



Distributed ECDSA

Fireblocks-3MI

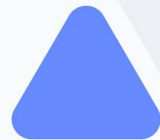
Speaker: Nikolaos Makriyannis

NIST MPTC Workshop
27/01/2026

Contents

- 1 Meet the team
- 2 ECDSA for Digital Assets
- 3 MPC ECDSA
& the CGGMP21 Protocol
- 4 2PC ECDSA
& the "BAM" Protocol

Meet the team



Fireblocks - 3MI

Meet the team

1. Fireblocks:
 - a. Michael Adjedj (Cryptography Engineer - Fireblocks)
 - b. Michael Gutkin (VP Research - Fireblocks)
 - c. Nikolaos Makriyannis (Research Scientist- Fireblocks)
2. 3MI Labs:
 - a. Tomer Ashur (CEO/Chief Scientist - 3MI Labs)
 - b. Cyprien de Saint Guilhem (Head of R&D - 3MI Labs)
 - c. Amit Singh Bhati (Research Scientist - 3MI Labs)
3. External Consultant:
 - a. Geoffroy Couteau (CNRS Research Scientist)



Fireblocks - 3MI

Meet the team

1. 100+ research papers
2. Decades of combined industry experience
3. Contributions to standards (NIST, ISO)



Fireblocks

Global platform to issue, custody, move and manage any digital asset and currencies



2,400+
Institutional Customers

110
Countries Supported

1bn+
In Market Reach

830+
Employees

\$1Bn+
In funding

FUNDED BY INDUSTRY LEADERS

Logos of industry leaders who have funded Fireblocks: BNY, SEQUOIA, swisscom Ventures, Ribbit Capital, ICONIQ Growth, DRIV VENTURE CAPITAL, SPARK CAPITAL, COATUE, STRIPES, and D1 CAPITAL PARTNERS.

Ecosystem and Network, touching 1B+ Consumers

Banks & Custodians

- Custody
- Asset tokenization
- Treasury operations
- Stablecoin issuance



Asset Managers

- Fund tokenization
- Custody
- Distribution



Fintechs and Neobanks

- Crypto trading
- Yield products
- Payments
- Treasury operations



Payment Service Providers

- Stablecoin payments
- Digital accounts
- Treasury operations
- Loyalty programs



Market Makers & OTCs

- Post-trade and settlement
- Treasury operations
- Collateral management



Exchanges

- Custody
- Treasury operations
- Collateral management



Official Institutions

- CBDC trials
- Permissioned DeFi
- On Chain FX
- Programmable money



Enterprises

- Web3 experiences
- Loyalty programs
- Stablecoin payments



Fireblocks

\$10T+

transactions secured

150

Blockchains supported

\$100b

Daily AUC

550m+

Wallets created



Wallets-as-a-Service



Treasury Management



Tokenization Engine



Payments Engine

VISA

 checkout.com

worldpay
from FIS

 ny bank

 BITSO

Revolut

eToro

 Stellar

 crypto.com

 TPG

Flipkart 

 Transfero
CRYPTO

ANZ 

 BNY MELLON

SCB 

VanEck

 BNP PARIBAS
SECURITIES SERVICES

 btg pactual

 MoonPay

 WISDOMTREE

What does it all boil down to

A look under the hood

ECDSA & Schnorr

1. Managing EC-signatures end-to-end

What does it all boil down to

A look under the hood

ECDSA & Schnorr

1. Managing EC-signatures end-to-end
2. Key management gives rise to operational/security/legal challenges
 - a. Where does the key material reside?
 - b. Is there a single point of failure?

