Generation of Applicative Attacks Scenarios Against Industrial Systems

Maxime Puys Marie-Laure Potet Jean-Louis Roch

VERIMAG, University of Grenoble Alpes / Grenoble-INP, France
Firstname.Name@univ-grenoble-alpes.fr

Dec. 7, 2017

MTV2/MFDL

This work was partially funded by the SACADE (ANR-16-ASTR-0023) and PEPS CRNS ASSI projects.

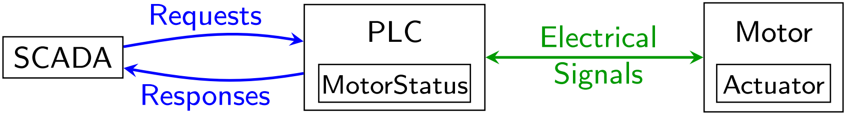
Industrial Systems 1/2

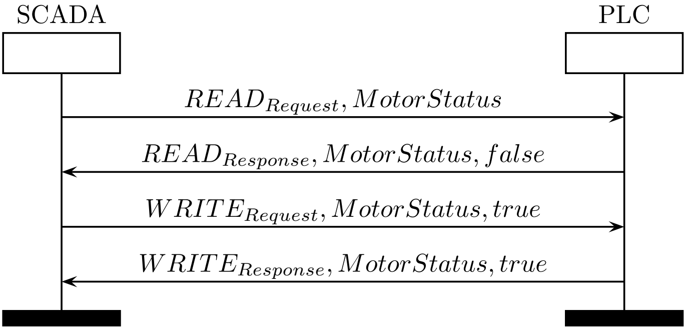
Hot topic

- Since Stuxnet (2009):
  - Complex attack ending up in increasing speed of Iranian centrifuges to damage them.
  - Also attacked the process monitoring to trick operators.
- Protection becoming a priority for government agencies.

Industrial Systems 2/2

- A SCADA controls a PLC which controls a motor.
- Variable *MotorStatus* on the PLC.





Industrial Communication Protocols

MODBUS (1979)

- No security at all.
- Some academic works to secure it (not used in practice):
  - Cryptographic asymmetric signatures [FCMT09]
  - Message Authentication Codes [HEK13]

OPC-UA (2006)

- Security layer: OPC-UA SecureConversation (similar to TLS).
- Three security modes:
  - None, Sign, SignAndEncrypt.

Table of Contents

Table of Contents

Cryptographic Protocols Verification

Mutual Authentication Protocol: Needham-Schroeder

1. A $\rightarrow$ B : $\{A, N_A\}_{KB}$
2. A $\leftarrow$ B : $\{N_A, N_B\}_{KA}$
3. A $\rightarrow$ B : $\{N_B\}_{KB}$

Designed and **proved** in 1978.
Broken in 1995 (17 years after) **with an automated tool**.

Man-In-The-Middle attack

1. A $\rightarrow$ I : $\{A, N_A\}_{KI}$

   1. I $\rightarrow$ B : $\{A, N_A\}_{KB}$
   2. I $\leftarrow$ B : $\{N_A, N_B\}_{KA}$

2. A $\leftarrow$ I : $\{N_A, N_B\}_{KA}$
3. A $\rightarrow$ I : $\{N_B\}_{KI}$

   3. I $\rightarrow$ B : $\{N_B\}_{KB}$

$\Rightarrow$ Need for automation: numerous tools exist (e.g.: Tamarin [MSCB13] or ProVerif [Bla01]).

Related Works on Verification of Industrial Protocols

Ref	Year	Studied Protocols	Analysis
[CRW04]	2004	DNP3, ICCP	Informal
[DNvHC05]	2005	OPC, MMS, IEC 61850 ICCP, EtherNet/IP	Informal
[GP05]	2005	DNP3	Formal (OFMC)
[IEC15]	2006	OPC-UA	Informal
[PY07]	2007	DNP3	Informal
[FCMT09]	2009	MODBUS	Informal
[HEK13]	2013	MODBUS	Informal
[WWSY15]	2015	MODBUS, DNP3, OPC-UA	Informal
[Amo16]	2016	DNP3	Formal (Petri nets)
[PPL16]	2016	OPC-UA	Formal (ProVerif)
[DPP+17]	2017	MODBUS, OPC-UA	Formal (Tamarin)

Table of Contents

Motivations on Studying OPC-UA Security

Probably next standard for industrial communications:

- Recent (2006).
- Designed by a consortium of key stakeholders.

Official specifications: 978 pages:

- Several terms redefined afterward.
- Highly context dependent.

$\Rightarrow$ Unclear on the use of some security features.

Objective: Propose a formal model of the handshake from the specifications.

Modeling Credentials in ProVerif

Login

Takes as parameter the public key of a host.
$\Rightarrow$ Anybody can usurp a login.

Passwd

Takes as parameter the private key of its owner.
Takes as parameter the public key of the server.

Equational Theory Added to ProVerif

verifyCreds(pk(S), Login(pk(C)),Passwd(sk(C), pk(S))) = true.
Allows to verify if a password and a login are matching and if password is the one the server knows (using its public key).

Key Takeaways on OPC-UA Analysis

Two attacks found when security features are removed

Possible reuse of cryptographic signatures (leads to replay attacks).
Possible attacks on passwords in absence of key-wrapping.
Specifications are elusive on purpose for interoperability.

Next steps

Test real implementations.
Application to other industrial protocols.
Model properties such as flow integrity, important for industry.

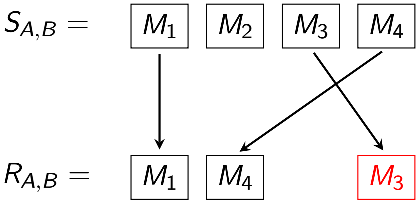
Table of Contents

Contributions

$\Rightarrow$ Main Objective: add properties adapted to industrial systems in automatic verification tools.

Contributions

- Formalization and implementation of properties for industrial systems in Tamarin
- Tested on 2 real industrial protocols and academic works



Properties and relations among them

$$
\begin{array}{ccccc}
(\mathsf{FD} & \wedge & \mathsf{FA}) & \Longleftrightarrow & \mathsf{FI} \\
\Downarrow & & \Downarrow & & \Downarrow \\
(\mathsf{IMD} & \wedge & \mathsf{IMA}) & \Longleftrightarrow & \mathsf{IMI} \\
\Downarrow & & \Downarrow & & \Downarrow \\
(\mathsf{NIMD} & \wedge & \mathsf{NIMA}) & \Longleftrightarrow & \mathsf{NIMI}
\end{array}
$$

Figure : Relationships: $A \Rightarrow B$ if a protocol ensuring A also ensures B.

- Classical network properties (e.g.: TCP sequence numbers)
  - $\Rightarrow$ Never implemented in protocol verification tools
- Can an intruder tamper with these sequence numbers?

Flow Authenticity (FA)

Property

« *All messages are received in the same order they have been sent.* »

$$
\begin{aligned}
&\forall i, j : time, A, B : agent, m, m_2 : msg.(\\
&\quad Received(A, B, m)@i \land Received(A, B, m_2)@j \land i \lessdot j\\
&) \Rightarrow (\exists k, l : time.\\
&\quad Sent(A, B, m)@k \land Sent(A, B, m_2)@l \land k \lessdot l\\
&)
\end{aligned}
$$

Key Takeaways on Flow Integrity

- Formalization of 9 Flow Integrity properties with various security levels

- Implementation in Tamarin
- No modification to Tamarin source code

- Tested on 2 real industrial protocols and academic works (16 models total)
- All models and attacks publicly available

Table of Contents

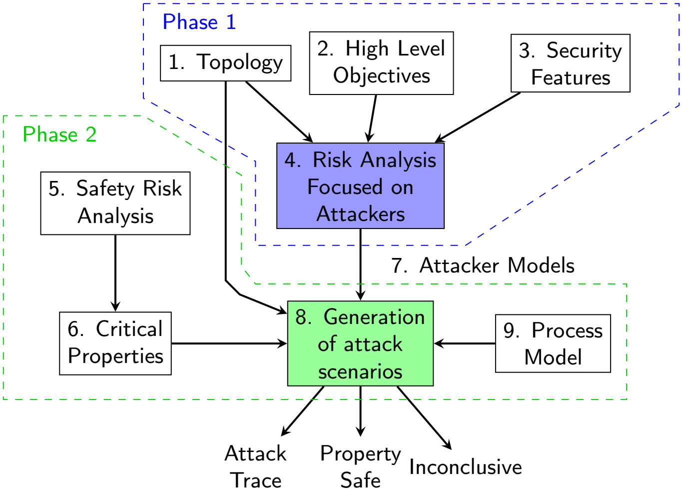
Idea & Contributions

- A^2SPICS: Find applicative attacks on industrial systems:
  - Considering an attacker already in the system;
  - What possible actions on the industrial process.
  - E.g.: Nozzle opens with no bottles under it.
- Implementation using the UPPAAL model-checker;
- Proof-of-concept on a case study.

