Proof of a mixnet in the Computationally Complete Symbolic Attacker (CCSA) model

Margot Catinaud [*] Caroline Fontaine [*] Guillaume Scerri [*]

[*] Université Paris-Saclay, CNRS, ENS Paris-Saclay,
Laboratoire Méthodes Formelles (LMF)

SVP day, 6th February 2024

Electronic voting protocol

Election protocol

Step 1: Election setup

Step 2: Voting phase

Step 3: Tally

Electronic voting protocol

Election protocol

Step 1: Election setup

Step 2: Voting phase

Step 3: Tally

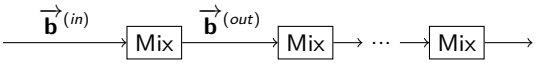
Two kinds of tally

Homomorphic encryption

Mix networks + Decrypt

Mix networks

Principle



Network of *mix-servers*

Algorithm : Mixing

let `mixing` $\overrightarrow{\mathbf{b}}^{(in)}$ **=**

$\pi \xleftarrow{\$} \mathfrak{S}_N$;

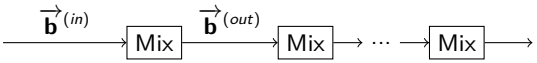
[*do some stuff...*] ;

return $\overrightarrow{\mathbf{b}}^{(out)}$

Mix-server in a nutshell

Mix networks

Principle



Network of *mix-servers*

Algorithm : Mixing

let `mixing` $\overrightarrow{\mathbf{b}}^{(in)}$ **=**

$\pi \xleftarrow{\$} \mathfrak{S}_N$;

[*do some stuff...*] ;

return $\overrightarrow{\mathbf{b}}^{(out)}$

Mix-server in a nutshell

Security properties for one router

Ballot privacy

Verifiability

Commitment schemes

Principle

Security properties

Hiding

Binding

Zero-knowledge proofs

Principle

- Two agents: a prover $\mathcal{P}$ and a verifier $\mathcal{V}$
- Goal: prove that $(\underbrace{x}_{\text{statement}} , \underbrace{w}_{\text{witness}}) \in \mathcal{R}$

Interactive vs. Non Interactive protocols

- Sigma-protocol: proof transcript

$$(\underbrace{p_0}_{\text{first message}} , \underbrace{c}_{\text{challenge}} , \underbrace{p_1}_{\text{response}})$$

- NIZK proof

Main security properties

Soundness

Zero-knowledge

Additionnal property

Extractibility

Terelius & Wikström protocol

Two parts ...

- *Offline phase:* Choose $\pi \xleftarrow{\$} \mathfrak{S}_N$ and output $\mathbf{a} = \mathrm{Com}_{\mathbb{Z}_{q_\eta}^{N \times N}}\big(ck, M_\pi\,;\,\mathbf{s}\big)$.
- *Online phase:* Output $\overrightarrow{\mathbf{b}}^{(out)} = \mathrm{ReRand}\big(M_\pi \cdot \overrightarrow{\mathbf{b}}^{(in)}\big)$

Terelius & Wikström protocol

Two parts ...

- *Offline phase:* Choose $\pi \xleftarrow{\$} \mathfrak{S}_N$ and output $\mathbf{a} = \mathsf{Com}_{\mathbb{Z}_{q_\eta}^{N\times N}}\big(ck, M_\pi\,;\,\mathbf{s}\big)$.
- *Online phase:* Output $\overrightarrow{\mathbf{b}}^{(out)} = \mathsf{ReRand}\big(M_\pi \cdot \overrightarrow{\mathbf{b}}^{(in)}\big)$

... with two distinct zero-knowledge proofs

- ZK proof of the offline phase

$$((ck, \mathbf{a}), (\pi, \mathbf{s})) \in \mathcal{R}_{\mathrm{com}} \iff \mathbf{a} = \mathsf{Com}_{\mathbb{Z}_{q_\eta}^{N\times N}}\big(ck, M_\pi\,;\,\mathbf{s}\big)$$

- ZK proof of the online phase

$$\Big((ck, \mathbf{a}, \overrightarrow{\mathbf{b}}^{(in)}, \overrightarrow{\mathbf{b}}^{(out)}), (\pi, \mathbf{r})\Big) \in \mathcal{R}_{\mathrm{mix}} \iff \overrightarrow{\mathbf{b}}^{(out)} = \mathsf{ReRand}\big(M_\pi \cdot \overrightarrow{\mathbf{b}}^{(in)}; \mathbf{r}\big)$$

- Almost sigma-protocols: there are 2 rounds.

Verifiability game

Cryptographic game - Mix-server verifiability.

