NIST WORKSHOP ON PRIVACY-ENHANCING CRYPTOGRAPHY

Private Sealed-bid Auctions at Scale

Alireza Kavousi

University of Cambridge, ak2829@cam.ac.uk

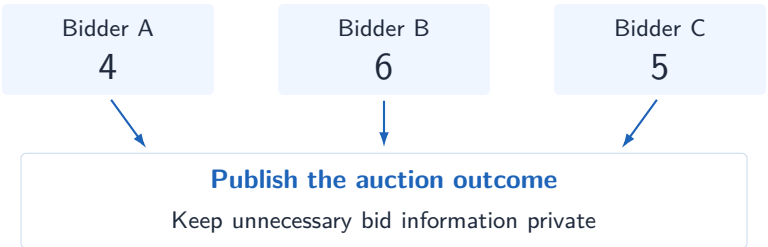
Joint works with: Kobi Gurkan, Philipp Jovanovic, Andrija Novakovic, and István András Seres

WPEC 2026 · 28 October 2026

The outcome is public. Must every bid be?

Bidder A	Bidder B	Bidder C
4	6	5

The outcome is public. Must every bid be?



Leaked bids reveal valuations and affect future auctions.

Auctions Are Everywhere — And Bid Privacy Is Critical

Where Digital Auctions Run Today

- **Blockchain / Web3:** Block-space allocation, Maximal Extractable Value (MEV) auctions, Ethereum Proposer-Builder Separation (PBS).
- **Web2 & Cloud:** Real-time online ad exchanges, cloud compute and decentralized storage allocation.

Auctions Are Everywhere — And Bid Privacy Is Critical

Where Digital Auctions Run Today

- **Blockchain / Web3:** Block-space allocation, Maximal Extractable Value (MEV) auctions, Ethereum Proposer-Builder Separation (PBS).
- **Web2 & Cloud:** Real-time online ad exchanges, cloud compute and decentralized storage allocation.

Why Losing Bids Must Stay Private

- **Preference Exposure:** Bid leakage allows competitors to infer valuations and adapt strategies in iterative runs.
- **Strategic Manipulation:** Winners learning the runner-up bid can shade future bids just above the second price.
- **Collusion & MEV:** Visible bids enable bribery, censoring, and front-running attacks.

Auctions Are Everywhere — And Bid Privacy Is Critical

Where Digital Auctions Run Today

- **Blockchain / Web3:** Block-space allocation, Maximal Extractable Value (MEV) auctions, Ethereum Proposer-Builder Separation (PBS).
- **Web2 & Cloud:** Real-time online ad exchanges, cloud compute and decentralized storage allocation.

Why Losing Bids Must Stay Private

- **Preference Exposure:** Bid leakage allows competitors to infer valuations and adapt strategies in iterative runs.
- **Strategic Manipulation:** Winners learning the runner-up bid can shade future bids just above the second price.
- **Collusion & MEV:** Visible bids enable bribery, censoring, and front-running attacks.

Takeaway: **Key Question:** How can auction outcomes be computed efficiently and verified publicly without exposing losing bids or trusting an auctioneer?

Five Desired Properties of a Sealed-Bid Auction

- **Post-Auction Bid Privacy:** Only the winning bid (and sale price) is revealed; **all losing bids remain confidential** both during and after the auction concludes.

Five Desired Properties of a Sealed-Bid Auction

- **Post-Auction Bid Privacy:** Only the winning bid (and sale price) is revealed; **all losing bids remain confidential** both during and after the auction concludes.
- **Public Verifiability:** All protocol steps and the final result can be publicly verified by third-party auditors to detect and penalize misbehavior.

Five Desired Properties of a Sealed-Bid Auction

- **Post-Auction Bid Privacy:** Only the winning bid (and sale price) is revealed; **all losing bids remain confidential** both during and after the auction concludes.
- **Public Verifiability:** All protocol steps and the final result can be publicly verified by third-party auditors to detect and penalize misbehavior.
- **Trust Minimization:** No honest-majority threshold committees or non-colluding server assumptions — rely at most on a **single untrusted auctioneer** for coordination.

Five Desired Properties of a Sealed-Bid Auction

- **Post-Auction Bid Privacy:** Only the winning bid (and sale price) is revealed; **all losing bids remain confidential** both during and after the auction concludes.
- **Public Verifiability:** All protocol steps and the final result can be publicly verified by third-party auditors to detect and penalize misbehavior.
- **Trust Minimization:** No honest-majority threshold committees or non-colluding server assumptions — rely at most on a **single untrusted auctioneer** for coordination.
- **Scalability:** Support hundreds of bidders m and large price ranges n with low computation and bandwidth costs.

Five Desired Properties of a Sealed-Bid Auction

- **Post-Auction Bid Privacy:** Only the winning bid (and sale price) is revealed; **all losing bids remain confidential** both during and after the auction concludes.
- **Public Verifiability:** All protocol steps and the final result can be publicly verified by third-party auditors to detect and penalize misbehavior.
- **Trust Minimization:** No honest-majority threshold committees or non-colluding server assumptions — rely at most on a **single untrusted auctioneer** for coordination.
- **Scalability:** Support hundreds of bidders m and large price ranges n with low computation and bandwidth costs.
- **Versatility:** Support first-price, second-price (Vickrey), uniform $(p+1)$-st-price, and sequential auctions with minimal protocol changes.

Five Desired Properties of a Sealed-Bid Auction

- **Post-Auction Bid Privacy:** Only the winning bid (and sale price) is revealed; **all losing bids remain confidential** both during and after the auction concludes.
- **Public Verifiability:** All protocol steps and the final result can be publicly verified by third-party auditors to detect and penalize misbehavior.
- **Trust Minimization:** No honest-majority threshold committees or non-colluding server assumptions — rely at most on a **single untrusted auctioneer** for coordination.
- **Scalability:** Support hundreds of bidders m and large price ranges n with low computation and bandwidth costs.
- **Versatility:** Support first-price, second-price (Vickrey), uniform $(p+1)$-st-price, and sequential auctions with minimal protocol changes.

Takeaway: Prior schemes sacrifice at least one of privacy, scalability, or trust minimization. **Cryptobazaar achieves all five simultaneously.**

Landscape I: Commit–Reveal & Timed-Release Cryptography

1. Classical Commit–Reveal

- **Mechanism:** Bidders post $\mathrm{Com}(b_i, r_i)$ in Round 1 and open (b_i, r_i) in Round 2.
- **Strengths:** Fast, simple, single untrusted coordinator.
- **Fatal Flaws:**
  1. **Selective Abort (Free-Option):** Bidders can refuse to open after observing others.
  2. **Zero Post-Auction Privacy:** Every opened bid is exposed to the public.