Generic verification tools vs. Protocol verification tools

- Generic tools: model-checkers, smt-solvers, etc.
- Protocol verification tools: embed attacker logic.
- Trade-off: tool optimized for verification with attackers vs. granularity.

The A²SPICS Approach

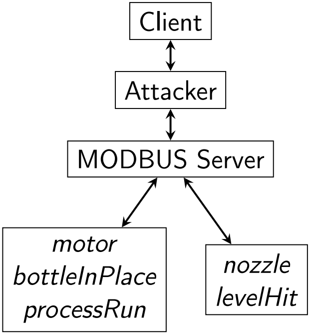


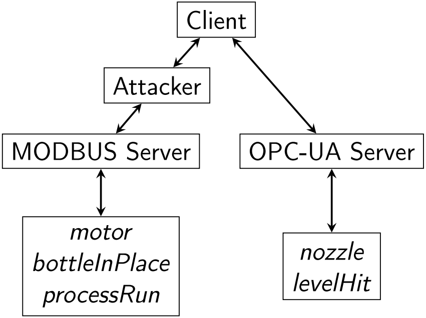
Phase 1 presented at AFADL/MTV2/MFDL 2016 in Besançon.

Topologies

Network topology of the system (expressed in CSP, π-calculus, etc):

- Communication channels between components;
- Position of attackers.



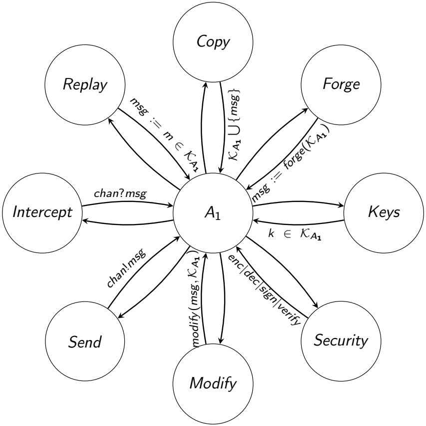


Attackers 1/2

Characterized by:

- Position in the topology:
  - On a channel (Man-In-The-Middle);
  - On a corrupted component (virus, malicious operator, etc).
- Capacities:
  - Possible actions on messages (intercept, modify, replay, etc);
  - Deduction system (deduce new information from knowledge, e.g.: encrypt/decrypt).
- Initial knowledge:
  - Other components;
  - Process behavior;
  - Cryptographic keys, etc.

Attackers 2/2

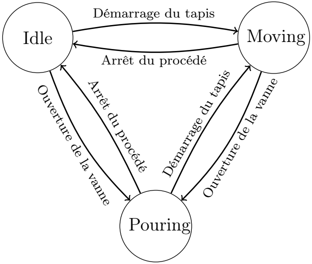


Four attackers:

- A_1 = close to Dolev-Yao;
- Other are subsets of A_1.

Attacker	Modify	Forge	Replay
A_1	✓	✓	✓
A_2	✓	✗	✗
A_3	✗	✓	✗
A_4	✗	✗	✓

Behaviors and Safety Properties



(a) Automaton of the behavior of the process

Current State	Next State	Guard	Actions
Idle	Moving	$processRun = true \wedge bottleInPlace = false$	$motor := true$
Idle	Pouring	$processRun = true \wedge bottleInPlace = true$	$nozzle := true$
Moving	Pouring	$bottleInPlace = true$	$motor := false \wedge nozzle := true$
Pouring	Moving	$levelHit = true$	$motor := true \wedge nozzle := false$
Moving	Idle	$processRun = false$	$motor := false \wedge nozzle := false$
Pouring	Idle	$processRun = false$	$motor := false \wedge nozzle := false$

(b) Transitions Details

Properties: CTL formula:

- Φ_1: At all time and on each path, *nozzle* is never *true* if *bottleInPlace* is *false*).
  $A\Box\neg(nozzle = true \text{ and } bottleInPlace = false)$
- Φ_2: $A\Box\neg(motor = true \text{ and } levelHit = false)$
- Φ_3: $A\Box\neg(nozzle = true \text{ and } motor = true)$

Results on the case study

All attackers on all properties (checked using UPPAAL):

- ✓ = attack found;
- ✗ = no attack found;
- $\mathcal{O}$ = inconclusive (here, out of memory).

Topologies	Properties	A_1	A_2	A_3	A_4
T_1	Φ_1	✓	✓	✓	✗
	Φ_2	✓	✓	✓	✗
	Φ_3	✓	✓	✓	✗
T_2	Φ_1	$\mathcal{O}$	$\mathcal{O}$	✗	✗
	Φ_2	✓	✓	✓	✗
	Φ_3	✓	✓	✓	✗

Related Works

- Survey on assessment of security in industrial system ([CBB+15, PCB13, KPCBH15]).
- Comparison criteria from [KPCBH15, CBB+15]:

Ref.	Type	Focus	Process model	Probabilistic	Automated
[BFM04]	Model	A	No	No	No
[MBFB06]	Model	A	No	Yes (E)	No
[PGR08]	Model	A	No	Yes (E,H)	No
[TML10]	Model	A	No	Yes (H)	Yes
[CAL+11]	Formula	N/A	Yes	Yes (N/C)	Yes
[KBL15]	Model	A	No	Yes (E)	Yes
[RT17]	Model	A,G	Yes	No	Yes
A^2SPICS	Model	A,G	Yes	No	Yes

- Rely on Cl-Atse (protocol verification tool)
  - Dolev-Yao intruder ⇒ less precise control on attacker capacities
- A^2SPICS aims at modeling attackers resulting on risk analysis

Limitations

- Time and state of the process are discretized (e.g.: the bottle is either empty or full).
- Number of actions per attack is bounded (configurable, classical limitation of model-checking).
- Model only considers logical state of variables:
  - real state (i.e.: if a bottle is physically present or not);
  - logical state (i.e.: if the variable *bottleInPlace* is set to *true*);
  - properties are verified on logical state;
  - if a captor is written, a decorrelation is introduced.

    $\Rightarrow$ Can lead to missed attacks (e.g.: Φ_1).

Perspectives

- Study how to address former model limitations.
- Assess example from [RT17] for a better comparison.
- Allow collusions between intruders.
- Consider resilience properties.

- Tentative of automation with ProVerif and Tamarin.
  - Apply formalisms of [RT17].

- Combine protocol and safety properties verification.

Conclusion

Thanks for your attention!

Maxime Puys
Maxime.Puys@univ-grenoble-alpes.fr

Table of Contents

Risk Analyzes 1/2

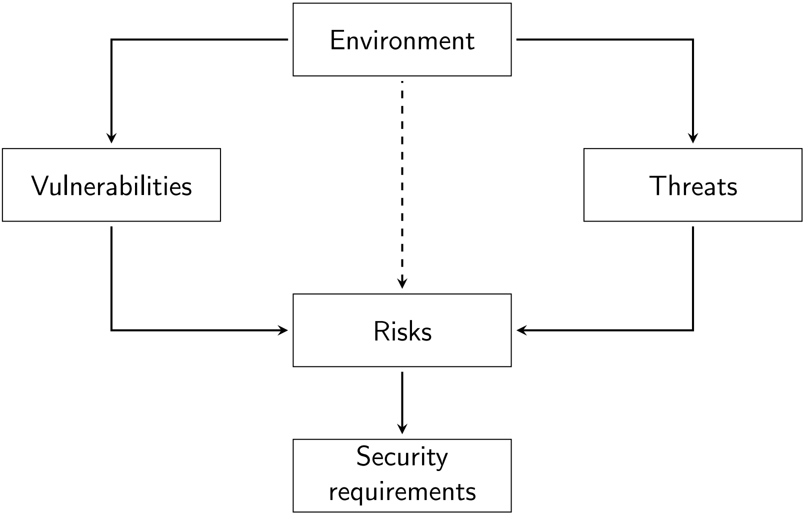


Figure : Functioning of EBIOS

Risk Analyzes 2/2

Nom	Year	Domain	Source
FMEA	196X	Safety	Industry
HAZOP	1977	Safety	Industry
IEC 61508	2010	Safety	Industry
CRAMM	1985	Security	Government (CCTA, UK)
EBIOS	1995	Security	Government (ANSSI)
MEHARI	1998	Security	Industry (CLUSIF)
OCTAVE	1999	Security	Academia (CMU)
FPIS199-220, SP800-53	2002(?)	Security	Government (NIST)
MORDA	?	Security	Government (NSA)
SQUARE	2005	Security	Academia (CMU)
ISO 2700X	2007	Security	Industry

Table : Non-Exhaustive List of Risk Analysis Methods

⇒ See : The SEMA referential framework: Avoiding ambiguities in the terms "security" and "safety", Piètre-Cambacédès and Chaudet, International Journal of Critical Infrastructure Protection, 2010.