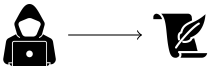
Context

Adversarial mix-server

Honest verifier $\mathcal{V}$

Game statement

Hypothesis



Proofs accepted by $\mathcal{V}$

$\Longrightarrow$

Conclusion

$$\left\{\text{Dec } \overrightarrow{\mathbf{b}}^{(in)}\right\} = \left\{\text{Dec } \overrightarrow{\mathbf{b}}^{(out)}\right\}$$

Equality of plaintexts lists as multisets

Sketch of proof

Extraction of sealed matrix M

- Witness extractor
- Collect enough witness
- Reconstruction of sealed informations

Is M a permutation matrix?

$\overrightarrow{\mathbf{b}}^{(out)} = \mathrm{ReRand}\left(M \cdot \overrightarrow{\mathbf{b}}^{(in)}\right)$**?**

Special-Soundness + Rewinding

Rewinding

Linear algebra

Sketch of proof

Extraction of sealed matrix M

- Witness extractor
- Collect enough witness
- Reconstruction of sealed informations

Special-Soundness + Rewinding
Rewinding
Linear algebra

Is M a permutation matrix?

- Witness extractor
- Witness consistency
- Generalization of equations on witness to equations on matrix
- Characterization of permutation matrix

Special-Soundness + Rewinding
Binding
Schwartz-Zippel lemma
Algebra

$\overrightarrow{\mathbf{b}}^{(out)} = \mathrm{ReRand}\left(M \cdot \overrightarrow{\mathbf{b}}^{(in)}\right)$**?**

Sketch of proof

Extraction of sealed matrix M

- Witness extractor — Special-Soundness + Rewinding
- Collect enough witness — Rewinding
- Reconstruction of sealed informations — Linear algebra

Is M a permutation matrix?

- Witness extractor — Special-Soundness + Rewinding
- Witness consistency — Binding
- Generalization of equations on witness to equations on matrix — Schwartz-Zippel lemma
- Characterization of permutation matrix — Algebra

$\overrightarrow{\mathbf{b}}^{(out)} = \mathrm{ReRand}\left(M \cdot \overrightarrow{\mathbf{b}}^{(in)}\right)$?

- Another witness extractor — Special-Soundness + Rewinding
- Consistency between the witness and the extracted matrix — Binding
- Generalization to the whole set of ciphertexts in/out pairs — Group theory algebra

Sketch of proof

Extraction of sealed matrix M

- *Witness extractor* — Special-Soundness + Rewinding
- *Collect enough witness* — Rewinding
- *Reconstruction of sealed informations* — Linear algebra

Is M a permutation matrix?

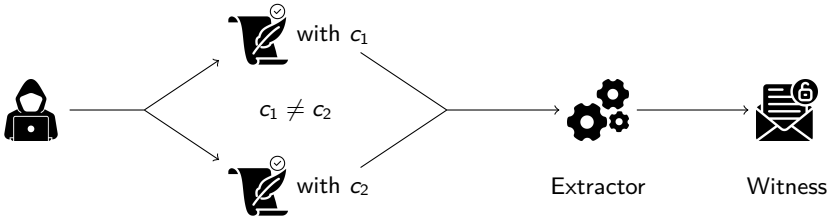
- Witness extractor — Special-Soundness + Rewinding
- Witness consistency — Binding
- Generalization of equations on witness to equations on matrix — Schwartz-Zippel lemma
- Characterization of permutation matrix — Algebra

$\overrightarrow{\mathbf{b}}^{(out)} = \mathrm{ReRand}\left(M \cdot \overrightarrow{\mathbf{b}}^{(in)}\right)$?

- Another witness extractor — Special-Soundness + Rewinding
- Consistency between the witness and the extracted matrix — Binding
- Generalization to the whole set of ciphertexts in/out pairs — Group theory algebra

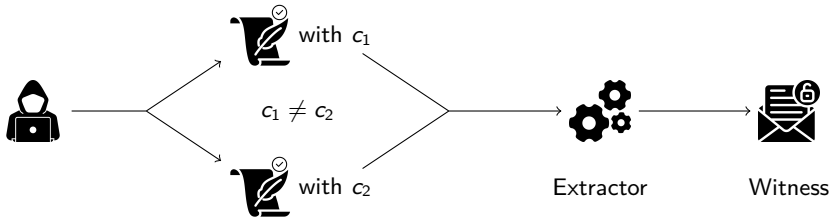
Zoom on the proof: Witness extraction

What we want to use: *Special-Soundness*

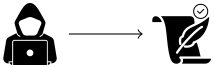


Zoom on the proof: Witness extraction

What we want to use: *Special-Soundness*



What we have



Only one proof!