Landscape I: Commit–Reveal & Timed-Release Cryptography

1. Classical Commit–Reveal

- **Mechanism:** Bidders post $\mathrm{Com}(b_i, r_i)$ in Round 1 and open (b_i, r_i) in Round 2.
- **Strengths:** Fast, simple, single untrusted coordinator.
- **Fatal Flaws:**
  1. **Selective Abort (Free-Option):** Bidders can refuse to open after observing others.
  2. **Zero Post-Auction Privacy:** Every opened bid is exposed to the public.

2. Time-Lock Puzzles (Riggs, Cicada, ZeroAuction)

- **Riggs [CCS'23] & ZeroAuction [ePrint'24]:** Seal bids in Time-Lock Puzzles (TLPs) so aborters are force-opened via sequential computation.
- **Cicada [ePrint'23]:** Uses **Linearly Homomorphic TLPs** [MT'19] to pack all bids into a *single* puzzle for a **1-round non-interactive** auction.
- **Limitations:** Relies on fragile sequential hardware speed assumptions and still reveals losing bids upon opening.

Landscape I: Commit–Reveal & Timed-Release Cryptography

1. Classical Commit–Reveal

- **Mechanism:** Bidders post $\mathrm{Com}(b_i, r_i)$ in Round 1 and open (b_i, r_i) in Round 2.
- **Strengths:** Fast, simple, single untrusted coordinator.
- **Fatal Flaws:**
  1. **Selective Abort (Free-Option):** Bidders can refuse to open after observing others.
  2. **Zero Post-Auction Privacy:** Every opened bid is exposed to the public.

2. Time-Lock Puzzles (Riggs, Cicada, ZeroAuction)

- **Riggs [CCS'23] & ZeroAuction [ePrint'24]:** Seal bids in Time-Lock Puzzles (TLPs) so aborters are force-opened via sequential computation.
- **Cicada [ePrint'23]:** Uses **Linearly Homomorphic TLPs** [MT'19] to pack all bids into a *single* puzzle for a **1-round non-interactive** auction.
- **Limitations:** Relies on fragile sequential hardware speed assumptions and still reveals losing bids upon opening.

Takeaway: Can we build **1-round non-interactive auctions** with $O(1)$ tallying *without* fragile sequential hardware assumptions?

Landscape II: Homomorphic Signature-based Witness Encryption

From Signature-Based Release to Homomorphic Evaluation

- **SWE:** Encrypt under a public tag ρ and verification key. A valid signature σ_ρ acts as the **decryption witness**.
- **Stateless baseline:** Signers release a signature on ρ without knowing which ciphertexts will be decrypted.
- **HSWE:** Publicly aggregate ciphertexts before release. For a **common tag**, decrypt one aggregate instead of opening every input separately.

Landscape II: Homomorphic Signature-based Witness Encryption

From Signature-Based Release to Homomorphic Evaluation

- **SWE:** Encrypt under a public tag ρ and verification key. A valid signature σ_ρ acts as the **decryption witness**.
- **Stateless baseline:** Signers release a signature on ρ without knowing which ciphertexts will be decrypted.
- **HSWE:** Publicly aggregate ciphertexts before release. For a **common tag**, decrypt one aggregate instead of opening every input separately.

Four Constructions Across Message Spaces

1. **Binary / classical:** Cocks IBE + Rabin signatures; XOR-homomorphic.
2. **Binary / lattice-based:** GPV-style extraction + LWE; XOR aggregation with a noise bound.
3. **Bounded additive messages:** Boneh–Franklin / BLS; recovery uses a bounded discrete log.
4. **Packed large message space:** Ring-LWE; d coefficients modulo t, giving $|R_t| = t^d$.

Alireza Kavousi and István András Seres — *Homomorphic Signature-based Witness Encryption and Applications* (`ia.cr/2025/443`).

An Overview on the Literature

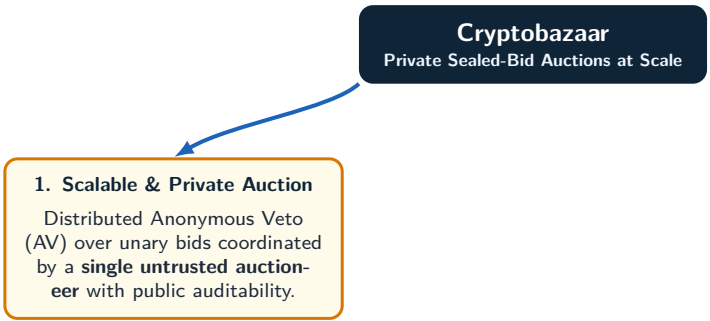
Protocol / Paradigm	Losing-Bid Privacy	Scalability	Trust Minimization	Versatility	Primary Trade-off / Bottleneck
Commit–Reveal	○ None	● Fast	● 1 Untrusted	● High	Exposes all losing bids; selective abort
Riggs [CCS'23] / **Cicada**	○ Pre-open only	◐ Moderate	● 1 Untrusted	◐ Limited	Sequential computation + bid exposure
HSWE	◐ Aggregate	● 1-Round, $O(1)$	◐ Threshold Committee	◐ Moderate	Relies on honest-majority quorum
Sako / **MPC & FHE**	● Full	○ Slow (min)	○ Committee	● High	Heavy multi-round MPC or FHE overhead
SEAL [TIFS'19]	● Full	○ $O(m^2)$ P2P	● Auctioneer-free	○ First-price	Bidders must stay online & self-adjust
Addax [NSDI'23]	● Full	● Fast (< 0.5 s)	○ ≥ 2 Non-colluding	○ First-price	Breaks if auctioneers collude; no integrity
Cryptobazaar [NDSS'26]	● **Full**	● < 0.5 **s**	● **1 Untrusted**	● **Multi-mode**	Linear $O(n)$ bandwidth per bidder (unary)

