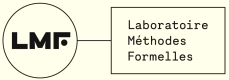
Make cryptographic proofs more formal: case of Zero-Knowledge proofs and rewinding

Margot Catinaud[1]

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



Seminar @ GRACE, July 2026, Palaiseau

Motivation – E-voting protocols and mixnets

Voting protocol chronology

Voting phase

Mix

Tally

Main security properties looked for

Vote privacy

Verifiability
(universal & individual)

Motivation – E-voting protocols and mixnets

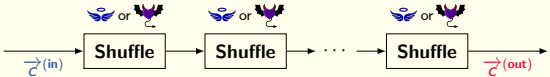
Voting protocol chronology

Voting phase

Mix

Tally

Mix-servers scheduling



Legend: honest agent – dishonest agent

Main security properties looked for

Vote privacy

Verifiability
(universal & individual)

Mixnet security properties

- **Vote confidentiality**

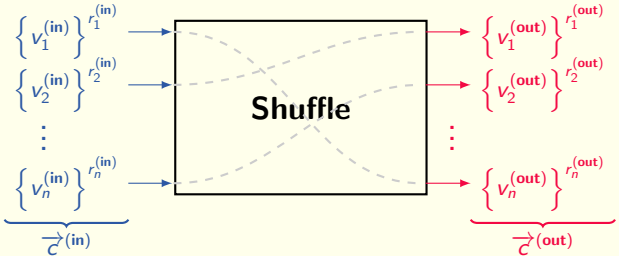
  Ensured provided that *at least* one mix-server is honest

- **Resistance to malicious mix-servers**

  No dishonest mix-server **should be able to** inject or dismiss votes

Motivation – Build mixnets through shuffle protocols

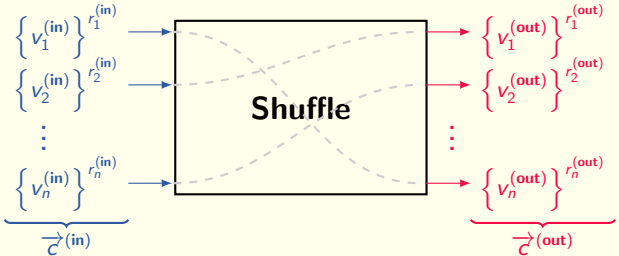
Main ingredients for mixnets: shuffle protocols



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

Motivation – Build mixnets through shuffle protocols

Main ingredients for mixnets: shuffle protocols



$$\exists \pi \in \mathfrak{S}_n,\ \forall i \in [\![1; n]\!],\ v_i^{(\mathbf{out})} = v_{\pi(i)}^{(\mathbf{in})}$$

Security properties looked for

Permutation secrecy

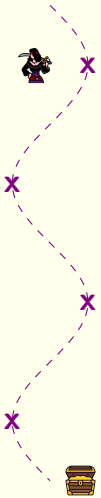
The attacker **must not guess** the permutation used to shuffle lists

Vote integrity

Plaintexts **must** remain **untouched**

$\Longrightarrow$ We need **zero-knowledge proofs**

Overview of the talk



I. Settings of zero-knowledge proofs

II. Brief overview of formal methods world – CCSA logic

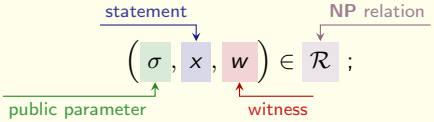
III. How to model knowledge soundness and rewinding in the CCSA logic

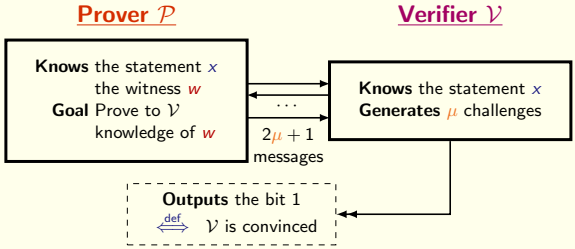
IV. Other contributions and conclusion

Settings of (Interactive) Zero-Knowledge proofs

Which kind of ZK proofs?

Goal: prove equation





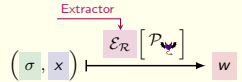
Which properties?

- **(Perfect completeness)**
  Nothing to say, functional correctness
- **Permutation secrecy – Zero-Knowledge**

Honest-Verifier
Zero-Knowledge

"Easy" \o/

- **Vote integrity – Knowledge soundness**

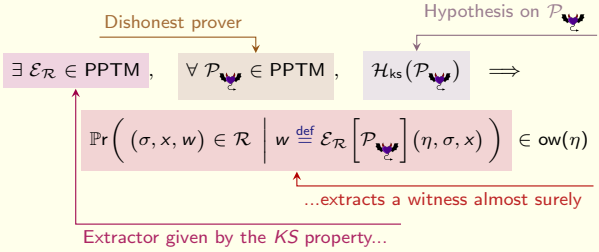


Work to do

Note: Used to prove *soundness* (implying *integrity*)

Focus on *Knowledge soundness* property

Basics of Knowledge soundness



Focus on *Knowledge soundness* property

Basics of Knowledge soundness

Dishonest prover

Hypothesis on $\mathcal{P}$

$$\exists\, \mathcal{E}_{\mathcal{R}} \in \mathsf{PPTM}\,, \quad \forall\, \mathcal{P} \in \mathsf{PPTM}\,, \quad \mathcal{H}_{\mathsf{ks}}(\mathcal{P}) \implies$$

$$\mathbb{P}\mathrm{r}\left((\sigma, x, w) \in \mathcal{R} \;\middle|\; w \overset{\text{def}}{=} \mathcal{E}_{\mathcal{R}}\left[\mathcal{P}\right](\eta, \sigma, x) \right) \in \mathsf{ow}(\eta)$$

...extracts a witness almost surely

Extractor given by the *KS* property...

Hypothesis $\mathcal{H}_{\mathsf{ks}}(\mathcal{P})$ we need

Honest verifier $\mathcal{V}$ accepts

interacts with

$$\mathbb{P}\mathrm{r}_{\gamma}\left(\mathcal{P} \rightleftharpoons^{(\mu)}_{\mathcal{R}} \mathcal{V}(\gamma) = 1 \right) \geqslant g > 0$$

Probability taken on challenges

$g \in]0, 1]$ is **non-negligible**

Focus on *Knowledge soundness* property

Basics of Knowledge soundness

Dishonest prover — Hypothesis on $\mathcal{P}$

$$\exists\, \mathcal{E}_{\mathcal{R}} \in \mathrm{PPTM}\,, \quad \forall\, \mathcal{P} \in \mathrm{PPTM}\,, \quad \mathcal{H}_{\mathsf{ks}}(\mathcal{P}) \implies$$

$$\mathbb{P}\mathrm{r}\left((\sigma, x, w) \in \mathcal{R} \;\middle|\; w \overset{\text{def}}{=} \mathcal{E}_{\mathcal{R}}\left[\mathcal{P}\right](\eta, \sigma, x) \right) \in \mathsf{ow}(\eta)$$

...extracts a witness almost surely

Extractor given by the *KS* property...

