# Security Protocols in CSP

Markus Roggenbach

Feburary 2021

# Introduction to the topics

A *protocol* is a series of steps carried out by two or more entities.
Examples: HTTP, TCP, SMTP

A *security protocol* is a protocol that runs in an untrusted environment and tries to achieve a security goal.
Examples of security goals:

- Authentication (guarantee that one speaks with a specific participant and not someone else),
- Untraceability (adversary can't tell if they have seen you before)

Slide uses material from Ton van Deursen, `https://satoss.uni.lu/members/patrick/1stsxxm.pdf`

# Basic principles

# Cryptography ("kryptos" – "hidden")

*Cryptosystem:*

- encryption(plain text) = cipher text
- decryption(cipher text) = plain text

We write: $\{m\}_k$ for message $m$ encrypted with key $k$

Here: Public Key Cryptosystem (as, e.g, RSA)
Every participant has

- public key $pk$
- secret key $sk$

such that

$$\{\{m\}_{pk}\}_{sk} = m$$
$$\{\{m\}_{sk}\}_{pk} = m$$

# Principles of Security - CIA

*Confidentiality*
Data is said to beconfidential to a set of entities if it is only available to those entities, andnot disclosed to any other outside of the set.

*Integrity*
Assurance that data is not modified or manipulated in anyway from inception.

*Availability*
Assuarance that data is available when needed.

# Security protocols

*Communication protocol:* agreed sequence of actions performed by two or more communicating entities in order to accomplish a purpose, e.g. fault tolerance over a noisy communication medium.

*Writing Convention:*

$$(i)\ A \to B : m$$

in the $i$th step, entity $A$ sends message $m$ destined for entity $B$.

*Security protocol:* communication protocol that provides assurance on security.

# Dolev/Yao Intruder Model

The intruder/adversary can

- block messages, where a message is withheld from recipients;
- replay messages, where an old message could be retransmitted to a recipient of choice;
- spoof messages, where messages are constructed to falsely come from a different source;
- manipulate messages, where multiple messages could be assembled into one or deassembled into fragments of choice; and
- encrypt or decrypt messages, however only where the intruder is in possession of the relevant keys (perfect encryption assumption).

# Objective of Security Protocols

Even in the presence of a Dolev/Yao intruder, a security protocol shall guarantee security goals.

# Needham-Schroeder Protocol for Authentication

# Needham-Schroeder Protocol (N-S protocol)

$$
\begin{aligned}
&(1)\ A \rightarrow B : \{N_A, A\}_{pk_B}\\
&(2)\ B \rightarrow A : \{N_A, N_B\}_{pk_A}\\
&(3)\ A \rightarrow B : \{N_B\}_{pk_B}
\end{aligned}
$$

$A, B$ : entities
$pk_A, pk_B$ public keys of $A$ and $B$, resp.
$N_A$ and $N_B$ are nonces: arbitrary values for single use.

- *fresh* every time they are generated;
- *unpredictable* such that no participant can determine the value of a nonce yet to appear; and
- *not able to reveal the identity* of the participant that produced the nonce.

# Purpose of the Needham-Schroeder Protocol

## Definition (Injective agreement – a special form of authentication)

We say that a protocol guarantees to an initiator $A$ *injective agreement* with a responder $B$ on a set of data items $ds$ if, whenever $A$ (acting as initiator) completes a run of the protocol, apparently with responder $B$, then $B$ has previously been running the protocol, apparently with $A$, and $B$ was acting as responder in his run, and the two agents agreed on the data values corresponding to all the variables in $ds$, and each such run of $A$ corresponds to a *unique* run of $B$.

N-S protocol shall guarantee that

- $A$ in injective agreement with $B$; and
- $B$ in injective agreement with $A$

# Lowe's attack on the N-S Protocol

$$(1.1)\ A \rightarrow I : \{N_A, A\}_{pk_I}$$

$$(2.1)\ I \rightarrow B : \{N_A, A\}_{pk_B}$$

$$(2.2)\ B \rightarrow I : \{N_A, N_B\}_{pk_A}$$

$$(1.2)\ I \rightarrow A : \{N_A, N_B\}_{pk_A}$$

$$(1.3)\ A \rightarrow I : \{N_B\}_{pk_I}$$

$$(2.3)\ I \rightarrow B : \{N_B\}_{pk_B}$$

# Needham-Schroeder-Lowe Protocol (N-S-L)

$$(1)\ A \rightarrow B : \{N_A, A\}_{pk_B}$$
$$(2)\ B \rightarrow A : \{B, N_A, N_B\}_{pk_A}$$
$$(3)\ A \rightarrow B : \{N_B\}_{pk_B}$$

Claim: this little repair in step (2) does the job :-)
Question: how can we know?