An Overview on the Literature

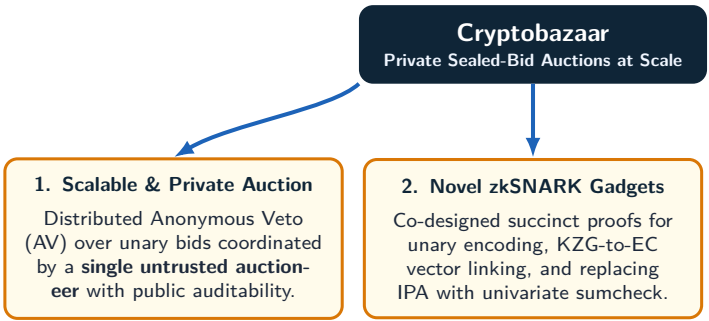
Protocol / Paradigm	Losing-Bid Privacy	Scalability	Trust Minimization	Versatility	Primary Trade-off / Bottleneck
Commit–Reveal	○ None	● Fast	● 1 Untrusted	● High	Exposes all losing bids; selective abort
Riggs [CCS'23] / **Cicada**	○ Pre-open only	◐ Moderate	● 1 Untrusted	◐ Limited	Sequential computation + bid exposure
HSWE	◐ Aggregate	● 1-Round, $O(1)$	◐ Threshold Committee	◐ Moderate	Relies on honest-majority quorum
Sako / **MPC & FHE**	● Full	○ Slow (min)	○ Committee	● High	Heavy multi-round MPC or FHE overhead
SEAL [TIFS'19]	● Full	○ $O(m^2)$ P2P	● Auctioneer-free	○ First-price	Bidders must stay online & self-adjust
Addax [NSDI'23]	● Full	● Fast (< 0.5 s)	○ ≥ 2 Non-colluding	○ First-price	Breaks if auctioneers collude; no integrity
Cryptobazaar [NDSS'26]	● **Full**	● < 0.5 **s**	● **1 Untrusted**	● **Multi-mode**	Linear $O(n)$ bandwidth per bidder (unary)

Takeaway: When **losing bids must remain confidential after the auction** with **no honest-majority or non-colluding server assumptions**, **Cryptobazaar** hits the sweet spot.

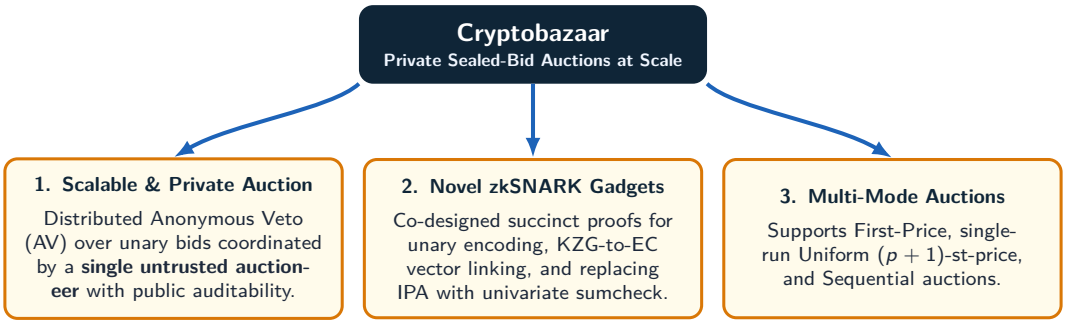
This Work: Cryptobazaar (NDSS 2026)



This Work: Cryptobazaar (NDSS 2026)

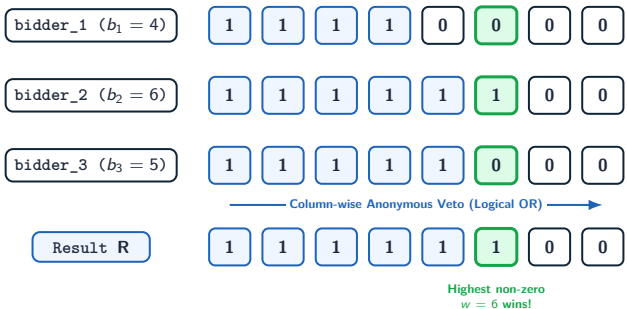


This Work: Cryptobazaar (NDSS 2026)



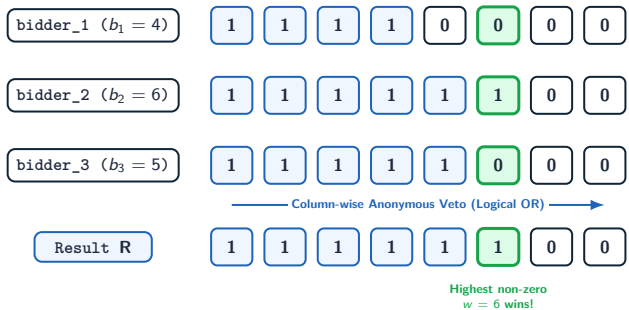
Cryptobazaar at a Glance: Unary Encoding + Distributed OR

1. Each bidder $i \in [m]$ chooses a bid $b_i \in [0, n]$ and encodes it in **unary**: b_i ones followed by $n - b_i$ zeros.



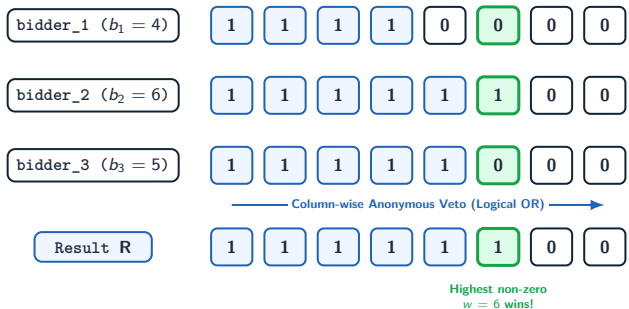
Cryptobazaar at a Glance: Unary Encoding + Distributed OR

1. Each bidder $i \in [m]$ chooses a bid $b_i \in [0, n]$ and encodes it in **unary**: b_i ones followed by $n - b_i$ zeros.
2. Bidders execute a 2-round distributed **logical-OR** (Anonymous Veto) on each price index $j \in [n]$ in parallel.



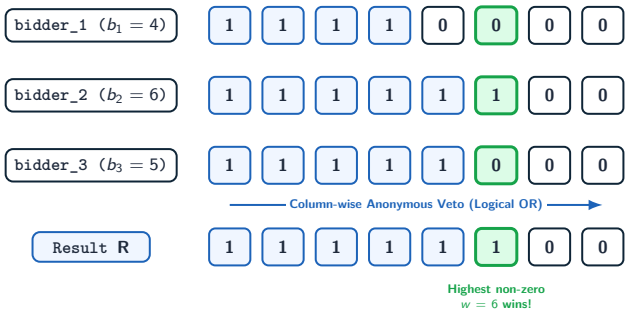
Cryptobazaar at a Glance: Unary Encoding + Distributed OR

1. Each bidder $i \in [m]$ chooses a bid $b_i \in [0, n]$ and encodes it in **unary**: b_i ones followed by $n - b_i$ zeros.
2. Bidders execute a 2-round distributed **logical-OR** (Anonymous Veto) on each price index $j \in [n]$ in parallel.
3. Orchestrated via a **star topology** by a single untrusted auctioneer.