Hypothesis $\mathcal{H}_{\mathsf{ks}}(\mathcal{P})$ we need

Honest verifier $\mathcal{V}$ accepts — interacts with

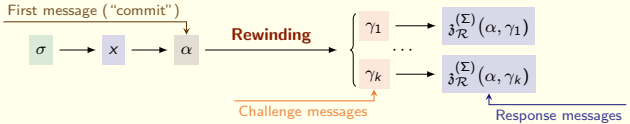
$$\mathbb{P}\mathrm{r}_{\gamma}\left(\mathcal{P} \rightleftharpoons^{(\mu)}_{\mathcal{R}} \mathcal{V}(\gamma) = 1 \right) \geqslant g > 0$$

Probability taken on challenges

$g \in]0, 1]$ is **non-negligible**

Usual way to build $\mathcal{E}_{\mathcal{R}}$

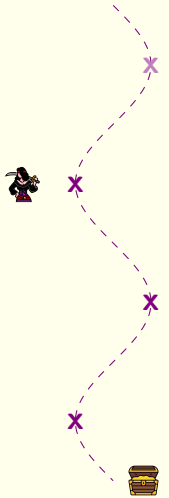
Use the rewinding technique – lightning example of Σ-protocols



Observations on Knowledge soundness

- $\mathcal{E}_{\mathcal{R}}$, a **probabilistic** extractor
- Probability taken on *honest* challenges (on random tape of $\mathcal{V}$)
- ⚠ $\mathcal{P}$ is **deterministic** (fixed random tape)

Overview of the talk



I. Settings of zero-knowledge proofs

II. Brief overview of formal methods world – CCSA logic

III. How to model knowledge soundness and rewinding in the CCSA logic

IV. Other contributions and conclusion

"More formal" you say?

What does it mean? – 2 examples

Computer-aided proofs
(proof assistants)

Fully automatic tools

In general, **increases**
complexity of doing proofs

"More formal" you say?

What does it mean? – 2 examples

Computer-aided proofs
(proof assistants)

Fully automatic tools

In general, **increases** complexity of doing proofs

Why bother to use them?

- **Ease** the scaling to wide and concrete protocols: TLS 1.3, Signal, E-voting protocols, ...
- **Modularity:** *Plug-and-play* cryptographic primitives
- **Translators** between cryptographers and other research communities, *e.g.* proofs of programs

"More formal" you say?

What does it mean? – 2 examples

Computer-aided proofs
(proof assistants)

Fully automatic tools

In general, **increases** complexity of doing proofs

Our specific needs

- Fine-grained **probabilistic** reasoning
- **Computational** guarantees
- Level of **abstraction** higher as possible

Why bother to use them?

- **Ease** the scaling to wide and concrete protocols: TLS 1.3, Signal, E-voting protocols, ...
- **Modularity:** *Plug-and-play* cryptographic primitives
- **Translators** between cryptographers and other research communities, *e.g.* proofs of programs

"More formal" you say?

What does it mean? – 2 examples

Computer-aided proofs
(proof assistants)

Fully automatic tools

In general, **increases** complexity of doing proofs

Why bother to use them?

- **Ease** the scaling to wide and concrete protocols: TLS 1.3, Signal, E-voting protocols, ...
- **Modularity:** *Plug-and-play* cryptographic primitives
- **Translators** between cryptographers and other research communities, *e.g.* proofs of programs

Our specific needs

- Fine-grained **probabilistic** reasoning
- **Computational** guarantees
- Level of **abstraction** higher as possible

A model at a crossroad of our needs

The *Computationally Complete Symbolic Attacker* (CCSA) model

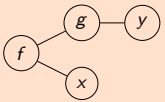
The `Squirrel` prover

Panorama of tools for cryptographic protocols security

The two complementary and idiomatic models of attacker

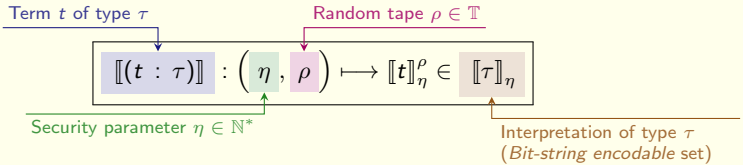
	Symbolic model	Computational model
Attacker capabilities	**Positive** definition – add capabilities (Dolev-Yao like)	**Negative** definition – remove capabilities
Strengths	**Easily** automated	More **precise** attacker
Weaknesses	**Weaker** attacker	**Hardly** automated
Idiomatic tools	ProVerif, Tamarin	CryptoVerif, Easycrypt
Suitable for	Cryptographic **protocols** – TLS 1.3, Signal, 5G – AKA, SwissPost	Cryptographic **primitives** – ▷ Encryption schemes ▷ **Low level protocols (ZK proofs)**

Introduction to CCSA – CCSA: An hybrid model ([Baelde, Koutsos, Lallemand'23])

	Syntax	Semantics ($[\![\cdot]\!]_\eta$)
Attacker model	Recursive functions (frame)	Probabilistic Polynomial-time Turing Machines (PPTMs)
Messages	Simply-typed term algebra ($t : \tau$)	Bitstrings ($[\![t]\!]^\rho_\eta \in \{0,1\}^*$)
Proof techniques	Rewriting / induction inference rules	Polynomial-time reductions
	 $f(x, g(y))$ $\dfrac{f(x, g(y)) \qquad y}{x}$	$\eta \longmapsto \mathbb{P}\mathsf{r}_\rho\Big([\![\phi]\!]^\rho_\eta = 1\Big)$ is negligible in η

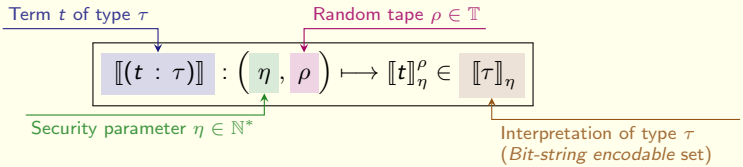
Introduction to CCSA – Deep into semantics subtleties

Semantics $[\![\cdot]\!] : \mathbb{N}^* \times \mathbb{T} \longrightarrow [\![\tau]\!]$ (interpretation of terms as random variables)

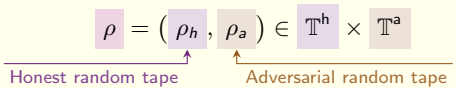


Introduction to CCSA – Deep into semantics subtleties

Semantics $[\![\cdot]\!] : \mathbb{N}^* \times \mathbb{T} \longrightarrow [\![\tau]\!]$ (interpretation of terms as random variables)



About random tapes of $\mathbb{T}$



- **Honest tape** –
  uniform and independent randomness
  e.g. random tape of the honest verifier $\mathcal{V}$
- **Adversarial tape** –
  all randomness can be *dependent*
  e.g. random tape of a PPTM adversarial prover $\mathcal{P}$

Introduction to CCSA – How to prove using this logic?

