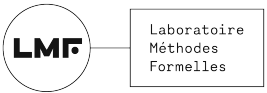
Insights from formal proofs of e-voting mixnets

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

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

PESTO seminar, 25th April 2025



Integration of our works in voting protocols

Voting protocol chronology

Election setup

Voting phase

Shuffle

Tally

Main security properties looked for

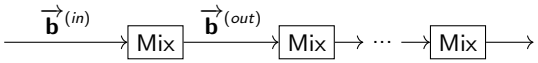
Ballot privacy

Universal verifiability

Individual verifiability

Mix networks in a nutshell

Principle



Network of *mix-servers*

Example: If $\overrightarrow{\mathbf{b}}^{(in)} = \left(v_1\ {}_{r_1}, v_2\ {}_{r_2}, v_3\ {}_{r_3} \right)$ then

$$\overrightarrow{\mathbf{b}}^{(out)} = \left(v_3\ {}_{s_1}, v_1\ {}_{s_2}, v_2\ {}_{s_3} \right)$$

Algorithm : Mixing

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

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

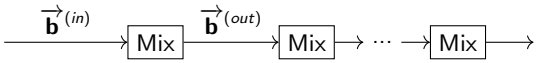
[*do some stuff...*] ;

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

Mix-server in a nutshell

Mix networks in a nutshell

Principle



Network of *mix-servers*

Example: If $\overrightarrow{\mathbf{b}}^{(in)} = \left(v_1\ {}_{r_1}, v_2\ {}_{r_2}, v_3\ {}_{r_3} \right)$ then $\overrightarrow{\mathbf{b}}^{(out)} = \left(v_3\ {}_{s_1}, v_1\ {}_{s_2}, v_2\ {}_{s_3} \right)$

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

Goal (privacy)

The attacker **can not** guess the permutation used to mix the ballot box

Constraints (verifiability)

Ballot content stays **untouched**

Commitment schemes

Principle

Security properties

Hiding

$\mathsf{Com}_{\mathcal{M}}$ behaves like an *one-way function*

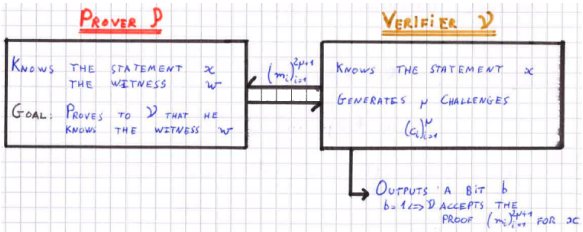
Binding

$\mathsf{Com}_{\mathcal{M}}$ is *resistent to collisions*

(Interactive) Zero-knowledge proofs

Principle

- **Two agents** a prover $\mathcal{P}$ and a verifier $\mathcal{V}$;
- **Goal** prove that ($\underbrace{\sigma}_{\text{public parameter}}$, $\underbrace{x}_{\text{statement}}$, $\underbrace{w}_{\text{witness}}$) $\in \mathcal{R}$;
- **Special cases:** NIZK ($\mu = 0$), Σ-protocols ($\mu = 1$).

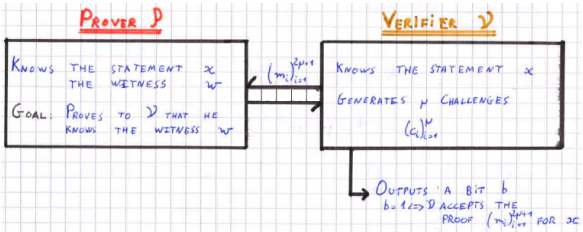


Basic properties

(Interactive) Zero-knowledge proofs

Principle

- **Two agents** a prover $\mathcal{P}$ and a verifier $\mathcal{V}$;
- **Goal** prove that ($\underbrace{\sigma}_{\text{public parameter}}$, $\underbrace{x}_{\text{statement}}$, $\underbrace{w}_{\text{witness}}$) $\in \mathcal{R}$;
- **Special cases:** NIZK ($\mu = 0$), Σ-protocols ($\mu = 1$).



