Composition Theorems for CryptoVerif and Application to TLS 1.3

Bruno Blanchet

INRIA Paris
Bruno.Blanchet@inria.fr

June 2018

Introduction

- Composition between
  - a key exchange protocol
  - a protocol that uses the key
- Results stated in the CryptoVerif framework:
  - protocol verifier in the computational model
  - formal framework for stating the composition theorem
  - prove bigger protocols in CryptoVerif
  - prove protocols with loops in CryptoVerif

Adapt and extend previous computational composition results by Brzuska, Fischlin et al. [CCS'11, CCS'14 and CCS'15]

Application to TLS 1.3

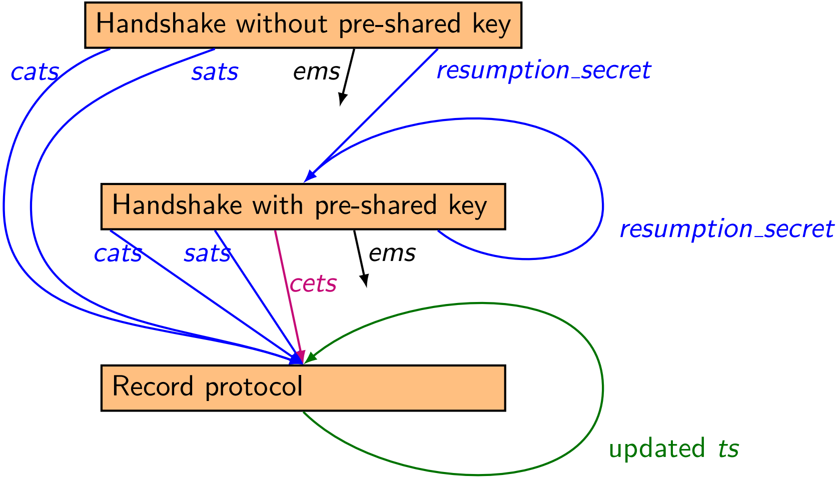
Why TLS 1.3 ?

- Important protocol, in the final stages of development
- Well designed to allow composition
- Contains loops:
  - Unbounded number of handshakes and key updates
- Variety of compositions:
  - In most cases, the key exchange provides injective authentication
  - For 0-RTT data = data sent by the client to the server immediately after the message (ClientHello):
    - possible replay, so non-injective authentication
    - variant for the case of altered ClientHello
  - Simpler composition theorem for key updates

Fills a gap in the proof of TLS 1.3 Draft 18 by Bhargavan et al [S&P'17]

- The composition was stated only informally.

TLS 1.3: Structure of the composition



CryptoVerif, http://cryptoverif.inria.fr/

CryptoVerif is a semi-automatic prover that:

- works in the computational model.
- generates proofs by sequences of games.
- provides a generic method for specifying properties of cryptographic primitives which handles MACs (message authentication codes), symmetric encryption, public-key encryption, signatures, hash functions, Diffie-Hellman key agreements, ...
- works for N sessions (polynomial in the security parameter), with an active adversary.
- gives a bound on the probability of an attack (exact security).

Notations

- CryptoVerif represents protocols using a process calculus.
- P, Q, S: processes
- C: context $=$ process with one or several holes $[\,]$
- $C[P_1, P_2]$: context C with P_1 in the first hole and P_2 in the second hole

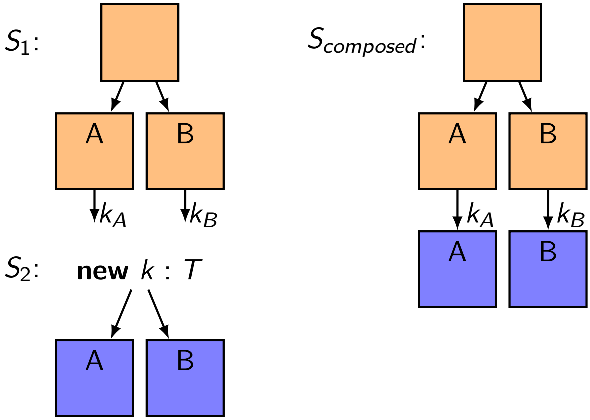
Security properties proved by CryptoVerif

