Reduction of equational theories for verification of trace equivalence: re-encryption, associativity and commutativity

Myrto Arapinis
joint work with Sergiu Bursuc and Mark Ryan

Process equivalence and formal verification

Authentication [Abadi, Gordon; Information and Computation'99]:

$$P_{\text{impl}} \sim P_{\text{spec}}$$

Strong secrecy [Blanchet; Security and Privacy'04]:

$$P[s] \sim P[s']$$

Guessing attacks [Corin, Doumen, Etalle; Elec. TCS'05]:

$$\nu s.P \sim P$$

Vote privacy [Kremer, Ryan; ESOP'05]:

$$P[\ \mathrm{V}(\mathrm{id}_1, a) \mid \mathrm{V}(\mathrm{id}_2, b)\ ] \sim P[\ \mathrm{V}(\mathrm{id}_1, b) \mid \mathrm{V}(\mathrm{id}_2, a)\ ]$$

Unlinkability [Arapinis, Chothia, Ritter, Ryan; CSF'10]:

$$P[Q] \sim P[!Q]$$

Automated verification of process equivalence

- Deciding framed bisimilarity [Huttel; Elec.TCS'02]
- Automatic testing equivalence verification from spi-calculus specifications [Durante, Sisto, Valenzano; ACM Trans.SEM'03]
- Deciding security of protocols against off-line guessing attacks [Baudet; ACM CCS'05].
- Automated verification of selected equivalences for the applied pi-calculus [Blanchet, Abadi, Fournet; LICS'05]
- A method for proving observational equivalence [Cortier, Delaune; CSF'09]
- Automating open bisimulation checking for the spi calculus [Tiu,Dawson; CSF'10]
- Automating security analysis: symbolic equivalence of constraint systems [Cheval, Comon, Delaune; IJCAR'10]
- Trace equivalence decision: negative tests and non-determinism [Cheval, Comon, Delaune; ACM CCS'11]
- Automated verification of equivalence properties of cryptographic protocols [Chadha, Ciobâcă, Kremer; ESOP'12]

Automated verification of process equivalence

`Examples not covered by current tools and algorithms:`

Re-encryption

$$\mathcal{E}_{\mathsf{renc}} : \begin{cases} \mathsf{renc}(\mathsf{enc}(x, y, z), z') = \mathsf{enc}(x, y, z \oplus z') \\ \mathsf{renc}(\mathsf{renc}(x, z), z') = \mathsf{renc}(x, z \oplus z') \end{cases}$$

Associativity and commutativity

$$\mathcal{E}_{\oplus} : \begin{cases} x \oplus (y \oplus z) = (x \oplus y) \oplus z \\ x \oplus y = y \oplus x \end{cases}$$

Homomorphic encryption

$$\mathcal{E}_{\mathsf{hom}} : \mathsf{enc}(x_1, y, z_1) \otimes \mathsf{enc}(x_2, y, z_2) = \mathsf{enc}(x_1 \oplus x_2, y, z_1 \oplus z_2)$$

Automated verification of process equivalence

`Examples not covered by current tools and algorithms:`

Re-encryption

$$\mathcal{E}_{\mathsf{renc}} : \begin{cases} \mathsf{renc}(\mathsf{enc}(x, y, z), z') = \mathsf{enc}(x, y, z \oplus z') \\ \mathsf{renc}(\mathsf{renc}(x, z), z') = \mathsf{renc}(x, z \oplus z') \end{cases}$$

Associativity and commutativity

$$\mathcal{E}_{\oplus} : \begin{cases} x \oplus (y \oplus z) = (x \oplus y) \oplus z \\ x \oplus y = y \oplus x \end{cases}$$

Homomorphic encryption

$$\mathcal{E}_{\mathsf{hom}} : \mathsf{enc}(x_1, y, z_1) \otimes \mathsf{enc}(x_2, y, z_2) = \mathsf{enc}(x_1 \oplus x_2, y, z_1 \oplus z_2)$$