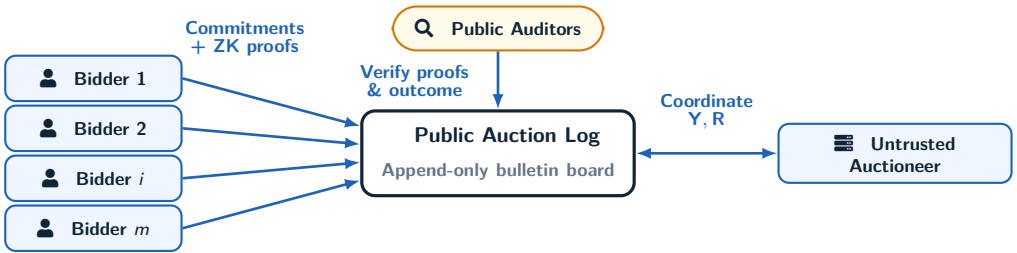


Cryptobazaar at a Glance: Unary Encoding + Distributed OR

1. Each bidder $i \in [m]$ chooses a bid $b_i \in [0, n]$ and encodes it in **unary**: b_i ones followed by $n - b_i$ zeros.
2. Bidders execute a 2-round distributed **logical-OR** (Anonymous Veto) on each price index $j \in [n]$ in parallel.
3. Orchestrated via a **star topology** by a single untrusted auctioneer.
4. The first non-zero position from the top (w) determines the **highest bid**, keeping all lower losing bids masked behind 1s!



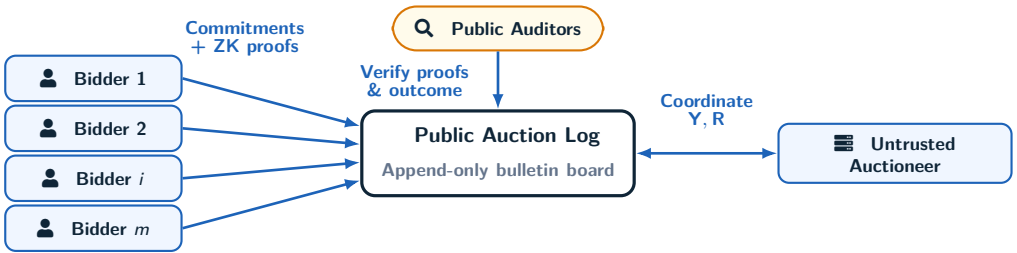
System and Threat Model: Star Topology with Public Auditability



Communication: Star Topology

Bidders do **not** interact P2P or need to know each other's identities. All messages are posted to the public log coordinated by the auctioneer.

System and Threat Model: Star Topology with Public Auditability



Communication: Star Topology

Bidders do **not** interact P2P or need to know each other's identities. All messages are posted to the public log coordinated by the auctioneer.

Adversary Model: Malicious ($k \leq m - 2$)

Auctioneer and up to $m - 2$ colluding bidders may deviate arbitrarily. Any invalid step fails public verification and triggers out-of-band slashing.

Building Block: Anonymous Veto (AV)

Two-Round AV Execution (Per Price Bit)

- **Round 1:** Each party $i \in [m]$ samples random $x_i \leftarrow \mathbb{F}$ and broadcasts $X_i = x_i G$. Define:

$$Y_i = \sum_{j=1}^{i-1} X_j - \sum_{j=i+1}^{m} X_j$$

- **Round 2:** Party i sends $R_i = x_i Y_i$ (**no veto / bit 0**) or $R_i = (x_i + r_i) Y_i$ for random $r_i \in \mathbb{F}^*$ (**veto / bit 1**).
- **Output:** $R = \sum_{i=1}^{m} R_i$. Since $\sum_{i=1}^{m} x_i Y_i = 0$:

$$R = 0 \iff \text{all bits 0}, \quad R \neq 0 \iff \text{at least one 1}.$$

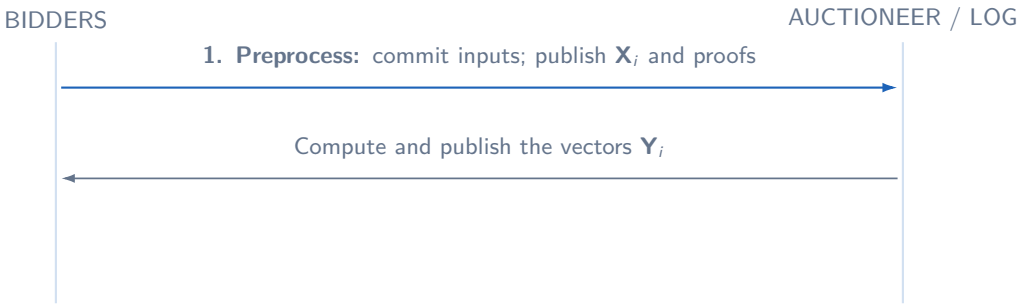
Takeaway: Under DDH, $(x_i + r_i) Y_i$ is indistinguishable from $x_i Y_i$, preserving input privacy as long as ≥ 2 parties are honest.

Two phases before evaluating the outcome



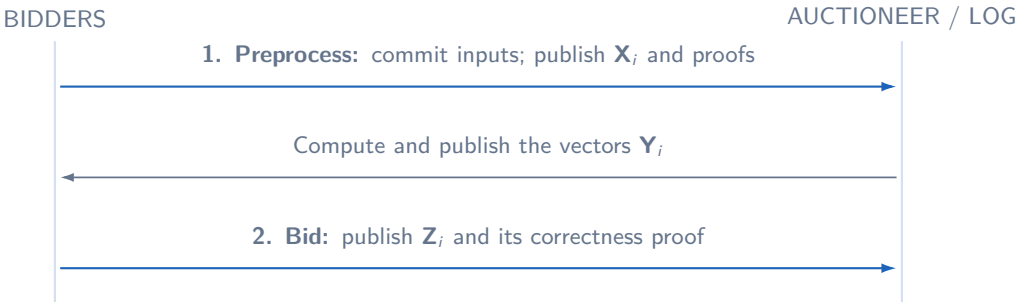
In the basic protocol, the bid commitment is already part of preprocessing.