- Indistinguishability: $Q \approx^V Q'$ when an adversary with access to the variables V has a negligible probability of distinguishing Q from Q'.
- Secrecy: Q preserves the secrecy of x with public variables V when an adversary with access to the variables V has a negligible probability of distinguishing the values of x in several sessions from independent random values.
- Correspondences: If some events have been executed, then other events have been executed. Example:

$$\textbf{event}(e_1(x)) \Longrightarrow \textbf{event}(e_2(x))$$

  Q satisfies the correspondence *corr* with public variables V when an adversary with access to the variables V has a negligible probability of breaking *corr*.

Main theorem



(S_1 may run several sessions of A and B.)

Replicating S_2

Consider:

$$S_2 = c(); \dots c_1(y : T) \dots \textbf{event } e(M) \dots$$

We want to replicate S_2:

$$!^{i \leq n}\, c(); \dots c_1(y : T) \dots \textbf{event } e(M) \dots$$

Replicating S_2

Consider:

$$S_2 = c(); \dots c_1(y : T) \dots \textbf{event } e(M) \dots$$

We want to replicate S_2:

$$!^{i \leq n}\, c(); \dots c_1(y[i] : T) \dots \textbf{event } e(M) \dots$$

Variables implicitly with indices of replication.

Replicating S_2

Consider:

$$S_2 = c(); \dots c_1(y : T) \dots \textbf{event } e(M) \dots$$

We want to replicate S_2:

$$!^{i \le n}\, c[i](); \dots c_1[i](y[i] : T) \dots \textbf{event } e(i, M) \dots$$

We could add indices to channels and events to distinguish the various sessions.

Replicating S_2

Consider:

$$S_2 = c(); \dots c_1(y : T) \dots \textbf{event } e(M) \dots$$

We want to replicate S_2:

$$!^{i \leq n} c[i](); \dots c_1[i](y[i] : T) \dots \textbf{event } e(i, M) \dots$$

Problem: this is not preserved by composition.
In the key exchange, partenered sessions exchange the same messages, but may not have the same replication indices.
Also in the composed system.

Replicating S_2

Consider:

$$S_2 = c(); \dots c_1(y : T) \dots \textbf{event } e(M) \dots$$

We want to replicate S_2:

$$!^{i \leq n} c[i](x : T_{\mathrm{sid}}); \dots c_1[i](y[i] : T) \dots \textbf{event } e(x, M) \dots$$

Partnered sessions can be determined by a session identifier computed from the messages in the protocol.
The protocol that uses the key receives the session identifier in a variable x.

Replicating S_2

Consider:

$$S_2 = c(); P$$
$$P = \dots c_1(y : T) \dots \textbf{event } e(M) \dots$$

We replicate S_2:

$$\begin{aligned} S_{2!} &= \mathrm{AddReplSid}(i \leq n, c', T_{\mathrm{sid}}, S_2) \\ &= !^{i \leq n}\, c'[i](x : T_{\mathrm{sid}}); \\ &\qquad \textbf{if } \textit{that value of } x \textit{ never used before } \textbf{then} \\ &\qquad \mathrm{AddIdxSid}(i \leq n, x : T_{\mathrm{sid}}, P) \end{aligned}$$

$$\mathrm{AddIdxSid}(i \leq n, x : T_{\mathrm{sid}}, P) = \dots c_1[i](y[i] : T) \dots \textbf{event } e(x, M) \dots$$

Never use the same session identifier twice.