Differences between Industrial and Business IT

- Really long-term installations, hard to patch, lot of legacy hosts.
- Security objectives are different from traditional systems:
  - Availability, integrity, authentication and non-repudiation.
- Messages are READ/WRITE commands to PLCs.
  - Sometimes SUBSCRIPTIONS, RPCs or grouped commands.
  - Industrial protocols: MODBUS, OPC-UA.
- Attack examples: change the value of a WRITE request to change a temperature, change a READ response to mislead operators.

Disambiguation

Security concepts

- Safety = Protection against identified/natural difficulties.
  - Historic industrial concern.
- Cybersecurity = Protection against malicious adversaries.
  - Often called Security.

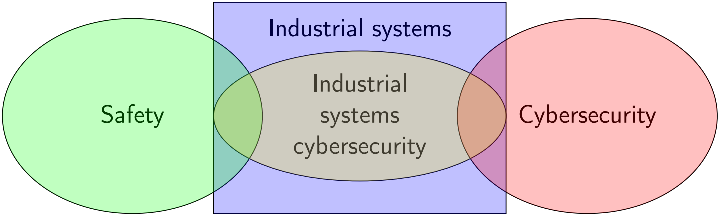


Figure : Relations among security concepts

- Ludovic Pietre-Cambacedes' thesis: On the relationships between safety and security, Telecom ParisTech and EDF, 2010.

Safety and Security

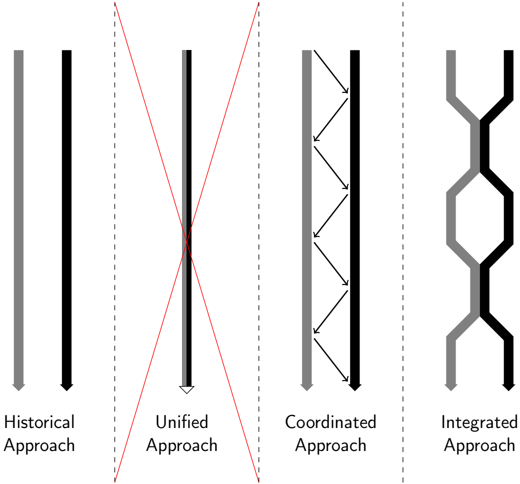


Figure : How to link safety and security [PC10]

Purdue Model

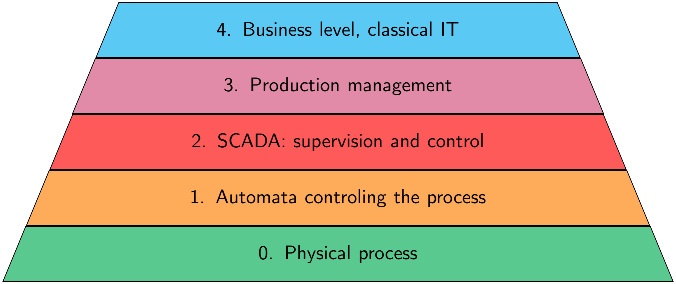


Figure : Purdue model [Wil91]

Motivations on Studying OPC-UA Security

Official specifications: 978 pages.

Several terms redefined afterward:

For this reason, the OpenSecureChannel Service **is not the same as the one specified in the Part 4.** – Part 6, Release 1.02, Page 41.

Highly context dependent:

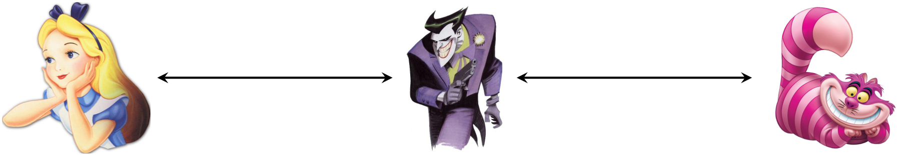
Some SecurityProtocols do not encrypt the entire Message with an asymmetric key. **Instead, they use the AsymmetricKeyWrapAlgorithm to encrypt a symmetric key** [...]. – Part 6, Release 1.02, Page 27.

The AsymmetricKeyWrapAlgorithm element of the SecurityPolicy structure defined in Table 22 **is not used by UASC implementations.** – Part 6, Release 1.02, Page 37.

Cryptographic Protocols Verification 2/2

Numerous tools exist (e.g.: Tamarin [MSCB13] or ProVerif [Bla01]):

- They automatically verify the protocol in presence of an intruder.



Dolev-Yao Intruder [DY81]

Controls the network.

Cryptography is supposed perfect.

Intruder is able to deduce possible messages from his knowledge:

- E.g.: If he has an encrypted message and the key, he can deduce the plaintext.

Open Secure Channel Sub-Protocol

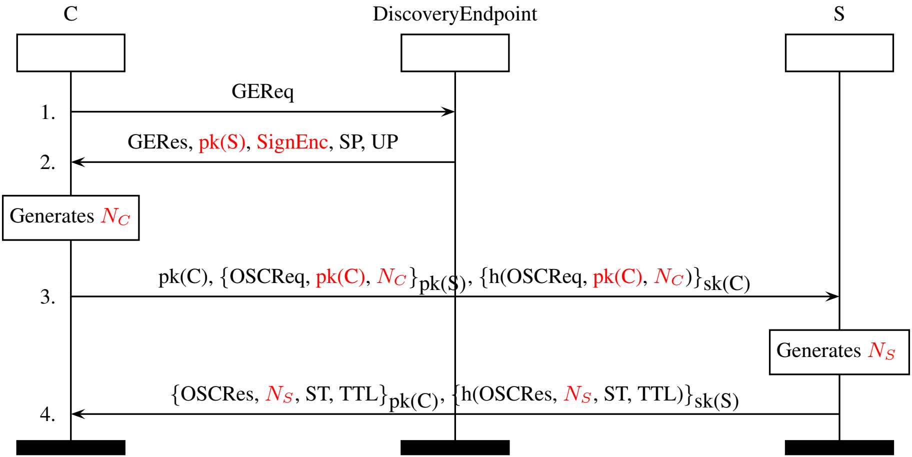


Figure : OPC-UA OpenSecureChannel

Nonce: random value for freshness or challenges/responses.

Modeling Hypotheses

- Normally, several responses to a GetEnpointRequest.
  - We suppose that the client receives and accepts a single one.
  - We tried all possible combinations.
- Client's and server's certificates are modeled by their public keys.
  - Common practice since other fields are out of the scope of tools.
- The intruder can be legitimate clients or servers (e.g.: corrupted devices, malicious operators, etc).
  - Increasing the power of the intruder.
- Objectives:
  - Secrecy of the generated keys (K_{CS}, K_{SC}) from N_C and N_S.
  - Authentication on exchanged nonces N_C and N_S.