Two phases before evaluating the outcome



In the basic protocol, the bid commitment is already part of preprocessing.

Two phases before evaluating the outcome



$$\mathbf{Z}_i = (\mathbf{x}_i + \mathbf{b}_i \circ \mathbf{r}_i) \circ \mathbf{Y}_i.$$

In the basic protocol, the bid commitment is already part of preprocessing.

Cryptobazaar Two-Phase Protocol Flow

Phase 1: Preprocessing Phase

Each bidder $i \in [m]$:

1. Sample non-zero random vectors $\mathbf{x}_i, \mathbf{r}_i \in \mathbb{F}^n$.
2. Compute KZG commitments $\mathrm{x}_i, \mathrm{r}_i$ and EC vector $\mathbf{X}_i = \mathbf{x}_i \cdot G$.
3. Encode bid $b_i \in [0, n]$ in unary $\mathbf{b}_i$ and commit to blinded b_i.
4. Post $(\mathrm{x}_i, \mathrm{r}_i, \mathrm{b}_i, \mathbf{X}_i, \pi_{\mathbf{x}_i}, \pi_{\mathbf{r}_i}, \pi_{\mathbf{b}_i})$ to public log.

Untrusted Auctioneer:

5. Compute $m \times n$ matrix $\mathbf{Y} = \mathbf{M} \cdot \mathbf{X}$ and post rows $\mathbf{Y}_i$.

Cryptobazaar Two-Phase Protocol Flow

Phase 1: Preprocessing Phase

Each bidder $i \in [m]$:

1. Sample non-zero random vectors $\mathbf{x}_i, \mathbf{r}_i \in \mathbb{F}^n$.
2. Compute KZG commitments $\mathrm{x}_i, \mathrm{r}_i$ and EC vector $\mathbf{X}_i = \mathbf{x}_i \cdot G$.
3. Encode bid $b_i \in [0, n]$ in unary $\mathbf{b}_i$ and commit to blinded b_i.
4. Post $(\mathrm{x}_i, \mathrm{r}_i, \mathrm{b}_i, \mathbf{X}_i, \pi_{\mathbf{x}_i}, \pi_{\mathbf{r}_i}, \pi_{\mathbf{b}_i})$ to public log.

Untrusted Auctioneer:

5. Compute $m \times n$ matrix $\mathbf{Y} = \mathbf{M} \cdot \mathbf{X}$ and post rows $\mathbf{Y}_i$.

Phase 2: Bidding & Winner Determination

Each bidder $i \in [m]$:

1. Compute $\mathbf{Z}_i = (\mathbf{x}_i + \mathbf{b}_i \circ \mathbf{r}_i) \circ \mathbf{Y}_i$ and proof $\pi_{\mathbf{Z}_i}$. $(b_{ij} = 1 \Rightarrow (x_{ij} + r_{ij})Y_{ij}; \quad b_{ij} = 0 \Rightarrow x_{ij} Y_{ij}.)$
2. Append $(\mathbf{Z}_i, \pi_{\mathbf{Z}_i})$ to the public log.

Auctioneer & Candidate Winner:

3. Compute AV output vector $\mathbf{R} = \sum_{i=1}^{m} \mathbf{Z}_i$.
4. Highest index $w = \max\{j \in [n] : R_j \neq 0\}$ is winning bid.
5. Winner submits ZK eligibility proof π_w (ties broken fairly via public randomness).

Cryptobazaar Two-Phase Protocol Flow

Phase 1: Preprocessing Phase

Each bidder $i \in [m]$:

1. Sample non-zero random vectors $\mathbf{x}_i, \mathbf{r}_i \in \mathbb{F}^n$.
2. Compute KZG commitments $\mathrm{x}_i, \mathrm{r}_i$ and EC vector $\mathbf{X}_i = \mathbf{x}_i \cdot G$.
3. Encode bid $b_i \in [0, n]$ in unary $\mathbf{b}_i$ and commit to blinded b_i.
4. Post $(\mathrm{x}_i, \mathrm{r}_i, \mathrm{b}_i, \mathbf{X}_i, \pi_{\mathbf{x}_i}, \pi_{\mathbf{r}_i}, \pi_{\mathbf{b}_i})$ to public log.

Untrusted Auctioneer:

5. Compute $m \times n$ matrix $\mathbf{Y} = \mathbf{M} \cdot \mathbf{X}$ and post rows $\mathbf{Y}_i$.

Phase 2: Bidding & Winner Determination

Each bidder $i \in [m]$:

1. Compute $\mathbf{Z}_i = (\mathbf{x}_i + \mathbf{b}_i \circ \mathbf{r}_i) \circ \mathbf{Y}_i$ and proof $\pi_{\mathbf{Z}_i}$. $(b_{ij} = 1 \Rightarrow (x_{ij} + r_{ij})Y_{ij}; \quad b_{ij} = 0 \Rightarrow x_{ij} Y_{ij}.)$
2. Append $(\mathbf{Z}_i, \pi_{\mathbf{Z}_i})$ to the public log.

Auctioneer & Candidate Winner:

3. Compute AV output vector $\mathbf{R} = \sum_{i=1}^{m} \mathbf{Z}_i$.
4. Highest index $w = \max\{j \in [n] : R_j \neq 0\}$ is winning bid.
5. Winner submits ZK eligibility proof π_w (ties broken fairly via public randomness).

Takeaway: $\mathbf{X}_i$ is completely independent of the bid b_i — AV randomness and $\mathbf{Y}$ can be precomputed offline!