Replicating S_2: transfer of security properties

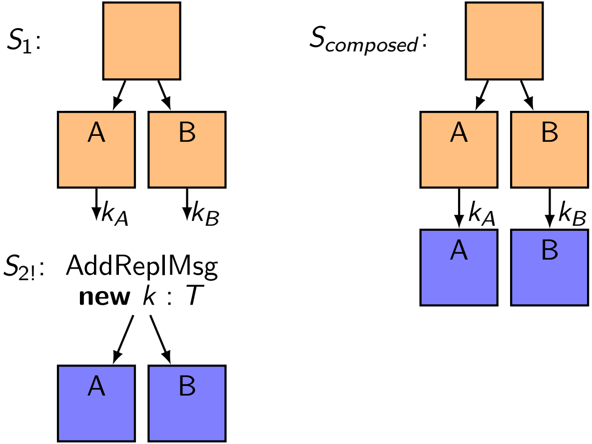
Theorem

Let $Q_! = \mathrm{AddReplSid}(i \leq n, c', T_{\mathsf{sid}}, Q)$
and $Q'_! = \mathrm{AddReplSid}(i \leq n, c', T_{\mathsf{sid}}, Q')$.

1. *If Q and Q' do not contain events and $Q \approx^V Q'$, then $Q_! \approx^V Q'_!$.*
2. *If Q preserves the secrecy of y with public variables V, then so does $Q_!$.*
3. *If Q satisfies* $\mathbf{event}(e_1(y)) \Longrightarrow \mathbf{event}(e_2(y))$ *with public variables V, then $Q_!$ satisfies* $\mathbf{event}(e_1(x, y)) \Longrightarrow \mathbf{event}(e_2(x, y))$ *with public variables V.*

   (Add a variable session identifier at the beginning of each event.)

Main composition theorem



(S_1 may run several sessions of A and B.)

Main composition theorem

Theorem (S_1 and $S_{2!}$)

$$S_1 = C[\textbf{event } e_A(\mathrm{sid}(\widetilde{msg}_A), k_A, i); \textbf{let } k'_A = k_A \textbf{ in } \overline{c_A[i]}\langle M_A\rangle; Q_{1A}, \\ \textbf{event } e_B(\mathrm{sid}(\widetilde{msg}_B), k_B); \overline{c_B[i']}\langle M_B\rangle; Q_{1B}]$$

$$S_2 = c_1(); \textbf{new } k : T; \overline{c_2}\langle\rangle; (Q_{2A} \mid Q_{2B})$$

$$S_{2!} = \mathrm{AddReplSid}(i \leq n, c'_1, T_{\mathrm{sid}}, S_2)$$

where

1. *S_1 and $S_{2!}$ have no common variable, channel, event;*
2. *Other syntactic conditions: see the paper.*

Main composition theorem

C is a context with two holes, with replication $!^{i \leq n}$ above the first hole and $!^{i' \leq n'}$ above the second hole

Theorem

$$S_1 = C[\textbf{event } e_A(\mathrm{sid}(\widetilde{msg}_A), k_A, i); \textbf{let } k'_A = k_A \textbf{ in } \overline{c_A[i]}\langle M_A\rangle; Q_{1A},\\ \textbf{event } e_B(\mathrm{sid}(\widetilde{msg}_B), k_B); \overline{c_B[i']}\langle M_B\rangle; Q_{1B}]$$

$$S_2 = c_1(); \textbf{new } k : T; \overline{c_2}\langle\rangle; (Q_{2A} \mid Q_{2B})$$

$$S_{2!} = \mathrm{AddReplSid}(i \leq n, c'_1, T_{\mathrm{sid}}, S_2)$$

where

1. S_1 *and* $S_{2!}$ *have no common variable, channel, event;*
2. *Other syntactic conditions: see the paper.*

Main composition theorem

Theorem (S_1 and $S_{2!}$)