Results

OPC-UA Security mode	Objectives			
	Sec K_{CS}	Sec K_{SC}	Auth N_S	Auth N_C
None	UNSAFE	UNSAFE	UNSAFE	UNSAFE
Sign	UNSAFE	UNSAFE	UNSAFE	UNSAFE
SignEnc	SAFE	SAFE	UNSAFE	UNSAFE

Table : Results for *OpenSecureChannel* sub-protocol

Attack on Authentication on N_C in SignAndEncrypt

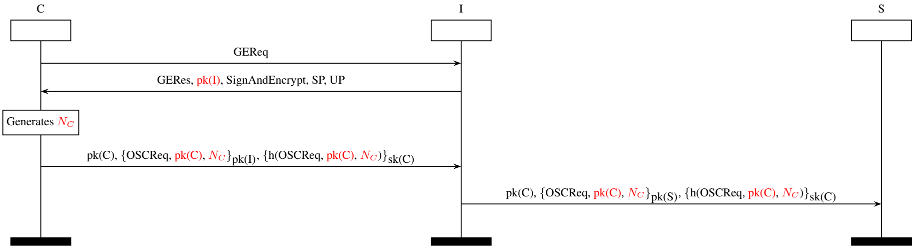


Figure : Attack on OPC-UA OpenSecureChannel

A message can be replayed because receiver is not mentioned in signature.

Counter-measure

Before counter-measure

$\{m\}_{pk(Rcv)}, \{h(m)\}_{sk(Snd)}$

Using key-wrapping and receivers identity

$\{m\}_{pk(Rcv)}, \{h(m), \mathbf{Rcv}\}_{sk(Snd)}$
Very similar counter-measure than in Needham-Schroeder's fixed version.

OPC-UA Security mode	Objectives			
	Sec K_{CS}	Sec K_{SC}	Auth N_S	Auth N_C
None	UNSAFE	UNSAFE	UNSAFE	UNSAFE
Sign	SAFE	SAFE	SAFE	SAFE
SignEnc	SAFE	SAFE	SAFE	SAFE

Table : Results for fixed *OpenSecureChannel* sub-protocol

Table of Contents

Create Session Sub-Protocol

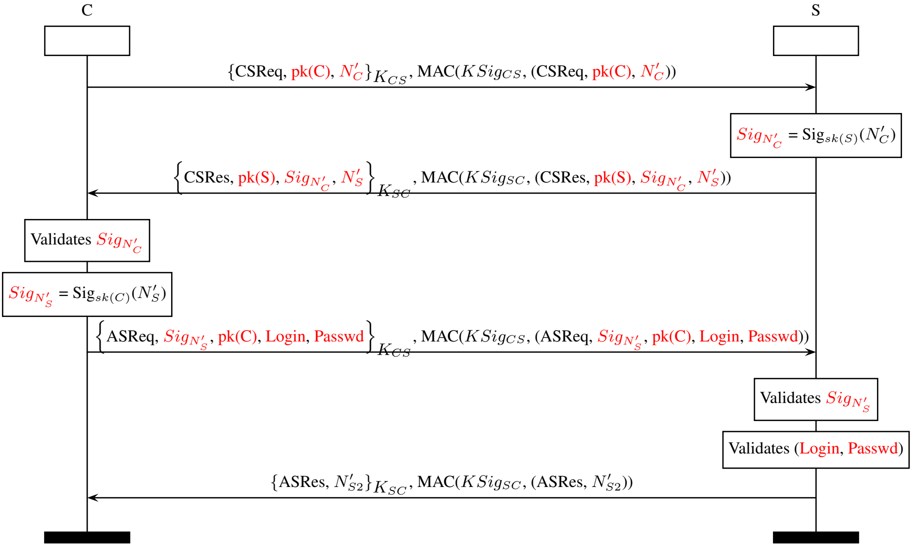


Figure : OPC-UA CreateSession

Results (CreateSession independently)

Without key-wrapping in Sign mode during OpenSecureChannel protocol, intruder obtains symmetric keys.

Results for OPC-UA CreateSession sub-protocol without key-wrapping

OPC-UA Security mode	Objectives			
	Sec *Passwd*	Auth *Passwd*	Auth Sig_{N_s}	Auth Sig_{N_c}
None	UNSAFE	UNSAFE	UNSAFE	UNSAFE
Sign	UNSAFE	UNSAFE	SAFE	SAFE
SignEnc	SAFE	SAFE	SAFE	SAFE

Results for OPC-UA CreateSession sub-protocol with key-wrapping and password encryption

OPC-UA Security mode	Objectives			
	Sec *Passwd*	Auth *Passwd*	Auth Sig_{N_s}	Auth Sig_{N_c}
None	UNSAFE	UNSAFE	UNSAFE	UNSAFE
Sign	SAFE	SAFE	SAFE	SAFE
SignEnc	SAFE	SAFE	SAFE	SAFE

Non-Injective Message Authenticity (NIMA)

Property

« **All messages received have been sent.** »
A protocol ensures Non-Injective Message Authenticity (NIMA) between sender A and receiver B if $\textbf{set}(R_{A,B}) \subseteq \textbf{set}(S_{A,B})$.

$$S_{A,B} = \boxed{M_1}\ \boxed{M_2}\ \boxed{M_3}\ \boxed{M_4}$$

$$R_{A,B} =$$

Non-Injective Message Authenticity (NIMA)

Property

« **All messages received have been sent.** »
A protocol ensures Non-Injective Message Authenticity (NIMA) between sender A and receiver B if $\textbf{set}(R_{A,B}) \subseteq \textbf{set}(S_{A,B})$.

$$S_{A,B} = \boxed{M_1}\ \boxed{M_2}\ \boxed{M_3}\ \boxed{M_4}$$

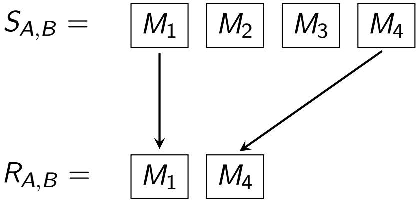
$$R_{A,B} = \boxed{M_1}$$

Non-Injective Message Authenticity (NIMA)

Property

« **All messages received have been sent.** »

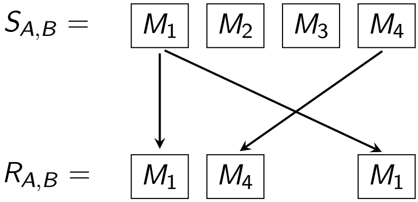
A protocol ensures Non-Injective Message Authenticity (NIMA) between sender A and receiver B if $\textbf{set}(R_{A,B}) \subseteq \textbf{set}(S_{A,B})$.



Non-Injective Message Authenticity (NIMA)

Property

« **All messages received have been sent.** »
A protocol ensures Non-Injective Message Authenticity (NIMA) between sender A and receiver B if $\mathbf{set}(R_{A,B}) \subseteq \mathbf{set}(S_{A,B})$.

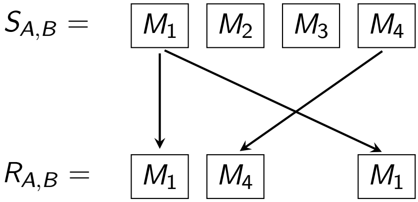


Non-Injective Message Authenticity (NIMA)

Property

« **All messages received have been sent.** »

A protocol ensures Non-Injective Message Authenticity (NIMA) between sender A and receiver B if $\mathbf{set}(R_{A,B}) \subseteq \mathbf{set}(S_{A,B})$.



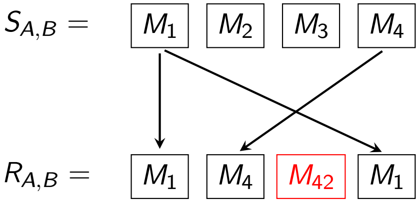
✓ NIMA verified

Non-Injective Message Authenticity (NIMA)

Property

« **All messages received have been sent.** »

A protocol ensures Non-Injective Message Authenticity (NIMA) between sender A and receiver B if $\textbf{set}(R_{A,B}) \subseteq \textbf{set}(S_{A,B})$.

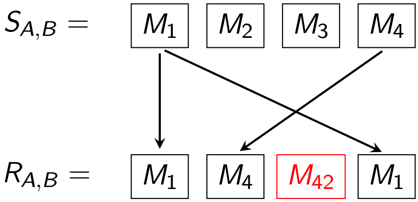


Non-Injective Message Authenticity (NIMA)

Property

« **All messages received have been sent.** »

A protocol ensures Non-Injective Message Authenticity (NIMA) between sender A and receiver B if $\textbf{set}(R_{A,B}) \subseteq \textbf{set}(S_{A,B})$.