Basic properties

Soundness

Settings: dishonest prover $\tilde{\mathcal{P}}$, honest verifier $\mathcal{V}$

Zero-Knowledge

Settings: honest prover $\mathcal{P}$, dishonest verifier $\tilde{\mathcal{V}}$

Extractibility

Idea: Extractor $\mathcal{E}_{\mathcal{R}}$ computing w for (σ, x) with black-box access to $\tilde{\mathcal{P}}$.

Mix networks - State of the art

Two quite mature shuffle protocols

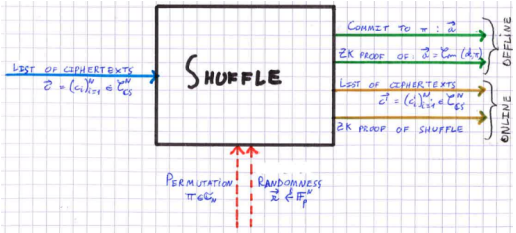
Belenios

Terelius & Wikström shuffle protocol [**TereliusW10**; **Wikstrom11**]

SwissPost

Bayer & Groth shuffle protocol [**BayerG12**]

Shuffle protocols basics



Two zero-knowledge proofs

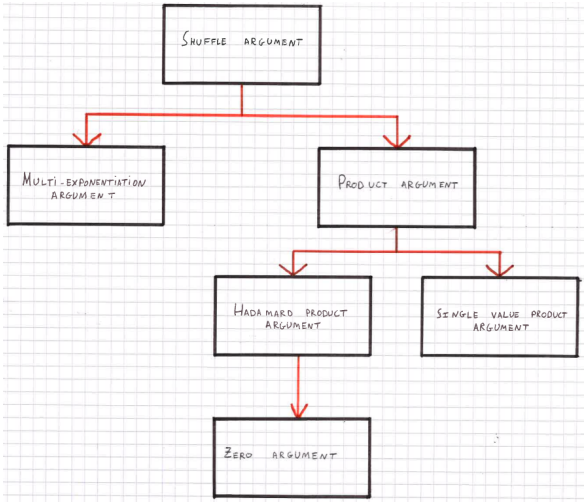
- **(Offline)** Commit to a permutation:

$$\mathbf{a} = \mathrm{Com}(\, ck \,,\, \pi \,;\, \mathbf{s} \,)\,;$$

- **(Online)** Commitment-consistent shuffle:

$$\forall\, i \in [\![1; N]\!],\ c'_{\pi(i)} = \phi_{\mathrm{CS}}(\, pk \,,\, c_i \,;\, r_i \,).$$

Structure of Bayer-Groth shuffle protocol

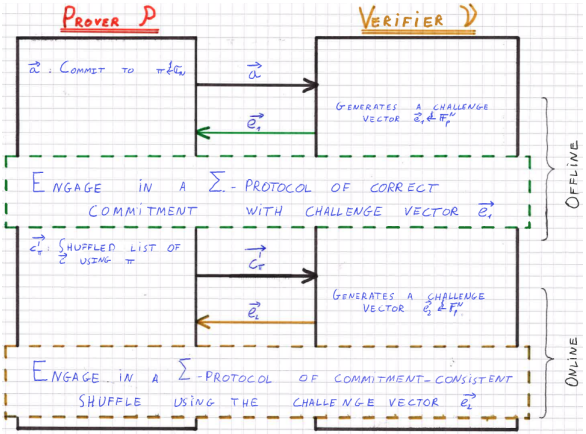


Argument protocol	Number of messages
Zero argument	3 ($\mu = 1$)
Hadamard product argument	$\mathbf{5}^{(*)}$ ($\mu = 2$)
Single value product argument	3 ($\mu = 1$)
Product argument	$\mathbf{7}^{(*)}$ ($\mu = 3$)
Multi-exponentiation argument	3 ($\mu = 1$)
Shuffle argument	$\mathbf{13}^{(*)}$ ($\mu = 6$)