$$S_1 = C[\textbf{event } e_A(\mathrm{sid}(\widetilde{msg}_A), k_A, i); \textbf{let } k'_A = k_A \textbf{ in } \overline{c_A[i]}\langle M_A\rangle; Q_{1A},$$
$$\textbf{event } e_B(\mathrm{sid}(\widetilde{msg}_B), k_B); \overline{c_B[i']}\langle M_B\rangle; Q_{1B}]$$

$$S_2 = c_1(); \textbf{new } k : T; \overline{c_2}\langle\rangle; (Q_{2A} \mid Q_{2B})$$
$$S_{2!} = \mathrm{AddReplSid}(i \leq n, c'_1, T_{\mathrm{sid}}, S_2)$$

where

1. S_1 *and* $S_{2!}$ *have no common variable, channel, event;*
2. *Other syntactic conditions: see the paper.*

Main composition theorem

sid is a function that takes a sequence of messages and returns a session identifier

Theorem (S_1 and S

$$S_1 = C[\textbf{event } e_A(\mathsf{sid}(\widetilde{msg}_A), k_A, i); \textbf{let } k'_A = k_A \textbf{ in } \overline{c_A[i]}\langle M_A\rangle; Q_{1A},$$
$$\textbf{event } e_B(\mathsf{sid}(\widetilde{msg}_B), k_B); \overline{c_B[i']}\langle M_B\rangle; Q_{1B}]$$

$$S_2 = c_1(); \textbf{new } k : T; \overline{c_2}\langle\rangle; (Q_{2A} \mid Q_{2B})$$
$$S_{2!} = \mathsf{AddReplSid}(i \leq n, c'_1, T_{\mathsf{sid}}, S_2)$$

where

1. *S_1 and $S_{2!}$ have no common variable, channel, event;*
2. *Other syntactic conditions: see the paper.*

Main composition theorem

$\widetilde{msg}_A$ *is a sequence of variables input or output by* C *above the first hole*

Theorem (S_1 and $S_{2!}$)

$$S_1 = C[\mathbf{event}\ e_A(\mathrm{sid}(\widetilde{msg}_A), k_A, i); \mathbf{let}\ k'_A = k_A\ \mathbf{in}\ \overline{c_A[i]}\langle M_A\rangle; Q_{1A},$$
$$\mathbf{event}\ e_B(\mathrm{sid}(\widetilde{msg}_B), k_B); \overline{c_B[i']}\langle M_B\rangle; Q_{1B}]$$

$$S_2 = c_1(); \mathbf{new}\ k : T; \overline{c_2}\langle\rangle; (Q_{2A} \mid Q_{2B})$$
$$S_{2!} = \mathrm{AddReplSid}(i \leq n, c'_1, T_{\mathrm{sid}}, S_2)$$

where

1. S_1 *and* $S_{2!}$ *have no common variable, channel, event;*
2. *Other syntactic conditions: see the paper.*

Main composition theorem

Theorem (S_1 and $S_{2!}$)

$\widetilde{msg}_B$ is a sequence of variables input or output by C above the second hole

$$S_1 = C[\textbf{event } e_A(\mathrm{sid}(\widetilde{msg}_A), k_A, i); \textbf{let } k'_A = k_A \textbf{ in } \overline{c_A[i]}\langle M_A\rangle; Q_{1A}, \quad \textbf{event } e_B(\mathrm{sid}(\widetilde{msg}_B), k_B); \overline{c_B[i']}\langle M_B\rangle; Q_{1B}]$$

$$S_2 = c_1(); \textbf{new } k : T; \overline{c_2}\langle\rangle; (Q_{2A} \mid Q_{2B})$$

$$S_{2!} = \mathrm{AddReplSid}(i \leq n, c'_1, T_{\mathrm{sid}}, S_2)$$

where

1. *S_1 and $S_{2!}$ have no common variable, channel, event;*
2. *Other syntactic conditions: see the paper.*