Validity Proofs Overview & Preprocessing Gadgets

Proof	Phase	Relation Enforced	Core Cryptographic Gadget
$\pi_{\mathbf{x}_i}$ (Π_{pv})	Preprocessing	KZG $\mathrm{x}_i = [f(\tau)]_1$ and $\mathbf{X}_i = \mathbf{x}_i \cdot G$ encode same $\mathbf{x}_i$	Verifier Lagrange basis MSM + KZG opening
$\pi_{\mathbf{r}_i}$ (Π_{nz})	Preprocessing	Randomness entries $r_{ij} \neq 0$ for all $j \in [n]$	Blinded inverse polynomial over subgroup $\mathbb{H}$
$\pi_{\mathbf{b}_i}$ (Π_{unary})	Preprocessing	$\mathbf{b}_i = (1, \ldots, 1, 0, \ldots, 0)$ is a valid unary vector	Adjacent differences Δ + Logarithmic derivatives
$\pi_{\mathbf{Z}_i}$ (Π_{sum}, Π_{pse})	Bidding	$\mathbf{Z}_i = (\mathbf{x}_i + \mathbf{b}_i \circ \mathbf{r}_i) \circ \mathbf{Y}_i$ matches commitments	Generalized IPA + Univariate sumcheck link
π_w $(\mathcal{R}_{win})$	Winner Claim	Candidate winner bid $b_{iw} = 1$ at winning price w	Merkle set membership + Nullifier $\mathrm{nul} = \mathcal{H}(x, r)$

Validity Proofs Overview & Preprocessing Gadgets

Proof	Phase	Relation Enforced	Core Cryptographic Gadget
$\pi_{\mathbf{x}_i}$ (Π_{pv})	Preprocessing	KZG $\mathsf{x}_i = [f(\tau)]_1$ and $\mathbf{X}_i = \mathbf{x}_i \cdot G$ encode same $\mathbf{x}_i$	Verifier Lagrange basis MSM + KZG opening
$\pi_{\mathbf{r}_i}$ (Π_{nz})	Preprocessing	Randomness entries $r_{ij} \neq 0$ for all $j \in [n]$	Blinded inverse polynomial over subgroup $\mathbb{H}$
$\pi_{\mathbf{b}_i}$ (Π_{unary})	Preprocessing	$\mathbf{b}_i = (1, \ldots, 1, 0, \ldots, 0)$ is a valid unary vector	Adjacent differences Δ + Logarithmic derivatives
$\pi_{\mathbf{Z}_i}$ (Π_{sum}, Π_{pse})	Bidding	$\mathbf{Z}_i = (\mathbf{x}_i + \mathbf{b}_i \circ \mathbf{r}_i) \circ \mathbf{Y}_i$ matches commitments	Generalized IPA + Univariate sumcheck link
π_w ($\mathcal{R}_{win}$)	Winner Claim	Candidate winner bid $b_{iw} = 1$ at winning price w	Merkle set membership + Nullifier $\mathsf{nul} = \mathcal{H}(x, r)$

Gadget 1 (Π_{pv}): Linking KZG to EC Vector X

1. Verifier sends random challenge $\gamma \leftarrow \mathbb{F}$; prover sends KZG opening quotient $q = \left[\frac{f(\tau) - f(\gamma)}{\tau - \gamma}\right]_1$.
2. **Trick:** Verifier reconstructs $[y]_1 = \sum_{i=1}^{n} L_i(\gamma) X_i$ directly from $\mathbf{X}$ via Lagrange MSM and checks $e(\mathsf{x} - [y]_1 + \gamma q, [1]_2) = e(q, [\tau]_2)$.

Validity Proofs Overview & Preprocessing Gadgets

Proof	Phase	Relation Enforced	Core Cryptographic Gadget
$\pi_{\mathbf{x}_i}$ (Π_{pv})	Preprocessing	KZG $\mathsf{x}_i = [f(\tau)]_1$ and $\mathbf{X}_i = \mathbf{x}_i \cdot G$ encode same $\mathbf{x}_i$	Verifier Lagrange basis MSM + KZG opening
$\pi_{\mathbf{r}_i}$ (Π_{nz})	Preprocessing	Randomness entries $r_{ij} \neq 0$ for all $j \in [n]$	Blinded inverse polynomial over subgroup $\mathbb{H}$
$\pi_{\mathbf{b}_i}$ (Π_{unary})	Preprocessing	$\mathbf{b}_i = (1, \ldots, 1, 0, \ldots, 0)$ is a valid unary vector	Adjacent differences Δ + Logarithmic derivatives
$\pi_{\mathbf{Z}_i}$ (Π_{sum}, Π_{pse})	Bidding	$\mathbf{Z}_i = (\mathbf{x}_i + \mathbf{b}_i \circ \mathbf{r}_i) \circ \mathbf{Y}_i$ matches commitments	Generalized IPA + Univariate sumcheck link
π_w ($\mathcal{R}_{win}$)	Winner Claim	Candidate winner bid $b_{iw} = 1$ at winning price w	Merkle set membership + Nullifier $\mathsf{nul} = \mathcal{H}(x, r)$

Gadget 1 (Π_{pv}): Linking KZG to EC Vector X

1. Verifier sends random challenge $\gamma \leftarrow \mathbb{F}$; prover sends KZG opening quotient $q = \left[\frac{f(\tau)-f(\gamma)}{\tau-\gamma}\right]_1$.
2. **Trick:** Verifier reconstructs $[y]_1 = \sum_{i=1}^{n} L_i(\gamma) X_i$ directly from $\mathbf{X}$ via Lagrange MSM and checks $e(\mathsf{x} - [y]_1 + \gamma q, [1]_2) = e(q, [\tau]_2)$.

Gadget 2 (Π_{nz}): Proving $\mathbf{r}_i$ Has Non-Zero Entries

1. $r(w^i) \neq 0$ on $\mathbb{H}$ iff there exists an inverse polynomial $s(w^i) = r(w^i)^{-1}$ on $\mathbb{H}$.
2. Prover commits to blinded $\hat{r}(X), \hat{\mathsf{s}}(X)$ and quotient $q(X)$ satisfying $\hat{r}(X)\hat{\mathsf{s}}(X) - 1 = q(X)z_{\mathbb{H}}(X)$, verified at random $\gamma \leftarrow \mathbb{F}$.

Gadget 3 ($\pi_{\mathbf{b}_i}$): Valid Unary Encoding via Log Derivatives

Step 1: Difference Vector Characterization

Goal: Prove $\mathbf{b} = (1, \ldots, 1, 0, \ldots, 0) \in \{0,1\}^n$ without revealing the bid value b.

Define the adjacent difference vector $\Delta \in \mathbb{F}^n$:

$$\Delta_i = \begin{cases} b_i - b_{i+1} & \text{if } i < n \\ b_n & \text{if } i = n \end{cases}$$

Lemma 6: $\mathbf{b}$ is a valid non-zero unary vector iff:

1. $b_1 = 1$, and
2. Δ contains $(n-1)$ **zeros and a single** 1 (at transition index $i = b$).

Gadget 3 ($\pi_{\mathbf{b}_i}$): Valid Unary Encoding via Log Derivatives

Step 1: Difference Vector Characterization

Goal: Prove $\mathbf{b} = (1, \ldots, 1, 0, \ldots, 0) \in \{0,1\}^n$ without revealing the bid value b.