What does it all boil down to

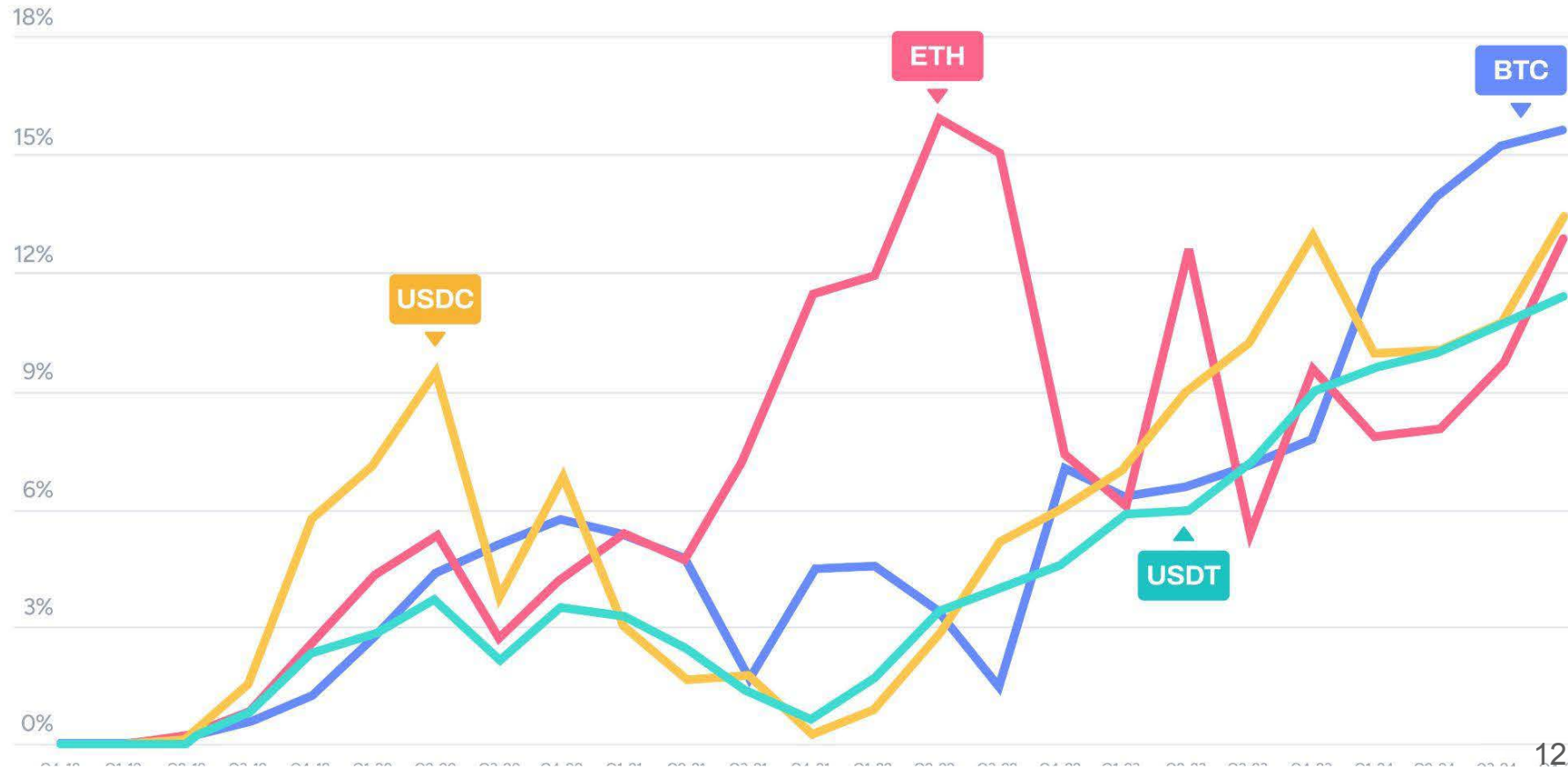
A look under the hood

ECDSA & Schnorr

1. Managing EC-signatures end-to-end
2. Key management gives rise to operational/security/legal challenges
 - a. Where does the key material reside?
 - b. Is there a single point of failure?
3. Fireblocks solves these challenges using MPC.
 - a. Key material is generated via DKG
 - b. Signatures are generated via distributed protocol



Fireblocks processes 10-15% of main blockchain transactions

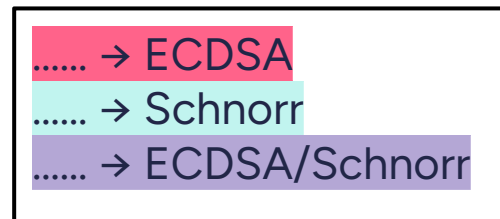


Fireblocks processes 10-15% of main blockchain transactions



Research Contributions

1. UC Non-Interactive, Proactive, Threshold ECDSA
 - *Canetti, M & Peled (eprint)*
2. UC Non-Interactive, Proactive, Threshold ECDSA with Identifiable Aborts
 - *Canetti, Gennaro, Goldfeder, M & Peled (CCS 2021)*
3. Highly Efficient OT-Based Multiplication Protocols
 - *Haitner, M, Ranellucci, Tsfadia (Eurocrypt 2022)*
4. On the Classic Protocol for MPC Schnorr Signatures
 - *M (eprint)*
5. Efficient Asymmetric Threshold ECDSA for MPC-based Cold Storage
 - *Blokh, M, Peled (eprint)*
6. Practical Key-Extraction Attacks in Leading MPC Wallets
 - *M, Yomtov, Galansky (CCS 2024)*
7. Two-Round 2PC ECDSA at the Cost of 1 OLE
 - *Adjedj, Blokh, Couteau, Galansky, Joux, M (eprint)*
8. From OT to OLE with Subquadratic Communication
 - *Doerner, Haitner, Ishai, M (CCS 2025)*
9. Tight Bounds on Uniform-Challenge Black-Box Reductions from Sigma Protocols
 - *Haitner, M (eprint)*
10. Integer Commitments, Old and New Tools
 - *Haitner, Lindell, M (eprint)*
11. Stateless 2PC Signatures for Internet-Scale Authentication and Authorization
 - *Adjedj, Couteau, Galansky, M, Yomtov (AsiaCCS 2026)*



Research Contributions

1. UC Non-Interactive, Proactive, Threshold ECDSA
 - *Canetti, M & Peled (eprint)*
2. UC Non-Interactive, Proactive, Threshold ECDSA with Identifiable Aborts
 - *Canetti, Gennaro, Goldfeder, M & Peled (CCS 2021)*
3. Highly Efficient OT-Based Multiplication Protocols
 - *Haitner, M, Ranellucci, Tsfadia (Eurocrypt 2022)*
4. On the Classic Protocol for MPC Schnorr Signatures
 - *M (eprint)*
5. Efficient Asymmetric Threshold ECDSA for MPC-based Cold Storage
 - *Blokh, M, Peled (eprint)*
6. Practical Key-Extraction Attacks in Leading MPC Wallets
 - *M, Yomtov, Galansky (CCS 2024)*
7. Two-Round 2PC ECDSA at the Cost of 1 OLE
 - *Adjedj, Blokh, Couteau, Galansky, Joux, M (eprint)*
8. From OT to OLE with Subquadratic Communication
 - *Doerner, Haitner, Ishai, M (CCS 2025)*
9. Tight Bounds on Uniform-Challenge Black-Box Reductions from Sigma Protocols
 - *Haitner, M (eprint)*
10. Integer Commitments, Old and New Tools
 - *Haitner, Lindell, M (eprint)*
11. Stateless 2PC Signatures for Internet-Scale Authentication and Authorization
 - *Adjedj, Couteau, Galansky, M, Yomtov (AsiaCCS 2026)*

Submission Packages
for NIST-MPTC



ECDSA Signatures for Digital Assets



ECDSA Signatures for Digital Assets

Background

1. EC-based signatures schemes dominate the ecosystem

ECDSA Signatures for Digital Assets

Background

1. EC-based signatures schemes dominate the ecosystem
2. ECDSA gained wide adoption thanks to:
 - a. Standardization
 - b. Security has stood the test of time
 - c. Ubiquitous support (OpenSSL, HSMs, hardware wallets)
 - d. Strong network effects / interoperability



ECDSA Signatures for Digital Assets

Background

1. EC-based signatures schemes dominate the ecosystem
2. ECDSA gained wide adoption thanks to:
 - a. Standardization
 - b. Security has stood the test of time
 - c. Ubiquitous support (OpenSSL, HSMs, hardware wallets)
 - d. Strong network effects / interoperability
3. ECDSA is the undisputable king of blockchain signatures
 - a. Bitcoin
 - b. Ethereum
 - c. Most blockchains emulate BTC and ETH



ECDSA Signatures

Definition

1. The public key is a random element X in the EC subgroup
 - a. The secret key is x such that $x \cdot G = X$
2. Signatures have the form (r, s)
 - a. $r = \text{conv}(k \cdot G)$ for a random point in the EC subgroup
 - b. $s = k^{-1}(m + r \cdot x)$

Message digest



ECDSA Signatures

Definition

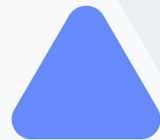
1. The public key is a random element X in the EC subgroup
 - a. The secret key is x such that $x \cdot G = X$
2. Signatures have the form (r, s)
 - a. $r = \text{conv}(k \cdot G)$ for a random point in the EC subgroup
 - b. $s = k^{-1}(m + r \cdot x)$

