

Modularity of mix-server security proofs in the CCSA logic: case of Bayer-Groth protocol

Myrto Arapinis¹ Margot Catinaud² Caroline Fontaine² Guillaume Scerri²

¹The University of Edinburgh
Edinburgh, Scotland



²Université Paris-Saclay, CNRS, ENS Paris-Saclay,
Laboratoire Méthodes Formelles (LMF)
Gif-sur-Yvette, France



Laboratoire
Méthodes
Formelles

GT MFS, March 2026, Luz-Saint-Sauveur



université
PARIS-SACLAY



Introduction – Overview of the talk

Voting protocol chronology



Voting phase



Mix



Tally

Main security properties looked for



Vote privacy



Verifiability
(universal & individual)

Introduction – Overview of the talk

Voting protocol chronology



Voting phase



Mix



Tally

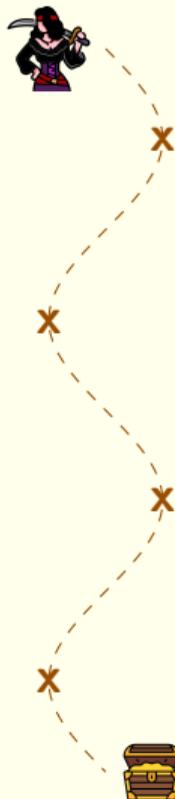
Main security properties looked for



Vote privacy



Verifiability
(universal & individual)

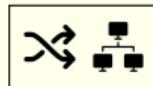


Cryptographic tools

First contribution:
Security proofs of mix-servers



Second contribution:
compose mix-servers
towards mixnets security



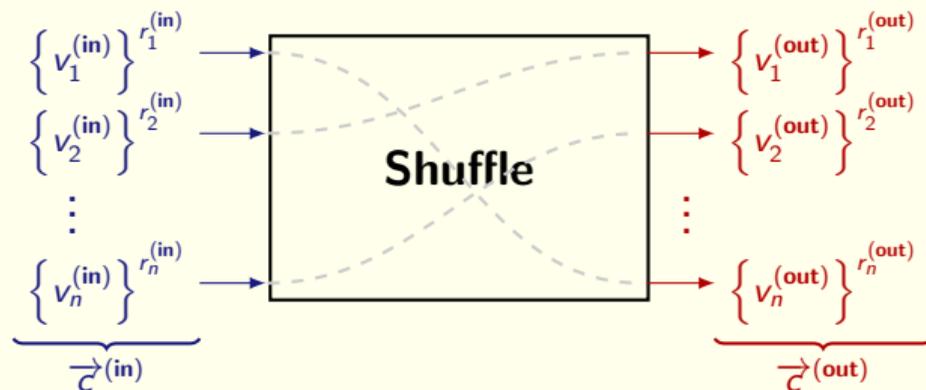
Third contribution:
From game secure mixnets
towards UC secure mixnets





– Mix-servers oriented to e-voting protocols

Mix-servers in a nutshell

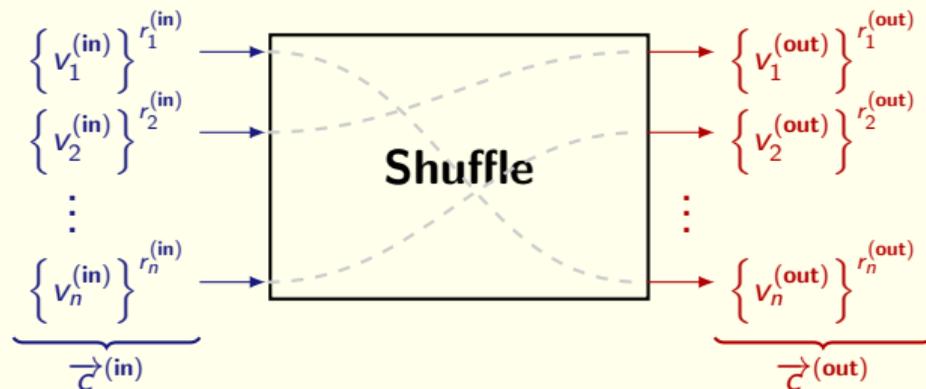


$$\exists \pi \in \mathfrak{S}_n, \forall i \in \llbracket 1; n \rrbracket, v_i^{(out)} = v_{\pi(i)}^{(in)}$$



– Mix-servers oriented to e-voting protocols

Mix-servers in a nutshell



$$\exists \pi \in \mathfrak{S}_n, \forall i \in \llbracket 1; n \rrbracket, v_i^{(out)} = v_{\pi(i)}^{(in)}$$