Research goal: reduction of equational theories

$$\mathcal{E}_c \rightsquigarrow \mathcal{E}_r$$

- Soundness

$$P \sim_{\mathcal{E}_r} Q \implies P \sim_{\mathcal{E}_c} Q$$

- Completeness

$$P \sim_{\mathcal{E}_c} Q \implies P \sim_{\mathcal{E}_r} Q$$

Re-encryption, associativity and commutativity

$$\mathcal{E}_{\mathsf{DY}} = \begin{cases} \mathsf{dec}(\mathsf{enc}(x, \mathsf{pub}(y), z), y) = x \\ \pi_1(\langle x, y \rangle) = x \\ \pi_2(\langle x, y \rangle) = y \end{cases}$$

$$\mathcal{E}_{\mathsf{renc}} = \begin{cases} \mathsf{renc}(\mathsf{enc}(x, y, z), z') = \mathsf{enc}(x, y, z \oplus z') \\ \mathsf{renc}(\mathsf{renc}(x, z), z') = \mathsf{renc}(x, z \oplus z') \end{cases}$$

$$\mathcal{E}_{\oplus} = \begin{cases} x \oplus (y \oplus z) = (x \oplus y) \oplus z \\ x \oplus y = y \oplus x \end{cases}$$

$$\mathcal{E}_{\mathsf{ver}} = \begin{cases} \mathsf{checkDec}(\mathsf{decPf}(\mathsf{enc}(x, \mathsf{pub}(y), z), x, y), \\ \qquad \mathsf{enc}(x, \mathsf{pub}(y), z), x, \mathsf{pub}(y)) = \mathsf{ok} \\ \mathsf{checkMix}(\mathsf{mixPf}(x, y, \mathsf{renc}(x, z_x), \mathsf{renc}(y, z_y), z_x, z_y), \\ \qquad x, y, \mathsf{renc}(x, z_x), \mathsf{renc}(y, z_y)) = \mathsf{ok} \\ \mathsf{checkMix}(\mathsf{mixPf}(x, y, \mathsf{renc}(y, z_y), \mathsf{renc}(x, z_x), z_y, z_x), \\ \qquad x, y, \mathsf{renc}(y, z_y), \mathsf{renc}(x, z_x)) = \mathsf{ok} \end{cases}$$

Re-encryption, associativity and commutativity

$$\mathcal{E}_{\mathsf{DY}} = \begin{cases} \mathsf{dec}(\mathsf{enc}(x, \mathsf{pub}(y), z), y) = x \\ \pi_1(\langle x, y \rangle) = x \\ \pi_2(\langle x, y \rangle) = y \end{cases}$$

$$\mathcal{E}_{\mathsf{renc}} = \begin{cases} \mathsf{renc}(\mathsf{enc}(x, y, z), z') = \mathsf{enc}(x, y, z \oplus z') \\ \mathsf{renc}(\mathsf{renc}(x, z), z') = \mathsf{renc}(x, z \oplus z') \end{cases}$$

$$\mathcal{E}_{\oplus} = \begin{cases} x \oplus (y \oplus z) = (x \oplus y) \oplus z \\ x \oplus y = y \oplus x \end{cases}$$

$$\mathcal{E}_{\mathsf{ver}} = \begin{cases} \mathsf{checkDec}(\mathsf{decPf}(\mathsf{enc}(x, \mathsf{pub}(y), z), x, y), \\ \qquad \mathsf{enc}(x, \mathsf{pub}(y), z), x, \mathsf{pub}(y)) = \mathsf{ok} \\ \mathsf{checkMix}(\mathsf{mixPf}(x, y, \mathsf{renc}(x, z_x), \mathsf{renc}(y, z_y), z_x, z_y), \\ \qquad x, y, \mathsf{renc}(x, z_x), \mathsf{renc}(y, z_y)) = \mathsf{ok} \\ \mathsf{checkMix}(\mathsf{mixPf}(x, y, \mathsf{renc}(y, z_y), \mathsf{renc}(x, z_x), z_y, z_x), \\ \qquad x, y, \mathsf{renc}(y, z_y), \mathsf{renc}(x, z_x)) = \mathsf{ok} \end{cases}$$

Re-encryption, associativity and commutativity

$$\mathcal{E}_{\mathsf{DY}} = \begin{cases} \mathsf{dec}(\mathsf{renc}^n(\mathsf{enc}(x, \mathsf{pub}(y), z), z_1, \ldots, z_n), y) = x \\ \pi_1(\langle x, y \rangle) = x \\ \pi_2(\langle x, y \rangle) = y \end{cases}$$

$$\mathcal{E}_{\mathsf{renc}} = \begin{cases} \mathsf{renc}(\mathsf{enc}(x, y, z), z') = \mathsf{enc}(x, y, z \oplus z') \\ \mathsf{renc}(\mathsf{renc}(x, z), z') = \mathsf{renc}(x, z \oplus z') \end{cases}$$