Two main predicates: $\sim$ and $[\cdot]$

- **Predicate for indistinguishability games**
  - ↪ *e.g. Honest-Verifier Zero-Knowledge*

Indistinguishability of terms predicate

$$u \sim v \quad \text{means} \quad \Big| \mathbb{P}\mathrm{r}_\rho\Big(\mathcal{A}_{\tilde{\ }}([\![u]\!]^\rho_\eta) = 1\Big) - \mathbb{P}\mathrm{r}_\rho\Big(\mathcal{A}_{\tilde{\ }}([\![v]\!]^\rho_\eta) = 1\Big) \Big| \in \mathsf{negl}(\eta)$$

- **Predicate for reachability games**
  - ↪ *e.g. Knowledge soundness*

Overwhelmingly true predicate

$$[\phi] \quad \text{means} \quad \mathbb{P}\mathrm{r}_\rho\Big(\big[\!\big[\ \phi\ \big]\!\big]^\rho_\eta = 1\Big) \in \mathsf{ow}(\eta)$$

Formula of type **bool**

Introduction to CCSA – How to prove using this logic?

Two main predicates: $\sim$ and $[\cdot]$

- **Predicate for indistinguishability games**
  - ↪ *e.g. Honest-Verifier Zero-Knowledge*

Indistinguishability of terms predicate

$$u \sim v \quad \text{means} \quad \left| \mathbb{P}\mathrm{r}_\rho\left(\mathcal{A}_{\tilde{\zeta}}([\![u]\!]^\rho_\eta) = 1\right) - \mathbb{P}\mathrm{r}_\rho\left(\mathcal{A}_{\tilde{\zeta}}([\![v]\!]^\rho_\eta) = 1\right) \right| \in \mathrm{negl}(\eta)$$

- **Predicate for reachability games**
  - ↪ *e.g. Knowledge soundness*

Overwhelmingly true predicate

$$[\phi] \quad \text{means} \quad \mathbb{P}\mathrm{r}_\rho\left(\left[\!\!\left[\phi \right]\!\!\right]^\rho_\eta = 1 \right) \in \mathrm{ow}(\eta)$$

Formula of type **bool**

Global logic vs Local logic

- **Global logic –**
  - ↪ Properties studied *on almost all* random tapes

Satisfiability **for almost all** random tapes in $\mathbb{T}$ — Global formula

$$\vDash_g \ [\phi]$$

- **Local logic –**
  - ↪ Properties studied *at fixed* random tape

Satisfiability **at fixed** random tape

$$\rho \quad \vDash^{(\eta)}_\ell \quad \phi$$

Fixed random tape — Local formula

⚠ $\vDash_g [b \vee \neg b]$ and $\forall \rho,\ \rho \vDash^{(\eta)}_\ell b \vee \neg b$ hold
but $\not\vDash_g [b] \ \tilde{\vee}\ [\neg b]$ does not
(where b : **bool** – random coin)

Overview of the talk

I. Settings of zero-knowledge proofs

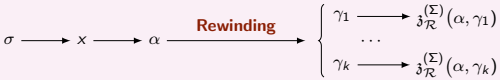
II. Brief overview of formal methods world – CCSA logic

III. How to model knowledge soundness and rewinding in the CCSA logic

IV. Other contributions and conclusion

Model knowledge soundness in CCSA – Set the scene

(Req. 1) – Build $\mathcal{E}_{\mathcal{R}}$ – case of Σ-protocols



(Req. 2) – Hypothesis $\mathcal{H}_{\mathsf{ks}}(\mathcal{P}_{\text{🦅}})$

$$\mathbb{P}\mathsf{r}_{\gamma}\left(\mathcal{P}_{\text{🦅}} \rightleftharpoons^{(\mu)}_{\mathcal{R}} \mathcal{V}_{\text{🐦}}(\gamma) = 1\right) \geqslant g$$

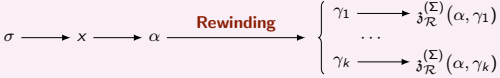
non-negligible parameter

Observations on Knowledge soundness

- **(Req. 3)** – $\mathcal{E}_{\mathcal{R}}$, a **probabilistic** extractor
- **(Req. 4)** – Probability taken on *honest* challenges
- **(Req. 5)** – ⚠ $\mathcal{P}_{\text{🦅}}$ is **deterministic**

Model knowledge soundness in CCSA – Set the scene

(Req. 1) – Build $\mathcal{E}_{\mathcal{R}}$ – case of Σ-protocols



(Req. 2) – Hypothesis $\mathcal{H}_{\mathsf{ks}}(\mathcal{P}_{\text{🦅}})$

$$\mathbb{P}\mathrm{r}_{\gamma}\left(\mathcal{P}_{\text{🦅}} \rightleftharpoons^{(\mu)}_{\mathcal{R}} \mathcal{V}_{\text{🐦}}(\gamma) = 1\right) \geqslant g$$

non-negligible parameter

Observations on Knowledge soundness

- **(Req. 3)** – $\mathcal{E}_{\mathcal{R}}$, a **probabilistic** extractor
- **(Req. 4)** – Probability taken on *honest* challenges
- **(Req. 5)** – ⚠ $\mathcal{P}_{\text{🦅}}$ is **deterministic**

Embed knowledge sound extractor in CCSA

Cryptography	CCSA logic
$\mathcal{E}_{\mathcal{R}}\left[\mathcal{P}_{\text{🦅}}\right] : (\sigma, x) \longmapsto w$	$\texttt{ks-extract}_{\mathcal{R}}/2$ (term)
Access to adversary $\mathcal{P}_{\text{🦅}}$ (explicit)	Access to $\mathcal{P}_{\text{🦅}}$ through ρ_a (implicit)

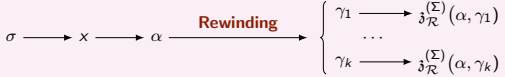
About deterministic or probabilistic PPTMs

