02244 Language-Based Security
Security Protocols:
Channels and Composition

Sebastian Mödersheim

April 30, 2018

Roadmap

1. **Channels as Assumptions**
2. **Channels as Goals**
3. **Channel Calculus**
4. **Pseudonymous Channels**
5. **Protocol Composition**

Motivation

Example (NSL)

$$\begin{array}{lcll} A & \rightarrow & B: & \{NA, A\}_{\mathsf{pk}(B)} \\ B & \rightarrow & A: & \{NA, NB, B\}_{\mathsf{pk}(A)} \\ A & \rightarrow & B: & \{NB\}_{\mathsf{pk}(B)} \end{array}$$

- Public-key cryptography is used here to ensure confidential transmission of messages.

Motivation

Example (NSL)

$$\begin{array}{lcll} A & \to\bullet & B: & NA, A \\ B & \to\bullet & A: & NA, NB, B \\ A & \to\bullet & B: & NB \end{array}$$

- Public-key cryptography is used here to ensure confidential transmission of messages.
- Abstraction: use a confidential channel.

Motivation

The Diffie-Hellman assumes an authentic exchange of the half-keys:

$$\begin{array}{lcll} A & \bullet\!\!\rightarrow & B: & \exp(g, X) \\ B & \bullet\!\!\rightarrow & A: & \exp(g, Y) \end{array}$$

- How this exchange is authenticated is not relevant for Diffie-Hellman!
- Many protocols use Diffie-Hellman, e.g. Station2Station, IKE/IKEv2/JFK, Kerberos, TLS, device-pairing. . . .
- Many different ways to authenticate the key-exchange:

  Cryptographically Digital signatures, symmetric/asymmetric encryption, MACs.

  Non-Cryptographically using a trusted third party, meeting face to face, using additional channels (SMS etc.)
- Using an authentic channel abstracts from the realization.

Motivation

- Channels can be both assumptions and goals of a protocol:

Example

$$
\begin{array}{cccll}
A & \bullet\!\!\rightarrow & B & : & \mathrm{exp}(g,X) \\
B & \bullet\!\!\rightarrow & A & : & \mathrm{exp}(g,Y) \\
A & \rightarrow & B & : & \{\!|Payload|\!\}_{\mathrm{exp}(\mathrm{exp}(g,X),Y)} \\
\hline
A & \bullet\!\!\rightarrow\!\!\bullet & B & : & Payload
\end{array}
$$

"Diffie-Hellman creates a secure channel from authentic channels."

- Actually, public-key cryptography could be defined in a broad sense as a mechanism to obtain secure channels from authentic channels.
- Very general way to see Diffie-Hellman.
- Good for system design and verification: reason about small components with a well-defined interface.

Towards a well-defined interface

What exactly does “authentic”, “confidential” and “secure” mean?

- Indeed there is not one right answer, there are many meaningful ways to define these notions.
- We define them here using asymmetric cryptography.
- There are many other ways to define them, e.g. by their behavior.

Realizing Channels with Cryptography

- Assume every agent A has two key pairs:
  - ★ $\langle \mathsf{ck}(A), \mathsf{inv}(\mathsf{ck}(A)) \rangle$ for asymmetric encryption.
  - ★ $\langle \mathsf{ak}(A), \mathsf{inv}(\mathsf{ak}(A)) \rangle$ for digital signatures.
- Assume that every agent knows the public keys $\mathsf{ck}(A)$ and $\mathsf{ak}(A)$ of every other agent A.
- Encode channels by encryption and signing:

Definition

$$\begin{array}{lllllllll}
A & \bullet\!\!\rightarrow & B: & M & \text{for} & A & \rightarrow & B: & \{\mathsf{atag}, B, M\}_{\mathsf{inv}(\mathsf{ak}(A))} \\
A & \rightarrow\!\!\bullet & B: & M & \text{for} & A & \rightarrow & B: & \{\mathsf{ctag}, M\}_{\mathsf{ck}(B)} \\
A & \bullet\!\!\rightarrow\!\!\bullet & B: & M & \text{for} & A & \rightarrow & B: & \{\{\mathsf{stag}, B, M\}_{\mathsf{inv}(\mathsf{ak}(A))}\}_{\mathsf{ck}(B)}
\end{array}$$

- atag, ctag, and stag are tags to distinguish the channel-encodings from other encryptions.

A Cryptographic Realization

Definition

$$
\begin{array}{llllllll}
A & \bullet\!\!\rightarrow & B: & M & \text{for} & A & \rightarrow & B: \{\mathsf{atag}, B, M\}_{\mathsf{inv}(\mathsf{ak}(A))} \\
A & \rightarrow\!\!\bullet & B: & M & \text{for} & A & \rightarrow & B: \{\mathsf{ctag}, M\}_{\mathsf{ck}(B)} \\
A & \bullet\!\!\rightarrow\!\!\bullet & B: & M & \text{for} & A & \rightarrow & B: \{\{\mathsf{stag}, B, M\}_{\mathsf{inv}(\mathsf{ak}(A))}\}_{\mathsf{ck}(B)}
\end{array}
$$

- This ensures the basic properties of channels:
  - ★ Only A can produce messages on the channel $A \bullet\!\!\rightarrow B$.
  - ★ Only B can read messages on the channel $A \rightarrow\!\!\bullet B$.
  - ★ Both restrictions on a secure channel.
- Note that the intruder can still intercept and replay messages!
- This model of channels can be used with all models and tools without extensions!

Authenticated Recipient

Definition

$$
\begin{array}{lllllllll}
A & \bullet\!\!\rightarrow & B: & M & \text{for} & A & \rightarrow & B: & \{\mathsf{atag}, B, M\}_{\mathsf{inv}(\mathsf{ak}(A))} \\
A & \rightarrow\!\!\bullet & B: & M & \text{for} & A & \rightarrow & B: & \{\mathsf{ctag}, M\}_{\mathsf{ck}(B)} \\
A & \bullet\!\!\rightarrow\!\!\bullet & B: & M & \text{for} & A & \rightarrow & B: & \{\{\mathsf{stag}, B, M\}_{\mathsf{inv}(\mathsf{ak}(A))}\}_{\mathsf{ck}(B)}
\end{array}
$$

Why is the recipient B contained in the signatures?

- Including the name B means to authenticate the intended recipient.
- This avoids a classical problem:
  - ★ Recall that $\{\{M\}_{\mathsf{inv}(\mathsf{ak}(A))}\}_{\mathsf{ck}(B)}$ does not give you a secure transmission.

Authenticated Recipient

Definition

$$
\begin{array}{lllllllll}
A & \bullet\!\!\rightarrow & B: & M & \text{for} & A & \rightarrow & B: & \{\mathsf{atag}, B, M\}_{\mathsf{inv}(\mathsf{ak}(A))} \\
A & \rightarrow\!\!\bullet & B: & M & \text{for} & A & \rightarrow & B: & \{\mathsf{ctag}, M\}_{\mathsf{ck}(B)} \\
A & \bullet\!\!\rightarrow\!\!\bullet & B: & M & \text{for} & A & \rightarrow & B: & \{\{\mathsf{stag}, B, M\}_{\mathsf{inv}(\mathsf{ak}(A))}\}_{\mathsf{ck}(B)}
\end{array}
$$

Why is the recipient B contained in the signatures?