Define the adjacent difference vector $\Delta \in \mathbb{F}^n$:

$$\Delta_i = \begin{cases} b_i - b_{i+1} & \text{if } i < n \\ b_n & \text{if } i = n \end{cases}$$

Lemma 6: $\mathbf{b}$ is a valid non-zero unary vector iff:

1. $b_1 = 1$, and
2. Δ contains $(n-1)$ **zeros and a single** 1 (at transition index $i = b$).

Step 2: Log-Derivative Multiset Lookup

By unique fractional decomposition of logarithmic derivatives [Haböck'22], Δ has multiset $\{0^{n-1}, 1^1\}$ iff:

$$\sum_{i=1}^{n} \frac{1}{X + \Delta_i} = \frac{n-1}{X+0} + \frac{1}{X+1}.$$

1. Interpolate $f(w^i) = u(w^i) - u(w^{i+1})$ over $\mathbb{H}$.
2. At random challenge $\gamma \leftarrow \mathbb{F}$, commit to blinded reciprocal $\hat{B}(w^i) = \frac{1}{\gamma + f(w^i)}$.
3. Run **univariate sumcheck** to prove $\sum_{h \in \mathbb{H}} \hat{B}(h) = \frac{n-1}{\gamma} + \frac{1}{\gamma+1}$ and check KZG opening $u(1) = 1$.

Gadget 3 ($\pi_{\mathbf{b}_i}$): Valid Unary Encoding via Log Derivatives

Step 1: Difference Vector Characterization

Goal: Prove $\mathbf{b} = (1, \ldots, 1, 0, \ldots, 0) \in \{0,1\}^n$ without revealing the bid value b.

Define the adjacent difference vector $\Delta \in \mathbb{F}^n$:

$$\Delta_i = \begin{cases} b_i - b_{i+1} & \text{if } i < n \\ b_n & \text{if } i = n \end{cases}$$

Lemma 6: $\mathbf{b}$ is a valid non-zero unary vector iff:

1. $b_1 = 1$, and
2. Δ contains $(n-1)$ **zeros and a single** 1 (at transition index $i = b$).

Step 2: Log-Derivative Multiset Lookup

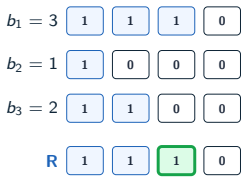
By unique fractional decomposition of logarithmic derivatives [Haböck'22], Δ has multiset $\{0^{n-1}, 1^1\}$ iff:

$$\sum_{i=1}^{n} \frac{1}{X + \Delta_i} = \frac{n-1}{X+0} + \frac{1}{X+1}.$$

1. Interpolate $f(w^i) = u(w^i) - u(w^{i+1})$ over $\mathbb{H}$.
2. At random challenge $\gamma \leftarrow \mathbb{F}$, commit to blinded reciprocal $\hat{B}(w^i) = \frac{1}{\gamma + f(w^i)}$.
3. Run **univariate sumcheck** to prove $\sum_{h \in \mathbb{H}} \hat{B}(h) = \frac{n-1}{\gamma} + \frac{1}{\gamma+1}$ and check KZG opening $u(1) = 1$.

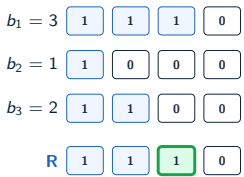
Protocol Variants: First-Price, Uniform & Sequential Auctions

1. First-Price (Unary)

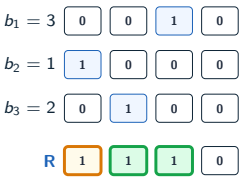


Protocol Variants: First-Price, Uniform & Sequential Auctions

1. First-Price (Unary)



2. Uniform $(p+1)$-st (One-Hot)

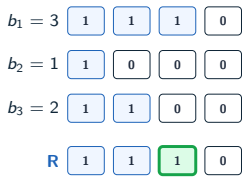


Uniform / Second-Price in a Single Run

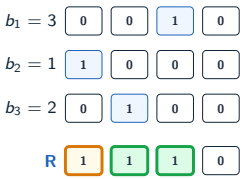
Encode bids in **one-hot Boolean** $\mathbf{b}_i = (0, \ldots, 1, \ldots, 0)$ (proved via log derivatives). Top p bidders win at the $(p+1)$-st price without re-running! Reveals anonymous bid set while keeping bidders unlinkable.

Protocol Variants: First-Price, Uniform & Sequential Auctions

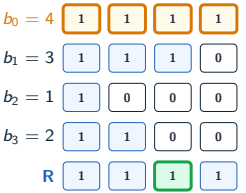
1. First-Price (Unary)



2. Uniform $(p+1)$-st (One-Hot)



3. Sequential (Masked $b_0 = n$)



Uniform / Second-Price in a Single Run

Encode bids in **one-hot Boolean** $\mathbf{b}_i = (0, \ldots, 1, \ldots, 0)$ (proved via log derivatives). Top p bidders win at the $(p+1)$-st price without re-running! Reveals anonymous bid set while keeping bidders unlinkable.

Sequential Auctions & Preprocessing Reuse

To stop the winner from testing its input to learn the runner-up bid, the auctioneer submits an all-1 masking bid $b_0 = n$ (with fraud proofs if it under-reports w).