Security properties looked for

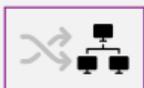
Permutation secrecy

The attacker **must not** guess the permutation used to shuffle lists



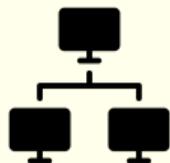
Vote integrity

Plaintexts **must stay untouched**

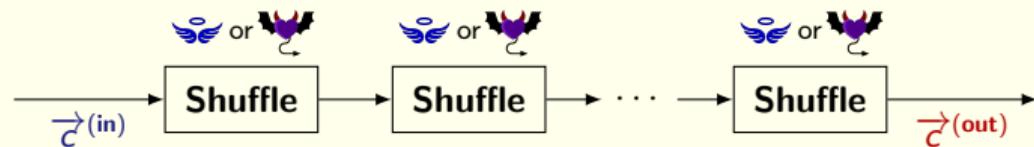


– Mixnets

Network of mix-servers



Mix-servers scheduling



Each mix-server can be

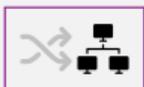


or



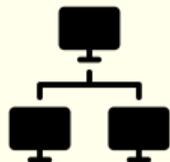
Honest

Dishonest

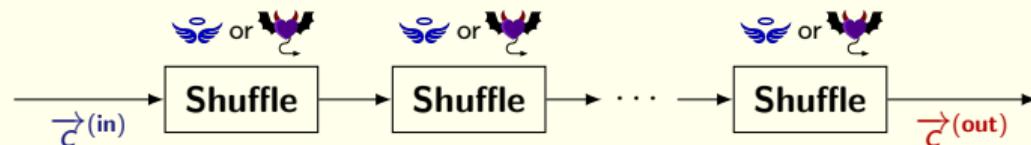


– Mixnets

Network of mix-servers



Mix-servers scheduling



Each mix-server can be



or



Honest

Dishonest

Mixnet security properties

▶ Vote confidentiality

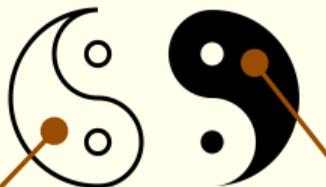
Ensured provided that *at least* one mix-server is
honest 

▶ Resistance to malicious mix-servers

Any dishonest  mix-server **should not be able**
to inject or dismiss votes



– Achieve both vote integrity and permutation secrecy: An oxymoron at first sight



Focus on vote integrity only

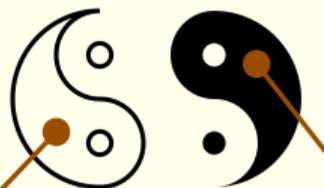
- **Idea:**
Reveal all shuffle parameters used
- **Pitfall:**
 **knows** everything
⇒ permutation secrecy hopeless

Focus on permutation secrecy only

- **Idea:**
Keep shuffle parameters
(π and $\vec{\tau}^{(out)}$) secret
- **Pitfall:**
 **can do** anything they wants
⇒ vote integrity hopeless



– Achieve both vote integrity and permutation secrecy: An oxymoron at first sight



Focus on vote integrity only

- **Idea:**
Reveal all shuffle parameters used
- **Pitfall:**
 **knows** everything
⇒ permutation secrecy hopeless

Focus on permutation secrecy only

- **Idea:**
Keep shuffle parameters
(π and $\vec{r}^{(out)}$) secret
- **Pitfall:**
 **can do** anything they wants
⇒ vote integrity hopeless

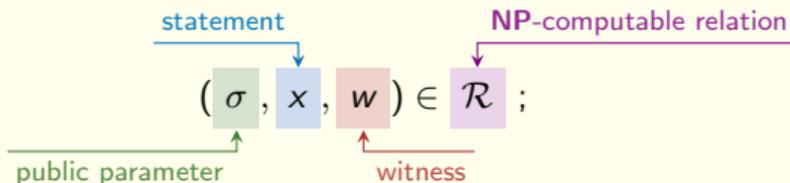
Solve the apparent unsolvable dilemma: “*prove*” shuffle **without leaking** permutation



– (Interactive) Zero-knowledge proofs

Principle

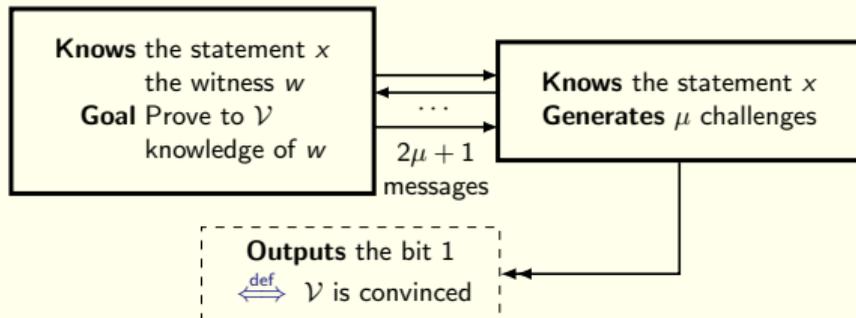
- **Two agents** a prover \mathcal{P} and a verifier \mathcal{V} ;
- **Goal:** prove equation



- **Special cases:** NIZK ($\mu = 0$), Σ -protocols ($\mu = 1$).