$$\mathcal{E}_{\oplus} = \begin{cases} x \oplus (y \oplus z) = (x \oplus y) \oplus z \\ x \oplus y = y \oplus x \end{cases}$$

$$\mathcal{E}_{\mathsf{ver}} = \begin{cases} \mathsf{checkDec}(\mathsf{decPf}(\mathsf{renc}^n(\mathsf{enc}(x, \mathsf{pub}(y), z), z_1, \ldots, z_n), x, y), \\ \qquad \mathsf{renc}^n(\mathsf{enc}(x, \mathsf{pub}(y), z), z_1, \ldots, z_n), x, \mathsf{pub}(y)) = \mathsf{ok} \\ \mathsf{checkMix}(\mathsf{mixPf}(x, y, \mathsf{renc}^n(x, \ldots), \mathsf{renc}^n(y, \ldots), z_x, z_y), \\ \qquad x, y, \mathsf{renc}^n(x, \ldots), \mathsf{renc}^n(y, \ldots)) = \mathsf{ok} \\ \mathsf{checkMix}(\mathsf{mixPf}(x, y, \mathsf{renc}^n(y, \ldots), \mathsf{renc}^n(x, \ldots), z_y, z_x), \\ \qquad x, y, \mathsf{renc}^n(y, \ldots), \mathsf{renc}^n(x, \ldots)) = \mathsf{ok} \end{cases}$$

Reduction of equational theories

$$\mathcal{E} = \mathcal{E}_0 \cup \mathcal{E}_{\mathsf{renc}} \cup \mathcal{E}_{\oplus} \quad \rightsquigarrow \quad \mathcal{E}_n$$

Reduction of equational theories

$$\mathcal{E} = \mathcal{E}_0 \cup \mathcal{E}_{\mathsf{renc}} \cup \mathcal{E}_{\oplus} \quad \rightsquigarrow \quad \mathcal{E}_n$$

Main theorem: $P \sim_{\mathcal{E}_n} Q \implies P \sim_{\mathcal{E}} Q$

where $n = 2 * \max(\sharp_{\mathsf{renc}}(P), \sharp_{\mathsf{renc}}(Q)) + 1$

Reduction of equational theories

$$\mathcal{E} = \mathcal{E}_0 \cup \mathcal{E}_{\mathsf{renc}} \cup \mathcal{E}_{\oplus} \quad \rightsquigarrow \quad \mathcal{E}_n$$

Main theorem: $P \sim_{\mathcal{E}_n} Q \implies P \sim_{\mathcal{E}} Q$

where $n = 2 * \max(\sharp_{\mathsf{renc}}(P), \sharp_{\mathsf{renc}}(Q)) + 1$

Main assumptions:

- $!R \in \mathsf{sp}(P, Q) \implies \mathsf{renc} \notin \mathsf{sig}(R)$
- $f \notin \mathsf{sig}(P) \cup \mathsf{sig}(\mathcal{E} \setminus \{\mathcal{E}_{\mathsf{renc}}, \mathcal{E}_{\oplus}\})$

Reduction of equational theories

$$\mathcal{E} = \mathcal{E}_0 \cup \mathcal{E}_{\mathsf{renc}} \cup \mathcal{E}_{\oplus} \quad \rightsquigarrow \quad \mathcal{E}_n$$

Main theorem: $P \sim_{\mathcal{E}_n} Q \implies P \sim_{\mathcal{E}} Q$

where $n = 2 * \max(\sharp_{\mathsf{renc}}(P), \sharp_{\mathsf{renc}}(Q)) + 1$

Main assumptions:

- $!R \in \mathsf{sp}(P, Q) \implies \mathsf{renc} \notin \mathsf{sig}(R)$
- $f \notin \mathsf{sig}(P) \cup \mathsf{sig}(\mathcal{E} \setminus \{\mathcal{E}_{\mathsf{renc}}, \mathcal{E}_{\oplus}\})$

Assumptions are satisfied by e.g. electronic voting protocols:

- bound on the number of processed ballots
- arithmetic operations confined to cryptographic primitives

