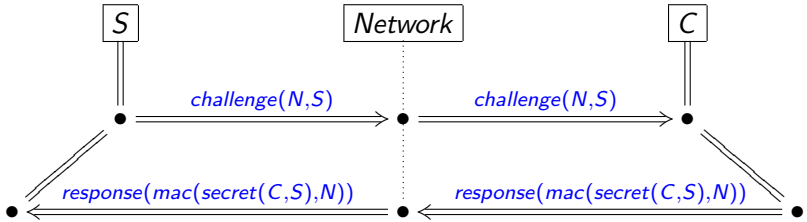
CSF 2021
Vertical Composition and Sound Payload Abstraction for Stateful Protocols

Sébastien Gondron and Sebastian Mödersheim

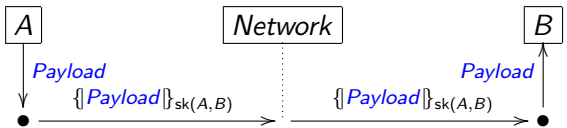
June 23, 2021

Motivations

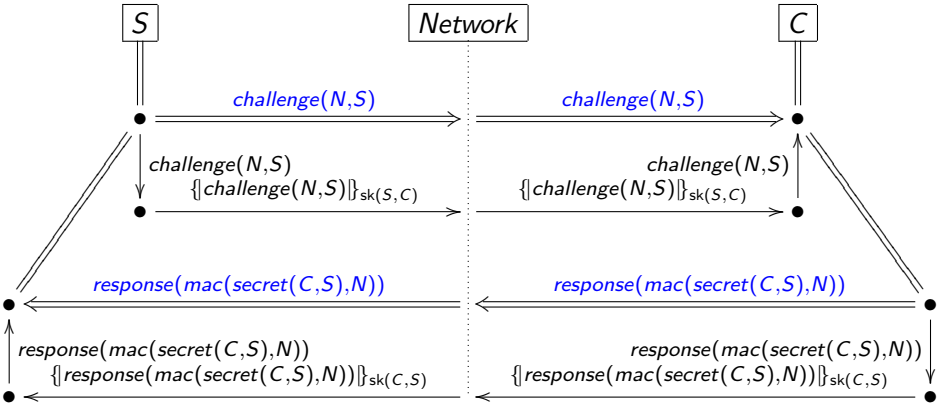
Login application



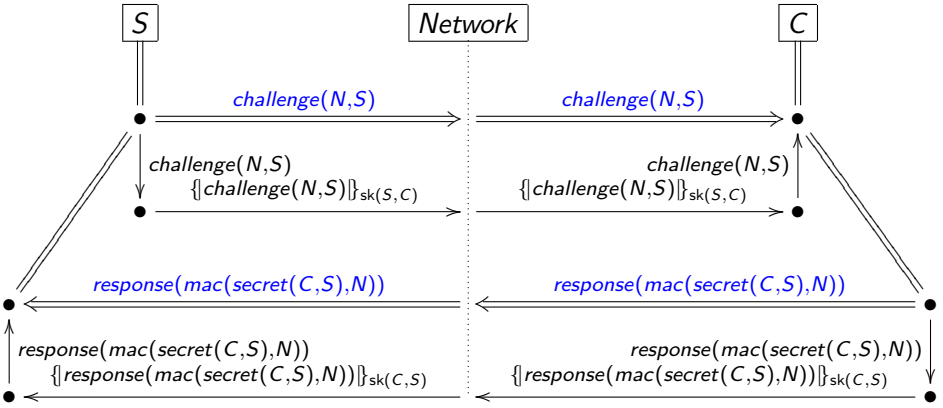
running over a secure channel:



Vertical Composition



Vertical Composition



Is such a composition secure?

- Can the channel be replaced by a different one?
- Can the application be replaced by a different one?

Vertical Composition

Composition of protocols with shared states is hard to get right.

Vertical Composition

Composition of protocols with shared states is hard to get right.

Vertical Composition

Given:

- an application App,
- a channel Ch protocols,
- they are secure in isolation,
- and some conditions (???),

is their vertical composition $\frac{\text{App}}{\text{Ch}}$ also secure?

Can we solve vertical composition of stateful protocols?

What about a parallel composition[1]?

