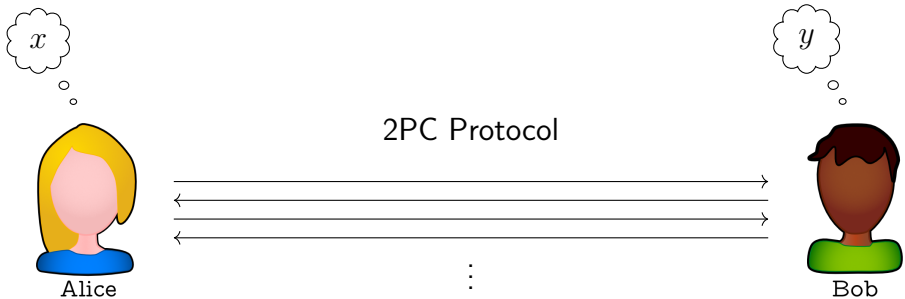
Communication-Efficient Secure Two-Party Computation From Minimal Assumptions

Lawrence Roy

Oregon State University

Secure Two-Party Computation (2PC)



Output:
$f(x, y)$

Cryptographic Assumptions

	Symmetric Key	**Public Key**
Assumptions:	One-way function, Pseudo-random function, Correlation-robust hash.	Diffie–Hellman, RSA, Learning Parity with Noise (LPN), Learning With Errors (LWE).
Common Instantiations:	AES, SHA, BLAKE, ...	Curve25519, RSA. PQC and FHE standards in progress.
Idealized models:	Random oracle.	Generic group.

Cost Tradeoffs

For a circuit with m inputs, n gates, and depth d:

	Costs			
	Computation	Communication (bits)	Rounds	Assumption
Secret Sharing	Low	$\sim 4n$	O(d)	None (+ OTs)
Garbled Circuits	Medium	$\sim 200n$	O(1)	Correlation-robust hash (+ OTs)
Fully Homomorphic Encryption	High	$O(m)$	O(1)	LWE

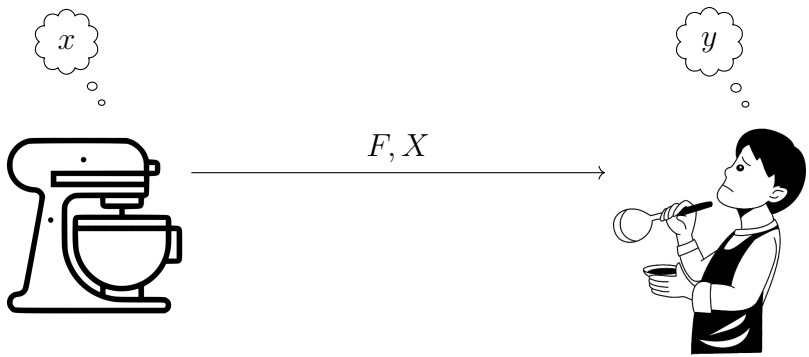
SecureML[1]

- 2PC technique: hybrid of secret sharing and garbled circuits.
- LAN connection: network delay 0.17 ms, bandwidth 1 GB/s.
- MNIST data set: $60\,000$ handwriting samples, 28×28 pixels each.
- NN architecture: two hidden layers of 128 neurons each, fully connected.
- Online runtime: 1.2 h. Mostly from garbled circuits.
- Offline precomputation: 80 h. Mostly from oblivious transfer.

[1] Mohassel and Zhang, "SecureML: A System for Scalable Privacy-Preserving Machine Learning", 2017.

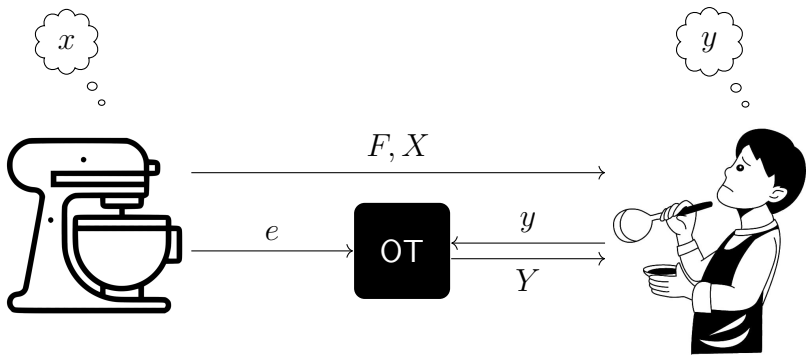
Garbling Overview

How to evaluate $f(x, y)$ when x and y are secret?