Technical details

Term algebra and static equivalence

$$\left.\begin{array}{rl} \mathcal{N} = & a, b, m, n, r, k_1, k_2, \ldots \\ \mathcal{X} = & x, y, z, \ldots \\ \mathcal{F} = & \mathsf{enc}, \mathsf{dec}, \mathsf{pub}, \mathsf{renc}, \langle ., . \rangle, \ldots \end{array}\right\} \rightsquigarrow \mathcal{T}(\mathcal{F}, \mathcal{N}, \mathcal{X})$$

$$u_1 = v_1, \ldots, u_k = v_k \rightsquigarrow \mathcal{E}$$

Frame:

$$\phi = \nu \tilde{n}.\{x_1 \mapsto u_1, \ldots, x_n \mapsto u_n\}$$
$$\psi = \nu \tilde{n}.\{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$$

Recipes

$$\Re(\phi) = \mathcal{T}(\mathcal{F}, \mathcal{N} \setminus \tilde{n}, \{x_1, \ldots, x_n\})$$

Static equivalence: $\phi \overset{s}{\sim} \psi$ iff

- $\mathsf{dom}(\phi) = \mathsf{dom}(\psi)$ and
- $\forall \zeta_1, \zeta_2 \in \Re(\phi).$

$$\zeta_1[\phi] =_{\mathcal{E}} \zeta_2[\phi] \Leftrightarrow \zeta_1[\psi] =_{\mathcal{E}} \zeta_2[\psi]$$

Process calculus and trace equivalence

$P, Q, R ::=$

0	null process
$P \mid Q$	parallel composition
$!P$	replication
$\nu n.P$	name restriction
$\text{if } u = v \text{ then } P \text{ else } Q$	conditional
$\mathsf{c}(x).P$	message input
$\overline{\mathsf{c}}\langle u \rangle.P$	message output
$\{x \mapsto u\}$	frame element

Process calculus and trace equivalence

$P, Q, R ::=$

0	null process
$P \mid Q$	parallel composition
$!P$	replication
$\nu n.P$	name restriction
$\text{if } u = v \text{ then } P \text{ else } Q$	conditional
$\mathsf{c}(x).P$	message input
$\overline{\mathsf{c}}\langle u \rangle.P$	message output
$\{x \mapsto u\}$	frame element

Traces:

$$
\begin{array}{rcl}
\tau & = & P_0 \xrightarrow{\alpha_0} P_1 \xrightarrow{\alpha_1} \ldots P_{n-1} \xrightarrow{\alpha_{n-1}} P_n \\
 & & \alpha_i \in \{\overline{\mathsf{c}}\langle x_i \rangle, c(\zeta_i), \epsilon \mid \zeta_i \in \Re(\mathsf{fr}(P_i))\} \\
\mathsf{obs}(\tau) & = & \alpha_{i_1} \ldots \alpha_{i_n} \text{ if } \alpha_1 \ldots \alpha_n = \epsilon^* \alpha_{i_1} \epsilon^* \ldots \alpha_{i_n} \epsilon^* \\
\mathsf{fr}(\tau) & = & \mathsf{fr}(P_n)
\end{array}
$$

Process calculus and trace equivalence

$P, Q, R ::=$

0	null process
$P \mid Q$	parallel composition
$!P$	replication
$\nu n.P$	name restriction
$\text{if } u = v \text{ then } P \text{ else } Q$	conditional
$\mathsf{c}(x).P$	message input
$\overline{\mathsf{c}}\langle u \rangle.P$	message output
$\{x \mapsto u\}$	frame element

Traces:

$$\begin{array}{rcl} \tau & = & P_0 \xrightarrow{\alpha_0} P_1 \xrightarrow{\alpha_1} \ldots P_{n-1} \xrightarrow{\alpha_{n-1}} P_n \\ & & \alpha_i \in \{\overline{\mathsf{c}}\langle x_i \rangle, \mathsf{c}(\zeta_i), \epsilon \mid \zeta_i \in \Re(\mathsf{fr}(P_i))\} \\ \mathsf{obs}(\tau) & = & \alpha_{i_1} \ldots \alpha_{i_n} \text{ if } \alpha_1 \ldots \alpha_n = \epsilon^* \alpha_{i_1} \epsilon^* \ldots \alpha_{i_n} \epsilon^* \\ \mathsf{fr}(\tau) & = & \mathsf{fr}(P_n) \end{array}$$

Trace equivalence

$$\begin{array}{rcl} \tau_1 \sim_{\mathcal{E}} \tau_2 & \Leftrightarrow & \mathsf{obs}(\tau_1) = \mathsf{obs}(\tau_2) \;\&\; \mathsf{fr}(\tau_1) \overset{\mathsf{s}}{\sim}_{\mathcal{E}} \mathsf{fr}(\tau_2) \\ P \sim_{\mathcal{E}} Q & \Leftrightarrow & \mathsf{tr}(P) \sim_{\mathcal{E}} \mathsf{tr}(Q) \end{array}$$