- *Product argument protocol:* commit to a permutation
- *Multi-exponentiation argument:* commitment-consistent shuffle

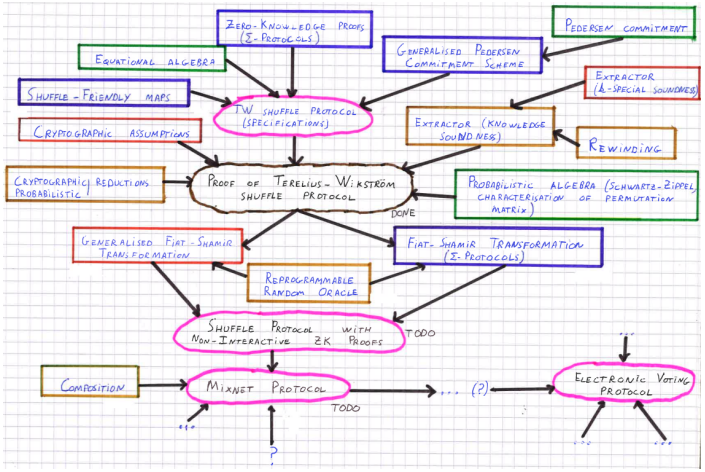
($*$) can be shorten

Structure of Terelius-Wikström shuffle protocol



Warning: Notice that both Σ-protocols are in fact a family of protocols $\left(\Sigma_{\mathsf{TW}}(\mathbf{e})\right)_{\mathbf{e}\in\mathbb{F}_{p\eta}^N}$

More details about Terelius-Wikström shuffle protocol



Proof of Terelius-Wikström shuffle protocol:

- Based on various advanced cryptographic constructions and their cryptographic properties, ...
- ... using (probabilistic) algebraic properties, ...
- ... and involving subtle cryptographic reductions techniques

We want to use a **logical framework** *expressive enough*, with strong *computational guarantees*, and *without making reductions explicit*.

State of the art of shuffle protocols proof

Authors	*Terelius & Wikström*	*Haines et al.*
References	[**TereliusW10**; **Wikstrom11**]	[**HainesGS21**; **HainesGT23**]
Kind of proof	 Pen-and-paper	 Rocq (Coq) prover
What is proved?	Correctness of "maths" (probabilistic and algebraic results)	*Zero-Knowledge* proof properties of protocols – Code extraction
Issues of the proof	No explicit cryptographic reductions, in particular, no explicit adversary advantage	Only "maths", it is **not** a formal proof of cryptographic reductions

Symbolic vs. computational models

	Symbolic models	Computational models
Attacker capabilities	Positive definition	Negative definition
Messages	Terms algebra	Bitstrings
Proof techniques	Rewriting / induction	Reductions (complexity-like)
Satisfying / Issues?	No computational guarantees	Explicit reductions and proofs

Remind our goal

We want to use a **logical framework** *expressive enough*, with strong *computational guarantees*, and *without making reductions explicit*.

Are we satisfied?

NO :(

An hybrid model: the *Computationally Complete Symbolic Attacker (CCSA)* model

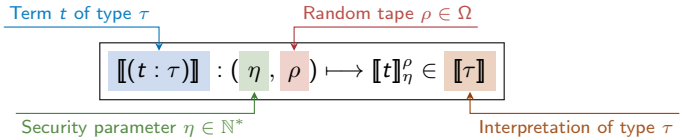
We want to use a **logical framework** which is *expressive enough*, with strong *computational guarantees*, and *without making reductions explicit*.