- Including the name B means to authenticate the intended recipient.
- This avoids a classical problem:
  - ★ Recall that $\{\{M\}_{\mathsf{inv}(\mathsf{ak}(A))}\}_{\mathsf{ck}(B)}$ does not give you a secure transmission.
  - ★ Think of a dishonest B!
- We want $\bullet\!\!\rightarrow + \rightarrow\!\!\bullet = \bullet\!\!\rightarrow\!\!\bullet$, so the intended recipient should be part of the authentication.
- Also later: relevant when relating to authentic channels as a goal.

Channels as Goals

- Consider payload messages in a protocol; the goal is the authentic and/or secure transmission of the payload:

Example

$$\begin{array}{cccll} A & \bullet\!\!\to & B & : & \mathrm{exp}(g, X) \\ B & \bullet\!\!\to & A & : & \mathrm{exp}(g, Y) \\ A & \to & B & : & \{\!|\mathit{Payload}|\!\}_{\mathrm{exp}(\mathrm{exp}(g,X),Y)} \\ \hline A & \bullet\!\!\to\!\!\bullet & B & : & \mathit{Payload} \end{array}$$

- Authentic transmission $A \bullet\!\!\to B : \mathit{Payload}$: standard non-injective authentication/agreement on *Payload*.
- Confidential transmission $A \to\!\!\bullet B : \mathit{Payload}$: standard secrecy goal (with earliest possible claim by the A).
- Secure transmission: both authentication and secrecy.
- In general, protocols will transmit many payload messages; here we consider only protocols with one payload per session.

Compositionality

Our Aim: A Compositionality Result

- Protocol P_1 realizes channel C as a goal.
- Protocol P_2 assumes channel C.
- Both P_1 and P_2 have been verified individually.
- Some conditions on P_1 and P_2 hold (details later).
- Then we can "plug" P_1 into P_2 to realize the channel C,
- and the resulting protocol $P_2[P_1]$ is correct.

Compositionality: Example

Example (P_1)

$$\begin{array}{rl} A \to s: & A, B, Payload, mac(sk(A,s), A, B, Payload) \\ s \to B: & A, B, Payload, mac(sk(B,s), A, B, Payload) \\ \hline Goal: \ A \bullet\!\!\to B: & Payload \end{array}$$

Example (P_2)

$$\begin{array}{rl} A \bullet\!\!\to B: & \exp(g, X) \\ B \bullet\!\!\to A: & \exp(g, Y) \\ A \to B: & \{\!|ApplicationPayload|\!\}_{\exp(\exp(g,X),Y)} \\ \hline Goal: \ A \bullet\!\!\to\!\!\bullet B: & ApplicationPayload \end{array}$$

Compositionality: Example

Example ($P_2[P_1]$)

$$
\begin{array}{rl}
A \to s: & A, B, \exp(g, X), mac(sk(A, s), A, B, \exp(g, X)) \\
s \to B: & A, B, \exp(g, X), mac(sk(B, s), A, B, \exp(g, X)) \\
B \to s: & B, A, \exp(g, Y), mac(sk(B, s), B, A, \exp(g, Y)) \\
s \to A: & B, A, \exp(g, Y), mac(sk(A, s), B, A, \exp(g, Y)) \\
A \to B: & \{\!|ApplicationPayload|\!\}_{\exp(\exp(g,X),Y)} \\
\hline
Goal: \; A \bullet\!\!\to\!\!\bullet B: & ApplicationPayload
\end{array}
$$

Authentication as an Assumption and as a Goal

Consider an alternative definition of an authentic channel as an assumption:

Definition (*)

$$A \bullet\!\!\rightarrow B : M \text{ for } A \rightarrow B : \{\mathsf{atag}, M\}_{\mathsf{inv}(\mathsf{ak}(A))}$$

This definition of an authentic channel as an assumption is not "compatible" with our definition of a channel as a goal, e.g.:

$$\frac{A \bullet\!\!\rightarrow B : \quad M}{Goal : \; A \bullet\!\!\rightarrow B : \quad M}$$

This protocol has an attack when using definition (*)!

Channel Calculus

Channel Game

- Given a set of channels between agents
- There is no further security relationship between the agents (no public/private keys, passwords,...)
- You can define a protocol to establish new channels between the agents that may use
  - the existing channels
  - create new keys, nonces, and
  - use standard cryptographic primitives
- Question: What new channels between agents can be achieved this way?

Channel Calculus

Channel Game

- Given a set of channels between agents
- There is no further security relationship between the agents (no public/private keys, passwords,...)
- You can define a protocol to establish new channels between the agents that may use
  - the existing channels
  - create new keys, nonces, and
  - use standard cryptographic primitives
- Question: What new channels between agents can be achieved this way?

- We can turn around the direction of any channel, e.g. given $A \bullet\!\!\rightarrow B$, we can achieve $B \rightarrow\!\!\bullet A$.
- From $A \bullet\!\!\rightarrow B$ and $A \rightarrow\!\!\bullet B$ we get $A \bullet\!\!\rightarrow\!\!\bullet B$.
- Nothing else.

Channel Calculus

Channel Game

- Given a set of channels between agents
- There is no further security relationship between the agents (no public/private keys, passwords,...)
- You can define a protocol to establish new channels between the agents that may use
  - the existing channels
  - create new keys, nonces, and
  - use standard cryptographic primitives
- Question: What new channels between agents can be achieved this way?

- We can turn around the direction of any channel, e.g. given $A \bullet\!\!\rightarrow B$, we can achieve $B \rightarrow\!\!\bullet A$.
- From $A \bullet\!\!\rightarrow B$ and $A \rightarrow\!\!\bullet B$ we get $A \bullet\!\!\rightarrow\!\!\bullet B$.
- Nothing else.

Additional question: what if we can assume some parties are honest?