- Probabilistic PPTM *for free* in CCSA (semantics)
- A deterministic PPTM *is always* explicit (syntax)

$$\vDash_g \mathtt{det}(f) \overset{\text{means}}{\Longrightarrow} \forall \eta,\ \rho \longmapsto [\![f]\!]^{\rho}_{\eta} \text{ is constant}$$

Model knowledge soundness in CCSA – Do we have tools? (1/2)

(Req. 1) – Build $\mathcal{E}_{\mathcal{R}}$ – case of Σ-protocols



(Req. 2) – Hypothesis $\mathcal{H}_{\mathsf{ks}}(\mathcal{P}_{\text{🦇}})$

$$\mathbb{P}\mathrm{r}_{\gamma}\left(\mathcal{P}_{\text{🦇}} \rightleftharpoons^{(\mu)}_{\mathcal{R}} \mathcal{V}_{\text{🧙}}(\gamma) = 1\right) \geqslant g$$

non-negligible parameter

Observations on Knowledge soundness

- **(Req. 3)** – $\mathcal{E}_{\mathcal{R}}$, a **probabilistic** extractor
- **(Req. 4)** – Probability taken on *honest* challenges
- **(Req. 5)** – ⚠ $\mathcal{P}_{\text{🦇}}$ is **deterministic**

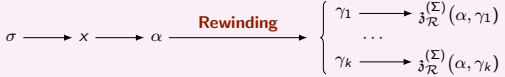
First naive hunch – local version

$$\mathtt{det}(\mathcal{P}_{\text{🦇}}) \vDash_{g} \left[\begin{array}{l} \mathtt{zkp\text{-}verif}_{\mathcal{R}}\left(\mathcal{P}_{\text{🦇}} \rightleftharpoons^{(\Sigma)}_{\mathcal{R}} \mathcal{V}_{\text{🧙}}(\gamma)\right) \\ \quad \rightarrow \Big(\underbrace{(\mathtt{ks\text{-}extract}_{\mathcal{R}}\ \sigma\ x)}_{\stackrel{\mathsf{def}}{=} w}\ \ \vdash_{\mathbb{ZK}}\ x \in \mathcal{L}_{\sigma}(\mathcal{R})\Big) \end{array}\right]$$

$\stackrel{\text{means}}{\Longrightarrow}\ (\sigma, x, w) \in \mathcal{R}$

Model knowledge soundness in CCSA – Do we have tools? (1/2)

(Req. 1) – Build $\mathcal{E}_{\mathcal{R}}$ – case of Σ-protocols



(Req. 2) – Hypothesis $\mathcal{H}_{\mathsf{ks}}(\mathcal{P}_{\text{😈}})$

$$\mathbb{P}\mathrm{r}_{\gamma}\left(\mathcal{P}_{\text{😈}} \rightleftharpoons^{(\mu)}_{\mathcal{R}} \mathcal{V}_{\text{😇}}(\gamma) = 1\right) \geqslant g$$

non-negligible parameter

Observations on Knowledge soundness

- **(Req. 3)** – $\mathcal{E}_{\mathcal{R}}$, a **probabilistic** extractor
- **(Req. 4)** – Probability taken on *honest* challenges
- **(Req. 5)** – ⚠ $\mathcal{P}_{\text{😈}}$ is **deterministic**

First naive hunch – local version

$$\mathtt{det}(\mathcal{P}_{\text{😈}}) \vDash_{g} \left[\begin{array}{l} \mathtt{zkp\text{-}verif}_{\mathcal{R}}\left(\mathcal{P}_{\text{😈}} \rightleftharpoons^{(\Sigma)}_{\mathcal{R}} \mathcal{V}_{\text{😇}}(\gamma)\right) \\ \quad \rightarrow \left(\underbrace{(\mathtt{ks\text{-}extract}_{\mathcal{R}}\ \sigma\ x)}_{\stackrel{\text{def}}{=} w}\ \vdash_{\mathbb{ZK}}\ x \in \mathcal{L}_{\sigma}(\mathcal{R})\right) \end{array}\right]$$

$$\stackrel{\text{means}}{\Longrightarrow} (\sigma, x, w) \in \mathcal{R}$$

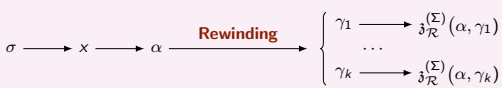
Does not work, but why?

If $\gamma \neq \gamma'$ then, in general,

$$\left(\mathcal{P}_{\text{😈}} \rightleftharpoons^{(\Sigma)}_{\mathcal{R}} \mathcal{V}_{\text{😇}}(\gamma)\right) = 1 \implies \left(\mathcal{P}_{\text{😈}} \rightleftharpoons^{(\Sigma)}_{\mathcal{R}} \mathcal{V}_{\text{😇}}(\gamma')\right) = 1$$

Model knowledge soundness in CCSA – Do we have tools? (2/2)

(Req. 1) – Build $\mathcal{E}_{\mathcal{R}}$ – case of Σ-protocols



(Req. 2) – Hypothesis $\mathcal{H}_{\mathsf{ks}}(\mathcal{P}_{\text{🦉}})$

$$\mathbb{P}\mathrm{r}_{\gamma}\left(\mathcal{P}_{\text{🦉}} \rightleftharpoons^{(\mu)}_{\mathcal{R}} \mathcal{V}_{\text{🐦}}(\gamma) = 1\right) \geqslant g$$

non-negligible parameter

Observations on Knowledge soundness

- **(Req. 3)** – $\mathcal{E}_{\mathcal{R}}$, a **probabilistic** extractor
- **(Req. 4)** – Probability taken on *honest* challenges
- **(Req. 5)** – ⚠ $\mathcal{P}_{\text{🦉}}$ is **deterministic**

Second hunch – global version

Embedding of hypothesis $\mathcal{H}_{\mathsf{ks}}(\mathcal{P}_{\text{🦉}})$

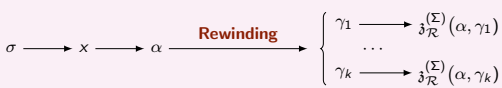
$$\mathtt{det}(\mathcal{P}_{\text{🦉}}) \vDash_{g} \tilde{\neg}\left[\neg\left(\mathtt{zkp\text{-}verif}_{\mathcal{R}}\ \left(\mathcal{P}_{\text{🦉}} \rightleftharpoons^{(\Sigma)}_{\mathcal{R}} \mathcal{V}_{\text{🐦}}(\gamma)\right)\right)\right]$$

$$\tilde{\rightarrow}\left[\left(\mathtt{ks\text{-}extract}_{\mathcal{R}}\ \sigma\ x\right)\ \vdash_{\mathbb{ZK}}\ x \in \mathcal{L}_{\sigma}(\mathcal{R})\right]$$

Are we done yet?

