Privacy issues: from protocols to applications

Stéphanie Delaune

LSV, CNRS & ENS Cachan & INRIA Saclay Île-de-France, France

⟶ joint work with Myrto Arapinis and Vincent Cheval (Birmingham University)

Wednesday, June 12th, 2013

Context: cryptographic protocols

PayPal™

Cryptographic protocols

- small programs designed to secure communication (*e.g.* confidentiality, authentication, ...)
- use cryptographic primitives (*e.g.* encryption, signature, ...)

The network is unsecure!

Communications take place over a public network like the Internet.

PayPal™

Cryptographic protocols

- small programs designed to secure communication (*e.g.* confidentiality, authentication, ...)
- use cryptographic primitives (*e.g.* encryption, signature, ...)

élections professionnelles 2011 cliquez, votez

Context: cryptographic protocols

PayPal™

Cryptographic protocols

- small programs designed to secure communication (*e.g.* confidentiality, authentication, ...)
- use cryptographic primitives (*e.g.* encryption, signature, ...)

It becomes more and more important to protect our privacy.

Composition issues

Some motivations:

- Existing tools allow us to verify relatively small protocols
- Most often, we verify them in isolation

Composition issues

Some motivations:

- Existing tools allow us to verify relatively small protocols
- Most often, we verify them in isolation

This is, of course, **not sufficient** !

Example:

$P_1 : \quad A \rightarrow B : \quad \{A\}^r_{\mathsf{pub}(B)}$

What about the anonymity of A?

Composition issues

Some motivations:

- Existing tools allow us to verify relatively small protocols
- Most often, we verify them in isolation

This is, of course, **not sufficient** !

Example:

$P_1: \quad A \rightarrow B: \quad \{A\}^r_{\mathsf{pub}(B)}$

$P_2: \quad A \rightarrow B: \quad \{N_a\}^r_{\mathsf{pub}(B)}$
$\quad\quad\; B \rightarrow A: \quad N_a$

What about the anonymity of A?

Case study: electronic passport

⟶ studied in [Arapinis *et al.*, 10]

An electronic passport is a passport with an RFID tag embedded in it.

The RFID tag stores:

- the information printed on your passport,
- a JPEG copy of your picture.

Case study: electronic passport

⟶ studied in [Arapinis *et al.*, 10]

An electronic passport is a passport with an RFID tag embedded in it.

The RFID tag stores:

- the information printed on your passport,
- a JPEG copy of your picture.

The ICAO standard specifies several sub-protocols, e.g.

1. the Basic Access Control (*BAC*) protocol;
2. the Passive Authentication (*PA*) protocol;
3. the Active Authentication (*AA*) protocol;
4. . . .

Case study: 3G mobile phones

$\longrightarrow$ studied in [Arapinis *et al.*, 12]

The UMTS standard specifies tens of sub-protocols running together in 3G mobile phone systems.

Case study: 3G mobile phones

⟶ studied in [Arapinis *et al.*, 12]

The UMTS standard specifies tens of sub-protocols running together in 3G mobile phone systems.

What about privacy guarantees provided by:

- the Authentication and Key Agreement protocol (AKA), and
- the Submit SMS procedure (sSMS)

when run in composition as specified in the standard?

Case study: 3G mobile phones

⟶ studied in [Arapinis *et al.*, 12]

The UMTS standard specifies tens of sub-protocols running together in 3G mobile phone systems.

What about privacy guarantees provided by:

- the Authentication and Key Agreement protocol (AKA), and
- the Submit SMS procedure (sSMS)

when run in composition as specified in the standard?

⟶ **A need of composition results !**

Our contributions

investigate sufficient conditions to ensure that protocols (that may share some keys) still preserve some privacy guarantees when used in an environment where some other protocols may be executed as well.

Our contributions

investigate sufficient conditions to ensure that protocols (that may share some keys) still preserve some privacy guarantees when used in an environment where some other protocols may be executed as well.

State of the art in 2012:

Several results already exist for sequential/parallel composition, *e.g.*:

- parallel composition using tagging
  $\longrightarrow$ [Guttman & Thayer, 2000], [Cortier *et al.*, 2007]
- sequential composition for arbitrary primitives
  $\longrightarrow$ [Ciobaca & Cortier, 2010]

Our contributions

investigate sufficient conditions to ensure that protocols (that may share some keys) still preserve some privacy guarantees when used in an environment where some other protocols may be executed as well.