✗ NIMA not verified

Injective Message Authenticity (IMA)

Property

« **All messages received have been sent only once.** »
A protocol ensures Injective Message Authenticity (IMA) between sender A and receiver B if $\textbf{multiset}(R_{A,B}) \subseteq \textbf{multiset}(S_{A,B})$.

$$S_{A,B} = \boxed{M_1}\ \boxed{M_2}\ \boxed{M_3}\ \boxed{M_4}$$

$$R_{A,B} =$$

Injective Message Authenticity (IMA)

Property

« **All messages received have been sent only once.** »

*A protocol ensures Injective Message Authenticity (IMA) between sender A and receiver B if **multiset**($R_{A,B}$) $\subseteq$ **multiset**($S_{A,B}$).*

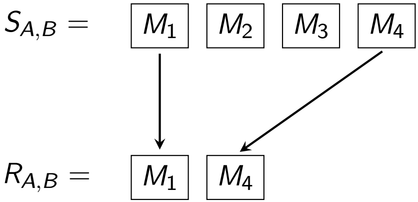
$$S_{A,B} = \boxed{M_1}\ \boxed{M_2}\ \boxed{M_3}\ \boxed{M_4}$$

$$R_{A,B} = \boxed{M_1}$$

Injective Message Authenticity (IMA)

Property

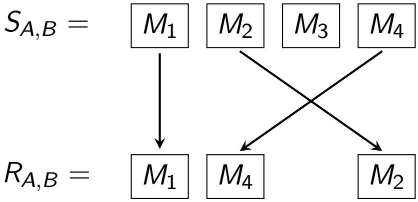
« **All messages received have been sent only once.** »
A protocol ensures Injective Message Authenticity (IMA) between sender A and receiver B if $\textbf{\textit{multiset}}(R_{A,B}) \subseteq \textbf{\textit{multiset}}(S_{A,B})$.



Injective Message Authenticity (IMA)

Property

« **All messages received have been sent only once.** »
A protocol ensures Injective Message Authenticity (IMA) between sender A and receiver B if $\boldsymbol{multiset}(R_{A,B}) \subseteq \boldsymbol{multiset}(S_{A,B})$.

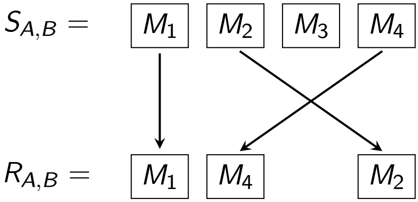


Injective Message Authenticity (IMA)

Property

« **All messages received have been sent only once.** »

A protocol ensures Injective Message Authenticity (IMA) between sender A and receiver B if $\textbf{multiset}(R_{A,B}) \subseteq \textbf{multiset}(S_{A,B})$.

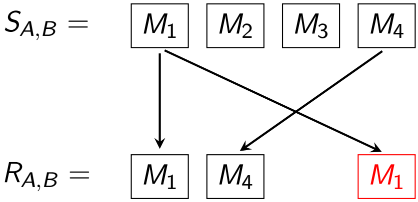


✓ IMA verified

Injective Message Authenticity (IMA)

Property

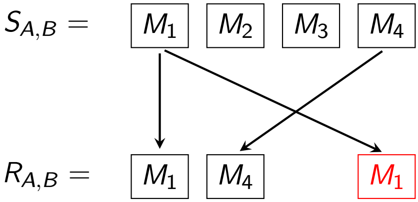
« **All messages received have been sent only once.** »
A protocol ensures Injective Message Authenticity (IMA) between sender A and receiver B if $\textbf{\textit{multiset}}(R_{A,B}) \subseteq \textbf{\textit{multiset}}(S_{A,B})$.



Injective Message Authenticity (IMA)

Property

« **All messages received have been sent only once.** »
A protocol ensures Injective Message Authenticity (IMA) between sender A and receiver B if $\boldsymbol{multiset}(R_{A,B}) \subseteq \boldsymbol{multiset}(S_{A,B})$.



✗ IMA not verified

Flow Authenticity (FA)

Property

« **All messages are received in the order they have been sent.** »
A protocol ensures Flow Authenticity (FA) between sender A and receiver B if $R_{A,B}$ **is a subchain of** $S_{A,B}$.

$$S_{A,B} = \boxed{M_1}\ \boxed{M_2}\ \boxed{M_3}\ \boxed{M_4}$$

$$R_{A,B} =$$

Flow Authenticity (FA)

Property

« **All messages are received in the order they have been sent.** »
A protocol ensures Flow Authenticity (FA) between sender A and receiver B if $R_{A,B}$ **is a subchain of** $S_{A,B}$.

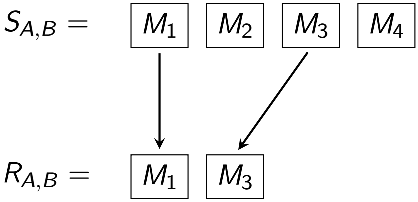
$$S_{A,B} = \quad \boxed{M_1} \; \boxed{M_2} \; \boxed{M_3} \; \boxed{M_4}$$

$$R_{A,B} = \quad \boxed{M_1}$$

Flow Authenticity (FA)

Property

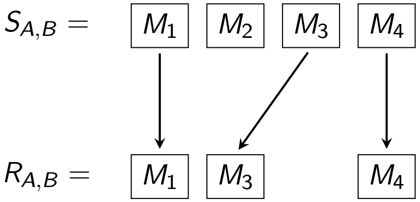
« **All messages are received in the order they have been sent.** »
A protocol ensures Flow Authenticity (FA) between sender A and receiver B if $R_{A,B}$ **is a subchain of** $S_{A,B}$.



Flow Authenticity (FA)

Property

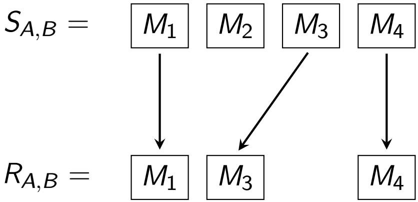
« **All messages are received in the order they have been sent.** »
A protocol ensures Flow Authenticity (FA) between sender A and receiver B if $R_{A,B}$ **is a subchain of** $S_{A,B}$.



Flow Authenticity (FA)

Property

« **All messages are received in the order they have been sent.** »
A protocol ensures Flow Authenticity (FA) between sender A and receiver B if $R_{A,B}$ **is a subchain of** $S_{A,B}$.

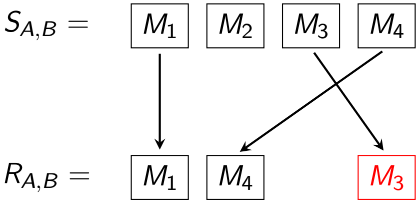


✓ FA verified

Flow Authenticity (FA)

Property

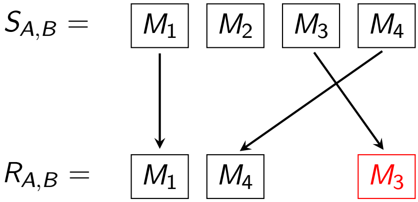
« **All messages are received in the order they have been sent.** »
A protocol ensures Flow Authenticity (FA) between sender A and receiver B if $R_{A,B}$ **is a subchain of** $S_{A,B}$.



Flow Authenticity (FA)

Property

« **All messages are received in the order they have been sent.** »
A protocol ensures Flow Authenticity (FA) between sender A and receiver B if $R_{A,B}$ **is a subchain of** $S_{A,B}$.



✗ FA not verified

Flow Authenticity (FA)

Property

« **All messages are received in the order they have been sent.** »
A protocol ensures Flow Authenticity (FA) between sender A and receiver B if $R_{A,B}$ **is a subchain of** $S_{A,B}$.

$$S_{A,B} = \quad \boxed{M_1} \; \boxed{M_2} \; \boxed{M_3} \; \boxed{M_4}$$

$$R_{A,B} =$$

Flow Authenticity (FA)

Property

« **All messages are received in the order they have been sent.** »
A protocol ensures Flow Authenticity (FA) between sender A and receiver B if $R_{A,B}$ **is a subchain of** $S_{A,B}$.