TLS

Consider the TLS handshake (simplified):

$$
\begin{array}{rl}
A \rightarrow B: & A, NA, Sid, PA \\
B \rightarrow A: & NB, Sid, PB, cert_B \\
A \rightarrow B: & cert_A, \{PMS\}_{\mathsf{pk}(B)}, \{hash(NB, B, PMS)\}_{\mathsf{inv}(\mathsf{pk}(A))}, \\
 & \{|hash(prf(PMS, NA, NB), msgs)|\}_{clientK(NA,NB,prf(PMS,NA,NB))} \\
B \rightarrow A: & \{|hash(prf(PMS, NA, NB), msgs)|\}_{serverK(NA,NB,prf(PMS,NA,NB))}
\end{array}
$$

where $msgs$ are all the previous messages of the protocol, $cert_A = \{A, pk(A), \ldots\}_{\mathsf{inv}(\mathsf{pk}(s))}$ is a public key certificate, and $hash$, prf, $clientK$, $serverK$ are hash functions.
Consider also the transmission of a payload with the created keys:

$$
\begin{array}{lrl}
 & A \rightarrow B: & \{|Payload_A|\}_{clientK(NA,NB,prf(PMS,NA,NB))} \\
 & B \rightarrow A: & \{|Payload_B|\}_{serverK(NA,NB,prf(PMS,NA,NB))} \\
\hline
Goal: & A \bullet\!\!\rightarrow\!\!\bullet B: & Payload_A \\
 & B \bullet\!\!\rightarrow\!\!\bullet A: & Payload_B
\end{array}
$$

TLS

$$
\begin{array}{rl}
A \to B : & A, NA, Sid, PA \\
B \to A : & NB, Sid, PB, cert_B \\
A \to B : & cert_A, \{PMS\}_{\mathsf{pk}(B)}, \{hash(NB, B, PMS)\}_{\mathsf{inv}(PK)}, \\
 & \{\!|hash(prf(PMS, NA, NB), msgs)|\!\}_{clientK(NA,NB,prf(PMS,NA,NB))} \\
B \to A : & \{\!|hash(prf(PMS, NA, NB), msgs)|\!\}_{serverK(NA,NB,prf(PMS,NA,NB))}
\end{array}
$$

- While a trustworthy server certificate is (not too un-) realistic, users usually do not have a client certificate!

TLS

$$
\begin{array}{ll}
A \to B: & A, NA, Sid, PA \\
B \to A: & NB, Sid, PB, cert_B \\
A \to B: & PK, \{PMS\}_{\mathrm{pk}(B)}, \{hash(NB, B, PMS)\}_{\mathrm{inv}(PK)}, \\
 & \{\!|hash(prf(PMS, NA, NB), msgs)|\!\}_{clientK(NA,NB,prf(PMS,NA,NB))} \\
B \to A: & \{\!|hash(prf(PMS, NA, NB), msgs)|\!\}_{serverK(NA,NB,prf(PMS,NA,NB))}
\end{array}
$$

- While a trustworthy server certificate is (not too un-) realistic, users usually do not have a client certificate!
- We model now that B has no way to authenticate A's public key:
- A simply generates a fresh public key PK
- What kind of channel do we get from this?

TLS

$$
\begin{array}{ll}
A \to B: & A, NA, Sid, PA \\
B \to A: & NB, Sid, PB, cert_B \\
A \to B: & PK, \{PMS\}_{\mathsf{pk}(B)}, \{hash(NB, B, PMS)\}_{\mathsf{inv}(PK)}, \\
& \{\!|hash(prf(PMS, NA, NB), msgs)|\!\}_{clientK(NA,NB,prf(PMS,NA,NB))} \\
B \to A: & \{\!|hash(prf(PMS, NA, NB), msgs)|\!\}_{serverK(NA,NB,prf(PMS,NA,NB))}
\end{array}
$$

- While a trustworthy server certificate is (not too un-) realistic, users usually do not have a client certificate!
- We model now that B has no way to authenticate A's public key:
- A simply generates a fresh public key PK
- What kind of channel do we get from this?
- The intruder can impersonate the client A towards B.
- But we still get something like a secure channel:
  - ★ B has a secure channel with the owner of PK, i.e., whoever knows $\mathsf{inv}(PK)$!
  - ★ Is is just not proved that this owner is A.

Secure Pseudonymous Channels

- Consider the public key $\mathsf{pk}(A)$ of agent A as a pseudonym of A.
- The link between A and $\mathsf{pk}(A)$ can be achieved by certificates.
- One can create any number of public keys/pseudonyms, which are a priori unlinkable to the creator/owner.
- One can authenticate as the owner of a pseudonym P by signing with $\mathsf{inv}(P)$.
- The pseudonym P cannot be stolen/hijacked because ownership is the knowledge of $\mathsf{inv}(P)$.

Secure Pseudonymous Channels

- Without authentication of the client, we can obtain secure channels with respect to a pseudonym.
- For the example of TLS w/o client authentication, we denote this as follows:

$$
\begin{array}{rrl}
 & \dots & \\
 & A \to B : & \{\!|Payload_A|\!\}_{clientK(NA,NB,prf(PMS,NA,NB))} \\
 & B \to A : & \{\!|Payload_B|\!\}_{serverK(NA,NB,prf(PMS,NA,NB))} \\
\hline
Goal: & [A] \bullet\!\!\to\!\!\bullet\, B : & Payload_A \\
 & B \bullet\!\!\to\!\!\bullet\, [A] : & Payload_B
\end{array}
$$

This is sometimes called sender/receiver invariance: B cannot be sure about A's real identity, but that it is the same entity in several transmissions (namely the owner of a certain key pair).

Good enough . . .

This kind of channel is good enough for many applications such as transmitting credit card data:

[A] $\bullet\!\!\rightarrow\!\!\bullet$ B : Order & Credit Card Data

The intruder can also make orders, or, as a dishonest merchant B receive credit card data, but cannot see the credit card data from an honest A sent to an honest B.
Example: B is a movie-server, we can ensure that the content is delivered to the paying customer (who sent the credit card data):

B $\bullet\!\!\rightarrow\!\!\bullet$ [A]: The-Movie

This does not prevent a dishonest A from sharing the movie with her friends, of course.

Good enough ...

The secure pseudonymous channel is also good enough for a login protocol:

$$\begin{array}{rrl} & [A] \bullet\!\!\rightarrow\!\!\bullet B : & A, password(A, B) \\ & B \bullet\!\!\rightarrow\!\!\bullet [A] : & Payload \\ \hline Goal : & B \bullet\!\!\rightarrow\!\!\bullet A : & Payload \end{array}$$

where $password(A, B)$ is A's password at server B.

- We establish a "classical" secure channel in two steps:
  1. We establish a secure pseudonymous channel $[A] \bullet\!\!\rightarrow\!\!\bullet B$ using TLS without client authentication.
  2. We use this channel to authenticate the client by a shared secret (which possibly has low entropy).
- Further replies (e.g. the data of client A stored on server B) are bound to this authentication.

Other realizations & applications

- Purpose-built keys (PBK): in mobile IP, a device creates a public key when entering a domain, so one can later prove to be the same device when leaving the domain.
- Protocol between a smart card and a card reader:
  - ★ The card is initially not authenticated.
  - ★ But an intruder cannot interfere between card and card reader.
  - ★ Thus $[Card] \bullet\!\!\rightarrow\!\!\bullet\, CardReader$ is an appropriate model of the communication channel!