The CCSA model

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

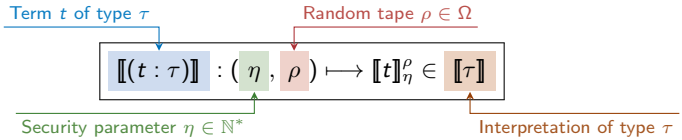
More about the CCSA model [**BanaC14**; **BaeldeKL23**]

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



More about the CCSA model [**BanaC14**; **BaeldeKL23**]

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



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

Indistinguishability of terms predicate

$u \sim v$ means $\left| \Pr_\rho\left(\mathcal{A}([\![u]\!]^\rho_\eta) = 1\right) - \Pr_\rho\left(\mathcal{A}([\![v]\!]^\rho_\eta) = 1\right) \right| \in \mathsf{negl}(\eta)$

Overwhelmingly true predicate

$[\phi]$ means $\Pr_\rho\left([\![\phi]\!]^\rho_\eta = 1\right) \in \mathsf{ow}(\eta)$

Formula of type **bool**

Global vs. local logic

Global logic

Satisfiability **for almost all** random tapes in Ω

$\vDash_g \ [\phi]$

Global formula

Global connector

$\vDash_g [\phi] \ \tilde{\rightarrow} \ [\psi]$ means

If $\mathsf{Pr}_\rho\left([\![\phi]\!]^\rho_\eta = 1\right) \in \mathsf{ow}(\eta)$

then $\mathsf{Pr}_\rho\left([\![\psi]\!]^\rho_\eta = 1\right) \in \mathsf{ow}(\eta)$

Local logic

Satisfiability **at fixed** random tape

$\rho \ \vDash_l^{(\eta)} \ \phi$

Fixed random tape

Local formula

Local connector

$\rho \vDash_l^{(\eta)} \phi \ \rightarrow \ \psi$ means $[\![\phi \rightarrow \psi]\!]^\rho_\eta = 1$

Relation between both logics

Local satisfiability

If $\mathsf{Pr}_\rho\left(\rho \vDash_l^{(\eta)} \phi\right) \in \mathsf{ow}(\eta)$ **then** $\vDash_g [\phi]$

Global satisfiability

Where are we with the CCSA logic?

Recall our goal

We want to use a **logical framework** which is (*iii*) *expressive enough*, with strong (*ii*) *computational guarantees*, and (*i*) *without making reductions explicit*.

Checkup so far

- (*i*) **Without making reductions explicit?**

  ✓ Yes, by definition of the syntax (reductions are hide in rules)

- (*ii*) **Strong computational guarantees?**

  ✓ Yes, because of
  - definitions of the semantics of our new predicates ($\sim$, $[\cdot]$), and
  - our attacker model (PPTM).

Where are we with the CCSA logic?

Recall our goal

We want to use a **logical framework** which is (*iii*) *expressive enough*, with strong (*ii*) *computational guarantees*, and (*i*) *without making reductions explicit*.

Checkup so far

- (*i*) **Without making reductions explicit?**

  ✓ Yes, by definition of the syntax (reductions are hide in rules)

- (*ii*) **Strong computational guarantees?**

  ✓ Yes, because of
  - definitions of the semantics of our new predicates ($\sim$, $[\cdot]$), and
  - our attacker model (PPTM).

- (*iii*) **Is CCSA logic expressive enough?**

  ? Not before us, core of our work

What exactly do we need for the proof?

- Probabilistic and equational algebraic properties

  Schwartz-Zippel lemma | Characterisation of permutation matrix | ...

- Advanced cryptographic constructions and their cryptographic properties

  Zero-Knowledge proofs | Commitment schemes | *Shuffle-Friendly* maps

What exactly do we need for the proof?

- Probabilistic and equational algebraic properties

  Schwartz-Zippel lemma | Characterisation of permutation matrix | ...