Solution

Rewinding

Zoom on the proof: Extraction of sealed matrix M

- Witness extraction algorithm:

Algorithm : Witness extraction

Input: Adversary $\mathcal{A}$ producing sometimes a proof accepted by the verifier $\mathcal{V}$.

$\mathcal{V}$ chooses a vector $\mathbf{e} \xleftarrow{\$} \mathbb{Z}_{q_\eta}^N$ and then sends it to $\mathcal{A}$;

repeat

  Run $\mathbf{p}_1 := (p_0, c_1, p_1) \leftarrow \mathcal{A}(x, \mathbf{e}, c_1)$;

  Rewind $\mathcal{A}$;

  Run $\mathbf{p}_2 := (p_0, c_2, p_2) \leftarrow \mathcal{A}(x, \mathbf{e}, c_2)$;

  Check if **true** $\leftarrow \mathcal{V}(x, \mathbf{p}_1)$ and **true** $\leftarrow \mathcal{V}(x, \mathbf{p}_2)$;

until $\mathbf{p}_1$ *and* $\mathbf{p}_2$ *are accepted by* $\mathcal{V}$ *and* $c_1 \neq c_2$;

return $w \leftarrow \mathtt{extract}_{\mathcal{R}}(x, \mathbf{p}_1, \mathbf{p}_2)$;

- Next, collect witnesses for enough vectors to have $(\mathbf{e}_1, \ldots, \mathbf{e}_N)$ a free family of $\mathbb{Z}_{q_\eta}^N$.
- Finally, use linear algebra (Gaussian elimination) to rebuild the matrix M

Why use the CCSA logic?

We need automatic tools...

- Proofs of mix-servers security are long and tough
- Existing pen-and-paper proofs are unsatisfying
- Need of computational guarantees

Why use the CCSA logic?

We need automatic tools...

- Proofs of mix-servers security are long and tough
- Existing pen-and-paper proofs are unsatisfying
- Need of computational guarantees

... But which ones?

- Hard to model rewinding technique in CryptoVerif
- Proofs in EasyCrypt too complicated

Why use the CCSA logic?

We need automatic tools...

- Proofs of mix-servers security are long and tough
- Existing pen-and-paper proofs are unsatisfying
- Need of computational guarantees

... But which ones?

- Hard to model rewinding technique in CryptoVerif
- Proofs in EasyCrypt too complicated

Solution! The *Computationally Complete Symbolic Attacker* (CCSA) logic

- Axioms model arguments of the proof
- Lemmas model steps of the proof
- Axioms quite easy to write and to handle

Rebuild the sealed matrix step in the CCSA logic

Lemmas

- Existence of a witness extractor
  If we have an adversary $\mathcal{A}$ which gaves *sometimes* a proof accepted by the verifier $\mathcal{V}$ *then* we can construct an extractor computing witnesses from this adversary $\mathcal{A}$.

- Rebuild the sealed matrix M
  If we have collected enough witnesses for a vector basis $(\mathbf{e}_1, \ldots, \mathbf{e}_N)$ *then* we can rebuild the sealed matrix M

Axioms

- Special-soundness
  There exists $\texttt{extract}_{\mathcal{R}}$ (ptime) such that

$$\frac{\bigwedge_{i \in \{1,2\}} \texttt{verify}_{\mathcal{R}}(x, \underbrace{(p_0, c_i, p_i)}_{\mathbf{p}_i}) \qquad c_1 \neq c_2}{(x, \texttt{extract}_{\mathcal{R}}(x, \mathbf{p}_1, \mathbf{p}_2)) \in \mathcal{R}}$$

- Basis axiom

$$\frac{\mathbf{indep}(\mathbf{e}_1, \ldots, \mathbf{e}_N)}{\mathbf{basis}_{\mathbb{Z}_{q_\eta}^N}(\mathbf{e}_1, \ldots, \mathbf{e}_N)}$$

Prospectives

Modularity ...

- Zero-knowledge proofs
- Commitment schemes
- ...

Prospectives

Modularity ...

- Zero-knowledge proofs
- Commitment schemes
- ...

... Towards formal proofs of e-voting protocols

- Sigma-protocols $\rightarrow$ NIZK proof
- Compose mix-servers security to obtain mixnets security
- Implement the proof in SQUIRREL

Prospectives

Modularity ...

- Zero-knowledge proofs
- Commitment schemes
- ...

... Towards formal proofs of e-voting protocols

- Sigma-protocols → NIZK proof
- Compose mix-servers security to obtain mixnets security
- Implement the proof in SQUIRREL

Thank you for your attention![1]

[1] Icons comes from the Flaticons website (https://www.flaticon.com/)