Cryptographic Model of Secure Pseudonymous Channels

We can extend our cryptographic model to secure pseudonymous channels:

Definition

$$
\begin{array}{cccccccccc}
[A]_P & \bullet\!\!\rightarrow & B: & M & \text{for} & A & \rightarrow & B: & \{\mathsf{atag}, B, M\}_{\mathsf{inv}(P)} \\
A & \rightarrow\!\!\bullet & [B]_P: & M & \text{for} & A & \rightarrow & B: & \{\mathsf{ctag}, M\}_P \\
[A]_P & \bullet\!\!\rightarrow\!\!\bullet & B: & M & \text{for} & A & \rightarrow & B: & \{\{\mathsf{stag}, B, M\}_{\mathsf{inv}(P)}\}_{\mathsf{ck}(B)} \\
A & \bullet\!\!\rightarrow\!\!\bullet & [B]_P: & M & \text{for} & A & \rightarrow & B: & \{\{\mathsf{stag}, P, M\}_{\mathsf{inv}(\mathsf{ak}(A))}\}_P
\end{array}
$$

where we explicitly annotate the pseudonym/public-key P being used. By default, we have a fresh public-key P for every protocol session and agent.

Channels out of thin air?

- TLS is an example for another rule for the channel calculus: From $A {\rightarrow}\!\bullet B$ we can get $[A] \bullet\!{\rightarrow}\!\bullet B$ (but not $A \bullet\!{\rightarrow}\!\bullet B$!)
- In general we can make secure channels with unauthenticated endpoints:
  - ★ From $A \rightarrow B$ we can get $[A] \bullet\!{\rightarrow}\!\bullet [B]$.
- We get this out of thin air, but it is only giving a weak property: sender/receiver invariance (we can be sure that we talk to the same end-point).
- This channel can easily be over-estimated as the following attack shows...

TLS Renegotiation Attack

A TLS Renegotiation Scenario

- When participants have established a TLS connection where the client is not authenticated
- The client issues a command that requires authentication
- Renegotiation: Run a new TLS handshake over the existing channel, producing new keys.
- After successful handshake, switch to the new keys.
- Now the server executes the command.

TLS Renegotiation Attack

A TLS Renegotiation Scenario

- When participants have established a TLS connection where the client is not authenticated
- The client issues a command that requires authentication
- Renegotiation: Run a new TLS handshake over the existing channel, producing new keys.
- After successful handshake, switch to the new keys.
- Now the server executes the command.

In Channel Notation:

- $[A] \bullet\!\!\to\!\!\bullet\ B$: critical command
- $B \bullet\!\!\to\!\!\bullet\ [A]$: require re-negotiate with client auth
- $[A] \bullet\!\!\leftrightarrow\!\!\bullet\ B$: $\boxed{\text{TLS-Handshake with client auth}}$
- B : execute command

Attack

- $[a(i)] \bullet\!\!\rightarrow\!\!\bullet\, b$: critical command
- $b \bullet\!\!\rightarrow\!\!\bullet [a(i)]$: require re-negotiate with client auth
- $a \leftrightarrow [a(i)] \bullet\!\!\leftrightarrow\!\!\bullet D$: $\boxed{\text{TLS-Handshake with client auth}}$
  - ★ The real a wants to start a session with b, the intruder relays every message from a into his channel with b and vice versa.
- b : execute command
  - ★ Even though a has never issued the command.

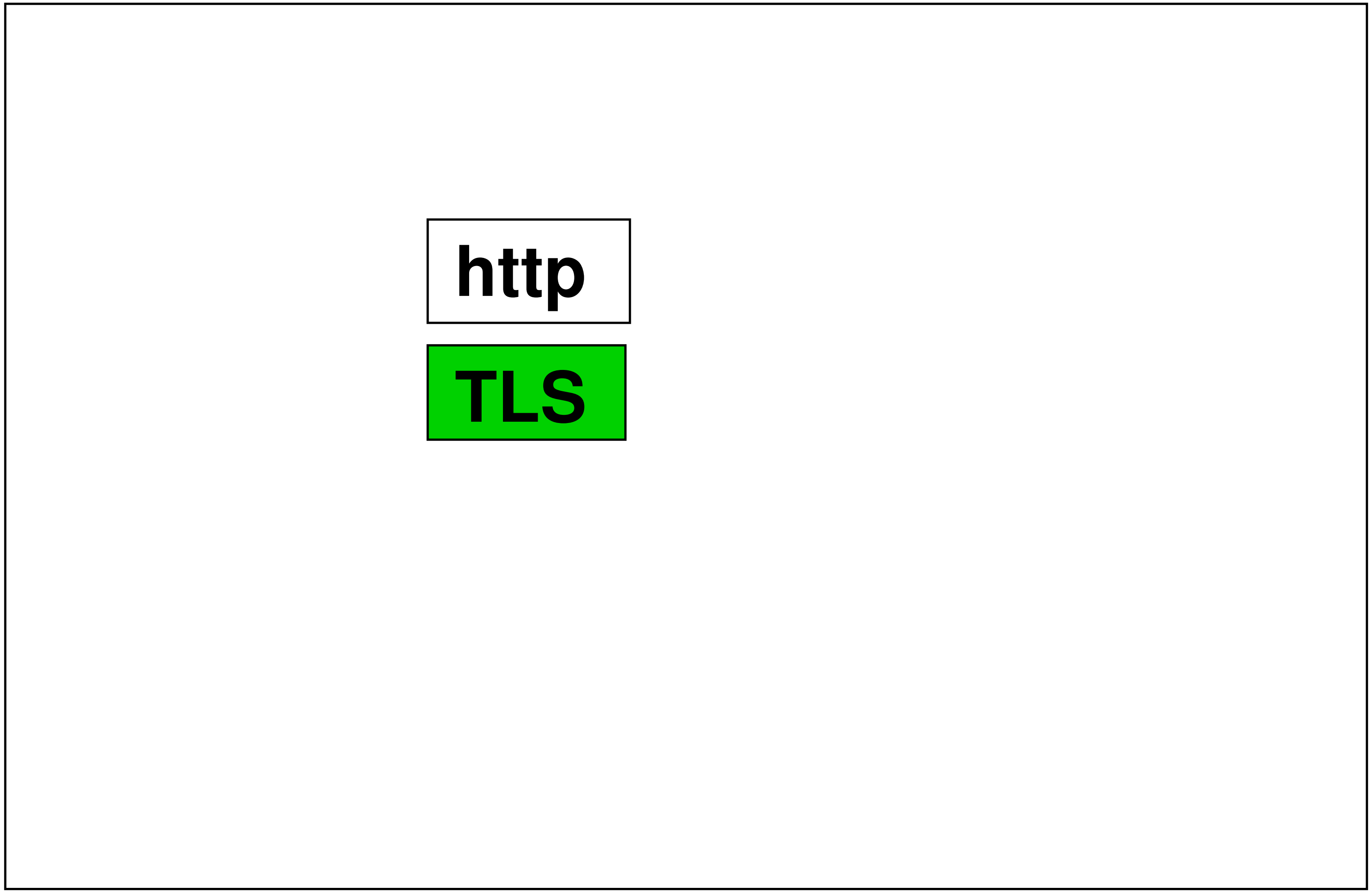
Similar for standard TLS channels (without client authentication) where the intruder is just able to *prefix* a session.