Prover \mathcal{P}

Verifier \mathcal{V}

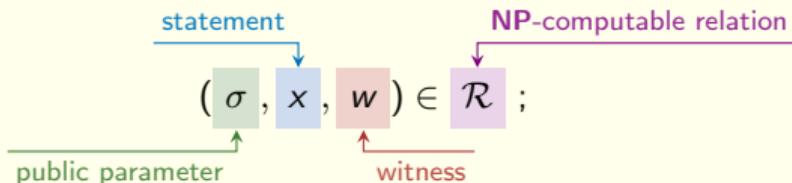




– (Interactive) Zero-knowledge proofs

Principle

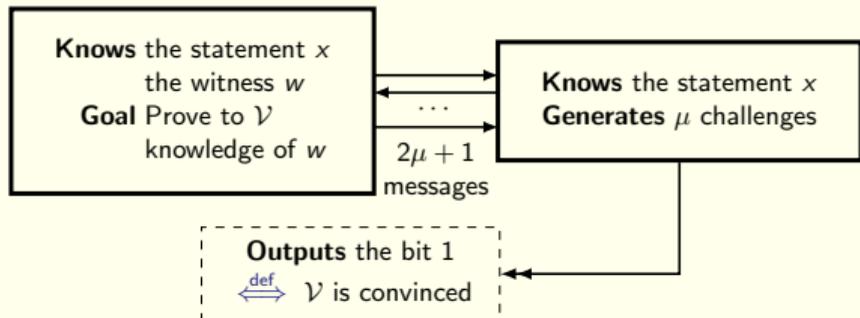
- **Two agents** a prover \mathcal{P} and a verifier \mathcal{V} ;
- **Goal:** prove equation



- **Special cases:** NIZK ($\mu = 0$), Σ -protocols ($\mu = 1$).

Prover \mathcal{P}

Verifier \mathcal{V}



Basic properties

Settings:



Zero-Knowledge



honest prover



dishonest verifier



Soundness

Settings:



dishonest prover

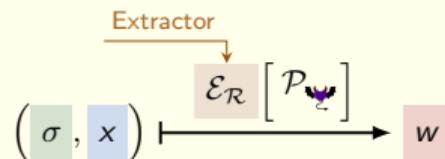


honest verifier

Usual way to prove soundness

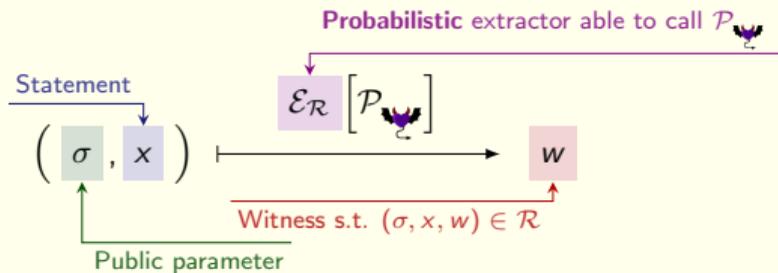


Knowledge soundness




 – Zoom on **knowledge soundness** property of ZK proofs and rewinding technique

Knowledge soundness



Warning 



Zero-Knowledge

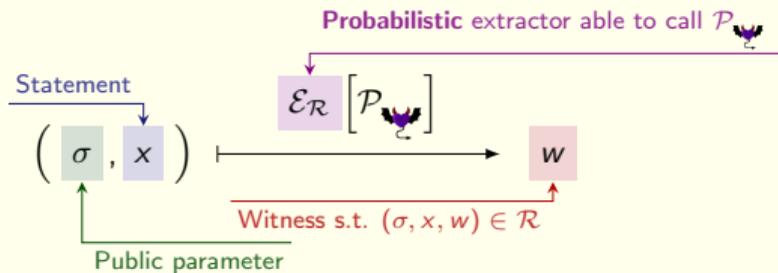


We need a "trick"



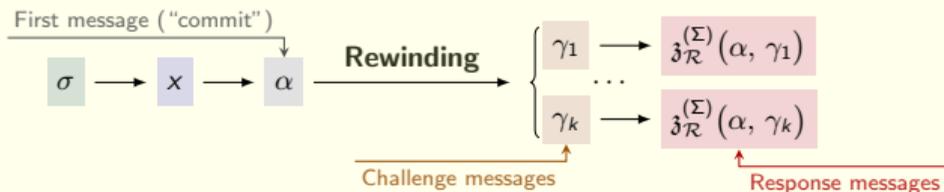
– Zoom on knowledge soundness property of ZK proofs and rewinding technique

Knowledge soundness

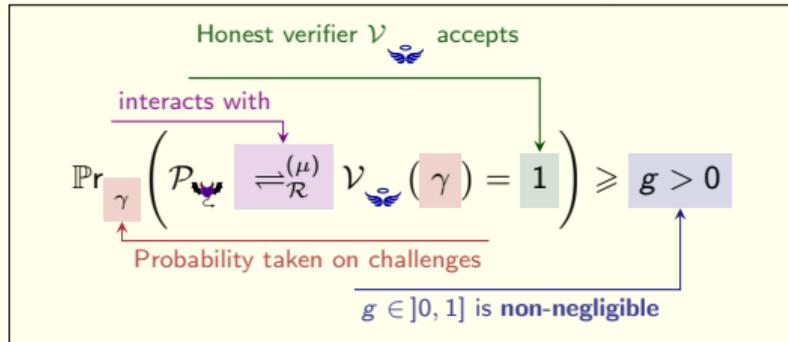


Usual “trick” for interactive ZK proofs

Use the rewinding technique – lightning example of Σ -protocols



Hypothesis made on $\mathcal{P}_{\mathcal{R}}$



Warning !