Reduction of equational theories

$$\mathcal{E} = \mathcal{E}_0 \cup \mathcal{E}_{\mathsf{renc}} \cup \mathcal{E}_{\oplus} \quad \rightsquigarrow \quad \mathcal{E}_n$$

Main theorem: $P \sim_{\mathcal{E}_n} Q \implies P \sim_{\mathcal{E}} Q$

where $n = 2 * \max(\sharp_{\mathsf{renc}}(P), \sharp_{\mathsf{renc}}(Q)) + 1$

Scheme of the proof:

$$P \sim_{\mathcal{E}_n} Q \implies P \simeq_{\mathcal{E}_n} Q \implies P \simeq_{\mathcal{E}} Q \implies P \sim_{\mathcal{E}} Q$$

Main assumptions:

- $!S \in \mathsf{sp}(P, Q) \implies \mathsf{renc} \notin \mathsf{sig}(S)$
- $\oplus \notin \mathsf{sig}(P, Q)$ and $\oplus \notin \mathsf{sig}(\mathcal{E}_0)$

Reduction of the set of traces

Reduction of the set of traces

Locality function $\mathcal{L}$: $\mathcal{L}(\phi) \subseteq \Re(\phi)$

Local static equivalence: $\phi \sim \psi$ iff

- $\mathsf{dom}(\phi) = \mathsf{dom}(\psi)$
- $\forall \zeta_1, \zeta_2 \in \mathcal{L}(\phi) \cap \mathcal{L}(\psi).\ \zeta_1[\phi] =_\mathcal{E} \zeta_2[\phi] \Leftrightarrow \zeta_1[\psi] =_\mathcal{E} \zeta_2[\psi]$

Local traces:

$$P_0 \xrightarrow{\alpha_0} P_1 \ldots \xrightarrow{\alpha_{n-1}} P_n$$
$$\alpha_i \in \{\overline{\mathsf{c}}\langle x_i \rangle, \mathsf{c}(\zeta_i), \epsilon \mid \zeta_i \in \mathcal{L}(\mathsf{fr}(P_i))\}$$

Local trace equivalence: $P \simeq_\mathcal{E} Q \Leftrightarrow \mathsf{tr}_\mathcal{L}(P) \sim_\mathcal{E} \mathsf{tr}_\mathcal{L}(Q)$

Scheme of the proof

$$P \sim_{\mathcal{E}_n} Q \implies P \simeq_{\mathcal{E}_n} Q \implies P \simeq_{\mathcal{E}} Q \implies P \sim_{\mathcal{E}} Q$$

Reduction of the set of traces

$$P \sim Q \implies P \simeq_{\mathcal{L}} Q$$

$$P \simeq_{\mathcal{L}} Q \implies P \sim Q$$

Reduction of the set of traces

Proposition 1. Let $\mathcal{L}$ be a locality function such that, for all frames ϕ, ψ that are issued from two statically equivalent traces, we have $\mathcal{L}(\phi) = \mathcal{L}(\psi)$. Then, for all plain processes P, Q, we have

$$P \sim Q \implies P \simeq_{\mathcal{L}} Q$$

$$P \simeq_{\mathcal{L}} Q \implies P \sim Q$$

Reduction of the set of traces

Proposition 1. Let $\mathcal{L}$ be a locality function such that, for all frames ϕ, ψ that are issued from two statically equivalent traces, we have $\mathcal{L}(\phi) = \mathcal{L}(\psi)$. Then, for all plain processes P, Q, we have

$$P \sim Q \implies P \simeq_{\mathcal{L}} Q$$

Normalization function $\mathsf{N} : \Re(\phi) \mapsto \mathcal{L}(\phi)$

$$\forall \zeta \in \Re(\phi). \quad \mathsf{N}(\zeta) \in \mathcal{L}(\phi) \ \&\ \mathsf{N}(\zeta)[\phi] =_{\mathcal{E}} \zeta[\phi]$$

$$P \simeq_{\mathcal{L}} Q \implies P \sim Q$$

Reduction of the set of traces

Proposition 1. Let $\mathcal{L}$ be a locality function such that, for all frames ϕ, ψ that are issued from two statically equivalent traces, we have $\mathcal{L}(\phi) = \mathcal{L}(\psi)$. Then, for all plain processes P, Q, we have

$$P \sim Q \implies P \simeq_{\mathcal{L}} Q$$

Normalization function $\mathsf{N} : \mathfrak{R}(\phi) \mapsto \mathcal{L}(\phi)$

$$\forall \zeta \in \mathfrak{R}(\phi). \quad \mathsf{N}(\zeta) \in \mathcal{L}(\phi) \ \& \ \mathsf{N}(\zeta)[\phi] =_{\mathcal{E}} \zeta[\phi]$$