General Problem

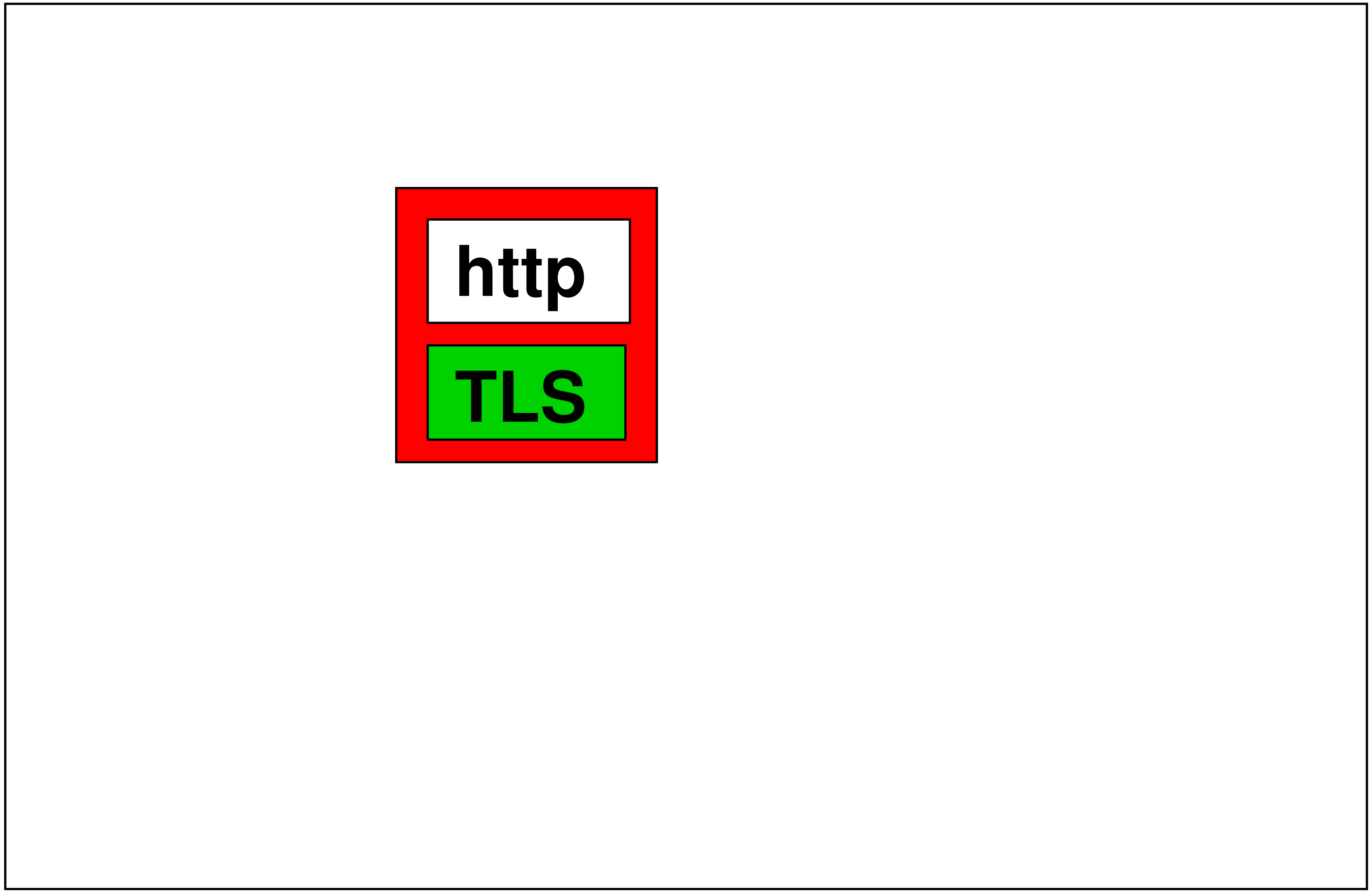
The re-negotiation gives no guarantee on messages exchanged before the re-negotiation.

In fact, this is not really a fault of TLS, but the way it is deployed!

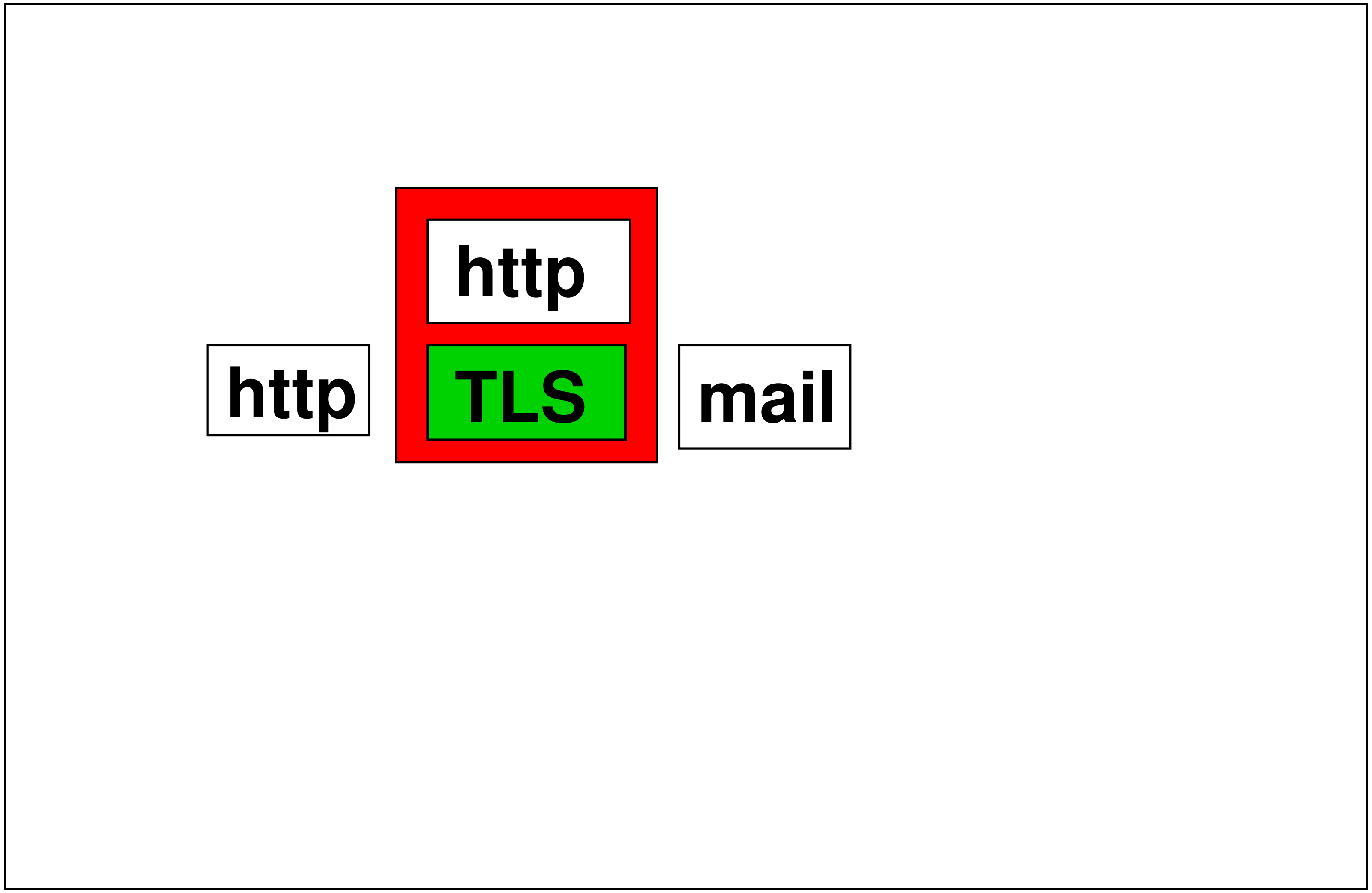
Example of Horizontal and Vertical Composition