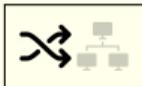
Zero-Knowledge



We need a “trick”

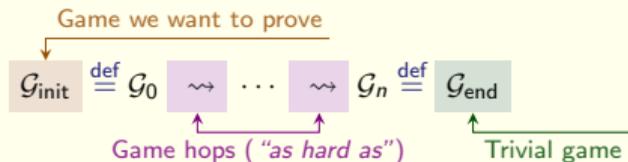


– Contributions going from *mix-servers* to *mixnets*



Mix-server security

Traditional framework used: Cryptographic games



First contribution

Are logical tools developed modular enough?

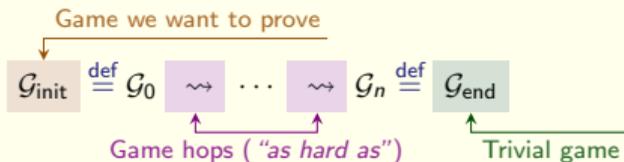


– Contributions going from *mix-servers* to *mixnets*



Mix-server security

Traditional framework used: Cryptographic games



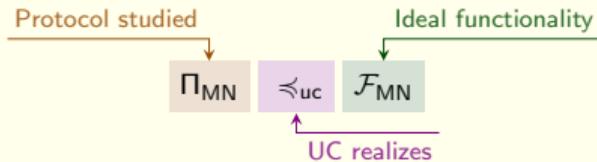
First contribution

Are logical tools developed modular enough?



Mixnet security

Traditional framework used: *Universal Composability* (UC)



Second contribution



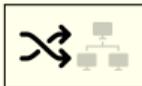
mix-server security



mixnet security

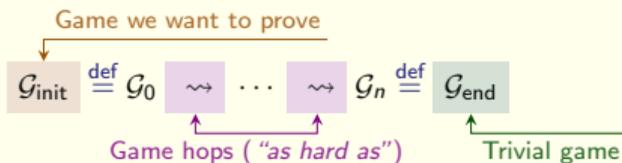


– Contributions going from *mix-servers* to *mixnets*



Mix-server security

Traditional framework used: Cryptographic games



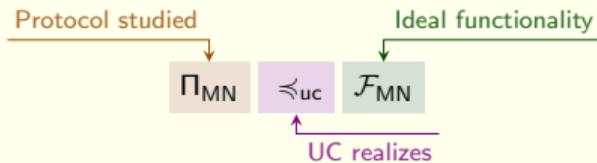
First contribution

Are logical tools developed modular enough?



Mixnet security

Traditional framework used: *Universal Composability* (UC)



Second contribution



mix-server security \implies



mixnet security



A link between frameworks

Game-based secure mixnets

UC secure mixnets

$$\Pi_{\text{MN}} \models_{\mathcal{G}} \mathcal{G}_{\text{priv}}^{(\text{MN})} \wedge \mathcal{G}_{\text{int}}^{(\text{MN})} \stackrel{\text{def}}{\iff} \Pi_{\text{MN}} \preceq_{\text{uc}} \mathcal{F}_{\text{MN}}$$

Third contribution

From **game-based secure** mixnets towards **UC secure** mixnets



– Computationally Complete Symbolic Attacker model

The CCSA model

		Syntax	Semantics $[\cdot]_{\eta}^{\rho}$
	Attacker model	Recursive function (frame)	Probabilistic Polynomial-time Turing Machines (PPTMs)
	Messages	Terms algebra (t)	Bitstrings ($[\![t]\!]_{\eta}^{\rho} \in \{0, 1\}^*$)
	Proof techniques	Rewriting / induction	Reductions (complexity-like)

Arguments in favour of this logic

- ▷ Best trade-off between **expressivity**¹ and **abstraction**;
- ▷ **Mechanisable**² in a proof assistant ($\text{SQUIRREL } \mathcal{P}$);
- ▷ Proof can be adapted to obtain **concrete security**³.



¹Baelde D., Koutsos A., Lallemand J., **A Higher-Order Indistinguishability Logic for Cryptographic Reasoning**, LICS, 2023

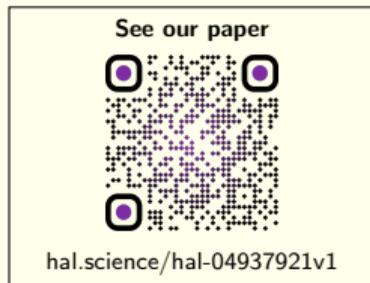
²Baelde D., Delaune S., Jacomme C., Koutsos A., Lallemand J., **The Squirrel Prover and its Logic**, ACM SIGLOG News, 2024

³Baelde D., Fontaine C., Koutsos A., Scerri G., Vignion T., **A Probabilistic Logic for Concrete Security**, CSF, 2024



– First contribution – State of the art of shuffle protocols

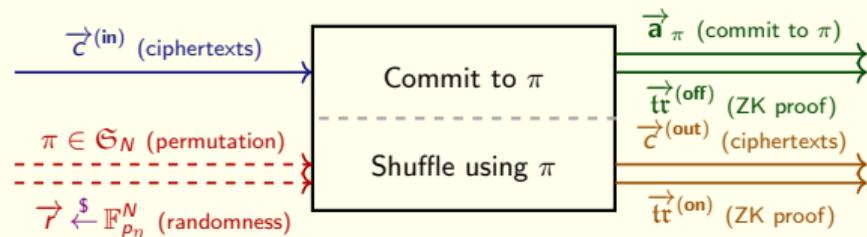
Where are we?