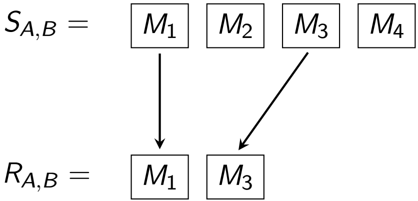
$$S_{A,B} = \quad \boxed{M_1}\ \boxed{M_2}\ \boxed{M_3}\ \boxed{M_4}$$

$$R_{A,B} = \quad \boxed{M_1}$$

Flow Authenticity (FA)

Property

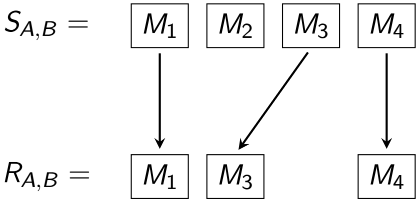
« **All messages are received in the order they have been sent.** »
A protocol ensures Flow Authenticity (FA) between sender A and receiver B if $R_{A,B}$ **is a subchain of** $S_{A,B}$.



Flow Authenticity (FA)

Property

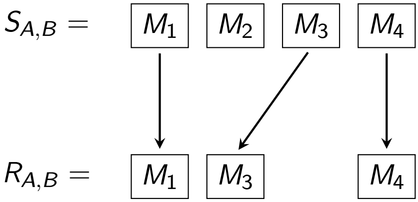
« **All messages are received in the order they have been sent.** »
A protocol ensures Flow Authenticity (FA) between sender A and receiver B if $R_{A,B}$ **is a subchain of** $S_{A,B}$.



Flow Authenticity (FA)

Property

« **All messages are received in the order they have been sent.** »
A protocol ensures Flow Authenticity (FA) between sender A and receiver B if $R_{A,B}$ **is a subchain of** $S_{A,B}$.

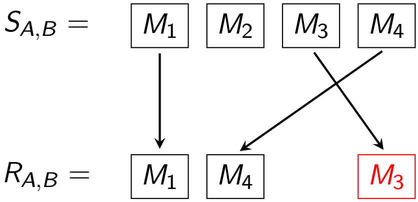


✓ FA verified

Flow Authenticity (FA)

Property

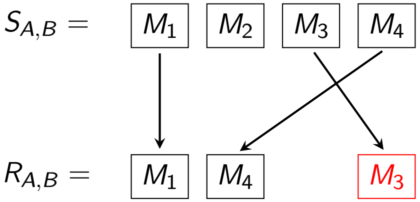
« **All messages are received in the order they have been sent.** »
A protocol ensures Flow Authenticity (FA) between sender A and receiver B if $R_{A,B}$ **is a subchain of** $S_{A,B}$.



Flow Authenticity (FA)

Property

« **All messages are received in the order they have been sent.** »
A protocol ensures Flow Authenticity (FA) between sender A and receiver B if $R_{A,B}$ **is a subchain of** $S_{A,B}$.



✗ FA not verified

Table of Contents

Tamarin Prover

- Automated cryptographic verification tool
- Developed since 2012 at ETH Zurich, Univ. of Oxford and Loria Nancy
- Protocols modeled using multiset rewritting rules
- Verified properties:
  - Trace properties: First order logical with time points
  - Observational equivalence

https://github.com/tamarin-prover/tamarin-prover

Non-Injective Message Authenticity (NIMA)

Property

« *All messages received have been sent.* »

$$\begin{aligned}
&\forall i : time, A, B : agent, m : msg. \\
&\quad Received(A, B, m)@i \Rightarrow (\\
&\qquad \exists j : time. Sent(A, B, m)@j \wedge j \lessdot i \\
&\quad)
\end{aligned}$$

Injective Message Authenticity (IMA)

Property

« All messages received have been sent only once. »

$$\begin{aligned}
&\forall i : time, A, B : agent, m : msg.\\
&\quad Received(A, B, m)@i \Rightarrow (\\
&\qquad \exists j.Sent(A, B, m)@j \wedge j \lessdot i \wedge \neg(\\
&\qquad\quad \exists i2 : time, A2, B2 : agent.\\
&\qquad\quad Received(A2, B2, m)@i2 \wedge \neg(i2 \doteq i)\\
&\qquad)\\
&\quad)
\end{aligned}$$

Resilient Channels

- Dolev-Yao intruder can block message, thus delivery is always false!
- Enforce intruder that all messages are eventually delivered.
- Security properties do not hold vacuously (still allows duplicating, reordering, delaying, forging).

$$\forall i : time, m : msg. Ch\_Sent(m)@i \\ \Rightarrow (\exists j. Ch\_Received(m)@j \wedge i \lessdot j)$$

Counters

- Usually modeled with Peano numbers, usually infinite loop
- Solution: let the intruder choose counter each time but must increment

$$\begin{aligned}
\forall i, j : time, A, B : agent, seq_1, seq_2 : msg. \\
(Seq_Sent(A, B, seq_1)@i \wedge Seq_Sent(A, B, seq_2)@j \\
\wedge i \lessdot j) \Rightarrow (\exists dif . seq_2 \approx seq_1 + dif)
\end{aligned}$$

Table of Contents

Studied Protocols

MODBUS (1979)

- No security at all.
- Some academic works to secure it:
  - Cryptographic asymmetric signatures [FCMT09]
  - Message Authentication Codes [HEK13]

OPC-UA (2006)

- Security layer: OPC-UA SecureConversation (similar to TLS).
- Three security modes:
  - None, Sign, SignAndEncrypt.

MODBUS

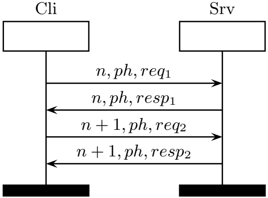


Figure : Textbook MODBUS [MOD04]

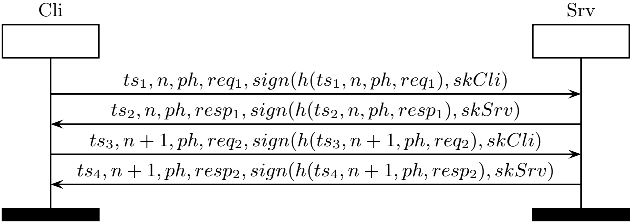


Figure : Secure MODBUS from [FCMT09]

OPC-UA

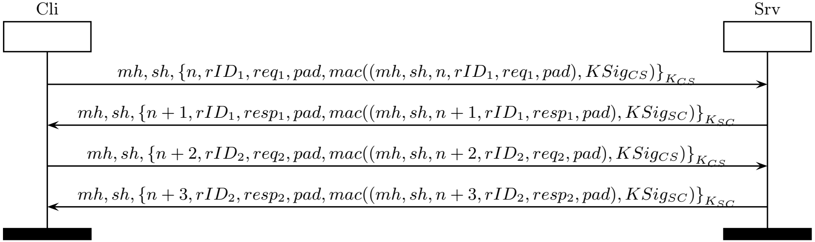


Figure : OPC-UA [IEC15]

Results on MODBUS and OPC-UA

Protocol	NIMI	IMI	FI
Textbook MODBUS [MOD04]	UNSAFE	UNSAFE	UNSAFE
MODBUS Sign [FCMT09]	UNSAFE	UNSAFE	UNSAFE
MODBUS MAC [HEK13]	SAFE	SAFE	SAFE

Table : Results for MODBUS assuming an resilient channel.

Protocol	NIMI	IMI	FI
OPC-UA None	UNSAFE	UNSAFE	UNSAFE
OPC-UA Sign	SAFE	SAFE	SAFE
OPC-UA SignAndEncrypt	SAFE	SAFE	SAFE

Table : Results for OPC-UA [IEC15], assuming a resilient channel.

Results on OPC-UA with bounded counters

- In real life, machine integers are bounded and wrap over.

Protocol	NIMA	IMA	FA	NIMD	IMD	FD
OPC-UA SignAndEncrypt with bounded numbers Insecure Channel	SAFE	SAFE	UNSAFE	UNSAFE	UNSAFE	UNSAFE

Table : Results for OPC-UA with bounded counters.

Results on OPC-UA with bounded counters

- In real life, machine integers are bounded and wrap over.

Protocol	NIMA	IMA	FA	NIMD	IMD	FD
OPC-UA SignAndEncrypt with bounded numbers Insecure Channel	SAFE	SAFE	UNSAFE	UNSAFE	UNSAFE	UNSAFE

Table : Results for OPC-UA with bounded counters.