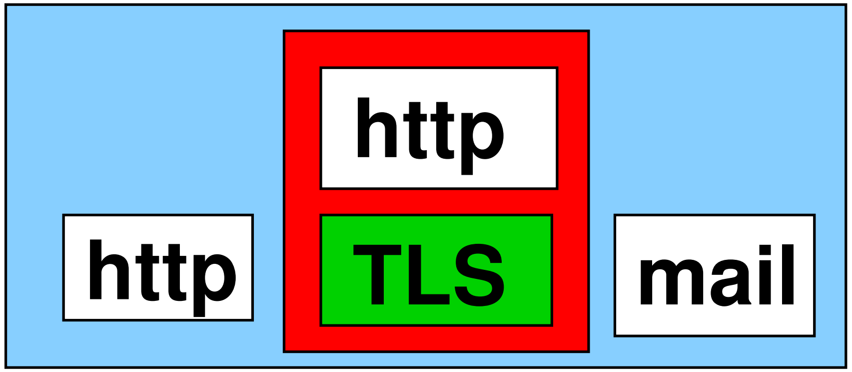
Example of Horizontal and Vertical Composition



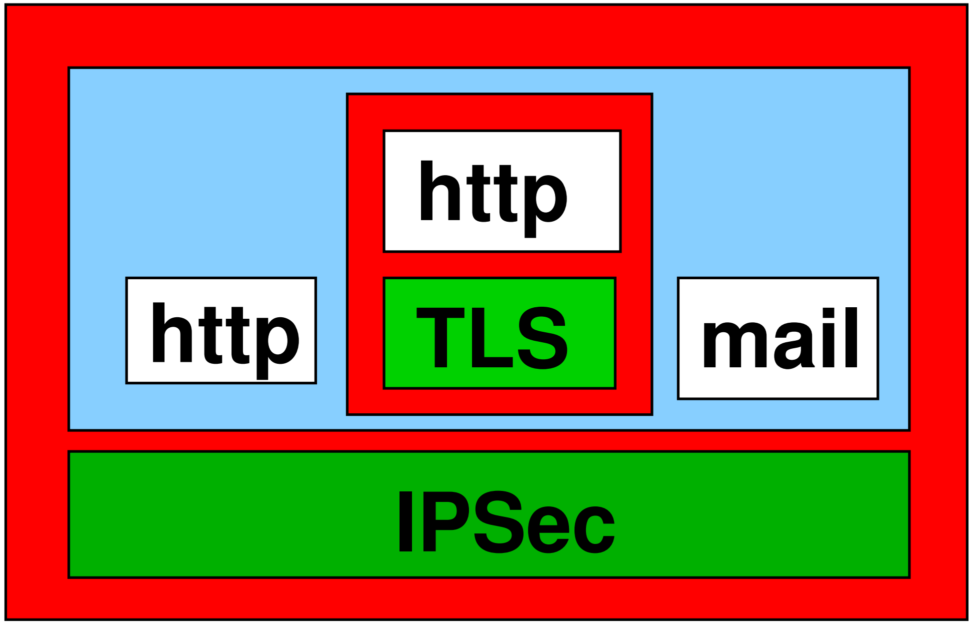
Example of Horizontal and Vertical Composition



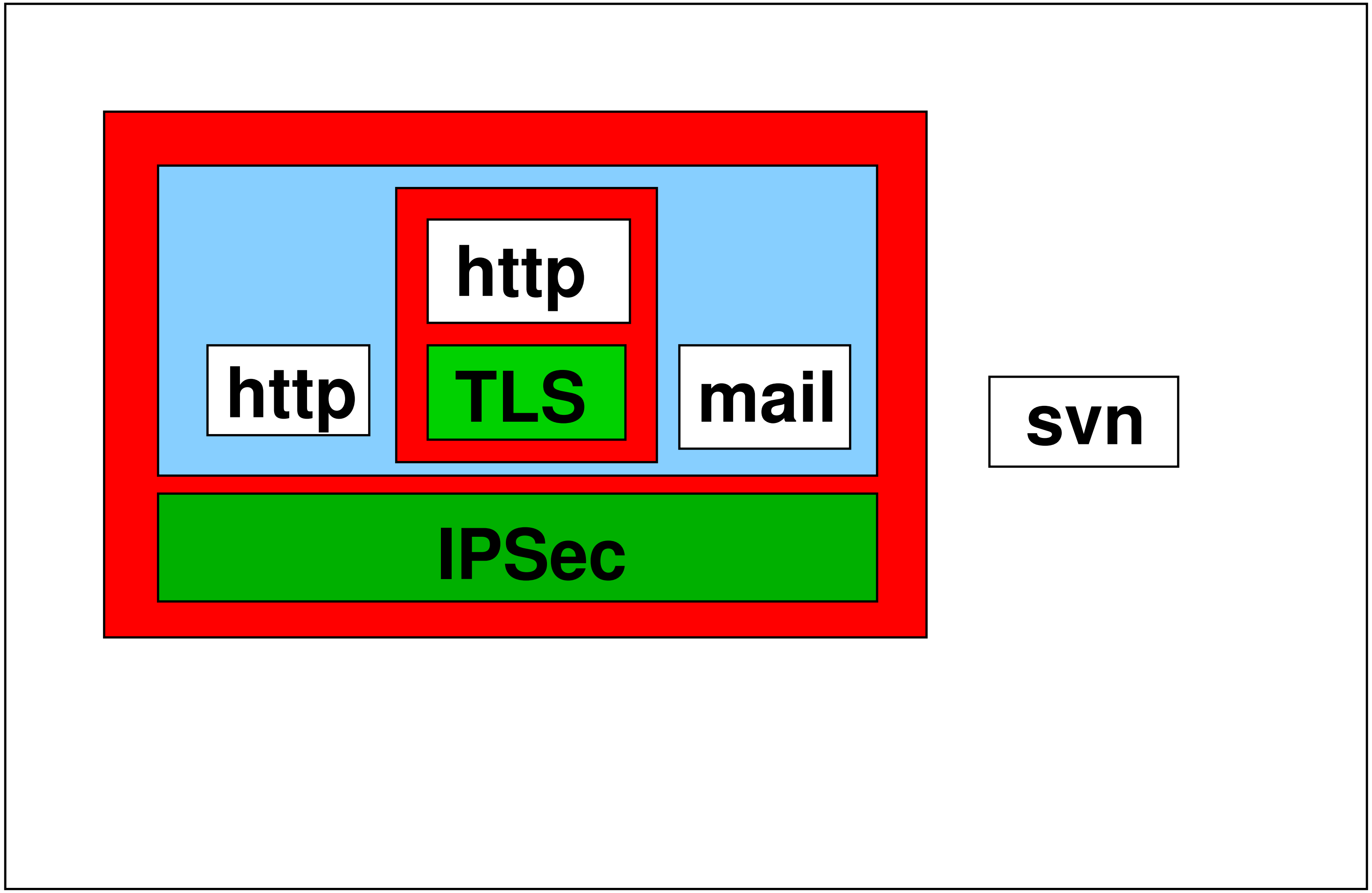
Example of Horizontal and Vertical Composition