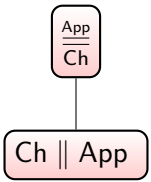
- we consider a channel protocol Ch and an application protocol App,
- they run in parallel and share sets as an interface, called inbox and outbox.

[1] Andreas Victor Hess, Sebastian Alexander Mödersheim, and Achim D. Brucker. "Stateful Protocol Composition". In: *ESORICS 2018*.

Can we solve vertical composition of stateful protocols?

What about a parallel composition[1]?

- we consider a channel protocol Ch and an application protocol App,
- they run in parallel and share sets as an interface, called inbox and outbox.

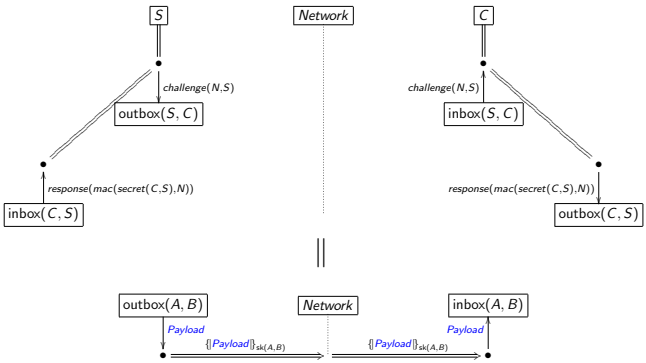


[1] Hess, Mödersheim, and Brucker, "Stateful Protocol Composition".

Can we solve vertical composition of stateful protocols?

What about a parallel composition[1]?

- we consider a channel protocol Ch and an application protocol App,
- they run in parallel and share sets as an interface, called inbox and outbox.

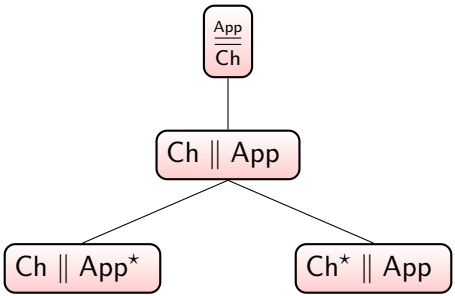


[1] Hess, Mödersheim, and Brucker, "Stateful Protocol Composition".

Can we solve vertical composition of stateful protocols?

What about a parallel composition[1]?

- we consider a channel protocol Ch and an application protocol App,
- they run in parallel and share sets as an interface, called inbox and outbox.



[1] Hess, Mödersheim, and Brucker, "Stateful Protocol Composition".

Channel Idealization

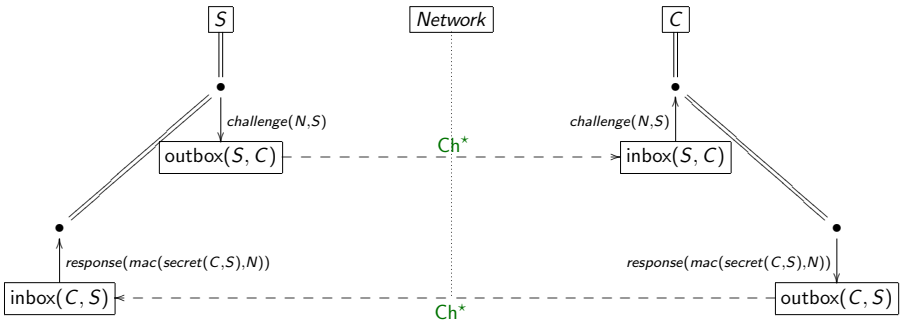
Verifying $\boxed{\mathsf{Ch}^\star \parallel \mathsf{App}}$ means that the application is secure and has no attack as long as:

- the channel does not manipulate the inbox and outbox sets in any other way than described in $\mathsf{Ch}^\star$, and
- the channel does not leak any messages except those explicitly declassified in $\mathsf{Ch}^\star$.

Channel Idealization

Verifying $\boxed{\text{Ch}^\star \parallel \text{App}}$ means that the application is secure and has no attack as long as:

- the channel does not manipulate the inbox and outbox sets in any other way than described in Ch*, and
- the channel does not leak any messages except those explicitly declassified in Ch*.