Message digest

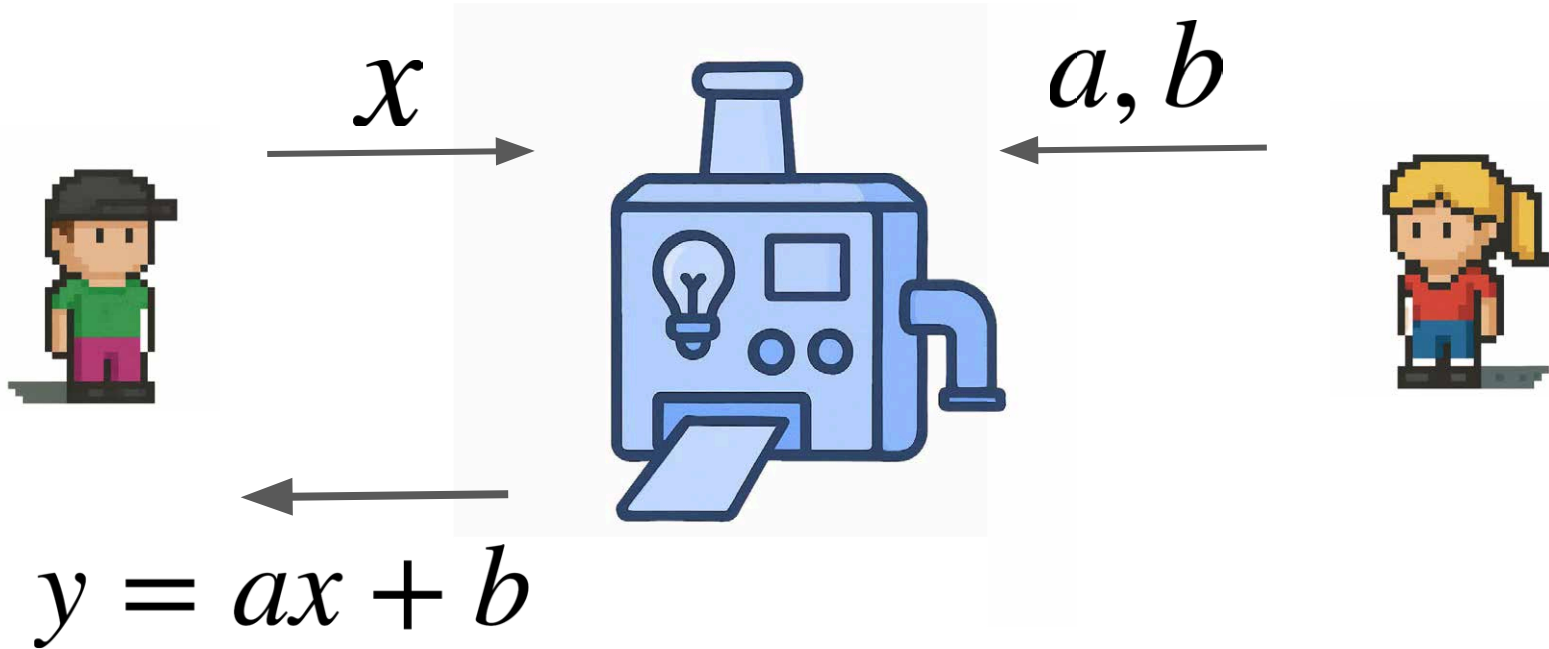
Naive sharing does not work



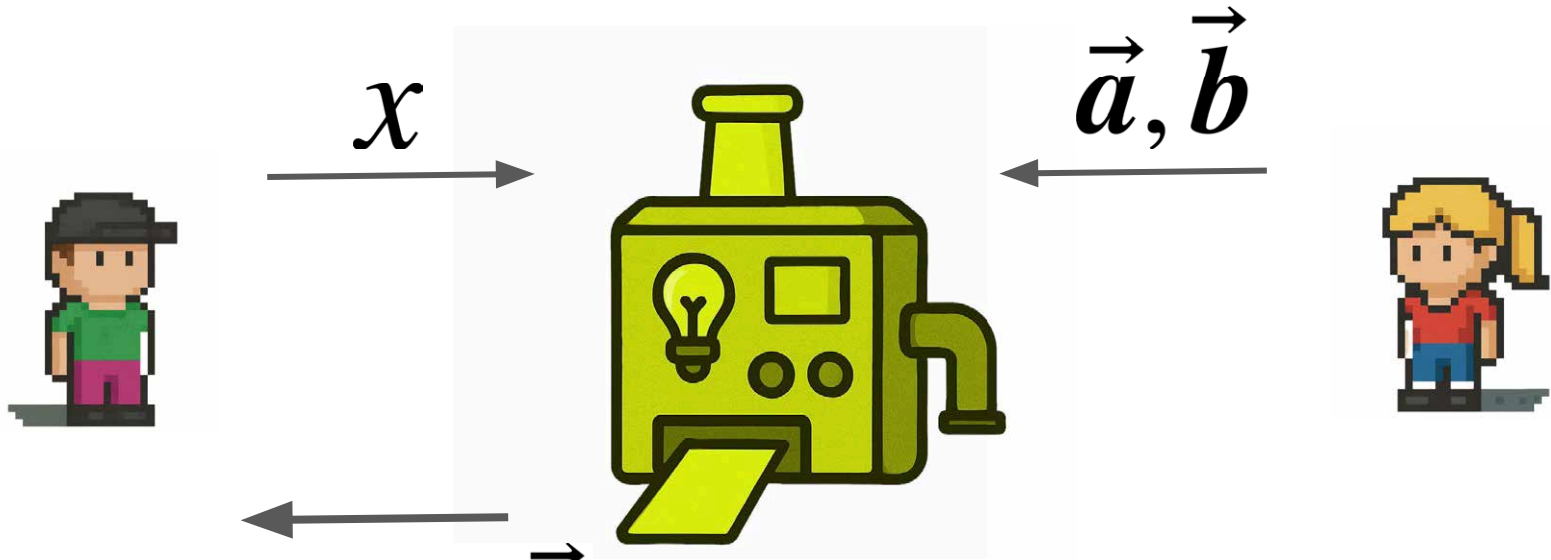
Background



Oblivious Linear Evaluation (OLE)



Vector Oblivious linear evaluation (VOLE)



$$\vec{y} = \vec{a} \cdot x + \vec{b}$$

The Beaver inversion trick



In order to compute shares of k^{-1} starting with shares of k



The Beaver inversion trick



In order to compute shares of k^{-1} starting with shares of k

1. Compute shares of γ and γk .
2. Reveal the shares of γk and sum them to obtain $\delta = \gamma k$
3. Multiply your local share of γ by δ^{-1} .



The Beaver inversion trick



In order to compute shares of k^{-1} starting with shares of k

1. Compute shares of γ and γk .
2. Reveal the shares of γk and sum them to obtain $\delta = \gamma k$
3. Multiply your local share of γ by δ^{-1} .

$$\sum_i \delta^{-1} \gamma_i = k^{-1}$$



The Beaver inversion trick



In order to compute shares of k^{-1} starting with shares of k

1. Compute shares of γ and γk .
2. Reveal the shares of γk and sum them to obtain $\delta = \gamma k$
3. Multiply your local share of γ by δ^{-1} .