State of the art in 2012:

Several results already exist for sequential/parallel composition, *e.g.*:

- parallel composition using tagging
  $\longrightarrow$ [Guttman & Thayer, 2000], [Cortier *et al.*, 2007]
- sequential composition for arbitrary primitives
  $\longrightarrow$ [Ciobaca & Cortier, 2010]

None of them are well-suited for analysing privacy-type properties

Our CSF 2012 paper

Main result

The first composition result that allows one to analyse privacy-type properties in a modular way.

- we consider processes that may share some keys and also some primitives provided that they are tagged (syntactic condition);
- we consider parallel composition only;

$\longrightarrow$ this allows us to analyse the passive/active authentication protocols of the e-passport application in a modular way

A first composition result – CSF 2012

Theorem (simplified version)

Let C and C' be two composition contexts. Let P_A/P'_A and P_B/P'_B be two pairs of processes built on disjoint signatures. Assume that $C[P_A]$, $C'[P'_A]$, $C[P_B]$, and $C'[P'_B]$ *do not reveal* any shared key. We have that:

$$C[P_A] \approx C'[P'_A] \;\wedge\; C[P_B] \approx C'[P'_B] \;\Rightarrow\; C[P_A \mid P_B] \approx C'[P'_A \mid P'_B]$$

In the full version, we consider:

- composition contexts with several holes;
- protocols may share some primitives in a fixed signature assuming some tagging;
- we may allow some keys (the public ones !) to be revealed

Main idea – CSF 2012

$\longrightarrow$ to go back to the disjoint case for which composition works well.

Main idea – CSF 2012

$\longrightarrow$ to go back to the disjoint case for which composition works well.

Step 1: Apply (disjoint) parallel composition

$$C[P_A] \mid C[P_B] \approx C'[P'_A] \mid C'[P'_B]$$

Main idea – CSF 2012

$\longrightarrow$ to go back to the disjoint case for which composition works well.

Step 1: Apply (disjoint) parallel composition

$$C[P_A] \mid C[P_B] \approx C'[P'_A] \mid C'[P'_B]$$

Step 2: Apply the following result on both sides:

Proposition

Let C be a composition context, and P_A (resp. P_B) be two processes built on disjoint signatures. Assume that $C[P_A]$ and $C[P_B]$ do not reveal any shared key. We have that:

$$C[P_A] \mid C[P_B] \approx C[P_A \mid P_B]$$

Main idea – CSF 2012

$\longrightarrow$ to go back to the disjoint case for which composition works well.

Step 1: Apply (disjoint) parallel composition

$$C[P_A] \mid C[P_B] \approx C'[P'_A] \mid C'[P'_B]$$

Step 2: Apply the following result on both sides:

Proposition

Let C be a composition context, and P_A (resp. P_B) be two processes built on disjoint signatures. Assume that $C[P_A]$ and $C[P_B]$ do not reveal any shared key. We have that:

$$C[P_A] \mid C[P_B] \approx C[P_A \mid P_B]$$

Step 3: We conclude that $C[P_A \mid P_B] \approx C'[P'_A \mid P'_B]$.

Beyond parallel composition

Our goal: a generic composition result to go beyond parallel composition for privacy-type properties

Beyond parallel composition

Our goal: a generic composition result to go beyond parallel composition for privacy-type properties

Targeted application: the case of key exchange protocol, i.e.

- $P = \texttt{new}\ \tilde{n}.(P_1 \mid P_2)$, a key establishment protocol between two participants.
- Q a two party protocol that uses the established key.

What about privacy guarantees offered by $\texttt{new}\ \tilde{n}.(P_1[Q_1] \mid P_2[Q_2])$?

Beyond parallel composition

Our goal: a generic composition result to go beyond parallel composition for privacy-type properties

Targeted application: the case of key exchange protocol, i.e.

- $P = \texttt{new}\ \tilde{n}.(P_1 \mid P_2)$, a key establishment protocol between two participants.
- Q a two party protocol that uses the established key.

What about privacy guarantees offered by $\texttt{new}\ \tilde{n}.(P_1[Q_1] \mid P_2[Q_2])$?

Case studies:

- E-passports: BAC followed by $PA \mid AA$;
- 3G mobile phones: AKA followed by $sSMS$;

Some difficulties

As usual, we have to assume that:

- protocols do not share any primitive
  $\longrightarrow$ we may assume some common primitives provided some tagging.
- shared keys are not revealed
  $\longrightarrow$ actually, for equivalence-based properties, revealing public keys already require some additional work (remember that we have to preserve static equivalence)

In addition, we rely on assignment variables:
$\longrightarrow$ the values of the shared keys are *not known a priori*, they are not atomic anymore, and they depend on the underlying execution.