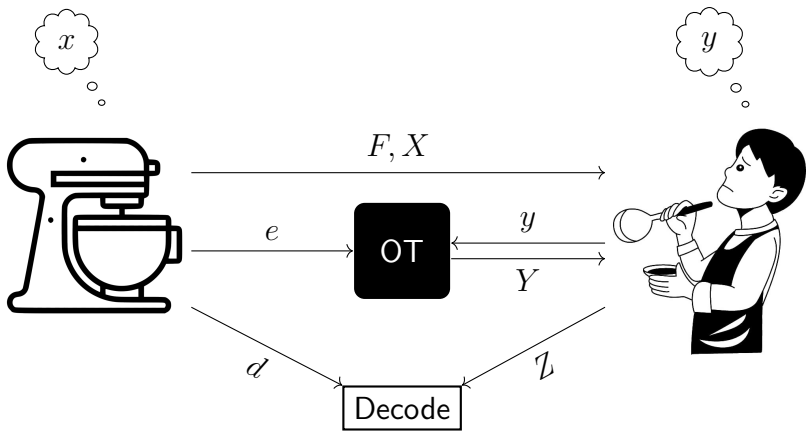
Garbling Overview

How to evaluate $f(x, y)$ when x and y are secret?



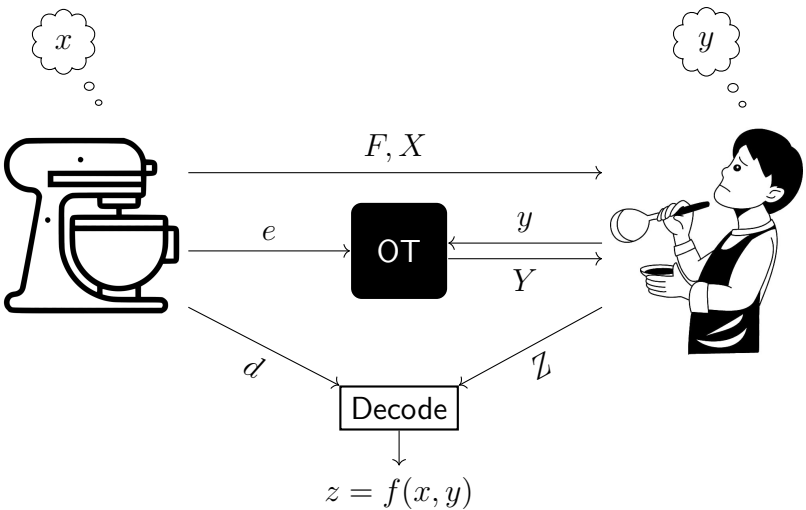
Garbling Overview

How to evaluate $f(x, y)$ when x and y are secret?



Garbling Overview

How to evaluate $f(x, y)$ when x and y are secret?



Oblivious Transfer (OT)

m_0, m_1

OT

x^*

m_{x^*}

Random Oblivious Transfer (OT)