Obtain shares of k^{-1} and $k^{-1}x$ using a VOLE

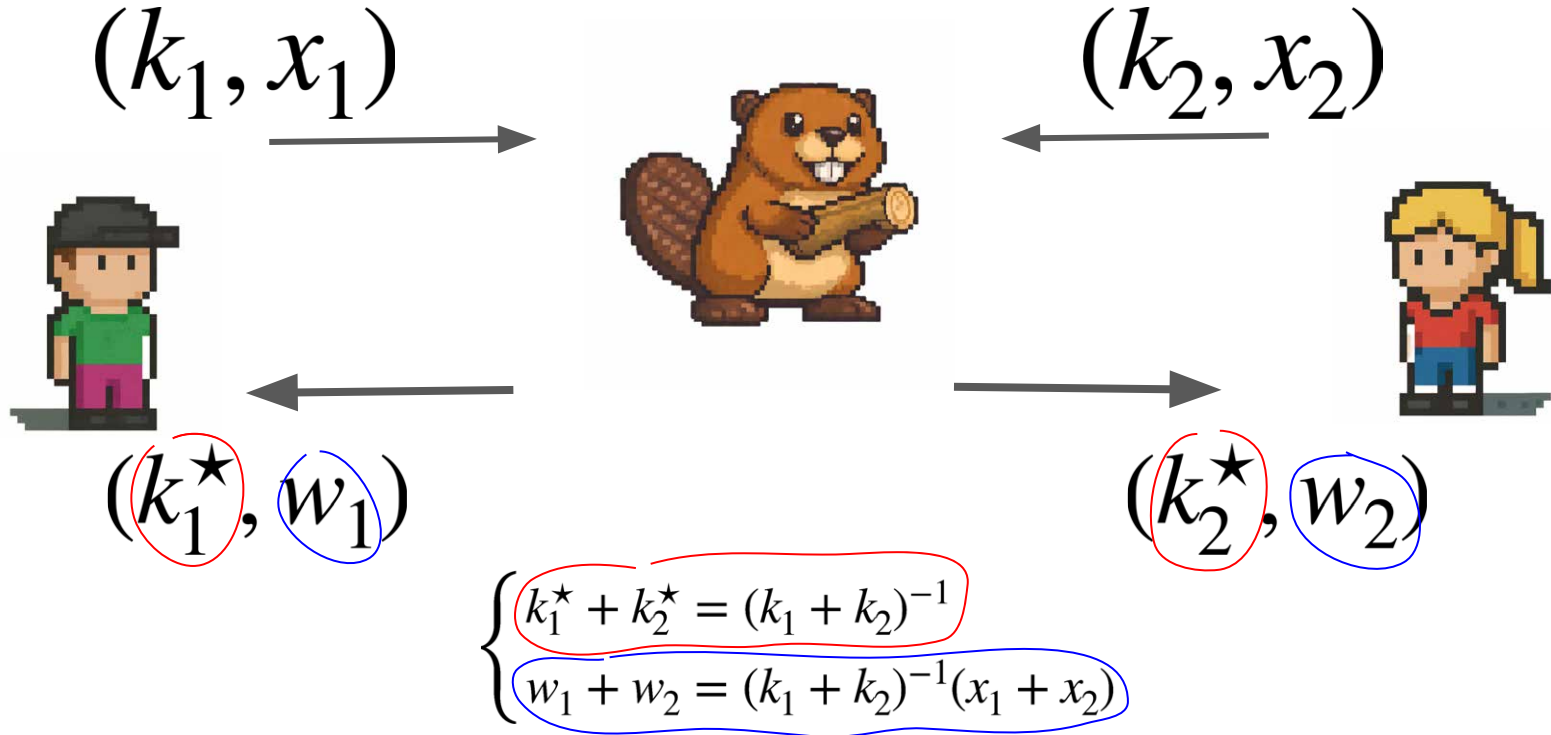
$$\sum_i \delta^{-1} \gamma_i = k^{-1}$$



Instantiating Beaver inversion using VOLE



Instantiating Beaver inversion using VOLE



MPC ECDSA & the CGGMP21 protocol

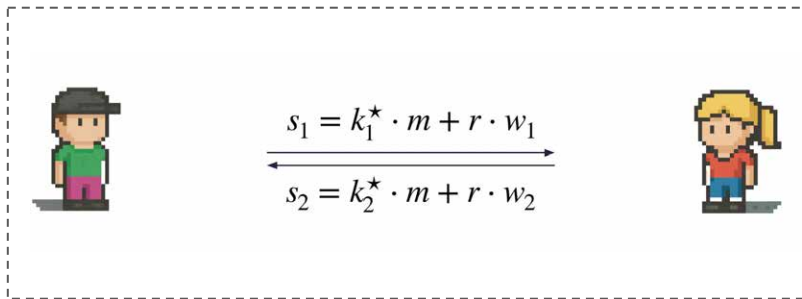
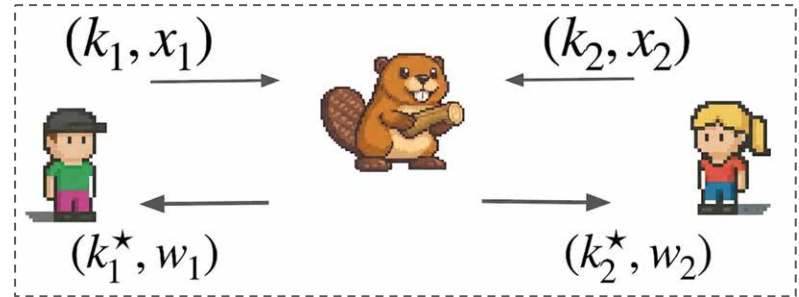
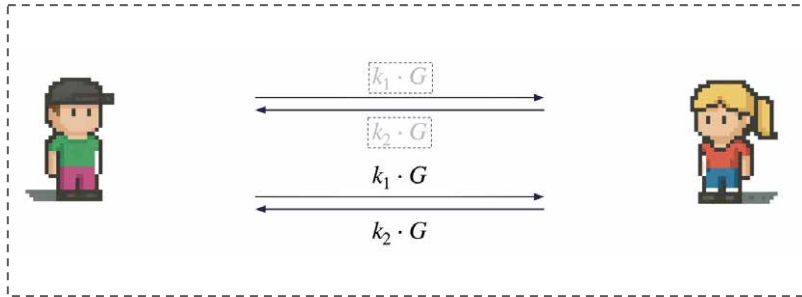