Trace equivalence does **not** compose well (in sequence)

This is wrong !

$$C[Q] \approx C[Q'] \Longrightarrow C[P[Q]] \approx C[P[Q']]$$

(develop the example on the board if needed)

Trace equivalence does **not** compose well (in sequence)

This is wrong !

$$C[Q] \approx C[Q'] \Longrightarrow C[P[Q]] \approx C[P[Q']]$$

(develop the example on the board if needed)

$\longrightarrow$ **We need to consider a stronger notion of equivalence**

Biprocesses and diff-equivalence

A biprocess is a pair of processes that have to evolve simultaneously ($\xrightarrow{\ell}_{\mathsf{bi}}$).

Definition (diff-equivalence)

A biprocess B_0 is in *diff-equivalence* if for every biprocess B such that $B_0 \stackrel{\mathsf{tr}}{\Rightarrow}_{\mathsf{bi}} B$ for some trace tr, we have that:

1. static equivalence: $\mathsf{fst}(B) \sim \mathsf{snd}(B)$;
2. if $\mathsf{fst}(B) \xrightarrow{\ell} A_L$ then there exists a biprocess B' such that $B \xrightarrow{\ell}_{\mathsf{bi}} B'$ and $\mathsf{fst}(B') = A_L$ (and similarly for snd).

We sometimes write $\mathsf{fst}(B_0) \approx_{\mathsf{diff}} \mathsf{snd}(B_0)$.

(Note that diff-equivalence rules out the previous counter-example)

Going back to the disjoint case (again)

Main idea: abstract the values of the assignment variables issued from the other process with fresh names preserving equalities/disequalities
$\longrightarrow$ contrary to the case of parallel composition, the abstraction ρ may vary form one execution trace to another.

Theorem

Let S_0 be a process "made up" of two processes built on disjoint signatures and sharing some values through assignment variables. Let ρ be an abstraction.

1. *For any process S such that $S_0 \stackrel{\mathsf{tr}}{\Rightarrow} S$ and compatible with ρ, we have that $\delta_\rho(S_0) \stackrel{\mathsf{tr}}{\Rightarrow} \delta_\rho(S)$, and $S \sim \delta_\rho(S)$.*
2. *For any process D such that $\delta(S_0) \stackrel{\mathsf{tr}}{\Rightarrow} D$ and compatible with ρ, we have that $S_0 \stackrel{\mathsf{tr}}{\Rightarrow} S$, $D = \delta_\rho(S)$, and $D \sim S$.*

Parallel composition (a variant of our CSF 2012 result)

Using the previous theorem, we can show that:

$$C[P_A \mid P_B] \approx_{\mathsf{diff}} C[P_A] \mid C[P_B].$$

Hence, we retrieve the same result as CSF 2012 but for diff-equivalence:

$$\frac{C[P_A] \approx_{\mathsf{diff}} C'[P'_A] \qquad C[P_B] \approx_{\mathsf{diff}} C'[P'_B]}{C[P_A \mid P_B] \approx_{\mathsf{diff}} C'[P'_A \mid P'_B]}$$

Sequential composition

We want to establish $C[P_1[Q_1] \mid P_2[Q_2]] \approx C'[P_1[Q_1] \mid P_2[Q_2]]$ in a modular way, i.e. from the smaller equivalences:

1. $C[P_1 \mid P_2] \approx C'[P_1 \mid P_2]$; and
2. $C[Q] \approx C'[Q]$ with $Q = \mathtt{new}\, k.[x_1, x_2 := k](Q_1 \mid Q_2)$.

Sequential composition

We want to establish $C[P_1[Q_1] \mid P_2[Q_2]] \approx C'[P_1[Q_1] \mid P_2[Q_2]]$ in a modular way, i.e. from the smaller equivalences:

1. $C[P_1 \mid P_2] \approx C'[P_1 \mid P_2]$; and
2. $C[Q] \approx C'[Q]$ with $Q = \mathtt{new}\, k.[x_1, x_2 := k](Q_1 \mid Q_2)$.

Of course, this is not for free !

Sequential composition

We want to establish $C[P_1[Q_1] \mid P_2[Q_2]] \approx C'[P_1[Q_1] \mid P_2[Q_2]]$ in a modular way, i.e. from the smaller equivalences:

1. $C[P_1 \mid P_2] \approx C'[P_1 \mid P_2]$; and
2. $C[Q] \approx C'[Q]$ with $Q = \mathtt{new}\, k.[x_1, x_2 := k](Q_1 \mid Q_2)$.

Of course, this is not for free !