Model knowledge soundness in CCSA – Do we have tools? (2/2)

(Req. 1) – Build $\mathcal{E}_{\mathcal{R}}$ – case of Σ-protocols



(Req. 2) – Hypothesis $\mathcal{H}_{\mathsf{ks}}(\mathcal{P}_{\text{🦉}})$

$$\mathbb{P}\mathrm{r}_{\gamma}\left(\mathcal{P}_{\text{🦉}} \rightleftharpoons^{(\mu)}_{\mathcal{R}} \mathcal{V}_{\text{🐍}}(\gamma) = 1\right) \geqslant g$$

non-negligible parameter

Observations on Knowledge soundness

- **(Req. 3)** – $\mathcal{E}_{\mathcal{R}}$, a **probabilistic** extractor
- **(Req. 4)** – Probability taken on *honest* challenges
- **(Req. 5)** – ⚠ $\mathcal{P}_{\text{🦉}}$ is **deterministic**

Second hunch – global version

Embedding of hypothesis $\mathcal{H}_{\mathsf{ks}}(\mathcal{P}_{\text{🦉}})$

$$\mathtt{det}(\mathcal{P}_{\text{🦉}}) \vDash_{g} \tilde{\neg}\left[\neg\left(\mathtt{zkp\text{-}verif}_{\mathcal{R}}\ \left(\mathcal{P}_{\text{🦉}} \rightleftharpoons^{(\Sigma)}_{\mathcal{R}} \mathcal{V}_{\text{🐍}}(\gamma)\right)\right)\right]$$

$$\tilde{\rightarrow}\left[\left(\mathtt{ks\text{-}extract}_{\mathcal{R}}\ \sigma\ x\right) \vdash_{\mathbb{ZK}} x \in \mathcal{L}_{\sigma}(\mathcal{R})\right]$$

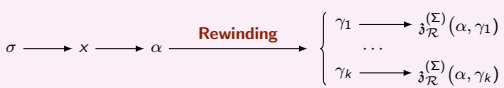
Are we done yet?

- Hypothesis $\mathtt{det}(\mathcal{P}_{\text{🦉}})$ too restrictive!

What if we delete it?

Model knowledge soundness in CCSA – Do we have tools? (2/2)

(Req. 1) – Build $\mathcal{E}_{\mathcal{R}}$ – case of Σ-protocols



(Req. 2) – Hypothesis $\mathcal{H}_{\mathsf{ks}}(\mathcal{P}_{\text{😈}})$

$$\mathbb{P}\mathrm{r}_{\gamma}\left(\mathcal{P}_{\text{😈}} \rightleftharpoons^{(\mu)}_{\mathcal{R}} \mathcal{V}_{\text{😇}}(\gamma) = 1\right) \geqslant g$$

non-negligible parameter

Observations on Knowledge soundness

- **(Req. 3)** – $\mathcal{E}_{\mathcal{R}}$, a **probabilistic** extractor
- **(Req. 4)** – Probability taken on *honest* challenges
- **(Req. 5)** – ⚠ $\mathcal{P}_{\text{😈}}$ is **deterministic**

Second hunch – global version

Embedding of hypothesis $\mathcal{H}_{\mathsf{ks}}(\mathcal{P}_{\text{😈}})$

$$\mathtt{det}(\mathcal{P}_{\text{😈}}) \vDash_{g} \rightsquigarrow \left[\neg\left(\mathtt{zkp\text{-}verif}_{\mathcal{R}}\ \left(\mathcal{P}_{\text{😈}} \rightleftharpoons^{(\Sigma)}_{\mathcal{R}} \mathcal{V}_{\text{😇}}(\gamma)\right)\right)\right]$$

$$\rightsquigarrow \left[\left(\mathtt{ks\text{-}extract}_{\mathcal{R}}\ \sigma\ x\right) \vdash_{\mathbb{ZK}}\ x \in \mathcal{L}_{\sigma}(\mathcal{R})\right]$$

Are we done yet?

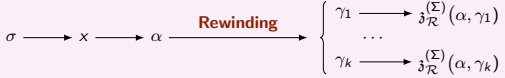
- Hypothesis $\mathtt{det}(\mathcal{P}_{\text{😈}})$ too restrictive!

  What if we delete it?
- $\mathcal{P}_{\text{😈}}$ becomes probabilistic

  Why is it a problem and how to fix it in CCSA?

Model knowledge soundness in CCSA – Lift the hood of $\texttt{ks-extract}_{\mathcal{R}}$

(Req. 1) – Build $\mathcal{E}_{\mathcal{R}}$ – case of Σ-protocols



(Req. 2) – Hypothesis $\mathcal{H}_{\mathsf{ks}}(\mathcal{P}_{\text{🦅}})$

$$\mathbb{P}\mathrm{r}_{\gamma}\left(\mathcal{P}_{\text{🦅}} \rightleftharpoons^{(\mu)}_{\mathcal{R}} \mathcal{V}_{\text{🐦}}(\gamma) = 1\right) \geqslant g$$

non-negligible parameter

Observations on Knowledge soundness

- **(Req. 3)** – $\mathcal{E}_{\mathcal{R}}$, a **probabilistic** extractor
- **(Req. 4)** – Probability taken on *honest* challenges
- **(Req. 5)** – ⚠ $\mathcal{P}_{\text{🦅}}$ is **deterministic**

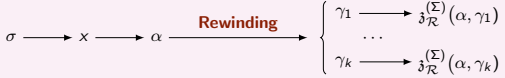
How $\texttt{ks-extract}_{\mathcal{R}}$ works?

Split in two steps

- **First step – rewinding**
  essentially, *resamples* challenges given by $\mathcal{V}_{\text{🐦}}$
- **Second step – extraction**
  computes the witness from all trace executions of $\mathcal{P}_{\text{🦅}}$

Model knowledge soundness in CCSA – Lift the hood of $\texttt{ks-extract}_{\mathcal{R}}$

(Req. 1) – Build $\mathcal{E}_{\mathcal{R}}$ – case of Σ-protocols



(Req. 2) – Hypothesis $\mathcal{H}_{\mathsf{ks}}(\mathcal{P}_{\text{🦅}})$

$$\mathbb{P}\mathsf{r}_{\gamma}\left(\mathcal{P}_{\text{🦅}} \rightleftharpoons^{(\mu)}_{\mathcal{R}} \mathcal{V}_{\text{🐳}}(\gamma) = 1\right) \geqslant g$$

non-negligible parameter

Observations on Knowledge soundness

- **(Req. 3)** – $\mathcal{E}_{\mathcal{R}}$, a **probabilistic** extractor
- **(Req. 4)** – Probability taken on *honest* challenges
- **(Req. 5)** – ⚠ $\mathcal{P}_{\text{🦅}}$ is **deterministic**

How $\texttt{ks-extract}_{\mathcal{R}}$ works?