The template for the “right” ECDSA protocol

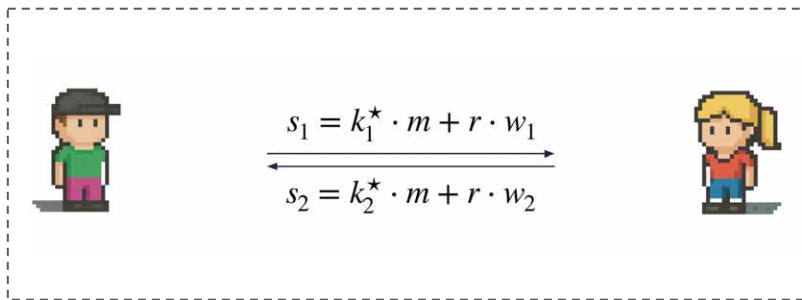
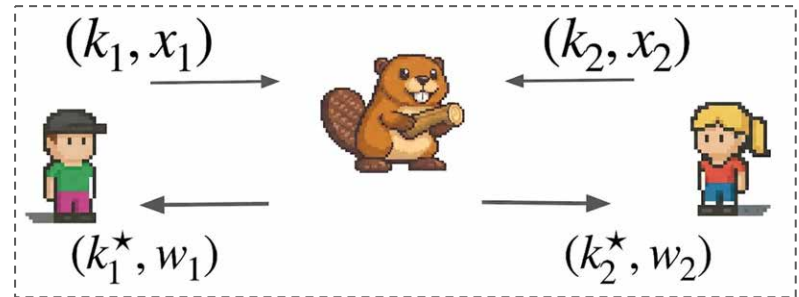
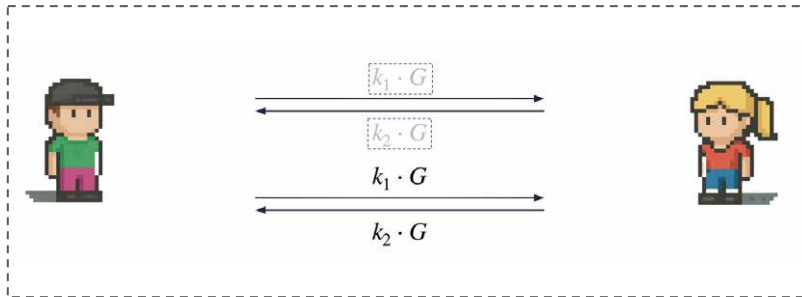
1. Sample a random group element via coin-toss (2 rounds)
2. Perform Beaver inversion on k and x (3 rounds)
3. Release the signature shares (1 round)



The template for the "right" ECDSA protocol

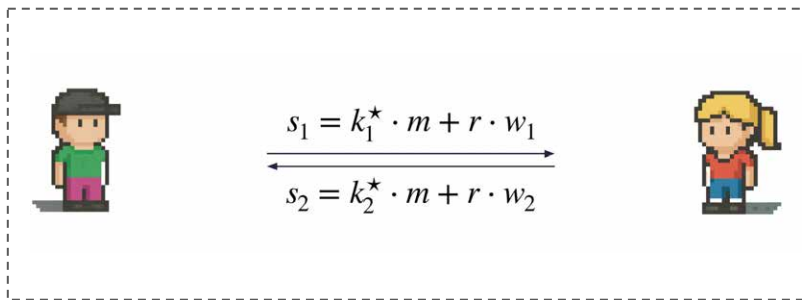
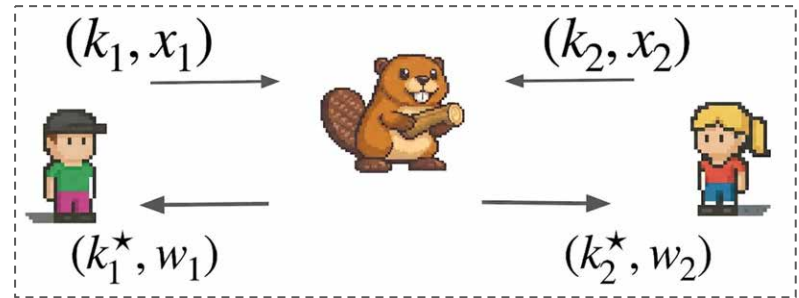
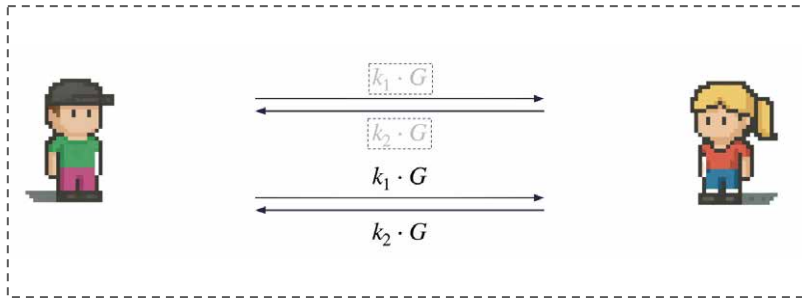


The template for the "right" ECDSA protocol



$$\begin{aligned} s_1 + s_2 &= (k_1^* + k_2^*) \cdot m + r \cdot (w_1 + w_2) \\ &= (k_1 + k_2)^{-1} \cdot m + r \cdot (k_1 + k_2)^{-1} (x_1 + x_2) \end{aligned}$$

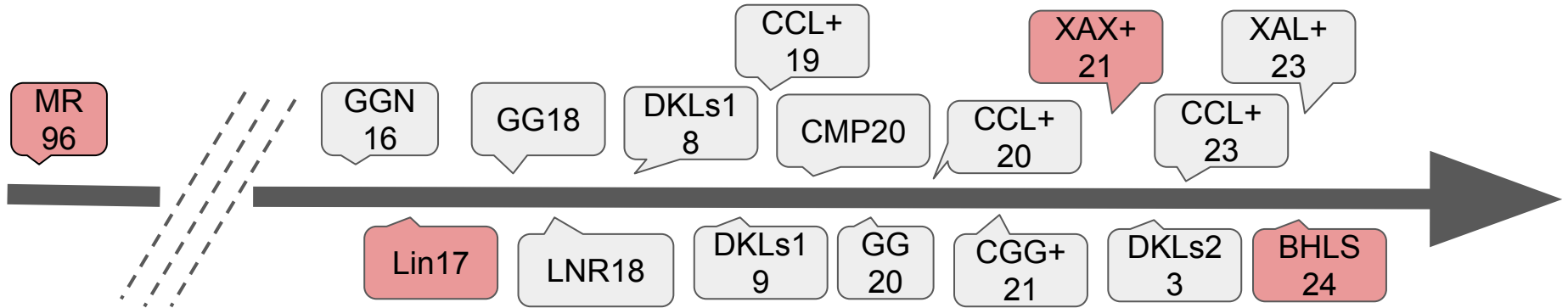
The template for the "right" ECDSA protocol



$$\begin{aligned}
 s_1 + s_2 &= (k_1^* + k_2^*) \cdot m + r \cdot (w_1 + w_2) \\
 &= (k_1 + k_2)^{-1} \cdot m + r \cdot (k_1 + k_2)^{-1} (x_1 + x_2) \\
 &= k^{-1} \cdot m + r \cdot k^{-1} x \\
 &= k^{-1} \cdot (m + rx)
 \end{aligned}$$



What's up with all the ECDSA protocols!?