Practical computation at 128 bidders

REPORTED CONFIGURATION: $m = 128$, $n = 1024$

$< 0.5\,\mathrm{s}$

Reported auction run

Not measured WAN latency

Rust / arkworks / BN254 · Apple M1 Max, 10 cores, 64 GB RAM

Practical computation at 128 bidders

REPORTED CONFIGURATION: $m = 128$, $n = 1024$

$< 0.5\,\mathrm{s}$

Reported auction run

Not measured WAN latency

$32\,\mathrm{KB}$

Quoted per-bidder scale

Phase and direction matter

Rust / arkworks / BN254 · Apple M1 Max, 10 cores, 64 GB RAM

Practical computation at 128 bidders

REPORTED CONFIGURATION: $m = 128$, $n = 1024$

$< 0.5\,\mathrm{s$

Reported auction run

Not measured WAN latency

32 KB

Quoted per-bidder scale

Phase and direction matter

263.82

ms bidder proving

Sum of proof costs

Rust / arkworks / BN254 · Apple M1 Max, 10 cores, 64 GB RAM

Performance Evaluation: Open-Source Rust Benchmarks

(a) Bidder Proof Generation (ms)				(b) Auctioneer Matrix Y (ms)				(c) Auctioneer Result R (ms)			
Proof	$n = 128$	1024	8192	$m \setminus n$	$n = 128$	1024	8192	$m \setminus n$	$n = 128$	1024	8192
$\pi_{\mathbf{x}_i}$ (Π_{pv})	13.59	101.51	807.21	$m = 32$	1.84	14.19	112.12	$m = 32$	0.30	6.36	50.48
$\pi_{\mathbf{r}_i}$ (Π_{nz})	2.38	10.25	58.38	$m = 128$	4.29	**38.87**	286.60	$m = 128$	1.95	**26.83**	145.06
$\pi_{\mathbf{b}_i}$ (Π_{unary})	2.53	10.68	62.03	$m = 256$	7.75	59.00	552.24	$m = 256$	4.01	32.90	265.14
$\pi_{\mathbf{z}_i}$ (Π_{sum})	27.28	141.38	953.36								
Total Proving	**45.78**	**263.82**	**1880.98**								

*Table (c) reports worst-case time (all bids minimal); expected time stops at highest w.

Sub-Second Latency & Low Bandwidth

For $m = 128$ bidders and $n = 1024$ prices, an auction finishes in < 0.5 **s** and each bidder sends/receives only ≈ 32 **KB** (4.2 MB total).

Performance Evaluation: Open-Source Rust Benchmarks

(a) Bidder Proof Generation (ms)

Proof	$n = 128$	1024	8192
$\pi_{\mathbf{x}_i}$ (Π_{pv})	13.59	101.51	807.21
$\pi_{\mathbf{r}_i}$ (Π_{nz})	2.38	10.25	58.38
$\pi_{\mathbf{b}_i}$ (Π_{unary})	2.53	10.68	62.03
$\pi_{\mathbf{z}_i}$ (Π_{sum})	27.28	141.38	953.36
Total Proving	**45.78**	**263.82**	**1880.98**

(b) Auctioneer Matrix Y (ms)

$m \setminus n$	$n = 128$	1024	8192
$m = 32$	1.84	14.19	112.12
$m = 128$	4.29	**38.87**	286.60
$m = 256$	7.75	59.00	552.24

(c) Auctioneer Result R (ms)

$m \setminus n$	$n = 128$	1024	8192
$m = 32$	0.30	6.36	50.48
$m = 128$	1.95	**26.83**	145.06
$m = 256$	4.01	32.90	265.14

*Table (c) reports worst-case time (all bids minimal); expected time stops at highest w.

Sub-Second Latency & Low Bandwidth

For $m = 128$ bidders and $n = 1024$ prices, an auction finishes in < 0.5 **s** and each bidder sends/receives only ≈ 32 **KB** (4.2 MB total).

Comparison with Addax [NSDI'23]

For $m = 128, n = 8192$, Cryptobazaar terminates in **431 ms** vs. 440 ms in Addax (96 bidders, $n = 10000$) *without* needing non-colluding auctioneers!

Conclusions, Trade-Offs, and Open Problems

Summary of Contributions

- **Cryptobazaar (NDSS 2026):** First sealed-bid auction achieving losing-bid confidentiality, public verifiability, scalability, and versatility with a **single untrusted coordinator**.
- **Reusable ZK Gadgets:** Log-derivative unary proofs (Π_{unary}), KZG-to-EC vector linking (Π_{pv}), and IPA-to-sumcheck reduction (Π_{sum}/Π_{pse}).
- **When 1-Round Non-Interactivity Is Paramount:** Use our companion **HSWE (`ia.cr/2025/443`)** construction with $O(1)$ tally decryption under a threshold beacon.

Conclusions, Trade-Offs, and Open Problems

Summary of Contributions

- **Cryptobazaar (NDSS 2026):** First sealed-bid auction achieving losing-bid confidentiality, public verifiability, scalability, and versatility with a **single untrusted coordinator**.
- **Reusable ZK Gadgets:** Log-derivative unary proofs (Π_{unary}), KZG-to-EC vector linking (Π_{pv}), and IPA-to-sumcheck reduction (Π_{sum}/Π_{pse}).
- **When 1-Round Non-Interactivity Is Paramount:** Use our companion **HSWE (`ia.cr/2025/443`)** construction with $O(1)$ tally decryption under a threshold beacon.

Limitations & Open Problems

1. **Sublinear Price-Range Scaling:** Unary encoding incurs $O(n)$ bandwidth per bidder; achieving polylog(n) communication with a single untrusted coordinator without heavy MPC/FHE remains open.
2. **Post-Quantum / Transparent Setups:** Instantiating Cryptobazaar's Polynomial IOPs with transparent or lattice/hash-based PCS.

Thank you.

Alireza Kavousi

University of Cambridge, ak2829@cam.ac.uk

Cryptobazaar Paper	ia.cr/2024/1410 (NDSS 2026)
Code	github.com/akinovak/cryptobazaar-impl
HSWE Paper	ia.cr/2025/443

Selected References

- **Novakovic, Kavousi, Gurkan, Jovanovic.** *Cryptobazaar: Private Sealed-bid Auctions at Scale*. NDSS 2026 (`https://eprint.iacr.org/2024/1410`).
- **Gurkan, Jovanovic, Kavousi, Novakovic, Seres.** *Homomorphic Signature-based Witness Encryption and Applications*. IACR ePrint 2025/443 (`https://eprint.iacr.org/2025/443`).
- **Hao and Zieliński.** *A 2-Round Anonymous Veto Protocol*. Security Protocols Workshop 2006.
- **Haböck.** *Multivariate Lookups Based on Logarithmic Derivatives*. IACR ePrint 2022/1530.
- **Bünz, Maller, Mishra, Tyagi, Vesely.** *Proofs for Inner Pairing Products and Applications*. ASIACRYPT 2021.
- **Zhong, Ma, Mao, Angel.** *Addax: A Fast, Private, and Accountable Ad Exchange Infrastructure*. USENIX NSDI 2023.
- **Bag, Hao, Shahandashti, Ray.** *SEAL: Sealed-Bid Auction Without Auctioneers*. IEEE TIFS 2019.
- **Tyagi, Arun, Freitag, Wahby, Bonneau, Mazières.** *Riggs: Decentralized Sealed-Bid Auctions*. ACM CCS 2023.
- **Glaeser, Seres, Zhu, Bonneau.** *Cicada: A Framework for Private Non-Interactive On-Chain Auctions and Voting*. IACR ePrint 2023.