Proposition 2. Let $\mathcal{L}$ be a locality function such that, for all frames ϕ, ψ that are issued from two $\mathcal{L}$-statically equivalent and $\mathcal{L}$-local traces, there exists a normalization function $\mathsf{N} \in \mathsf{norm}_{\mathcal{L}}(\phi) \cap \mathsf{norm}_{\mathcal{L}}(\psi)$. Then, for all processes P, Q, we have

$$P \simeq_{\mathcal{L}} Q \implies P \sim Q$$

Scheme of the proof

$$P \sim_{\mathcal{E}_n} Q \implies P \simeq_{\mathcal{E}_n} Q \implies P \simeq_{\mathcal{E}} Q \implies P \sim_{\mathcal{E}} Q$$

Scheme of the proof

$$P \sim_{\mathcal{E}_n} Q \implies P \simeq_{\mathcal{E}_n} Q \implies P \simeq_{\mathcal{E}} Q \implies P \sim_{\mathcal{E}} Q$$

$$\mathcal{L}_{\mathsf{renc},\oplus}(\phi) = \mathcal{L}_{\mathsf{renc}}(\phi) \cap \mathcal{L}_{\oplus}(\phi)$$

Locality for associativity and commutativity

Goal: ensure a canonical use of $\oplus$

Total ordering on recipes: $\prec$

Minimal recipes: $t \mapsto \min_{\Re}(t)$

Convergent rewrite system: $\mathcal{E} \setminus \mathcal{E}_{\oplus} \rightsquigarrow \mathcal{R}$

Locality function for AC: $\forall \zeta \in \mathsf{st}(\mathcal{L}_{\oplus}(\phi))$

$$
\begin{array}{lcl}
\zeta[\phi]{\downarrow_{\mathcal{R}}} & = & C_{\oplus}[t_1, \ldots, t_n] \text{ and } \top(t_1) \neq \oplus, \ldots, \top(t_n) \neq \oplus \\
& \Longrightarrow & \\
& 1. & C_{\oplus} = (\epsilon_1 \oplus (\ldots \oplus (\epsilon_{n-1} \oplus \epsilon_n) \ldots) \\
& 2. & \min_{\Re}(t_1) \prec \ldots \prec \min_{\Re}(t_n)
\end{array}
$$

Normalization function: $\mathsf{N}_{\oplus}(\zeta) = (\zeta_1 \oplus (\ldots \oplus (\zeta_{n-1} \oplus \zeta_n))$

Locality for re-encryption

Goal: bound the number of re-encryptions applied to any ciphertext

Recipes to avoid:

$$\mathsf{RR}(\phi) \quad = \quad \text{“nested re-encryptions by the environment”}$$
$$\text{Example:} \qquad \mathsf{renc}(C[\mathsf{renc}(\zeta_1, \zeta_2)], \zeta_3)$$

Locality function for re-encryption: $\mathsf{RR}(\phi) \cap \mathsf{st}(\mathcal{L}_{\mathsf{renc}}(\phi)) = \emptyset$

Normalization function: $\mathsf{N}_{\mathsf{renc}}$

Example: $\mathsf{N}_{\mathsf{renc}}(\mathsf{renc}(C[\mathsf{renc}(\zeta_1, \zeta_2)], \zeta_3)) = \mathsf{renc}(\zeta_1, \zeta_2 \oplus \zeta_3)$

Reduction of equational theories (last step)

$$P \simeq_{\mathcal{E}_n} Q \implies P \simeq_{\mathcal{E}} Q$$

Some ingredients:

- $u =_{\mathcal{E}} v \Leftrightarrow u =_{\mathcal{E} \setminus \mathcal{E}_\oplus} v' \;\&\; v' =_{\mathcal{E}_\oplus} v$
- Randoms used by the protocol are kept secret
- Re-encryption witness and re-encryption depth
- $u \to_{\mathcal{R}} v \implies u \to_{\mathcal{R}_n} v\rho$

Application

Vote privacy in Prêt à Voter with ProVerif