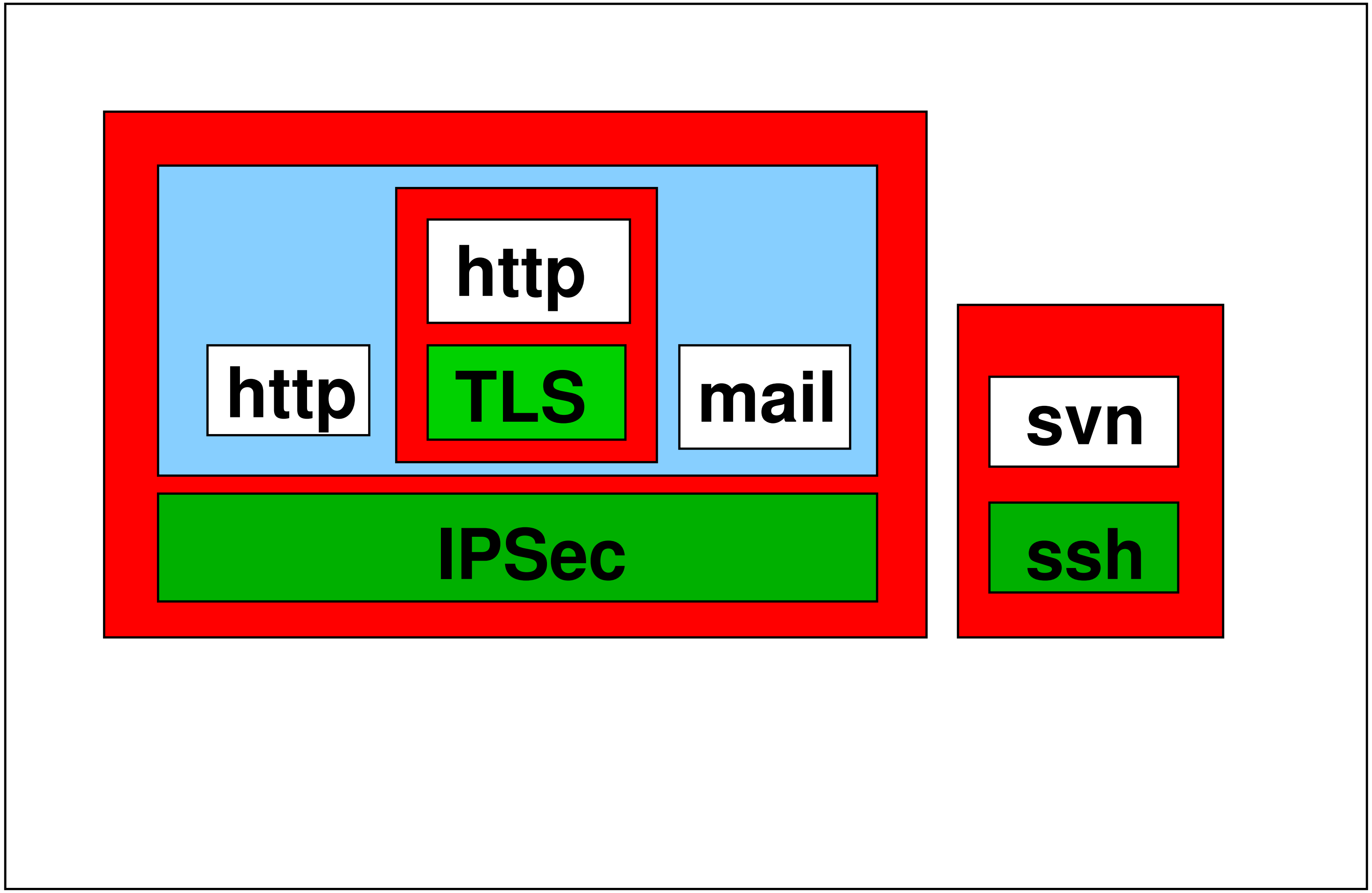
Example of Horizontal and Vertical Composition