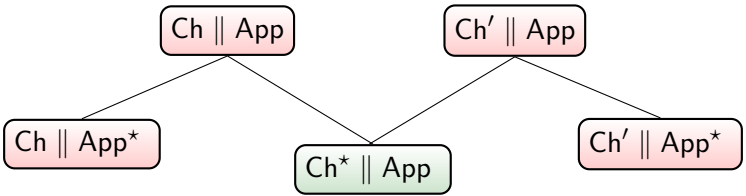


Channel Idealization

Verifying $\mathsf{Ch}^\star \parallel \mathsf{App}$ means that the application is secure and has no attack as long as:

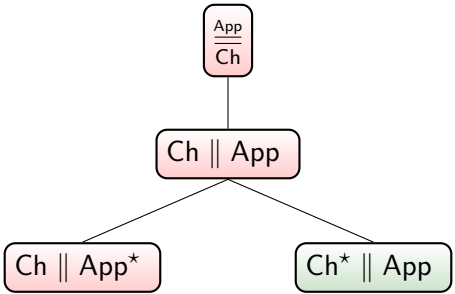
- the channel does not manipulate the inbox and outbox sets in any other way than described in $\mathsf{Ch}^\star$, and
- the channel does not leak any messages except those explicitly declassified in $\mathsf{Ch}^\star$.

First success: any channel Ch' with $\mathsf{Ch}'^\star = \mathsf{Ch}^\star$ works!



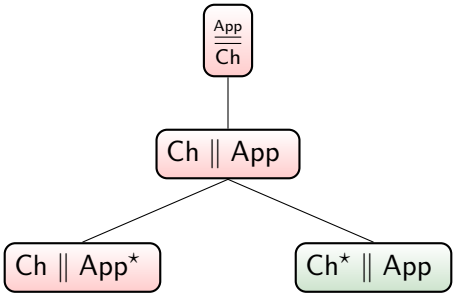
This is not enough!

Let us take stock!



This is not enough!

Let us take stock!



We still need to solve the other problem: $\mathsf{Ch} \parallel \mathsf{App}^\star$.

Abstracting the Payload

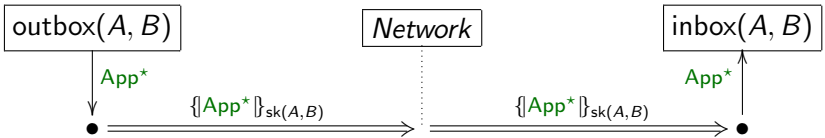
A concrete execution of Ch $\|$ App* has the concrete messages from the application:

- in the outbox and inbox sets, and
- as subterms of the messages that the channel transmits.

Abstracting the Payload

A concrete execution of Ch $\|$ App* has the concrete messages from the application:

- in the outbox and inbox sets, and
- as subterms of the messages that the channel transmits.



Abstracting the Payload

A concrete execution of Ch $\|$ App* has the concrete messages from the application:

- in the outbox and inbox sets, and
- as subterms of the messages that the channel transmits.

But it should be

- simpler: we do not want the complexity of the messages of App, and
- more general: we do not want to verify the channel again when considering a different application

Abstracting the Payload

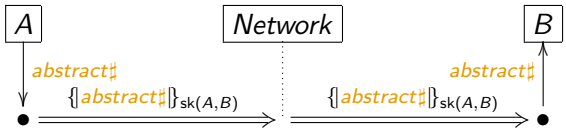
The main idea is to transform Ch into an abstract channel $\mathsf{Ch}^\sharp$:

- we remove outbox and inbox interface, and
- we replace payload variable with abstract constant

Abstracting the Payload

The main idea is to transform Ch into an abstract channel $\mathrm{Ch}^\sharp$:

- we remove outbox and inbox interface, and
- we replace payload variable with abstract constant



abstract$\sharp$ can be

- known to the intruder or not, and
- fresh or reused.

How to prove the security of $\frac{\mathsf{App}}{\mathsf{Ch}}$

To prove the security of $\frac{\mathsf{App}}{\mathsf{Ch}}$, it is enough to prove the security of $\mathsf{Ch}^\sharp$ and of $\mathsf{Ch}^\star \parallel \mathsf{App}$ (given that App and Ch respects a number of syntactic conditions such as disjointness).