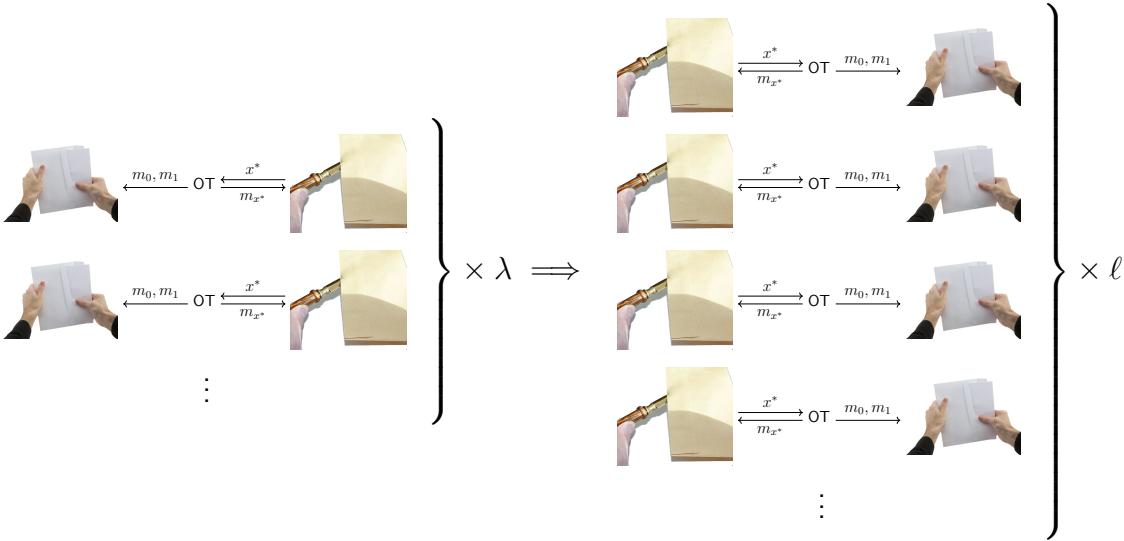
m_0, m_1

OT

x^*

m_{x^*}

OT Extension



$3/2$ Garbling[2]

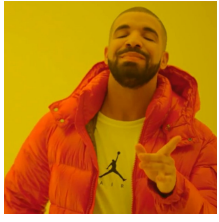
- New garbling techniques: 🔪 & 🎲.
- Compatible with free XOR.
- AND gates cost $\frac{3}{2}\lambda + O(1)$ bits — a 25% improvement.
- Garbling now takes 6 hash evaluations, 50% worse.

[2] Rosulek and Roy, "Three Halves Make a Whole? Beating the Half-Gates Lower Bound for Garbled Circuits", 2021. Honorable Mention for Best Paper, Crypto 2021.

Half-Gates

For evaluating a garbled gate on input (A, B):

$$H(A, B)$$

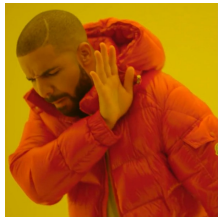


$$H(A) \oplus H(B)$$

Compression: 3 ciphertexts $\Rightarrow$ 2 ciphertexts (33% improvement).

Slicing

For evaluating a garbled gate on input (A, B):



$$H(A) \oplus H(B)$$

$$\begin{bmatrix} H(A) \oplus H(A \oplus B) \\ H(B) \oplus H(A \oplus B) \end{bmatrix}$$

Compression: 2 ciphertexts $\Rightarrow 1.5$ ciphertexts (25% improvement).

SoftSpokenOT: Comparison

	Assumptions	Computation	Communication (bits)
IKNP[3]	CR Hash	2ℓ PRG bits	$\lambda\ell$
Silent OT[4]	LPN	Syndrome Evaluation	$\Theta(\lambda \log(\ell))$
SoftSpokenOT[5]	CR Hash	$2^k\ell/k$ PRG bits	$\lambda\ell/k$

[3] Ishai et al., "Extending Oblivious Transfers Efficiently", 2003.

[4] Boyle et al., "Compressing Vector OLE", 2018.

[5] Roy, *SoftSpokenOT: Communication–Computation Tradeoffs in OT Extension*, 2022. Submitted to Crypto.