Attack on FA with bounded counters (modulo 4)

$$S_{A,B} = \boxed{\begin{array}{c} M_1 \\ \text{seq}=1 \end{array}} \quad \boxed{\begin{array}{c} M_2 \\ \text{seq}=2 \end{array}} \quad \boxed{\begin{array}{c} M_3 \\ \text{seq}=3 \end{array}} \quad \boxed{\begin{array}{c} M_4 \\ \text{seq}=4 \end{array}} \quad \boxed{\begin{array}{c} M_5 \\ \text{seq}=1 \end{array}}$$

$$R_{A,B} =$$

Results on OPC-UA with bounded counters

- In real life, machine integers are bounded and wrap over.

Protocol	NIMA	IMA	FA	NIMD	IMD	FD
OPC-UA SignAndEncrypt with bounded numbers Insecure Channel	SAFE	SAFE	UNSAFE	UNSAFE	UNSAFE	UNSAFE

Table : Results for OPC-UA with bounded counters.

Attack on FA with bounded counters (modulo 4)

$S_{A,B} =$ $\boxed{M_1 \text{ seq}=1}$ $\boxed{M_2 \text{ seq}=2}$ $\boxed{M_3 \text{ seq}=3}$ $\boxed{M_4 \text{ seq}=4}$ $\boxed{M_5 \text{ seq}=1}$

$R_{A,B} =$ $\boxed{M_5 \text{ seq}=1}$

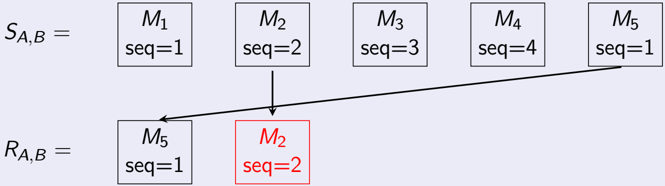
Results on OPC-UA with bounded counters

- In real life, machine integers are bounded and wrap over.

Protocol	NIMA	IMA	FA	NIMD	IMD	FD
OPC-UA SignAndEncrypt with bounded numbers Insecure Channel	SAFE	SAFE	UNSAFE	UNSAFE	UNSAFE	UNSAFE

Table : Results for OPC-UA with bounded counters.

Attack on FA with bounded counters (modulo 4)



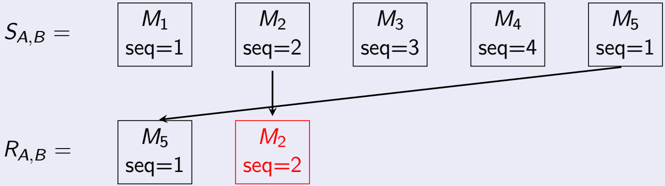
Results on OPC-UA with bounded counters

- In real life, machine integers are bounded and wrap over.

Protocol	NIMA	IMA	FA	NIMD	IMD	FD
OPC-UA SignAndEncrypt with bounded numbers Insecure Channel	SAFE	SAFE	UNSAFE	UNSAFE	UNSAFE	UNSAFE

Table : Results for OPC-UA with bounded counters.

Attack on FA with bounded counters (modulo 4)



- In practice, OPC-UA renegociates keys when sequence numbers wrap.
- Attack disapears, with this counter measure.

Table of Contents

Table of Contents

Identification of Attack Vectors

- Global analysis of attacker's objectives and communication protocols to reduce the number of possible scenarios

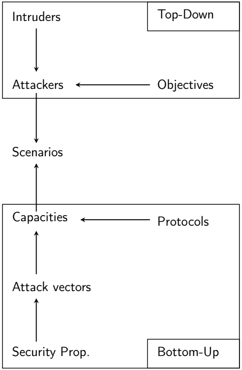


Figure : Attack vector analysis

- Top-down step:
  - Identify attacker's position and objectives
  - Similar to risk analysis methods
- Bottom-Up step:
  - Identify attacker's capacities given protocols counter-measure (encryption, signatures, etc)
- Combine both to obtain possible attack vectors

Top-Down Example

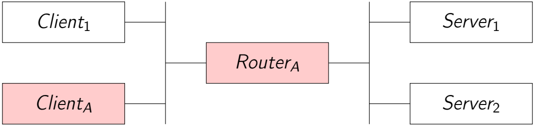


Figure : Infrastructure example

Possible security objectives:

- $IdTh$ = Identity theft,
- $AuthBP$ = Authentication by-pass,

$\mathcal{R}_{Obj}$	$IdTh$	$AuthBP$
$Client_A$	✗	✓
$Router_A$	✓	✗

Table : Objectives for each attacker

Bottom-Up Example

Possible realisation of objectives:

- $Real(IdTh) = \{\{Spy\}\}$
- $Real(AuthBP) = \{\{Usurp\}, \{Replay\}\}$

$Atk.vectors$	Spy	$Usurp$	$Replay$
FTP_{Auth}	✓	✗	✓
$\text{OPC-UA}_{SignEnc}$	✗	✗	✗

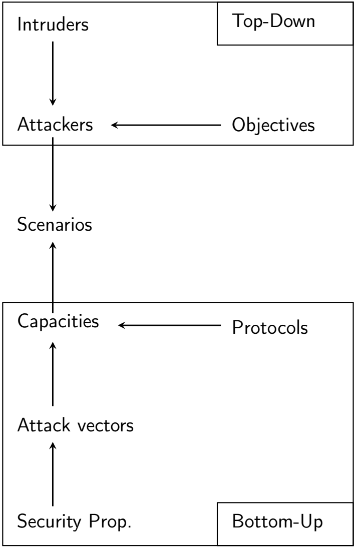
Table : Atk. vectors for each protocol

Results:

- $\mathcal{S}_{Client_A,\text{FTP}_{Auth}} = \{(AuthBP, Replay)\}$
- $\mathcal{S}_{Client_A,\text{OPC-UA}_{SignEnc}} = \emptyset$
- $\mathcal{S}_{Router_A,\text{FTP}_{Auth}} = \{(IdTh, Spy)\}$
- $\mathcal{S}_{Router_A,\text{OPC-UA}_{SignEnc}} = \emptyset$

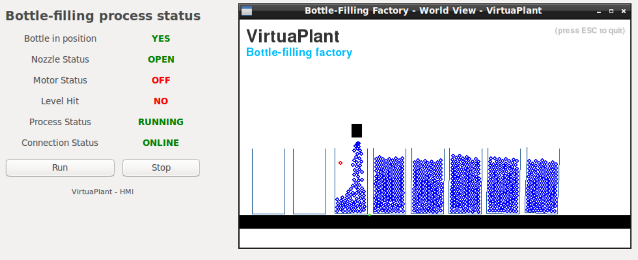
Phase 1: Attacker Models

- Presented at AFADL 2016, Besançon.
- Risk analysis focused on attackers.
- Based on:
  - Topology of the system;
  - Attacker objectives;
  - Security features of protocols.
- Objectives are security vuln., e.g.:
  - Modify a message;
  - Circumvent authentication.
- Yields attacker models in terms of:
  - Position in the topology;
  - Capacities (actions and deduction).



Case Study: Bottle-filling Factory

- Process simulator: `https://github.com/jseidl/virtuaplant`



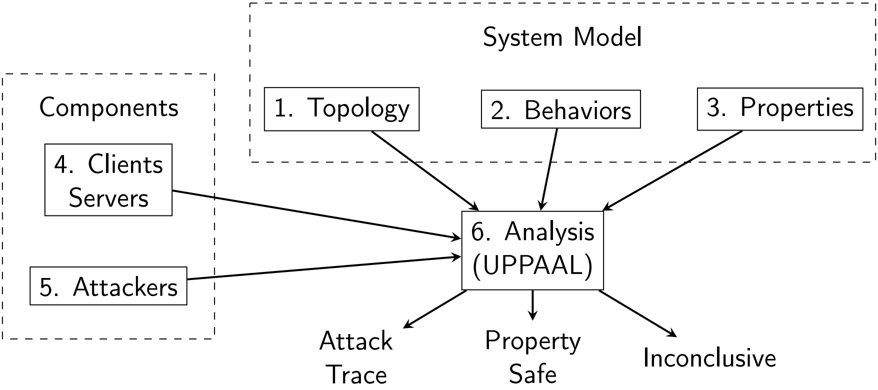
Variables:

- Conveyor belt
- Nozzle
- Position captor
- Level captor
- On/Off Switch

Properties:

- Nozzle only opens when a bottle is detected.
- Conveyor belt only starts when the bottle is full.
- Nozzle only opens when conveyor belt is stopped.