Split in two steps

- **First step – rewinding**
  essentially, *resamples* challenges given by $\mathcal{V}_{\text{🐳}}$
- **Second step – extraction**
  computes the witness from all trace executions of $\mathcal{P}_{\text{🦅}}$

⚠ Trace executions differs only from *challenges*!

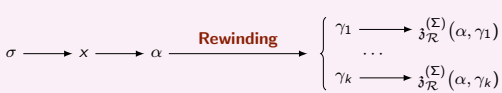
$\Longrightarrow$ **We need to find a way to**

- Fix adversarial random tape ρ_a...
- ...but with honest random tape ρ_h still free

$\Longrightarrow$ Mix reasoning between global and local logics

Model knowledge soundness in CCSA – How to mix global and local logics? (1/3)

(Req. 1) – Build $\mathcal{E}_{\mathcal{R}}$ – case of Σ-protocols



(Req. 2) – Hypothesis $\mathcal{H}_{\mathsf{ks}}(\mathcal{P}_{\text{😈}})$

$$\mathbb{P}\mathrm{r}_{\gamma}\left(\mathcal{P}_{\text{😈}} \rightleftharpoons^{(\mu)}_{\mathcal{R}} \mathcal{V}_{\text{😇}}(\gamma) = 1\right) \geqslant g$$

non-negligible parameter

Observations on Knowledge soundness

- **(Req. 3)** – $\mathcal{E}_{\mathcal{R}}$, a **probabilistic** extractor
- **(Req. 4)** – Probability taken on *honest* challenges
- **(Req. 5)** – ⚠ $\mathcal{P}_{\text{😈}}$ is **deterministic**

$\Longrightarrow$ **We need to find a way to**

- Fix adversarial random tape ρ_a...
- ...but with honest random tape ρ_h still free

How to precise non-negligible parameter in CCSA?

Local formula ϕ : **bool**

Non-negligible formulas in CCSA $\tilde{\neg}\left[\neg\ \phi\right]$

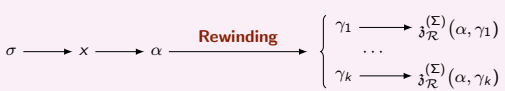
$\tilde{\neg}\left[\neg\phi\right] \leftrightarrows \tilde{\exists} e : \mathbf{real}.\ \texttt{non-negl}(e)\ \tilde{\wedge}\ {}_e[\phi]$

Predicate for non-negligible terms

where $\left[\!\left[{}_e[\phi] \right]\!\right]_\eta \overset{\text{means}}{\Longrightarrow} \mathbb{P}\mathrm{r}_\rho\left([\![\phi]\!]^\rho_\eta = 1 \right) \geqslant [\![e]\!]_\eta$

Model knowledge soundness in CCSA – How to mix global and local logics? (2/3)

(Req. 1) – Build $\mathcal{E}_{\mathcal{R}}$ – case of Σ-protocols



(Req. 2) – Hypothesis $\mathcal{H}_{\mathsf{ks}}(\mathcal{P}_{\text{😈}})$

$$\mathbb{Pr}_{\gamma}\left(\mathcal{P}_{\text{😈}} \rightleftharpoons^{(\mu)}_{\mathcal{R}} \mathcal{V}_{\text{😇}}(\gamma) = 1\right) \geqslant g$$

non-negligible parameter

Observations on Knowledge soundness

- **(Req. 3)** – $\mathcal{E}_{\mathcal{R}}$, a **probabilistic** extractor
- **(Req. 4)** – Probability taken on *honest* challenges
- **(Req. 5)** – ⚠ $\mathcal{P}_{\text{😈}}$ is **deterministic**

$\Longrightarrow$ **We need to find a way to**

- Fix adversarial random tape ρ_a...
- ...but with honest random tape ρ_h still free

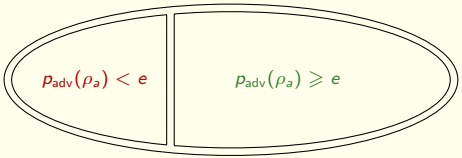
Local formula we rewind

Honest random tape (free)

$$\phi_\eta : \left(\rho_a, \rho_h\right) \longmapsto \phi_\eta\left((\rho_a, \rho_h)\right) \in \{0,1\}$$

Adversarial random tape (fixed)

We study the function $p_{\mathsf{adv}} \stackrel{\mathsf{def}}{=} \rho_a \longmapsto \mathbb{Pr}_{\rho_h}\left(\phi_\eta\left((\rho_a, \rho_h)\right) = 1\right)$



Model knowledge soundness in CCSA – How to mix global and local logics? (3/3)

(Req. 1) – Build $\mathcal{E}_{\mathcal{R}}$ – case of Σ-protocols

$$\sigma \longrightarrow x \longrightarrow \alpha \xrightarrow{\text{Rewinding}} \begin{cases} \gamma_1 \longrightarrow \mathfrak{z}^{(\Sigma)}_{\mathcal{R}}(\alpha, \gamma_1) \\ \cdots \\ \gamma_k \longrightarrow \mathfrak{z}^{(\Sigma)}_{\mathcal{R}}(\alpha, \gamma_k) \end{cases}$$

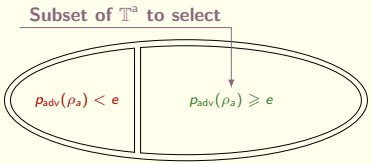
(Req. 2) – Hypothesis $\mathcal{H}_{\mathsf{ks}}(\mathcal{P}_{\text{🦇}})$

$$\mathbb{P}\mathsf{r}_{\gamma}\left(\mathcal{P}_{\text{🦇}} \rightleftharpoons^{(\mu)}_{\mathcal{R}} \mathcal{V}_{\text{🐍}}(\gamma) = 1\right) \geqslant g$$

g: non-negligible parameter

Observations on Knowledge soundness

- **(Req. 3)** – $\mathcal{E}_{\mathcal{R}}$, a **probabilistic** extractor
- **(Req. 4)** – Probability taken on *honest* challenges
- **(Req. 5)** – ⚠ $\mathcal{P}_{\text{🦇}}$ is **deterministic**



$$p_{\mathsf{adv}} : \rho_a \longmapsto \mathbb{P}\mathsf{r}_{\rho_h}\left(\phi_\eta\left((\rho_a, \rho_h)\right)\right)$$

How to "select" random tapes subset in CCSA?