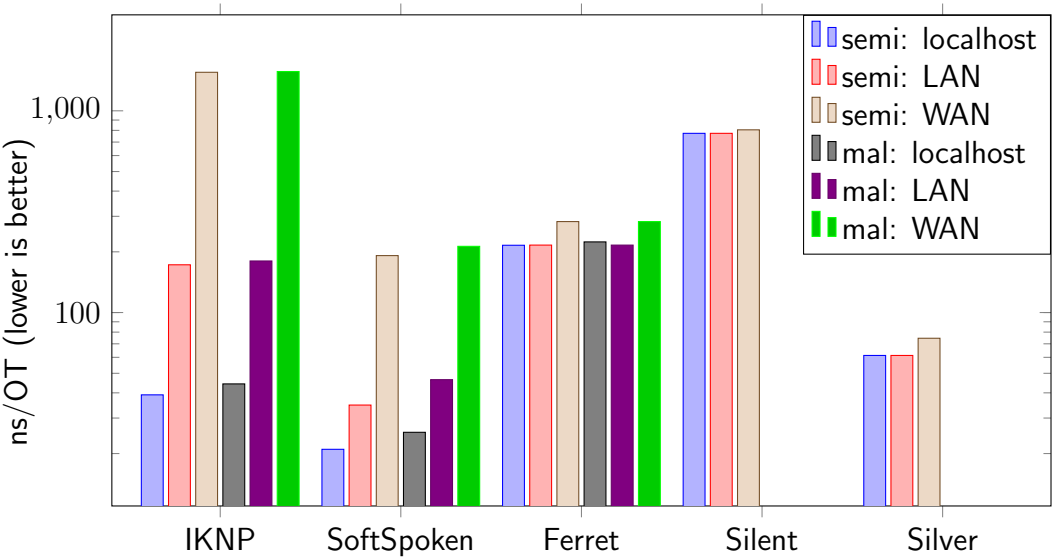
Implementation

Protocol	Semi-honest Security					Malicious Security		
	Communication		Time (ms)			Time (ms)		
	KB	bits/OT	localhost	LAN	WAN	localhost	LAN	WAN
IKNP / KOS	160010	128	391	1725	15525	443	1802	15662
SoftSpoken ($k = 1$)	160009	128	243	1590	15420	298	1637	15648
SoftSpoken ($k = 2$)	80009	64	210	815	7730	255	893	7985
SoftSpoken ($k = 3$)	53759	43	223	568	5208	322	677	5419
SoftSpoken ($k = 4$)	40008	32	261	433	3995	311	530	4114
SoftSpoken ($k = 5$)	32510	26	337	348	3271	454	465	3447
SoftSpoken ($k = 6$)	27509	22	471	488	2811	588	613	2985
SoftSpoken ($k = 7$)	23760	19	777	843	2380	899	966	2554
SoftSpoken ($k = 8$)	20008	16	1259	1314	1916	1293	1322	2130
SoftSpoken ($k = 9$)	18759	15	2302	2338	2439	2460	2457	2590
SoftSpoken ($k = 10$)	16259	13	3984	3983	4097	4126	4132	4223
Ferret[6]	2976	2.38	2156	2160	2825	2240	2242	3108
Silent (Quasi-cyclic)[7]	127	0.10	7735	7736	8049	Not tested		
Silent (Silver, weight 5)[8]	127	0.10	613	613	746	Not tested		

[6] Yang et al., "Ferret: Fast Extension for Correlated OT with Small Communication", 2020.

[7] Boyle et al., "Efficient Two-Round OT Extension and Silent Non-Interactive Secure Computation", 2019.

[8] Couteau, Rindal, and Raghuraman, "Silver: Silent VOLE and Oblivious Transfer from Hardness of Decoding Structured LDPC Codes", 2021.


ns/OT (lower is better)
1,000
100
semi: localhost
semi: LAN
semi: WAN
mal: localhost
mal: LAN
mal: WAN
IKNP
SoftSpoken
Ferret
Silent
Silver

Random Oblivious Transfer $\left(\binom{2}{1}\text{-OT}\right)$

m_0, m_1 ⟵ OT ⟶ $x^* \in \{0,1\}$

m_{x^*}

All-But-One Oblivious Transfer $\left(\binom{2}{2-1}\text{-OT}\right)$

m_0, m_1 ⟵ OT ⟶ $\Delta \in \{0,1\}$

$m_{1-\Delta}$

All-But-One Oblivious Transfer $\left(\binom{N}{N-1}\text{-OT}\right)$

$m_0, \ldots, m_{2^k-1}$

$\{m_x\}_{x\in[N]}$

OT

$\Delta \in [N]$

$\{m_x\}_{x\in[N]\setminus\{\Delta\}}$

$m_0, \ldots, m_{\Delta-1},$

$m_{\Delta+1}, \ldots, m_{2^k-1}$

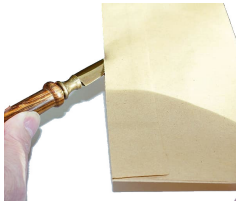
Random Oblivious Linear Evaluation (OLE)

u, v ⟵ OLE ⟶ Δ, w

$$w = u\Delta + v$$

Random Oblivious Linear Evaluation (OLE)

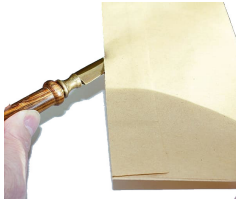
u, v ⟵ OLE ⟶ Δ, w



$$w - v = u\Delta$$

Random Oblivious Linear Evaluation (OLE)

u, v ⟵ OLE ⟶ Δ, w



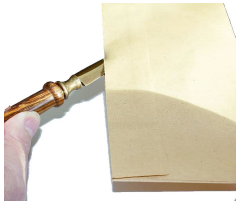
$$w - v = u\Delta$$

$$[\,\cdot\,] - [\,\cdot\,] = [\,\cdot\,]\,[\,\cdot\,]$$

OLE: $u, v, \Delta, w \in \mathbb{F}$.