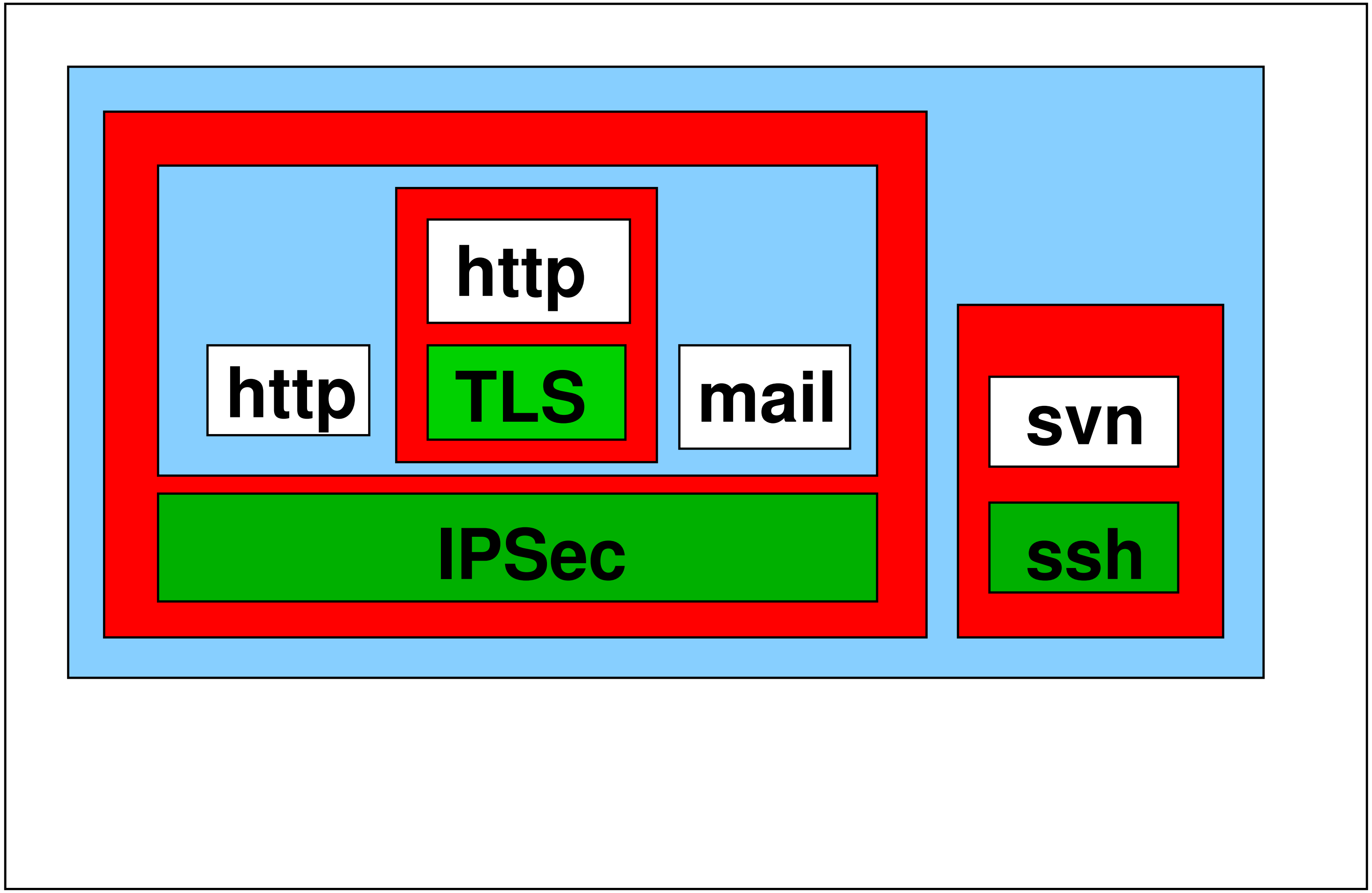
Example of Horizontal and Vertical Composition



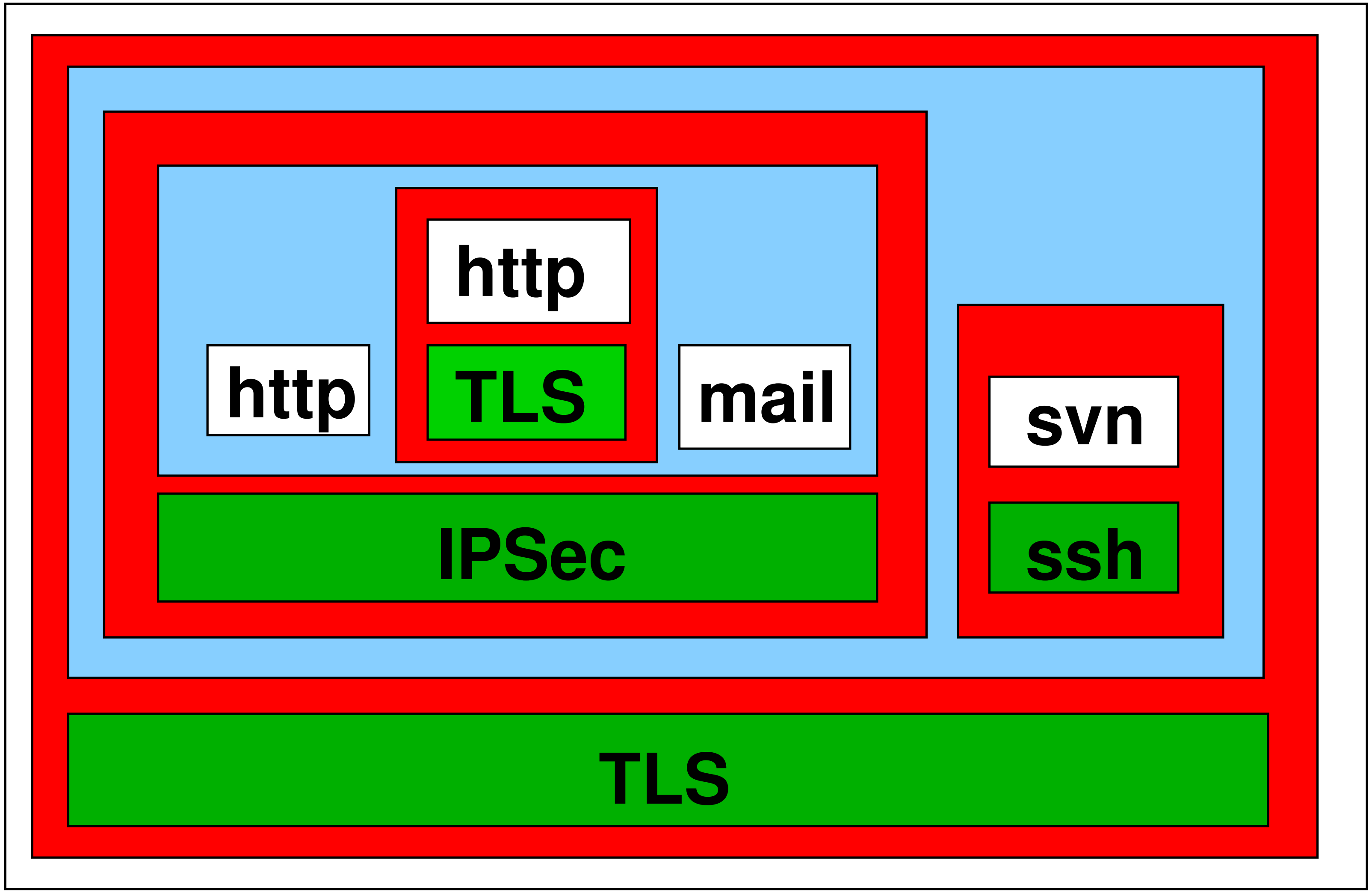
Example of Horizontal and Vertical Composition



Example of Horizontal and Vertical Composition



Example of Horizontal and Vertical Composition



Is this secure?

Compositionality in General

- Design and analyze/verify small systems in isolation.
- Compose them to a secure system using compositionality results
- Ideally, the designer of an application does not need to worry about security—it follows from composition with a security layer
- Can we really achieve this ideal situation?
  - ★ The designer needs to understand what properties a secure channel gives (and which not).
  - ★ Problem: endpoints may still be dishonest and try to attack on the application layer (e.g. injection)
- However, systematic verified solutions tend to reduce the vulnerabilities.