Local formula we rewind

$$\phi : \textbf{bool} \text{ s.t. } \forall \eta, \rho_a, \rho_h, \ [\![\phi]\!]^{(\rho_a,\rho_h)}_{\eta} \stackrel{\text{def}}{=} \phi_\eta\left((\rho_a, \rho_h)\right)$$

Local predicate

$$\left[\!\!\left[\texttt{select-tape } e\ \phi \right]\!\!\right]^{(\rho_a,\rho_h)}_{\eta} = 1 \stackrel{\text{def}}{\iff}$$

$$\mathbb{P}\mathsf{r}_{\rho'_h}\left(\phi_\eta\left((\rho_a, \rho'_h)\right)\right) \geqslant [\![e]\!]_\eta$$

Formal model of knowledge soundness and rewinding

Obtain Knowledge soundness – Definitive version

Hypothesis required by knowledge soundness

$$\vDash_g \left[\begin{array}{l} \texttt{select-tape}\ e\ \phi \\ \quad \rightarrow \Big(\big(\texttt{ks-extract}_{\mathcal{R}}\ \sigma\ x \big) \vdash_{\mathbb{ZK}} x \in \mathcal{L}_{\sigma}(\mathcal{R}) \Big) \end{array} \right]$$

Formal model of knowledge soundness and rewinding

Obtain Knowledge soundness – Definitive version

Hypothesis required by knowledge soundness

$$\vDash_g \left[\begin{array}{l} \texttt{select-tape}\ e\ \phi \\ \quad \rightarrow \Big(\big(\texttt{ks-extract}_{\mathcal{R}}\ \sigma\ x \big) \vdash_{\mathbb{ZK}} x \in \mathcal{L}_\sigma(\mathcal{R}) \Big) \end{array} \right]$$

Model rewinding in CCSA (simplified)

To obtain $n \in \mathbb{N}^*$ challenges generated by $\mathtt{r}_s^{(\tau)} : \mathbf{nat} \rightarrow \tau$ (uniform source of challenges)

Polynomial bound on indexes given to $\mathtt{r}_s^{(\tau)}$

$$\vDash_g \tilde{\exists}\, \mathbf{k_g} : \mathbf{nat}\ . \left[\begin{array}{l} \texttt{select-tape}\ e\ \phi \rightarrow \\ \quad \forall\, \mathtt{r}_s^{(\tau)}\ t.\ \texttt{choose}_\tau^{(n)}\ \mathbf{k_g}\ \mathtt{r}_s^{(\tau)}\ . \\ \qquad \phi\ \big(\mathtt{r}_s^{(\tau)}\ t\big) \end{array} \right]$$

Calls $\mathtt{r}_s^{(\tau)}$ until n challenges are obtained...

...s.t. $\phi : \tau \rightarrow \mathbf{bool}$ holds on $\mathtt{r}_s^{(\tau)}\ t$

Formal model of knowledge soundness and rewinding

Obtain Knowledge soundness – Definitive version

Hypothesis required by knowledge soundness

$$\models_g \left[\begin{array}{l} \texttt{select-tape}\ e\ \phi \\ \quad \rightarrow \left(\left(\texttt{ks-extract}_{\mathcal{R}}\ \sigma\ x \right) \vdash_{\mathbb{ZK}} x \in \mathcal{L}_\sigma(\mathcal{R}) \right) \end{array} \right]$$

Model rewinding in CCSA (simplified)

To obtain $n \in \mathbb{N}^*$ challenges generated by $\mathtt{r}_s^{(\tau)} : \mathbf{nat} \rightarrow \tau$ (uniform source of challenges)

Polynomial bound on indexes given to $\mathtt{r}_s^{(\tau)}$

$$\models_g \tilde{\exists}\, \mathbf{k_g} : \mathbf{nat}\ . \left[\begin{array}{l} \texttt{select-tape}\ e\ \phi \rightarrow \\ \quad \forall\, \mathtt{r}_s^{(\tau)}\ t.\ \texttt{choose}_\tau^{(n)}\ \mathbf{k_g}\ \mathtt{r}_s^{(\tau)}. \\ \quad\quad \phi\ \left(\mathtt{r}_s^{(\tau)}\ t \right) \end{array} \right]$$

Calls $\mathtt{r}_s^{(\tau)}$ until n challenges are obtained...

...s.t. $\phi : \tau \rightarrow \mathbf{bool}$ holds on $\mathtt{r}_s^{(\tau)}\ t$

Glue split parts back together

To **prove** $\models_g \left[\mathcal{H}\ \gamma \rightarrow \phi\ \gamma \right]$, we **prove**

$\models_g \tilde{\forall}\, e : \mathbf{real}$ [non-negl].

$$\left[\texttt{select-tape}\ e\ \mathcal{H} \rightarrow \left(\mathcal{H}\ \gamma \rightarrow \phi\ \gamma \right) \right].$$

Formal model of knowledge soundness and rewinding

Obtain Knowledge soundness – Definitive version

Hypothesis required by knowledge soundness

$$\models_g \left[\begin{array}{l} \texttt{select-tape}\ e\ \phi \\ \quad \rightarrow \Big(\big(\texttt{ks-extract}_{\mathcal{R}}\ \sigma\ x \big) \vdash_{\mathbb{ZK}} x \in \mathcal{L}_\sigma(\mathcal{R}) \Big) \end{array} \right]$$

Model rewinding in CCSA (simplified)

To obtain $n \in \mathbb{N}^*$ challenges generated by $\mathtt{r}_s^{(\tau)} : \mathbf{nat} \rightarrow \tau$ (uniform source of challenges)

Polynomial bound on indexes given to $\mathtt{r}_s^{(\tau)}$

$$\models_g \tilde{\exists}\, \mathbf{k_g} : \mathbf{nat}\ . \left[\begin{array}{l} \texttt{select-tape}\ e\ \phi \rightarrow \\ \quad \forall\, \mathtt{r}_s^{(\tau)}\ t.\ \texttt{choose}_\tau^{(n)}\ \mathbf{k_g}\ \mathtt{r}_s^{(\tau)}. \\ \qquad \phi\ \left(\mathtt{r}_s^{(\tau)}\ t\right) \end{array} \right]$$

Calls $\mathtt{r}_s^{(\tau)}$ until n challenges are obtained...