Random Oblivious Linear Evaluation (OLE)

u, v ← VOLE → Δ, w



$$w - v = u\Delta$$

$$\begin{bmatrix} \cdot \\ \vdots \\ \cdot \end{bmatrix} - \begin{bmatrix} \cdot \\ \vdots \\ \cdot \end{bmatrix} = \begin{bmatrix} \cdot \\ \vdots \\ \cdot \end{bmatrix} \begin{bmatrix} \cdot \end{bmatrix}$$

Vector OLE (VOLE): $\vec{u}, \vec{v}, \vec{w} \in \mathbb{F}^\ell$ are vectors. I.e., ℓ OLEs with the same Δ.

Random Oblivious Linear Evaluation (OLE)

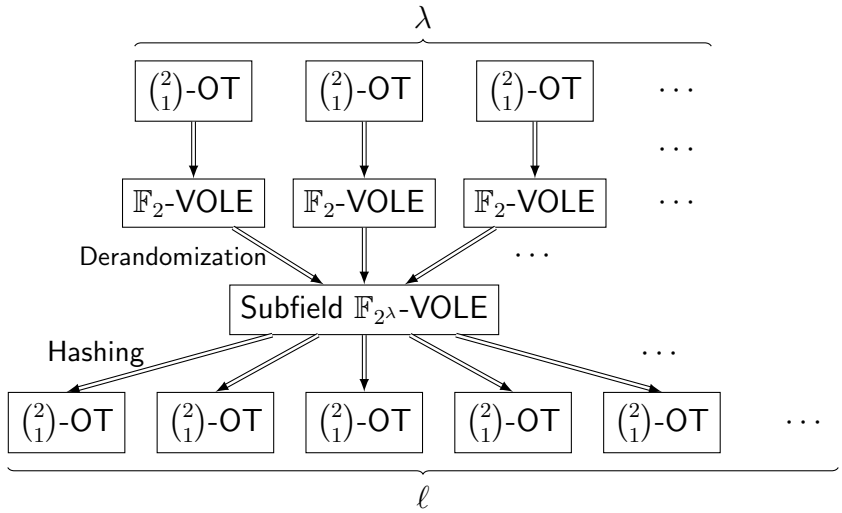
u, v ⟵ Subfield VOLE ⟶ Δ, w

$$w - v = u\Delta$$

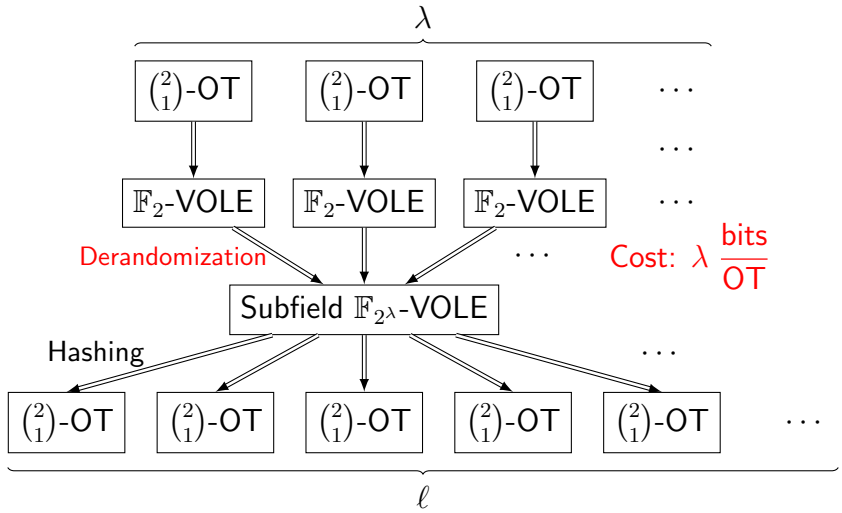
$$\begin{bmatrix} \cdot & \cdots & \cdot \\ \vdots & \ddots & \vdots \\ \cdot & \cdots & \cdot \end{bmatrix} - \begin{bmatrix} \cdot & \cdots & \cdot \\ \vdots & \ddots & \vdots \\ \cdot & \cdots & \cdot \end{bmatrix} = \begin{bmatrix} \cdot \\ \vdots \\ \cdot \end{bmatrix} \begin{bmatrix} \cdot & \cdots & \cdot \end{bmatrix}$$

Subfield VOLE: $u \in \mathbb{F}_2^\ell$, while $v, w, \Delta \in \mathbb{F}_{2^k}^\ell$.