CCSA proof of
Terelius & Wikström
shuffle protocol
([TW10], [CFS25])

Shuffle protocols basics

Shuffle



Two zero-knowledge proofs

- Commit to a permutation:

$$\vec{tr}^{(off)} \models_{\text{ZK}} \vec{a}_\pi = \text{Com}(ck, \pi; \mathbf{s});$$

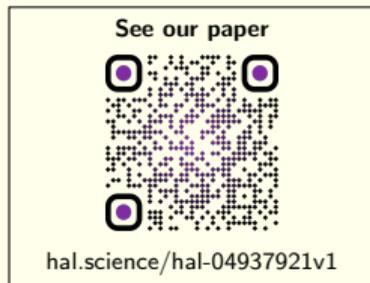
- Commitment-consistent shuffle:

$$\vec{tr}^{(on)} \models_{\text{ZK}} \forall i \in \llbracket 1; N \rrbracket, c_{\pi(i)}^{(out)} = \text{reenc} \left(c_i^{(in)}; r_i \right)$$



– First contribution – State of the art of shuffle protocols

Where are we?



CCSA proof of
Terelius & Wikström
shuffle protocol
([TW10], [CFS25])

Next target

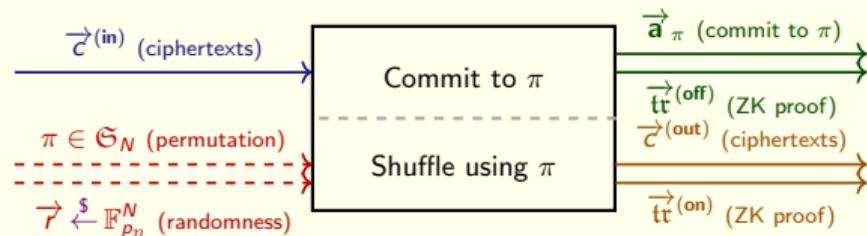


SWISSPOST

Bayer & Groth
shuffle protocol
([BG12])

Shuffle protocols basics

Shuffle



Two zero-knowledge proofs

- Commit to a permutation:

$$\vec{tr}^{(\text{off})} \models_{\text{ZK}} \vec{a}_\pi = \text{Com}(ck, \pi; \mathbf{s});$$

- Commitment-consistent shuffle:

$$\vec{tr}^{(\text{on})} \models_{\text{ZK}} \forall i \in \llbracket 1; N \rrbracket, c_{\pi(i)}^{(\text{out})} = \text{reenc} \left(c_i^{(\text{in})}; r_i \right)$$

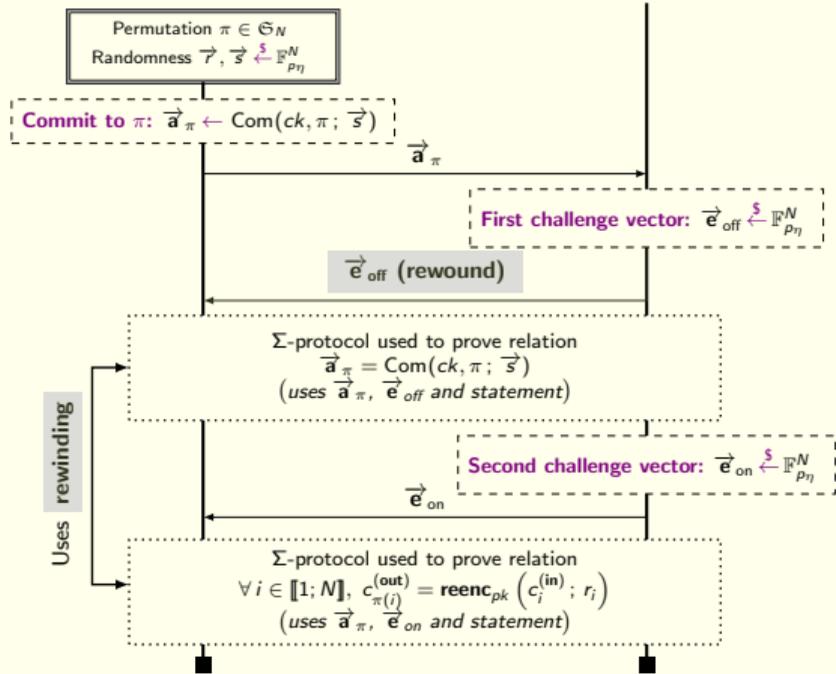


– First contribution – Comparison between TW and BG shuffle protocols

Terelius-Wikström shuffle protocol* ($\mu_{TW} = 4$)