We have to assume that:

1. processes P_1/P_2 and Q_1/Q_2 are built on disjoint signatures (or shared primitives are tagged);
2. shared keys that occurs in C (and also k) are not revealed;
3. we rely on diff-equivalence;

Sequential composition

We want to establish $C[P_1[Q_1] \mid P_2[Q_2]] \approx C'[P_1[Q_1] \mid P_2[Q_2]]$ in a modular way, i.e. from the smaller equivalences:

1. $C[P_1 \mid P_2] \approx C'[P_1 \mid P_2]$; and
2. $C[Q] \approx C'[Q]$ with $Q = \mathtt{new}\, k.[x_1, x_2 := k](Q_1 \mid Q_2)$.

Of course, this is not for free !

We have to assume that:

1. processes P_1/P_2 and Q_1/Q_2 are built on disjoint signatures (or shared primitives are tagged);
2. shared keys that occurs in C (and also k) are not revealed;
3. we rely on diff-equivalence;
4. P is a good key-exchange protocol (freshness/agreement property).

Freshness/agreement property

We say that $C[P_1 \mid P_2]$ satisfies the freshness/agreement property when $P_{fresh/agree}$ does not reveal the name *bad*.

Process $P_{fresh/agree}$

$$
\begin{array}{l}
\mathtt{new}\ bad.\ \mathtt{new}\ d.\\
\quad C[\mathtt{new}\ id.(P_1[\mathsf{out}(d, \langle x_1, id\rangle)] \mid P_2[\mathsf{out}(d, \langle x_2, id\rangle)])]\\
\quad \mid \mathsf{in}(d, x).\mathsf{in}(d, y).\\
\qquad \mathtt{if}\ \mathsf{proj}_1(x) = \mathsf{proj}_1(y) \wedge \mathsf{proj}_2(x) \neq \mathsf{proj}_2(y)\ \mathtt{then}\ \mathsf{out}(c, bad)\\
\qquad \mathtt{else}\ \mathtt{if}\ \mathsf{proj}_1(x) \neq \mathsf{proj}_1(y) \wedge \mathsf{proj}_2(x) = \mathsf{proj}_2(y)\ \mathtt{then}\ \mathsf{out}(c, bad)
\end{array}
$$

Freshness/agreement property

We say that $C[P_1 \mid P_2]$ satisfies the freshness/agreement property when $P_{fresh/agree}$ does not reveal the name *bad*.

Process $P_{fresh/agree}$

$$\begin{array}{l}\mathtt{new}\ bad.\ \mathtt{new}\ d.\\ \quad C[\mathtt{new}\ id.(P_1[\mathsf{out}(d,\langle x_1, id\rangle)] \mid P_2[\mathsf{out}(d,\langle x_2, id\rangle)])]\\ \quad \mid \mathsf{in}(d,x).\mathsf{in}(d,y).\\ \qquad \mathtt{if}\ \mathsf{proj}_1(x) = \mathsf{proj}_1(y) \wedge \mathsf{proj}_2(x) \neq \mathsf{proj}_2(y)\ \mathtt{then}\ \mathsf{out}(c, bad)\\ \qquad \mathtt{else\ if}\ \mathsf{proj}_1(x) \neq \mathsf{proj}_1(y) \wedge \mathsf{proj}_2(x) = \mathsf{proj}_2(y)\ \mathtt{then}\ \mathsf{out}(c, bad)\end{array}$$

$\longrightarrow$ too strong as soon as we want to consider a composition context of the form $C'[!\_]$

Freshness/ (weak) agreement property

We say that $C[P_1 \mid P_2]$ satisfies the weak agreement property when $P_{weak-agree}$ does not reveal the name *bad*.

$P_{weak-agree}$

$$\texttt{new}\ bad.\ \texttt{new}\ d.$$
$$C'[\texttt{new}\ id.!(P_1[\mathsf{out}(d, \langle x_1, id\rangle)] \mid P_2[\mathsf{out}(d, \langle x_2, id\rangle)]$$
$$\mid \mathsf{in}(d, x).\mathsf{in}(d, y).\texttt{if}\ \mathsf{proj}_1(x) = \mathsf{proj}_1(y)$$
$$\wedge \mathsf{proj}_2(x) \neq \mathsf{proj}_2(y)\ \texttt{then}\ \mathsf{out}(c, bad)$$

Freshness/ (weak) agreement property

We say that $C[P_1 \mid P_2]$ satisfies the weak agreement property when $P_{weak-agree}$ does not reveal the name *bad*.