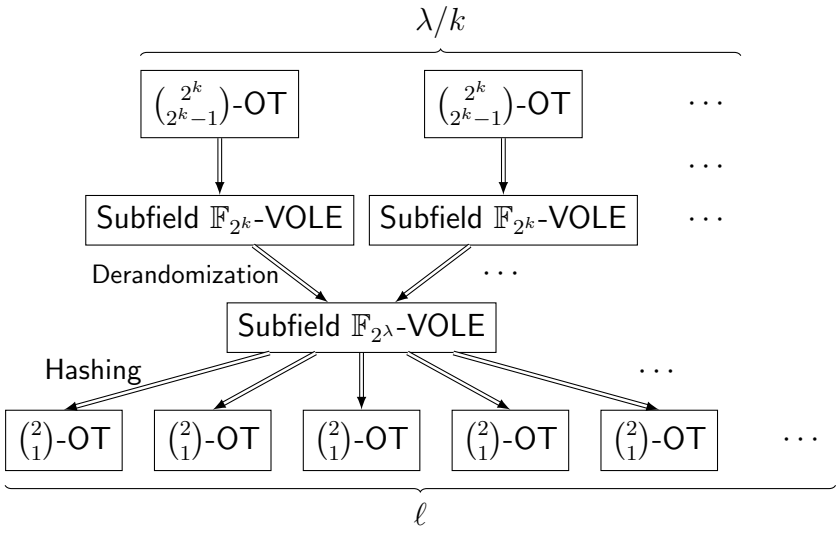
IKNP Overview



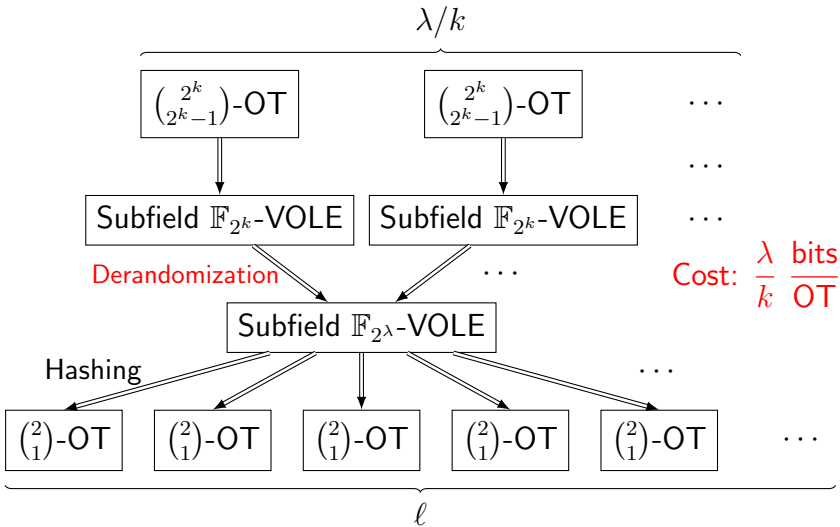
IKNP Overview



SoftSpokenOT Overview



SoftSpokenOT Overview



Consistency Check Comparision

Paper	Bound	Attack		Notes
		Computation	Probabillity	
SoftSpokenOT	2^{-s+1}	Secure		
CCG[9]	$2^{-s/2}$	Secure		
OOS[10]	Flawed	Proof Flaw Only		
KOS[11]	Flawed	λ^2	$\lambda^2 2^{-\lambda-1}$	only violates a Lemma.
		$2^{\lambda/5}$	$2^{-3\lambda/5}$	special $\mathbb{F}_{2^\lambda}$ where $20 \mid \lambda$.
PSS[12]	Flawed	$a2^{\lambda/a-1}$	2^{-a}	

[9] Cascudo, Christensen, and Gundersen, "Actively Secure OT-Extension from q-ary Linear Codes", 2018.

[10] Orrù, Orsini, and Scholl, "Actively Secure 1-out-of-N OT Extension with Application to Private Set Intersection", 2017.

[11] Keller, Orsini, and Scholl, "Actively Secure OT Extension with Optimal Overhead", 2015.