Prover \mathcal{P}

Verifier \mathcal{V}

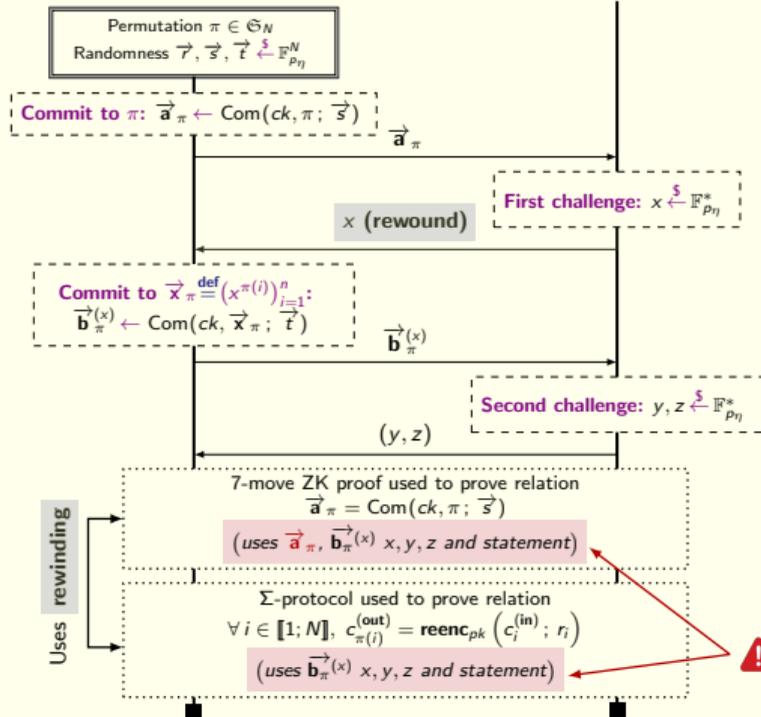


*Simplified to make them clearer

Bayer-Groth shuffle protocol* ($\mu_{BG} = 6$)

Prover \mathcal{P}

Verifier \mathcal{V}



– Second contribution – From **mix-servers** to **mixnet** security: Difficulties

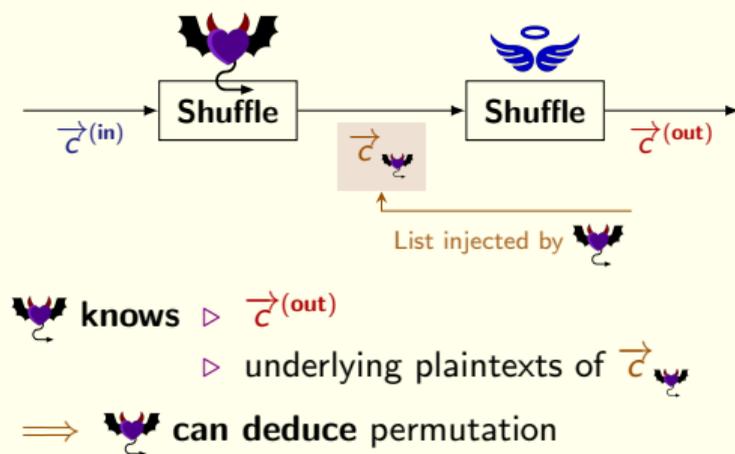
	 Mix-server	 Mixnet
Vote privacy game $\mathcal{G}_{\text{priv}}$	“Easy” (ZK property)	Hard (coming next)
Verifiability game \mathcal{G}_{int}	Hard (rewinding)	“Easy” (composition)



– Second contribution – From **mix-servers** to **mixnet** security: Difficulties

	 Mix-server	 Mixnet
Vote privacy game $\mathcal{G}_{\text{priv}}$	“Easy” (ZK property)	Hard (coming next)
Verifiability game \mathcal{G}_{int}	Hard (rewinding)	“Easy” (composition)

An attack on **confidentiality** without resistance to malicious mix-servers

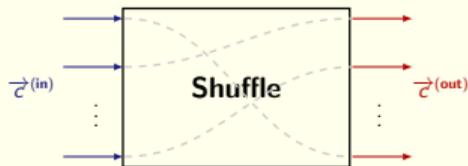




– Second contribution – Mix-servers towards mixnet security games

Mix-server security cryptographic games

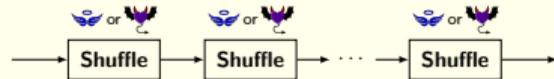
$\hookrightarrow \mathcal{G}_{\text{int}}^{(\text{MS})}$ – Integrity game



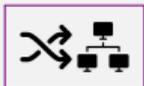
$$\mathcal{G}_{\text{int}}^{(\text{MS})} \text{ means } \left(\text{valid}_n^{(\text{in})} \xrightarrow{c^{(\text{in})}} v_{\text{in}} \right) \rightarrow \left(\text{valid}_n^{(\text{out})} \xrightarrow{c^{(\text{out})}} v_{\text{out}} \right)$$

Mixnet security cryptographic games

$\hookrightarrow \mathcal{G}_{\text{int}}^{(\text{MN})}$ – Integrity game



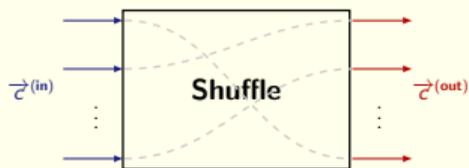
- ▶ Either for and , integrity game $\mathcal{G}_{\text{int}}^{(\text{MS})}$ holds;
- ▶ Good composition of valid_n predicates $\implies \mathcal{G}_{\text{int}}^{(\text{MN})}$ holds.