$P_{weak-agree}$

$$
\begin{array}{l}
\mathtt{new}\ bad.\ \mathtt{new}\ d.\\
C'[\mathtt{new}\ id.!(P_1[\mathsf{out}(d, \langle x_1, id\rangle)] \mid P_2[\mathsf{out}(d, \langle x_2, id\rangle)]\\
\mid \mathsf{in}(d,x).\mathsf{in}(d,y).\mathtt{if}\ \mathsf{proj}_1(x) = \mathsf{proj}_1(y)\\
\qquad\qquad \wedge \mathsf{proj}_2(x) \neq \mathsf{proj}_2(y)\ \mathtt{then}\ \mathsf{out}(c, bad)
\end{array}
$$

Sufficient for composition context of the form $C'[!\_]$ assuming in addition a freshness property, and a stronger result on the protocol P:

$$
\begin{array}{l}
P^+ = C[\mathtt{new}\ d.(P_1[\mathsf{out}(d,x_1)] \mid P_2[\mathsf{out}(d,x_2)]\\
\quad \mid \mathsf{in}(d,x).\mathsf{in}(d,y).\mathtt{if}\ x = y\ \mathtt{then}\ 0\ \mathtt{else}\ 0)]
\end{array}
$$

$\longrightarrow$ P^+ (and not only P) has to be in diff-equivalence.

Case study: e-passport

Parallel composition of *PA* and *AA*

Assuming a tagged version of these protocols, we derive privacy guarantees (anonymity, unlinkability) of $PA \mid AA$ from the results obtained on both protocols studied in isolation.

$\longrightarrow$ we use our CSF 2012 result

Sequential composition of *BAC* and $PA \mid AA$

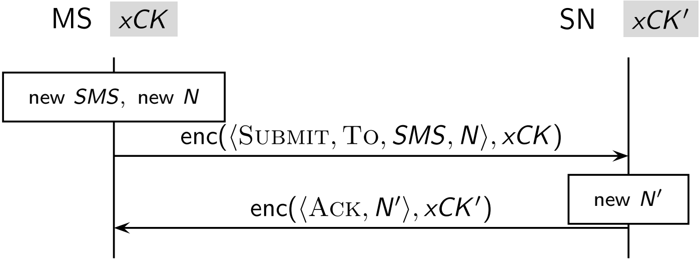
Still out of reach of our composition results !!

$\longrightarrow$ *BAC* (even the fixed version) does not satisfy diff-equivalence

Case study: 3G mobile phones

Sequential composition of *AKA* and *sSMS*

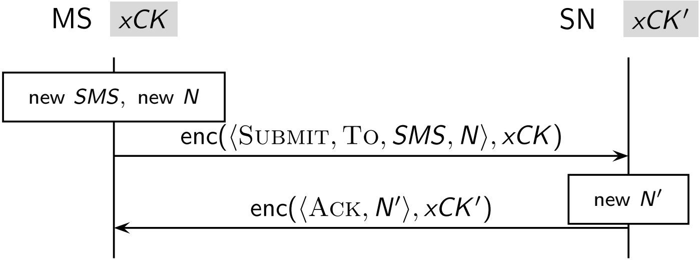
- *AKA* protocol: we need to consider a tagged version of the fixed proposed by Arapinis et al. [CCS'12];
- *sSMS* protocol allows a Mobile Station (MS) to send an SMS to another MS through the Service Network (SN).



Case study: 3G mobile phones

Sequential composition of *AKA* and *sSMS*

- *AKA* protocol: we need to consider a tagged version of the fixed proposed by Arapinis et al. [CCS'12];
- *sSMS* protocol allows a Mobile Station (MS) to send an SMS to another MS through the Service Network (SN).



⟶ we derive confidentiality (strong secrecy) and unlinkability using our new result (the version requiring the strong agreement property)

Conclusion

A generic composition result that is quite powerful

- we maintain a strong relationship between the shared case and *a* disjoint case
- we consider arbitrary primitives + some standard primitives provided some tagging;
- we considera large class of processes (in particular we have else branches).

This result:

- on traces can be reused to establish some other composition results.
- generalizes our CSF 2012 result, and also the main result of Ciobaca and Cortier, CSF 2010

Conclusion

A generic composition result that is quite powerful

- we maintain a strong relationship between the shared case and *a* disjoint case
- we consider arbitrary primitives + some standard primitives provided some tagging;
- we considera large class of processes (in particular we have else branches).

This result:

- on traces can be reused to establish some other composition results.
- generalizes our CSF 2012 result, and also the main result of Ciobaca and Cortier, CSF 2010

Unfortunately, the sequential composition results derived form this generic result are still quite limited.