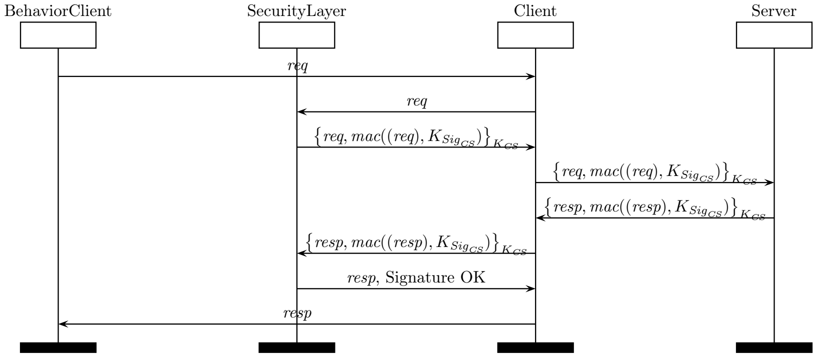
Phase 2: Generation of Attack Scenarios



Clients and Servers

For a transport protocol:

- Encapsulate and decapsulate applicative message into packets.
- Reusable for a model to another.
- BehaviorClient generates applicative messages.
- SecurityLayer performs cryptographic operations.



Timings

Topologies	Properties	A_1	A_2	A_3	A_4
T_1	Φ_1	0.43 s	0.07 s	1.05 s	0.84 s
	Φ_2	0.52 s	0.10 s	0.69 s	0.35 s
	Φ_3	0.47 s	0.04 s	0.37 s	0.42 s
T_2	Φ_1	Out of memory		601 s	31.55 s
	Φ_2	0.66 s	0.23 s	2.17 s	35.20 s
	Φ_3	0.78 s	0.21 s	2.35 s	34.85 s

Observations on results on the POC:

- A_2 obtains same results as A_1 faster (not all capacities of Dolev-Yao are needed to find attacks in this case);
- A_3 globally needs more time but is able to conclude on Φ_1 (less state-space needed);
- A_4 is globally the slowest: as it does not find any attacks, UPPAAL explores all paths.

References I

Raphael Amoah, *Formal security analysis of the dnp3-secure authentication protocol*, Ph.D. thesis, Queensland University of Technology, 2016.

Eric J Byres, Matthew Franz, and Darrin Miller, *The use of attack trees in assessing vulnerabilities in scada systems*, Proceedings of the international infrastructure survivability workshop, 2004.

Bruno Blanchet, *An efficient cryptographic protocol verifier based on Prolog rules*, Proceedings of the 14th IEEE Workshop on Computer Security Foundations (Washington, DC, USA), CSFW '01, IEEE Computer Society, 2001, pp. 82–.

References II

- Alvaro A Cárdenas, Saurabh Amin, Zong-Syun Lin, Yu-Lun Huang, Chi-Yen Huang, and Shankar Sastry, *Attacks against process control systems: risk assessment, detection, and response*, Proceedings of the 6th ACM symposium on information, computer and communications security, ACM, 2011, pp. 355–366.
- Yulia Cherdantseva, Pete Burnap, Andrew Blyth, Peter Eden, Kevin Jones, Hugh Soulsby, and Kristan Stoddart, *A review of cyber security risk assessment methods for SCADA systems*, Computers & Security **56** (2015), 1 – 27.
- Gordon R Clarke, Deon Reynders, and Edwin Wright, *Practical modern scada protocols: Dnp3, 60870.5 and related systems*, Newnes, 2004.
- D. Dzung, M. Naedele, T.P. von Hoff, and M. Crevatin, *Security for industrial communication systems*, Proceedings of the IEEE **93** (2005), no. 6, 1152–1177.

References III

Jannik Dreier, Maxime Puys, Marie-Laure Potet, Pascal Lafourcade, and Jean-Louis Roch, *Formally verifying flow integrity properties in industrial systems*, SECRYPT 2017 - 14th International Conference on Security and Cryptography (Madrid, Spain), July 2017, p. 12.

D. Dolev and Andrew C. Yao, *On the security of public key protocols*, Information Theory, IEEE Transactions on **29** (1981), no. 2, 198–208.

IgorNai Fovino, Andrea Carcano, Marcelo Masera, and Alberto Trombetta, *Design and implementation of a secure MODBUS protocol*, Critical Infrastructure Protection III (Charles Palmer and Sujeet Shenoi, eds.), IFIP Advances in Information and Communication Technology, vol. 311, Springer Berlin Heidelberg, 2009, pp. 83–96 (English).

References IV

- JH Graham and SC Patel, *Correctness proofs for SCADA communication protocols*, Proceedings of the Ninth World Multi-Conference on Systemics, Cybernetics and Informatics, 2005, pp. 392–397.
- G. Hayes and K. El-Khatib, *Securing MODBUS transactions using hash-based message authentication codes and stream transmission control protocol*, Communications and Information Technology (ICCIT), 2013 Third International Conference on, June 2013, pp. 179–184.
- IEC-62541, *OPC Unified Architecture*, International Electrotechnical Commission, August 2015.
- S Kriaa, M Bouissou, and Y Laarouchi, *A model based approach for SCADA safety and security joint modelling: S-Cube*, IET System Safety and Cyber Security, IET Digital Library, 2015.

References V

Siwar Kriaa, Ludovic Pietre-Cambacedes, Marc Bouissou, and Yoran Halgand, *A survey of approaches combining safety and security for industrial control systems*, Reliability Engineering & System Safety **139** (2015), 156–178.

Miles A McQueen, Wayne F Boyer, Mark A Flynn, and George A Beitel, *Quantitative cyber risk reduction estimation methodology for a small scada control system*, System Sciences, 2006. HICSS'06. Proceedings of the 39th Annual Hawaii International Conference on, vol. 9, IEEE, 2006, pp. 226–226.

MODBUS, *MODBUS IDA, MODBUS messaging on TCP/IP implementation guide v1.0a*, 2004.

References VI

- Simon Meier, Benedikt Schmidt, Cas Cremers, and David Basin, *The tamarin prover for the symbolic analysis of security protocols*, Computer Aided Verification (Natasha Sharygina and Helmut Veith, eds.), Lecture Notes in Computer Science, vol. 8044, Springer Berlin Heidelberg, 2013, pp. 696–701 (English).
- Ludovic Piètre-Cambacédès, *The relationships between safety and security*, Theses, Télécom ParisTech, November 2010.
- Ludovic Piètre-Cambacédès and Marc Bouissou, *Cross-fertilization between safety and security engineering*, Reliability Engineering & System Safety **110** (2013), 110–126.

References VII

- Sandip C Patel, James H Graham, and Patricia AS Ralston, *Quantitatively assessing the vulnerability of critical information systems: A new method for evaluating security enhancements*, International Journal of Information Management **28** (2008), no. 6, 483–491.
- Maxime Puys, Marie-Laure Potet, and Pascal Lafourcade, *Formal analysis of security properties on the OPC-UA SCADA protocol*, Computer Safety, Reliability, and Security - 35th International Conference, SAFECOMP 2016, Trondheim, Norway, September 21-23, 2016, Proceedings, 2016, pp. 67–75.
- Sandip C Patel and Yingbing Yu, *Analysis of SCADA security models*, International Management Review **3** (2007), no. 2, 68.

References VIII

Marco Rocchetto and Nils Ole Tippenhauer, *Towards formal security analysis of industrial control systems*, Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, ACM, 2017, pp. 114–126.

Chee-Wooi Ten, Govindarasu Manimaran, and Chen-Ching Liu, *Cybersecurity for critical infrastructures: Attack and defense modeling*, IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans **40** (2010), no. 4, 853–865.

Theodore J Williams, *A reference model for computer integrated manufacturing (cim): A description from the viewpoint of industrial automation: Prepared by cim reference model committee international purdue workshop on industrial computer systems*, Instrument Society of America, 1991.

References IX

Qu Wanying, Wei Weimin, Zhu Surong, and Zhao Yan, *The study of security issues for the industrial control systems communication protocols*, Joint International Mechanical, Electronic and Information Technology Conference (JIMET 2015) (2015).