– Second contribution – Mix-servers towards mixnet security games

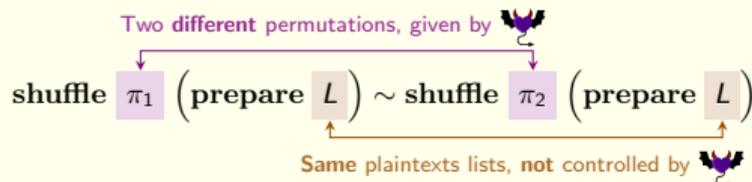
Mix-server security cryptographic games

↪ $\mathcal{G}_{\text{int}}^{(\text{MS})}$ – Integrity game



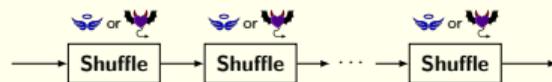
$$\mathcal{G}_{\text{int}}^{(\text{MS})} \text{ means } \left(\text{valid}_n^{(\text{in})} \xrightarrow{\mathcal{C}^{(\text{in})}} v_{\text{in}} \right) \rightarrow \left(\text{valid}_n^{(\text{out})} \xrightarrow{\mathcal{C}^{(\text{out})}} v_{\text{out}} \right)$$

↪ $\mathcal{G}_{\text{priv}}^{(\text{MS})}$ – Privacy game



Mixnet security cryptographic games

↪ $\mathcal{G}_{\text{int}}^{(\text{MN})}$ – Integrity game



- ▶ Either for and , integrity game $\mathcal{G}_{\text{int}}^{(\text{MS})}$ holds;
- ▶ Good composition of valid_n predicates $\implies \mathcal{G}_{\text{int}}^{(\text{MN})}$ holds.

↪ $\mathcal{G}_{\text{priv}}^{(\text{MN})}$ – Privacy game

- ▶ Can be reduced to the case of **only one** honest mix-server
-



– Third contribution – From games to UC

Which ideal functionality do we talk about?



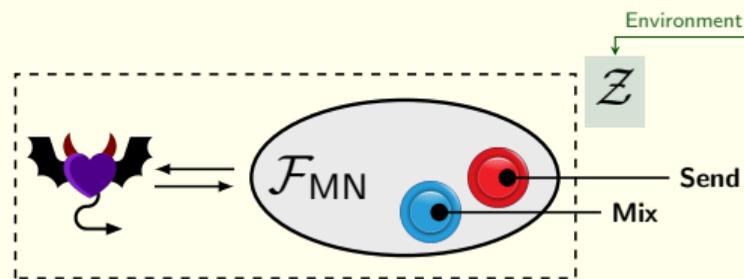
Why not mix-servers?

More naturally expressed (and easier to do)

through games: $\mathcal{G}_{\text{int}}^{(\text{MS})}$ and $\mathcal{G}_{\text{priv}}^{(\text{MS})}$



Traditional way: ideal mixnet



¹Wikström D., A Universally Composable Mixnet, TCC, 2004



– Third contribution – From games to UC

Which ideal functionality do we talk about?

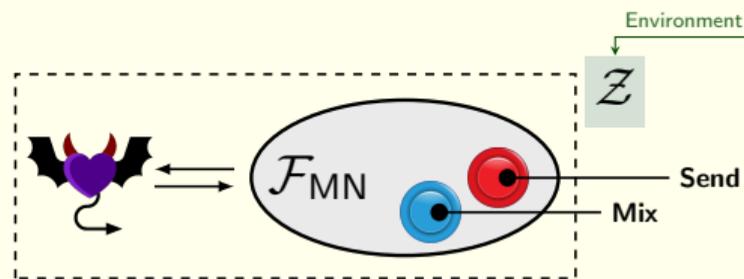


Why not mix-servers?

More naturally expressed (and easier to do)
through games: $\mathcal{G}_{\text{int}}^{(\text{MS})}$ and $\mathcal{G}_{\text{priv}}^{(\text{MS})}$



Traditional way: ideal mixnet



Ideal functionality studied¹

Ideal functionality – Ideal mixnet \mathcal{F}_{MN}

- Parameters:** $n, p \in \mathbb{N}^*$; $\theta \in]0, 1]$ mix threshold.
- Agents:** p mix-servers $(M_j)_{j=1}^p$, n senders $(S_i)_{i=1}^n$,
 ideal adversary.
- Commands:** \hookrightarrow **Send** – acts on senders;
 \hookrightarrow **Mix** – acts on mix-servers.
- Notable facts:** \triangleright Environment schedules agents;
 \triangleright Includes decryption;
 \triangleright Environment knows votes.