RED means 2PC

Non-exhaustive list!

Why it's not so simple

1. How to realize the VOLE
2. How to “glue” the coin toss and the VOLE together
 - a. i.e. input consistency
3. Most prior work essentially provide a toolbox for (1) and (2)



Why it's not so simple

1. How to realize the VOLE
2. How to “glue” the coin toss and the VOLE together
 - a. i.e. input consistency
3. Most prior work essentially provide a toolbox for (1) and (2)

Paillier & Integer Commitments	OT-based VOLE & OT Extension	CL Encryption (Class Groups)
CGGMP21	DKLs23	CCL+20



Additional Preliminaries—Paillier Encryption

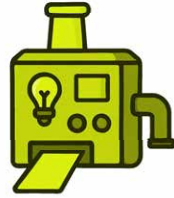
Parameters: $u, v \in \mathbb{Z}_{N^2}^*$

Enc:

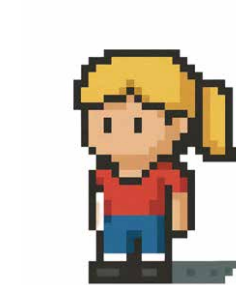
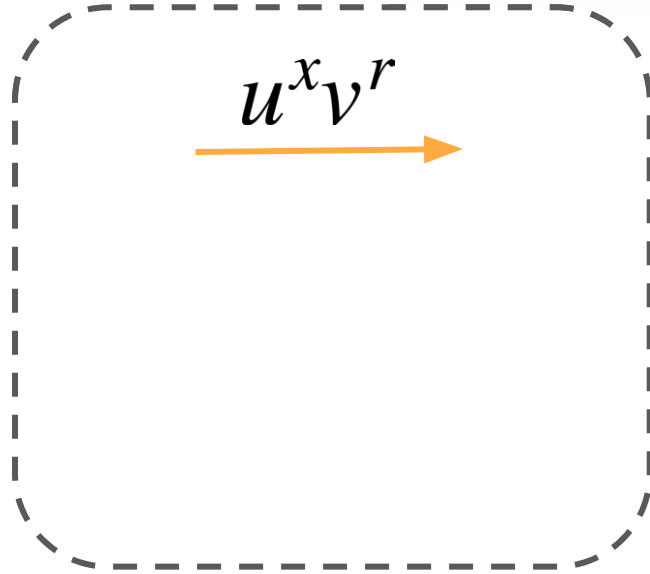
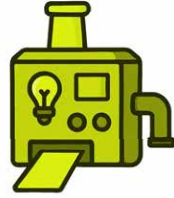
$$(m, r) \mapsto u^m v^r \pmod{N^2}$$



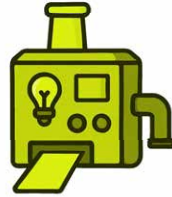
Realizing VOLE using Paillier





Realizing VOLE using Paillier



Realizing VOLE using Paillier

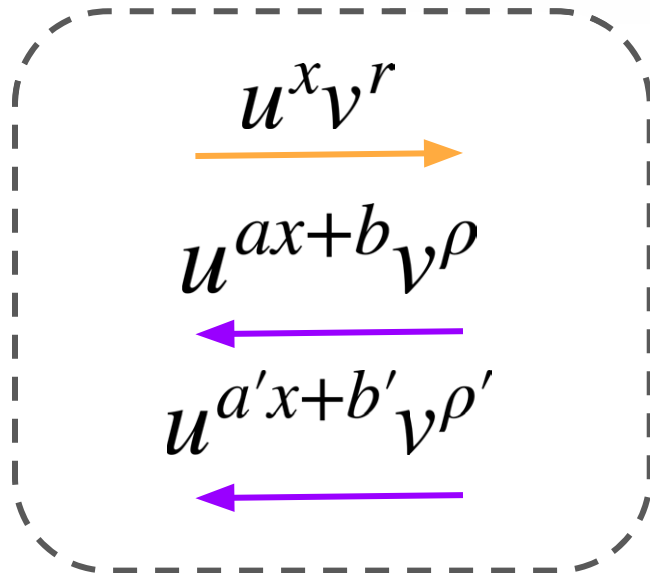


$$u^x v^r$$

$$u^{ax+b} v^\rho$$




$$y = \text{Dec}(u^{ax+b} v^\rho)$$

Realizing VOLE using Paillier



$$y = \text{Dec}(u^{ax+b} v^\rho)$$
$$y' = \text{Dec}(u^{a'x+b'} v^{\rho'})$$

The CGGMP21 Protocol



The CGGMP21 Protocol

1. The CGGMP21 protocol is essentially a suite of techniques for Paillier-based VOLE and DF commitments.



The CGGMP21 Protocol

1. The CGGMP21 protocol is essentially a suite of techniques for Paillier-based VOLE and DF commitments.
2. Instruction manual of necessary ZK proofs
 - a. Paillier modulus is a Paillier-compatible biprime
 - b. Paillier modulus does not admit small factors
 - c. DF commitments are statistically hiding
 - d. Well-formedness of VOLE messages



The CGGMP21 Protocol

1. The CGGMP21 protocol is essentially a suite of techniques for Paillier-based VOLE and DF commitments.
2. Instruction manual of necessary ZK proofs
 - a. Paillier modulus is a Paillier-compatible biprime
 - b. Paillier modulus does not admit small factors
 - c. DF commitments are statistically hiding
 - d. Well-formedness of VOLE messages

*Extractible/
Straightline Extractible*

The CGGMP21 Protocol

Security Analysis

1. Within the provable security framework, each property (unforgeability, id-abort, adaptive security) can be proven to be satisfied via its own game-based security definition.
2. Security guarantees can be “lifted” into UC theorem showing that the protocol realizes a threshold signatures functionality.



