', x) - \Delta \cdot H(x))$

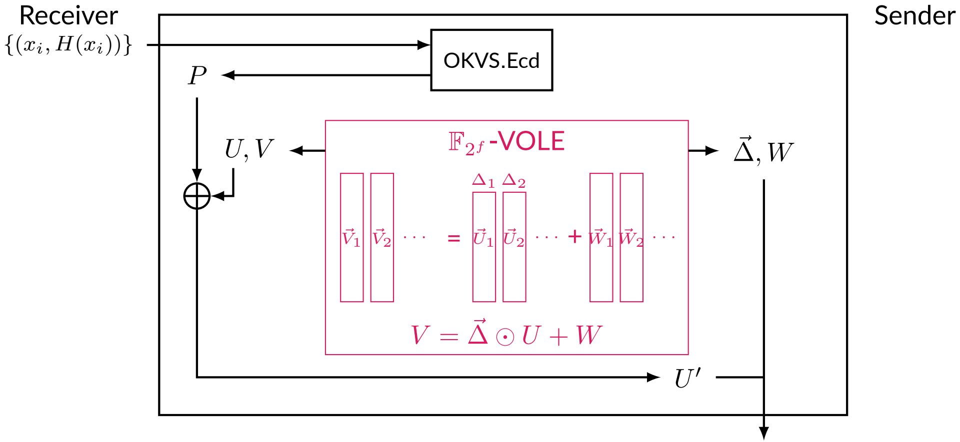
# OPRF in VOLE-PSI




(Locally) compute
$H_2(x_i, \mathsf{Dcd}(V, x_i))$

Define $W' := W + \Delta \cdot U'$
and $F_\Delta(x) := H_2(x, \mathsf{Dcd}(W', x) - \Delta \cdot H(x))$

# VOLE $\to$ SoftSpoken-VOLE



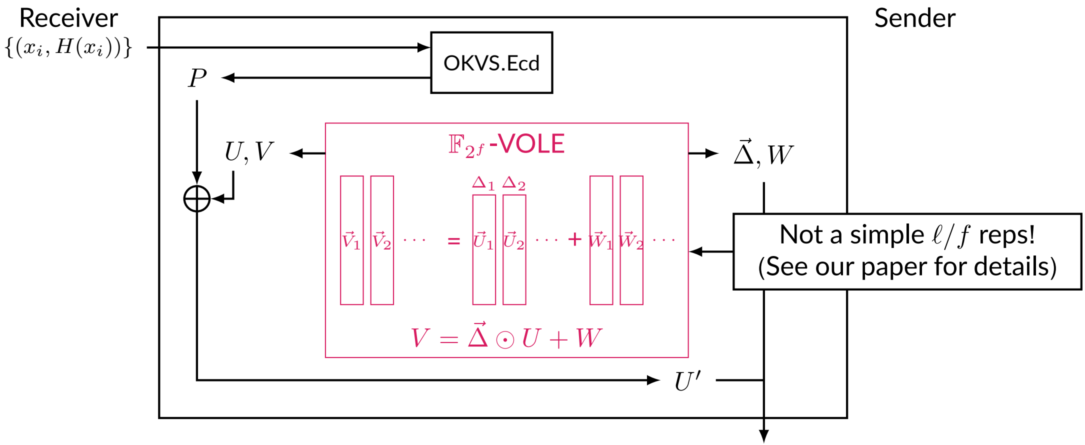

(Locally) compute

$H_2(x_i, \mathsf{Dcd}(V, x_i))$

Define $W' := W + \vec{\Delta} \odot U'$

and $F_\Delta(x) := H_2(x, \mathsf{Dcd}(W', x) - \vec{\Delta} \odot H(x))$

# VOLE $\to$ SoftSpoken-VOLE




(Locally) compute
$H_2(x_i, \mathsf{Dcd}(V, x_i))$

Define $W' := W + \vec{\Delta} \odot U'$
and $F_\Delta(x) := H_2(x, \mathsf{Dcd}(W', x) - \vec{\Delta} \odot H(x))$

# Performance

| $n = 2^{20}$ OPRFs | | Silent | Ours $(f=6)$ | [PRTY20] $(f=1)$ |
|---|---|---|---|---|
| Comm. (MB) | | 22.8 | 32.7 | 93.6 |
| Time (sec) | 5Gbps | 0.98 | 2.74 | 2.03 |
| | 100Mbps | 4.65 | 7.78 | 14.2 |
| Assumption | | dual-LPN | Minicrypt | |

- Previous best Minicrypt [PRTY20] : '$f = 1$' of ours

  (with minor differences)

- Narrow the gap between Minicrypt & LPN-based one!

Thank you!

# References


[PRTY20] B. Pinkas, M. Rosulek, N. Trieu, and A. Yanai. *PSI from PaXoS: Fast, Malicious Private Set Intersection*. Eurocrypt 2020.

[RS21] P. Rindal, and P. Schoppmann. *VOLE-PSI: Fast OPRF and Circuit-PSI from Vector-OLE*. Eurocrypt 2021.

[GPRTY21] G. Garimella, B. Pinkas, M. Rosulek, N. Trieu, and A. Yanai. *Oblivious Key-Value Stores and Amplification for Private Set Intersection*. Crypto 2021.

[Roy22] L. Roy. *SoftSpokenOT: Communication–Computation Tradeoffs in OT Extension*. Crypto 2022.

[RR22] S. Raghuraman, and P. Rindal. *Blazing Fast PSI from Improved OKVS and Subfield VOLE*. ACM CCS 2022.

[BPSY23] A. Bienstock, S. Patel, J. Seo, and K. Yeo. *Near-Optimal Oblivious Key-Value Stores for Efficient PSI, PSU and Volume-Hiding Multi-Maps*. USENIX Security 2023.

[RRT23] S. Raghuraman, P. Rindal, T. Tanguy. *Expand-Convolute Codes for Pseudorandom Correlation Generators from LPN*. CRYPTO 2023.

# Acknowledgment

- We appreciate Peter Rindal for valuable comments.
- Some illustrations were created using fontawesome5 (`https://fontawesome.com/`) free version latex package.