- Advanced cryptographic constructions and their cryptographic properties

  Zero-Knowledge proofs | Commitment schemes | *Shuffle-Friendly* maps

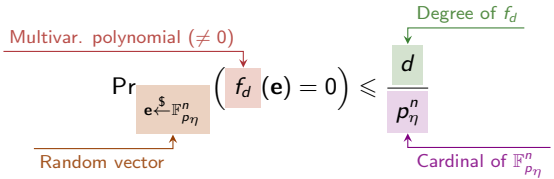
- Subtle cryptographic reductions techniques

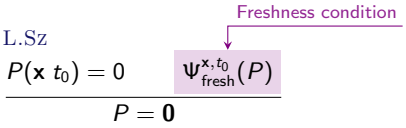
  Rewinding | Adversarial selection function | Conditional probabilities + Non-negligible formulas

Some flavor of our new rules

Probabilistic algebraic properties

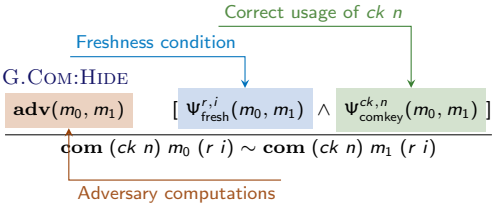
Schwartz-Zippel lemma





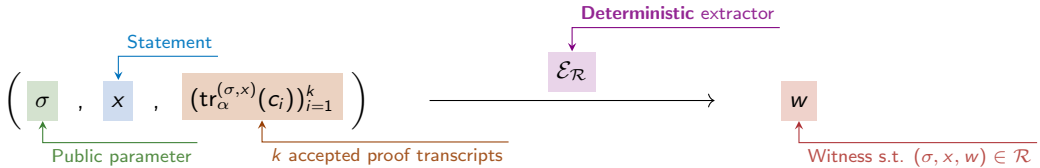
Cryptographic properties

Hiding property of commitment scheme



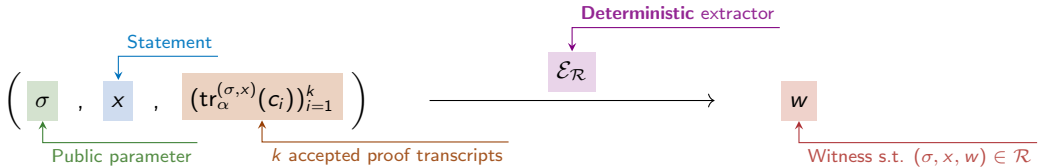
Cryptographic constructions – Σ-protocols

k-Special Soundness



Cryptographic constructions – Σ-protocols

k-Special Soundness

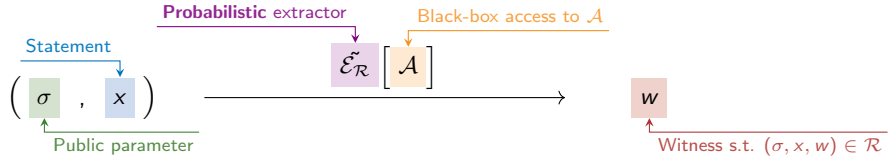


Observations

- Extractor $\mathcal{E}_{\mathcal{R}}$ is deterministic
- Strong assumption on the adversary: $\mathcal{A}$ has to produce $(\mathrm{tr}_{\alpha}^{(\sigma,x)}(c_i))_{i=1}^{k}$

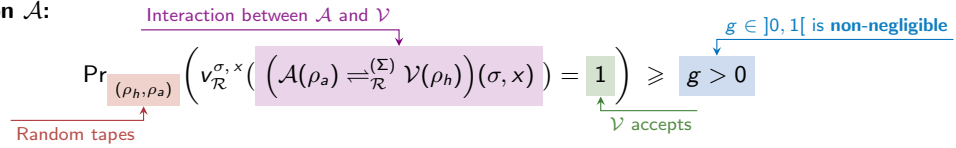
Σ-protocols – Knowledge soundness

Knowledge soundness



Observations

- Now, $\tilde{\mathcal{E}}_{\mathcal{R}}\left[\mathcal{A}\right]$ is *probabilistic*, with an overwhelming probability of success
- **Hypothesis on $\mathcal{A}$:**