Main composition theorem

Theorem (S_1 and $S_{2!}$)

$$S_1 = C[\textbf{event } e_A(\mathrm{sid}(\widetilde{msg}_A), k_A, i); \textbf{let } k'_A = k_A \textbf{ in } \overline{c_A[i]}\langle M_A\rangle; Q_{1A},$$
$$\textbf{event } e_B(\mathrm{sid}(\widetilde{msg}_B), k_B); \overline{c_B[i']}\langle M_B\rangle; Q_{1B}]$$

$$S_2 = c_1(); \textbf{new } k : T; \overline{c_2}\langle\rangle; (Q_{2A} \mid Q_{2B})$$
$$S_{2!} = \mathrm{AddReplSid}(i \leq n, c'_1, T_{\mathrm{sid}}, S_2)$$

where

1. *S_1 and $S_{2!}$ have no common variable, channel, event;*
2. *Other syntactic conditions: see the paper.*

Main composition theorem

Theorem ($S_{composed}$)

Let $Q'_{2A} = \mathrm{AddIdxSid}(i \leq n, x : T_{\mathrm{sid}}, Q_{2A})$
and $Q'_{2B} = \mathrm{AddIdxSid}(i' \leq n', x : T_{\mathrm{sid}}, Q_{2B})$.

$$\begin{aligned} S_{composed} = C[&\mathbf{event}\ e_A(\mathrm{sid}(\widetilde{msg}_A), k_A, i); \overline{c_A[i]}\langle M_A\rangle; (Q_{1A} \mid Q'_{2A}\{k_A/k, \mathrm{sid}(\widetilde{msg}_A)/x\}),\\ &\mathbf{event}\ e_B(\mathrm{sid}(\widetilde{msg}_B), k_B); \overline{c_B[i']}\langle M_B\rangle; (Q_{1B} \mid Q'_{2B}\{k_B/k, \mathrm{sid}(\widetilde{msg}_B)/x\})] \end{aligned}$$

Main composition theorem

Theorem (First conclusion)

1. *If S_1 satisfies*
   - *secrecy of k'_A with public variables V ($V \subseteq \mathrm{var}(S_1) \setminus \{k_A, k'_A\}$),*
   - *injective authentication of A to B:*
     $\textbf{inj-event}(e_B(sid, k)) \Longrightarrow \textbf{inj-event}(e_A(sid, k, i))$
     with public variables $V \cup \{k'_A\}$,
   - *single e_A for each session identifier:*
     $\textbf{event}(e_A(sid, k_1, i_1)) \wedge \textbf{event}(e_A(sid, k_2, i_2)) \Longrightarrow i_1 = i_2$
     with public variables $V \cup \{k'_A\}$,

   then we can transfer security properties from $S_{2!}$ to $S_{composed}$.

   Up to renumbering of variable indices,
   $S_{composed}$ with the events of S_1 removed
   is indistinguishable with public variables $V \cup (\mathrm{var}(S_2) \setminus \{k\})$
   from an adversary interacting with $S_{2!}$.

Main composition theorem

Theorem (Second conclusion)

2. We can transfer security properties from S_1 to $S_{composed}$, provided they are proved with public variables k'_A, k_B.

   $S_{composed}$
   is indistinguishable with public variables $\mathrm{var}(S_{composed}) \setminus \{k'_A\}$
   from an adversary interacting with S_1 with access to k'_A, k_B.

Further results in the paper

- Exact security.
- New: Shared hash oracles between the key exchange and the protocol that uses the key.
- New: Variant with non-injective authentication.
- New: Variant for modified ClientHello messages.

Conclusion

- Composition theorems for CryptoVerif
  - computational
  - easy to apply when the protocol pieces are proved secure in CryptoVerif
  - flexible: hash oracles, injective and non-injective authentication
- Application to TLS 1.3
  - important protocol
  - would be out of scope of CryptoVerif without composition because of loops
- Applicable to other protocols