Main objective (still under work)

Theorem – Game-based secure toward UC secure mixnets

$$\Pi_{\text{MN}} \vDash_{\mathcal{G}} \mathcal{G}_{\text{int}}^{(\text{MN})} \wedge \mathcal{G}_{\text{priv}}^{(\text{MN})} \iff \Pi_{\text{MN}} \preceq_{\text{uc}} \mathcal{F}_{\text{MN}}$$

¹Wikström D., A Universally Composable Mixnet, TCC, 2004

Key take-aways and prospectives



Key take aways

3 main research axis

- ▷ **(Mix-servers)** Security proofs of shuffle protocols;
- ▷ **(Composition)** From mix-servers security to mixnets security;
- ▷ **(Mixnets)** UC realization of ideal functionality \mathcal{F}_{MN} .



Prospectives

Key take-aways and prospectives



Key take aways

3 main research axis

- ▷ **(Mix-servers)** Security proofs of shuffle protocols;
- ▷ **(Composition)** From mix-servers security to mixnets security;
- ▷ **(Mixnets)** UC realization of ideal functionality \mathcal{F}_{MN} .

2 cryptographic/logic frameworks used

- ▷ *Universal composability* (UC)
- ▷ *Computationally Complete Symbolic Attacker* (CCSA)

2 security properties studied

Vote privacy & Verifiability



Prospectives

Key take-aways and prospectives



Key take aways

3 main research axis

- ▷ **(Mix-servers)** Security proofs of shuffle protocols;
- ▷ **(Composition)** From mix-servers security to mixnets security;
- ▷ **(Mixnets)** UC realization of ideal functionality \mathcal{F}_{MN} .

2 cryptographic/logic frameworks used

- ▷ *Universal composability* (UC)
- ▷ *Computationally Complete Symbolic Attacker* (CCSA)

2 security properties studied

Vote privacy & Verifiability

Success

- ▷ CCSA axiom of **rewinding** technique quite **suitable** (\o/ yay!).



Prospectives

Key take-aways and prospectives



Key take aways

3 main research axis

- ▷ **(Mix-servers)** Security proofs of shuffle protocols;
- ▷ **(Composition)** From mix-servers security to mixnets security;
- ▷ **(Mixnets)** UC realization of ideal functionality \mathcal{F}_{MN} .

2 cryptographic/logic frameworks used

- ▷ *Universal composability* (UC)
- ▷ *Computationally Complete Symbolic Attacker* (CCSA)

2 security properties studied

Vote privacy & Verifiability

Success

- ▷ CCSA axiom of **rewinding** technique quite **suitable** (\o/ yay!).



Prospectives

Zero-knowledge proofs

- ▷ From interactive ZK proofs to Non-Interactive ones (through **Fiat-Shamir transform**).

CCSA logic

- ▷ Handle **re-programmable Random Oracles**

E-voting protocols

- ▷ Look at **interactions** between mixnets and other parts of e-voting protocols

Key take-aways and prospectives



Key take aways

3 main research axis

- ▷ **(Mix-servers)** Security proofs of shuffle protocols;
- ▷ **(Composition)** From mix-servers security to mixnets security;
- ▷ **(Mixnets)** UC realization of ideal functionality \mathcal{F}_{MN} .

2 cryptographic/logic frameworks used

- ▷ *Universal composability* (UC)
- ▷ *Computationally Complete Symbolic Attacker* (CCSA)

2 security properties studied

Vote privacy & Verifiability

Success

- ▷ CCSA axiom of **rewinding** technique quite **suitable** (\o/ yay!).



Prospectives

Zero-knowledge proofs

- ▷ From interactive ZK proofs to Non-Interactive ones (through **Fiat-Shamir transform**).

CCSA logic

- ▷ Handle **re-programmable Random Oracles**

E-voting protocols

- ▷ Look at **interactions** between mixnets and other parts of e-voting protocols



Thank you for your attention!



mcatinaud.gitlab.io

Bibliography

CCSA logic related

- [BKL23] | *Baelde D., Koutsos A., Lallemand J., A Higher-Order Indistinguishability Logic for Cryptographic Reasoning*, LICS, 2023
- [BDJKL24] | *Baelde D., Delaune S., Jacomme C., Koutsos A., Lallemand J., The Squirrel Prover and its Logic*, ACM SIGLOG News, 2024
- [BFKSV24] | *Baelde D., Fontaine C., Koutsos A., Scerri G., Vignon T., A Probabilistic Logic for Concrete Security*, CSF, 2024

Cryptographic papers

- [TW10] | *Terelius B., Wikström D., Proof of restricted shuffles*, AFRICACRYPT, 2010
- [BG12] | *Bayer S., Groth J., Efficient zero-knowledge argument for correctness of a shuffle*, EUROCRYPT, 2012

E-voting protocols papers

- | *Glondou S., Belenios specification*, Version 3.0
- | *Swiss Post, Swiss Post Voting System – System specification*, Version 1.5.2, 2025

UC related papers

- [W04] | *Wikström D., A Universally Composable Mix-Net*, TCC, 2004
- [CS25] | *Chevalier C., Sageloli E., Diving Deep Into UC: Uncovering and Resolving Issues in Universal Composability*, IACR Cryptol. ePrint Arch, 2025

Our paper

- [CFS25] | *Catinaud M., Fontaine C., Scerri G., Proving E-voting Mixnets in the CCSA model: Zero-Knowledge proofs and Rewinding*, HAL Arch, 2025

Most of icons come from FLATICON