The CGGMP21 Protocol

Performance

Computation	200-500ms per party
Communication	15-23KB per party
Rounds	3



2PC ECDSA & the BAM protocol



Another look at the ECDSA formula



Another look at the ECDSA formula



$$\begin{aligned}s &= k^{-1} \cdot (m + rx) \\ &= k^{-1} \cdot (m + r \cdot (x_1 + x_2))\end{aligned}$$



Another look at the ECDSA formula



$$\begin{aligned} s &= k^{-1} \cdot (m + rx) \\ &= k^{-1} \cdot (m + r \cdot (x_1 + x_2)) \\ &= (k_1 \cdot k_2)^{-1} \cdot (m + r \cdot (x_1 + x_2)) \end{aligned}$$



Another look at the ECDSA formula



$$\begin{aligned} s &= k^{-1} \cdot (m + rx) \\ &= k^{-1} \cdot (m + r \cdot (x_1 + x_2)) \\ &= (k_1 \cdot k_2)^{-1} \cdot (m + r \cdot (x_1 + x_2)) \end{aligned}$$



⇒

$$\begin{aligned} s \cdot k_1 &= k_2^{-1} \cdot (m + r \cdot (x_1 + x_2)) \\ &= k_2^{-1} r \cdot x_1 + k_2^{-1} m + r x_2 \end{aligned}$$

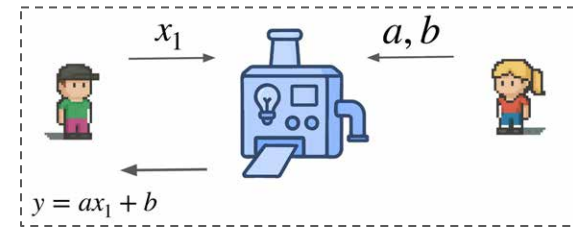
Another look at the ECDSA formula



$$\begin{aligned} s &= k^{-1} \cdot (m + rx) \\ &= k^{-1} \cdot (m + r \cdot (x_1 + x_2)) \\ &= (k_1 \cdot k_2)^{-1} \cdot (m + r \cdot (x_1 + x_2)) \end{aligned}$$

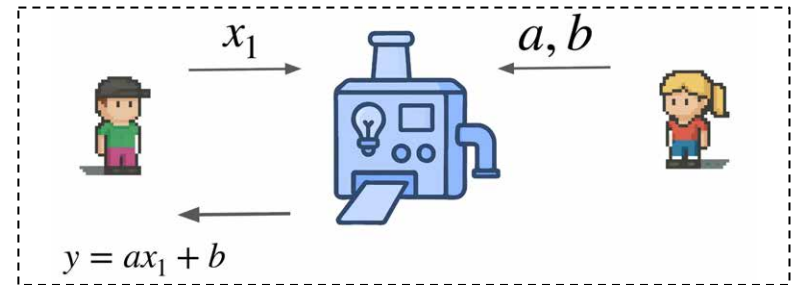
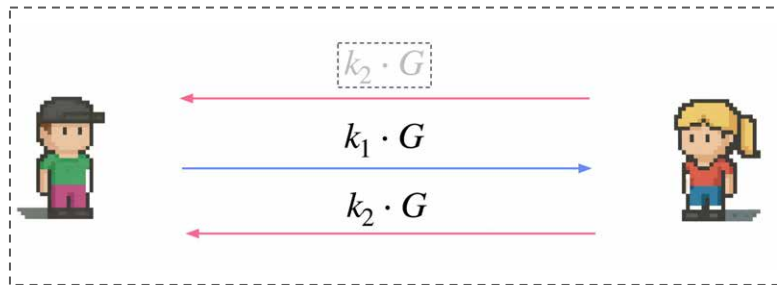


$$\begin{aligned} s \cdot k_1 &= k_2^{-1} \cdot (m + r \cdot (x_1 + x_2)) \\ &= \underbrace{k_2^{-1} r}_{a} \cdot x_1 + \underbrace{k_2^{-1} m + r x_2}_{b} \end{aligned}$$



The template for the "right" 2PC ECDSA

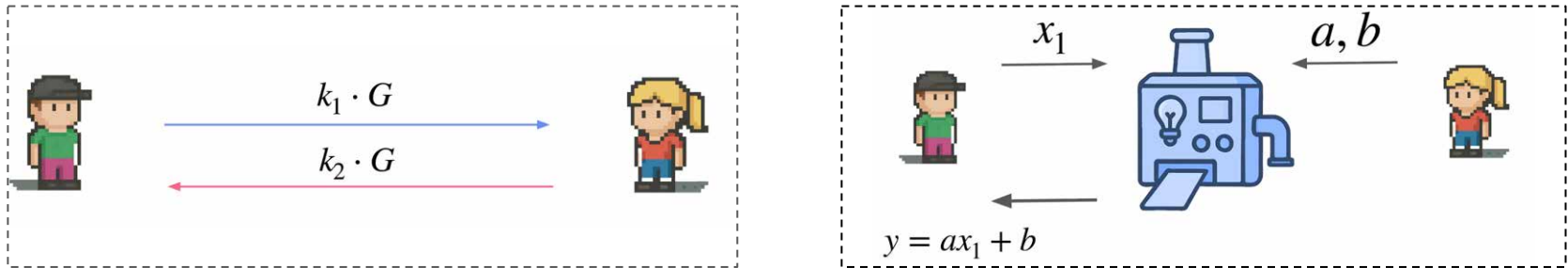
1. Sample a random group element via coin-toss (3 rounds*)
2. Execute OLE on inputs x_1 and a, b (2 rounds*)



The template for the "right" 2PC ECDSA

1. ~~Sample a random group element via coin-toss (3 rounds*)~~
2. Execute OLE on inputs x_1 and a, b (2 rounds*)

2-round *FROST-like* biased point selection



BAM under the hood

1. Additionally, BAM offers a suite of **highly optimized** techniques for Paillier-based (V)OLE and DF commitments.
 - a. New highly efficient DF-compatible RSA modulus
 - b. Systematic and ubiquitous use of small exponents
2. Instruction manual of necessary ZK proofs
 - a. New proofs using many new and old tricks
 - b. Check out the paper!