...s.t. $\phi : \tau \rightarrow \mathbf{bool}$ holds on $\mathtt{r}_s^{(\tau)}\ t$

Glue split parts back together

To **prove** $\models_g \Big[\mathcal{H}\ \gamma \rightarrow \phi\ \gamma\Big]$, we **prove**

$\models_g \tilde{\forall}\, e : \mathbf{real}$ [non-negl].

$$\Big[\texttt{select-tape}\ e\ \mathcal{H} \rightarrow \big(\mathcal{H}\ \gamma \rightarrow \phi\ \gamma\big)\Big].$$

Sketch of key ideas for the "glue" property proof

By contraposition – **assume** $\models_{g\ 2\cdot e} \Big[\mathcal{H}\ \gamma \wedge \neg\, \phi\ \gamma\Big]$ for some e : **real** and **prove**

$$\models_{g\ e} \Big[\big(\texttt{select-tape}\ e\ \mathcal{H}\big) \wedge \big(\mathcal{H}\ \gamma \wedge \neg\, \phi\ \gamma\big)\Big]$$

Formal model of knowledge soundness and rewinding

Obtain Knowledge soundness – Definitive version

Hypothesis required by knowledge soundness

$$\vDash_g \left[\begin{array}{l} \texttt{select-tape}\ e\ \phi \\ \quad \rightarrow \Big(\big(\texttt{ks-extract}_{\mathcal{R}}\ \sigma\ x\big) \vdash_{\mathbb{ZK}} x \in \mathcal{L}_\sigma(\mathcal{R}) \Big) \end{array} \right]$$

Model rewinding in CCSA (simplified)

To obtain $n \in \mathbb{N}^*$ challenges generated by $\mathtt{r}_s^{(\tau)} : \mathbf{nat} \rightarrow \tau$ (uniform source of challenges)

Polynomial bound on indexes given to $\mathtt{r}_s^{(\tau)}$

$$\vDash_g \tilde{\exists}\, \mathbf{k_g} : \mathbf{nat}\ . \left[\begin{array}{l} \texttt{select-tape}\ e\ \phi \rightarrow \\ \quad \forall\, \mathtt{r}_s^{(\tau)}\ t.\ \texttt{choose}_\tau^{(n)}\ \mathbf{k_g}\ \mathtt{r}_s^{(\tau)}. \\ \qquad \phi\ \left(\mathtt{r}_s^{(\tau)}\ t\right) \end{array} \right]$$

Calls $\mathtt{r}_s^{(\tau)}$ until n challenges are obtained...

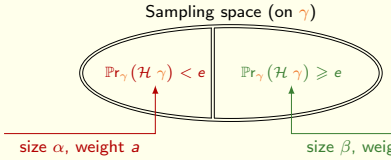
...s.t. $\phi : \tau \rightarrow \mathbf{bool}$ holds on $\mathtt{r}_s^{(\tau)}\ t$

Glue split parts back together

Assume $\vDash_{g\ 2\cdot e} \Big[\mathcal{H}\ \gamma \wedge \neg\, \phi\ \gamma\Big]$ and **prove**

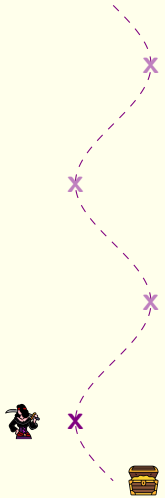
$$\vDash_{g\ e} \Big[\big(\texttt{select-tape}\ e\ \mathcal{H}\big) \wedge \big(\mathcal{H}\ \gamma \wedge \neg\, \phi\ \gamma\big)\Big]$$

Sketch of key ideas for the "glue" property proof



$$\frac{a \leqslant e \qquad b \leqslant \beta \qquad a + b \geqslant e}{\beta \geqslant e}$$

Overview of the talk



I. Settings of zero-knowledge proofs

II. Brief overview of formal methods world – CCSA logic

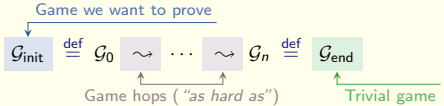
III. How to model knowledge soundness and rewinding in the CCSA logic

IV. Other contributions and conclusion

Other contributions going from mix-servers to mixnets

Mix-server security

Traditional framework used: Cryptographic games



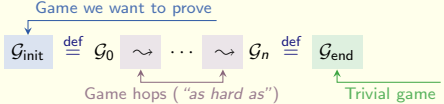
First contribution

Are logical tools developped modular enough?

Other contributions going from mix-servers to mixnets

Mix-server security

Traditional framework used: Cryptographic games



First contribution

Are logical tools developped modular enough?

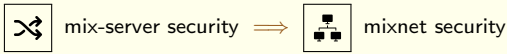
Mixnet security

Traditional framework used: *Universal Composability* (UC)

Secure shuffle protocol — Secure mixnet protocol

$$\Pi_{\text{MS}} \vDash_{\mathcal{G}} \mathcal{G}^{(\text{MS})}_{\text{priv}} \wedge \mathcal{G}^{(\text{MS})}_{\text{int}} \implies \Pi_{\text{MN}}(\Pi_{\text{MS}}) \vDash_{\mathcal{G}} \mathcal{G}^{(\text{MN})}_{\text{priv}} \wedge \mathcal{G}^{(\text{MN})}_{\text{int}}$$

Second contribution



Other contributions going from mix-servers to mixnets

Mix-server security

Traditional framework used: Cryptographic games

Game we want to prove

$\mathcal{G}_{\text{init}} \overset{\text{def}}{=} \mathcal{G}_0 \rightsquigarrow \cdots \rightsquigarrow \mathcal{G}_n \overset{\text{def}}{=} \mathcal{G}_{\text{end}}$

Game hops (*"as hard as"*) — Trivial game

First contribution

Are logical tools developped modular enough?

Mixnet security

Traditional framework used: *Universal Composability* (UC)

Secure shuffle protocol — Secure mixnet protocol

$\Pi_{\text{MS}} \vDash_{\mathcal{G}} \mathcal{G}^{(\text{MS})}_{\text{priv}} \wedge \mathcal{G}^{(\text{MS})}_{\text{int}} \implies \Pi_{\text{MN}}(\Pi_{\text{MS}}) \vDash_{\mathcal{G}} \mathcal{G}^{(\text{MN})}_{\text{priv}} \wedge \mathcal{G}^{(\text{MN})}_{\text{int}}$

Second contribution

 mix-server security $\implies$  mixnet security

A link between frameworks

Game-based secure mixnets — UC secure mixnets

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

Third contribution

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

Key take-aways and prospectives

Key take aways

3 main research axes

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

Prospectives

Key take-aways and prospectives

Key take aways

3 main research axes

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

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 axes

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

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 axes

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

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 axes

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

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

— *Glondu 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