[12] Patra, Sarkar, and Suresh, "Fast Actively Secure OT Extension for Short Secrets", 2017.

[13] Masny and Rindal, "Endemic Oblivious Transfer", 2019.

Consistency Check Comparision

Paper	Bound	Attack		Notes
		Computation	Probabillity	
SoftSpokenOT	2^{-s+1}	Secure		
CCG[9]	$2^{-s/2}$	Secure		
OOS[10]	Flawed	Proof Flaw Only		
KOS[11]	Flawed	λ^2	$\lambda^2 2^{-\lambda-1}$	only violates a Lemma.
		$2^{\lambda/5}$	$2^{-3\lambda/5}$	special $\mathbb{F}_{2^\lambda}$ where $20 \mid \lambda$.
PSS[12]	Flawed	$a2^{\lambda/a-1}$	2^{-a}	
Endemic OT[13]	Flawed	Near-Practical		heuristic.

[9] Cascudo, Christensen, and Gundersen, "Actively Secure OT-Extension from q-ary Linear Codes", 2018.

[10] Orrù, Orsini, and Scholl, "Actively Secure 1-out-of-N OT Extension with Application to Private Set Intersection", 2017.

[11] Keller, Orsini, and Scholl, "Actively Secure OT Extension with Optimal Overhead", 2015.

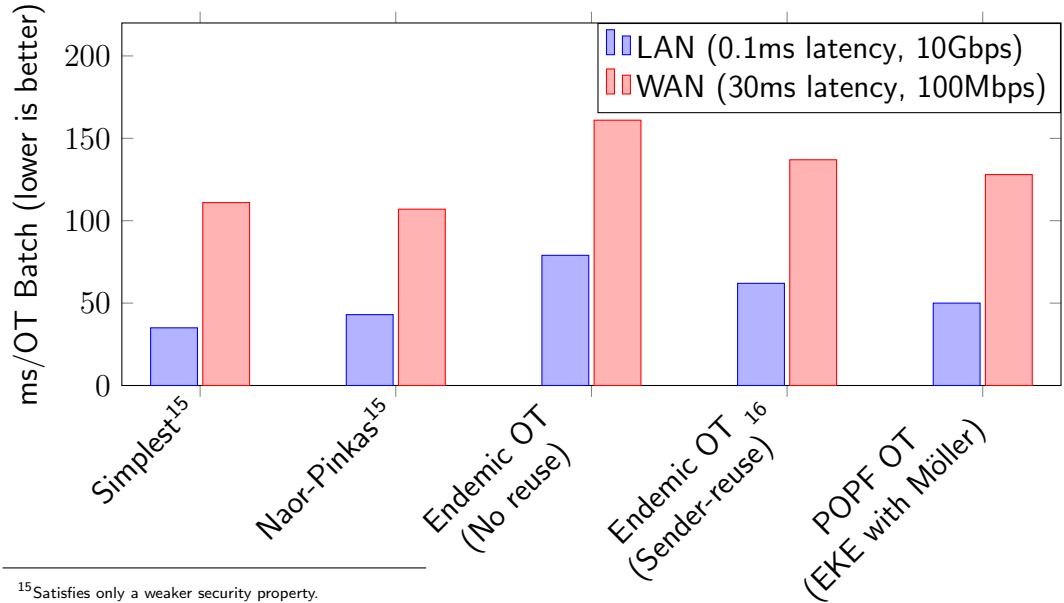
[12] Patra, Sarkar, and Suresh, "Fast Actively Secure OT Extension for Short Secrets", 2017.

[13] Masny and Rindal, "Endemic Oblivious Transfer", 2019.

Batching Base Oblivious Transfers (POPF OT)[14]

- POPF OT: Efficient base OT based on Möller key agreement and Programmable-Once Public Functions (POPF).
- Naive batching: a natural method of batching base OTs, used by 4 libraries and 3 papers. Unfortunately, it's insecure.
- We patched naive batching, and used it for batching POPF OT.

[14] McQuoid, Rosulek, and Roy, "Batching Base Oblivious Transfers", 2021.



[15] Satisfies only a weaker security property.

[16] Was vulnerable until our patch, because it used naive batching.

Acknowledgments

Thanks to:

- Mike Rosulek (advisor)
- Others at the OSU crypto group
- Elliott Slaughter (practicum advisor)
- Krell Institute and the DOE CSGF