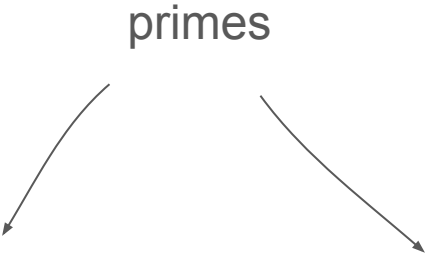


RSA moduli

Definition

$$N = p_1 \cdot p_2$$

primes




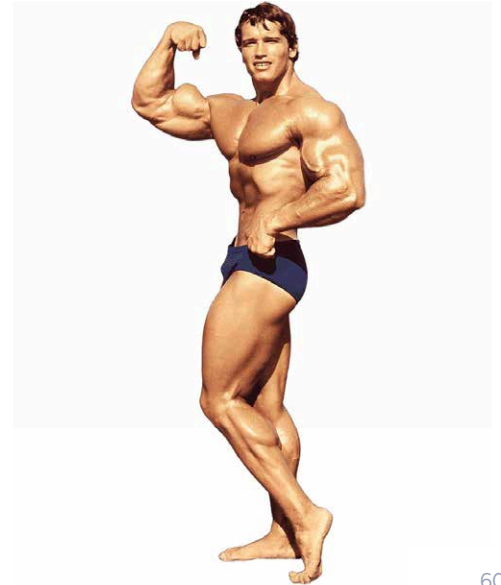
Strong RSA moduli

Primes have a special structure

$$p = 2p_0 + 1$$

primes



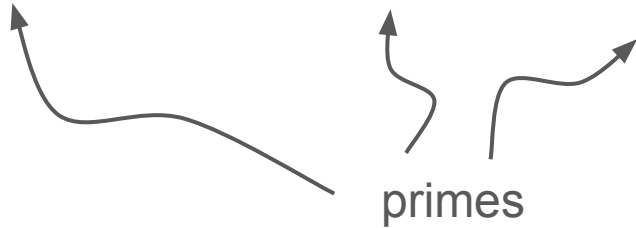


TOUGH

Strong RSA moduli

Primes have a special structure

$$p = 2p_1 \cdots p_\ell + 1$$



$$|p_i| = \text{sec_param}$$

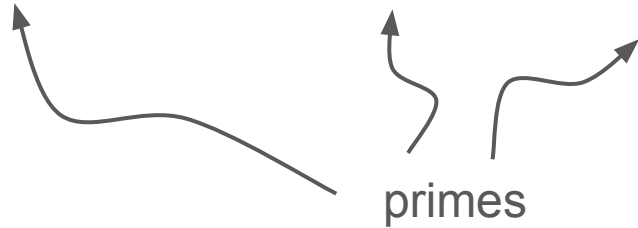


TOUGH

Strong RSA moduli

Primes have a special structure

$$p = 2p_1 \cdots p_\ell + 1$$

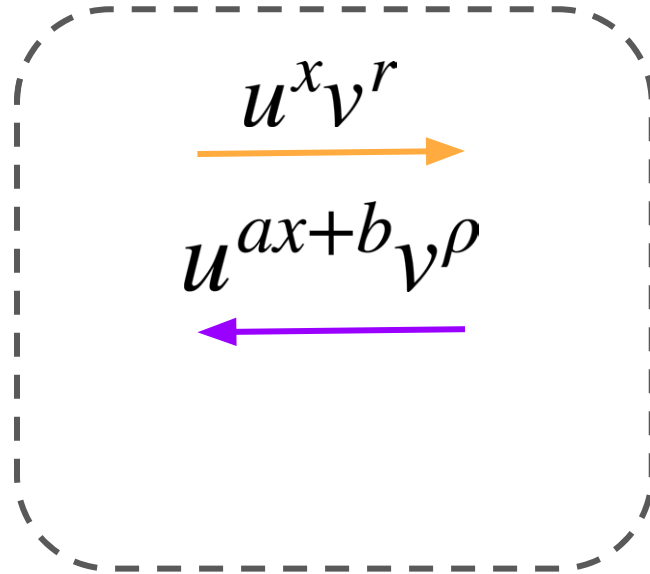
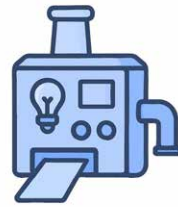


$$|p_i| = \text{sec_param}$$

**32X
FASTER!**

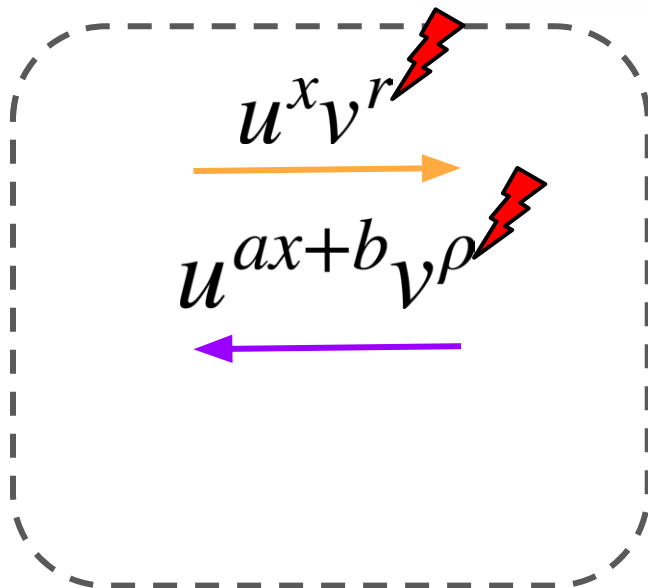
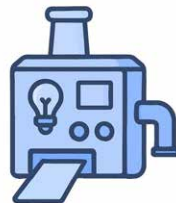


Paillier Small-Exponent OLE



$$y = \text{Dec}(u^{ax+b} v^\rho)$$

Paillier Small-Exponent OLE



v^{rand} dominates the computation as $\text{rand} > 2^{2000}$

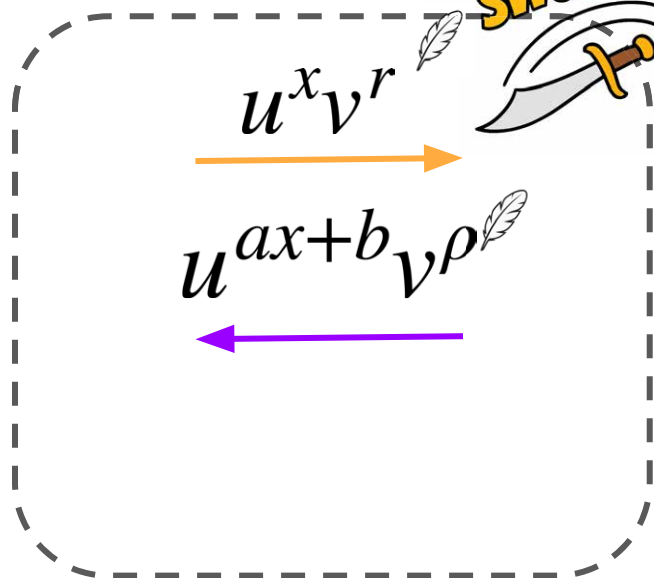


$$y = \text{Dec}(u^{ax+b} v^\rho)$$

Paillier Small-Exponent OLE



Improve comp. X10 with
 $\text{rand} \approx 2^{200}$



$$y = \text{Dec}(u^{ax+b} v^\rho)$$

BAM

Performance

Computation	50ms (40-60 split)
Communication	2KB
Rounds	2



Opening the floor to discussion