# Protocol Modelling in CSP

# Alphabet: all messages of the protocol

Given

- $\mathcal{U}$: set of protocol participants
- $\mathcal{N}$: set of all nonces
- $\mathcal{K}$: set of all encryption keys

Set of all atoms $\mathcal{A}$ :

$$\mathcal{A} ::= \mathcal{U} \mid \mathcal{N} \mid \mathcal{K}$$

Message space $\mathcal{M}$ :

$$\mathcal{M} ::= \mathcal{A} \mid \{\mathcal{M}\}_{\mathcal{K}} \mid \mathcal{M}.\mathcal{M}$$

# Example: Message space of the N-S Protocol

For the N-S Protocol the atoms of the message space are given as follows:

$$\mathcal{U} = \{A, B, I\}$$
$$\mathcal{N} = \{N_A, N_B, N_I\}$$
$$\mathcal{K} = \{pk_A, pk_B, pk_I, sk_A, sk_B, sk_I\}$$

Examples of messages: $A$, $\{N_A.A\}_{pk_B}$, and $A.B$.

The message space is infinite thanks to both, encryption and pairing.

# N-S Protocol in CSP: participants (single run)

$$
\begin{aligned}
A = \Box_{b \in \mathcal{U}, b \neq A}\ & \mathit{send}.A!b!\{N_A, A\}_{pk_b} \rightarrow \\
& \mathit{receive}.A.b?\{N_A, n\}_{pk_A} \rightarrow \\
& \mathit{send}.A.b.\{n\}_{pk_b} \rightarrow \mathit{STOP}
\end{aligned}
$$

$$
\begin{aligned}
B =\ & \mathit{receive}.B?a?\{n, a\}_{pk_B} \rightarrow \\
& \mathit{send}.B.a!\{n, N_B\}_{pk_a} \rightarrow \\
& \mathit{receive}.B.a.\{N_B\}_{pk_B} \rightarrow \mathit{STOP}
\end{aligned}
$$

# Putting things together in a reliable network

$$Network = \Box_{i,j \in \mathcal{U}, m \in \mathcal{M}} \; send?i?j?m \rightarrow receive!i!j!m \rightarrow Network$$

$$System = (|||_{U \in \{A,B\}}) \, [| \, send, receive \, |] \; Network$$

# Modelling the intruder: generates relation

Given a set $S \subseteq \mathcal{M}$ of messages, the generates relation $\vdash \subseteq \mathcal{P}(\mathcal{M}) \times \mathcal{M}$ is the smallest relation closed under:

1. $m \in S$ then $S \vdash m$
2. $S \vdash m$ and $S \vdash k$ then $S \vdash \{m\}_k$
3. $S \vdash \{m\}_k$ and $S \vdash k$ then $S \vdash m$
4. $S \vdash m_1.m_2$ then $S \vdash m_1$ and $S \vdash m_2$
5. $S \vdash m_1$ and $S \vdash m_2$ then $S \vdash m_1.m_2$

# Example: generates relation in the N-S Protocol

Initial set of knowledge

$$S = \{pk_A, pk_B, pk_I, sk_I\}.$$

On step (1.1) of Lowe's attack, the intruder receives the message $\{N_A, A\}_{pk_I}$.
This inreases the intruder's knowledge to a set

$$S' = S \cup \{\{N_A, A\}_{pk_I}\}.$$

With $S'$, the intruder can decrypt the message $\{N_A, A\}_{pk_I}$ :

- we have $\{N_A, A\}_{pk_I} \in S'$ and thus $S' \vdash \{N_A, A\}_{pk_I}$, by Rule 1,
- we have $sk_I \in S'$ and thus $S' \vdash sk_I$, by Rule 1,
- finally, we obtain $S' \vdash \{N_A, A\}$, by Rule 2.
  (reminder: $\{\{m\}_{pk_A}\}_{sk_A} = \mathrm{m}$ holds in the cryptosystem)

# Intruder and insecure network

$$\begin{aligned} \mathit{Intruder}(\mathit{IK}) = & ((\Box_{i,j \in \mathcal{U}, m \in \mathcal{M}}\ \mathit{send}?i?j?m \rightarrow \mathit{Intruder}(\mathit{IK} \cup m)) \\ & \Box \\ & (\Box_{i,j \in \mathcal{U}, \mathit{IK} \vdash m}\ \mathit{receive}!i!j!m \rightarrow \mathit{Intruder}(\mathit{IK})) \end{aligned}$$

$$\mathit{NET} = (|||_{U \in \mathcal{U}, U \neq \mathit{Intruder}} U) \, [| \, \mathit{send}, \mathit{receive} \, |] \, \mathit{Intruder}$$

# Encoding in CSP-M

It is possible to encode N-S, N-S-L, and injective agreement in CSP-M. (It's lot's of work ...)

FDR4

- finds Lowe's attack on N-S (instantly)
- proves N-S-L to do authentication as expected (in about 6 minutes of time)

# Summary of these two weeks

# Summary

- Formal methods allow for modelling and verifying systems
- CSP is a FM for concurrent systems, comes with tools for
  - Simulation
  - Model checking
  - Theorem proving
- Security protocols
  - allow to achieve security goals
  - tricky to design
  - Formal Methods like CSP can help with verification