Σ-protocols – Witness extraction

What we have as hypothesis

Deterministic extractor

$$\mathcal{E}_{\mathcal{R}} : \left(\sigma, x, (\mathrm{tr}_{\alpha}^{(\sigma,x)}(c_i))_{i=1}^{k}\right) \longmapsto w$$

k-**Special soundness**

used to build $\Longrightarrow$

What we want

Probabilistic extractor

$$\tilde{\mathcal{E}}_{\mathcal{R}}[\mathcal{A}] : (\sigma, x) \longmapsto w$$

Knowledge soundness

Σ-protocols – Witness extraction

What we have as hypothesis

Deterministic extractor

$$\mathcal{E}_{\mathcal{R}} : \left(\sigma, x, (\mathrm{tr}_{\alpha}^{(\sigma,x)}(c_i))_{i=1}^{k}\right) \longmapsto w$$

k-**Special soundness**

used to build ⟹

What we want

Probabilistic extractor

$$\tilde{\mathcal{E}}_{\mathcal{R}}[\mathcal{A}] : (\sigma, x) \longmapsto w$$

Knowledge soundness

Idea

- Build $\tilde{\mathcal{E}}_{\mathcal{R}}[\mathcal{A}]$ on top of $\mathcal{E}_{\mathcal{R}}$
- **How?** By using the *rewinding* technique

Upcoming next

- Non-negligible formulas
- Conditional probabilities
- Adversarial selection function
- Rewinding

Non-negligible formulas

Hypothesis on $\mathcal{A}$:

$$\Pr_\rho\left(v_{\mathcal{R}}^{\sigma,x}\left(\mathfrak{p}_{\mathcal{A}}^{(\sigma,x)}(\rho)\right) = 1\right) \geqslant g$$

where $\mathfrak{p}_{\mathcal{A}}^{(\sigma,x)}(\rho_h,\rho_a) \stackrel{\text{def}}{=} \left(\mathcal{A}(\rho_a) \rightleftharpoons_{\mathcal{R}}^{(\Sigma)} \mathcal{V}(\rho_h)\right)(\sigma,x)$

Non-negligible predicate

Formula: $[\phi]$ $\stackrel{\text{def}}{\Longleftrightarrow}$ $\Pr_\rho\left([\![\phi]\!]_\eta^\rho = 1\right) \geqslant [\![g]\!]_\eta$ (g is deterministic)

g : **real** such that **non-negl** g

Non-negligible definition

$$\textbf{non-negl}\ g \stackrel{\text{means}}{\Longleftrightarrow} \exists\, P \text{ poly}\ ,\ \forall\, \eta \in \mathbb{N}^*,\ \exists\, \eta' > \eta,\ [\![g]\!]_{\eta'} \geqslant \frac{1}{P(\eta')}$$

Properties on the non-negligible predicate

Link with the globally true predicate (**characterisation**)

(1) $\tilde{\neg}\,[\neg\,\phi] \Leftrightarrow \tilde{\exists}\, g : \textbf{real}.\ (\textbf{non-negl}\ g)\ \tilde{\wedge}\ {}_g[\phi]$

(2) ${}_g[\phi\ r] \tilde{\rightarrow} [\phi\ (\textbf{resample}\ r)]$

Make *non-negl* formula overwhelmingly true (**resampling**)

Rewinding lemma (1/2)

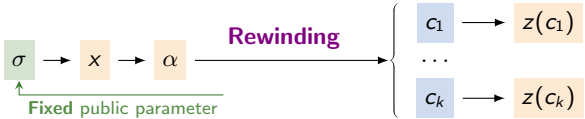
Look at random tapes

Honest random tape

$$\rho = (\rho_h, \rho_a)$$

Adversarial random tape

- **Honest tape:** uniform and independent randomness
- **Adversarial tape:** all randomness can be dependent



Key idea: We fix ρ_a and operate on ρ_h

Rewinding lemma (1/2)

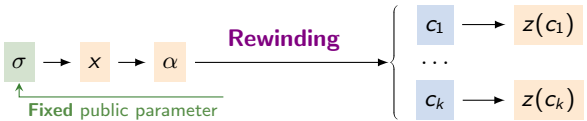
Look at random tapes

Honest random tape

$$\rho = (\rho_h, \rho_a)$$

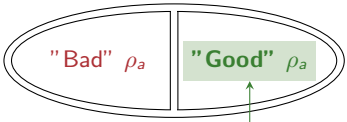
Adversarial random tape

- **Honest tape:** uniform and independent randomness
- **Adversarial tape:** all randomness can be dependent



Key idea: We fix ρ_a and operate on ρ_h

But we have to be careful!



$\Longrightarrow$ We want to *select* the "good" space

New predicate (conditional probabilities)

Formula studied

$$[\![\text{select-tape } g\ \phi]\!]_\eta^{(\rho_h,\rho_a)} = 1$$

Fixed tape

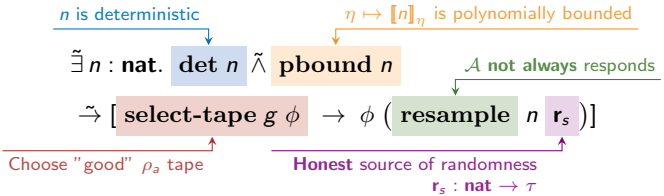
means $\Longleftrightarrow$ $\Pr_{\rho_h'}\left([\![\phi]\!]_\eta^{(\rho_h', \rho_a)} \right) \geqslant [\![g]\!]_\eta$

Tape we study

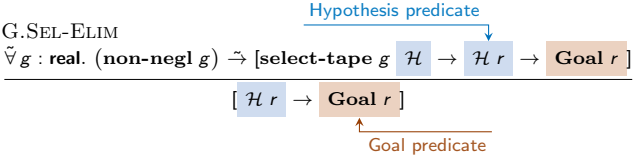
Non-negligible

Rewinding lemma (2/2)

Rewinding lemma



Glue splitted parts back together



Key take-aways and future works

Key take-aways

Context

- More complex protocols are, harder is security analysis
- $\Longrightarrow$ we need formal proof with computational guarantees
- The CCSA logic is promissing and at the good level of abstraction

Our contributions

- First **complete** cryptographic proof of Terelius-Wikström shuffle protocol
- Additions to the expressivity of the CCSA logic: handle conditional probabilities and non-negligible formulas

Key take-aways and future works

Key take-aways

Context

- More complex protocols are, harder is security analysis
- $\Longrightarrow$ we need formal proof with computational guarantees
- The CCSA logic is promissing and at the good level of abstraction

Our contributions

- First **complete** cryptographic proof of Terelius-Wikström shuffle protocol
- Additions to the expressivity of the CCSA logic: handle conditional probabilities and non-negligible formulas

Future works

- Σ-protocols $\longrightarrow$ NIZK proofs (Fiat-Shamir transformation)
- Handle Reprogrammable Random Oracle Model
- Modularity (proof of Bayer-Groth shuffle protocol)

Key take-aways and future works

Key take-aways

Context

- More complex protocols are, harder is security analysis
- $\Longrightarrow$ we need formal proof with computational guarantees
- The CCSA logic is promissing and at the good level of abstraction

Our contributions

- First **complete** cryptographic proof of Terelius-Wikström shuffle protocol
- Additions to the expressivity of the CCSA logic: handle conditional probabilities and non-negligible formulas

Future works

- Σ-protocols $\longrightarrow$ NIZK proofs (Fiat-Shamir transformation)
- Handle Reprogrammable Random Oracle Model
- Modularity (proof of Bayer-Groth shuffle protocol)

Thank you for your attention!