SCADA systems security: verifying integrity properties

Maxime Puys and Jean-Louis Roch

Grenoble INP - Ensimag, VERIMAG, University of Grenoble Alpes, France

April 24, 2019

SEIT'2019, INPT Rabat

Cybersecurity Institute
Univ. Grenoble Alpes

Industrial Systems

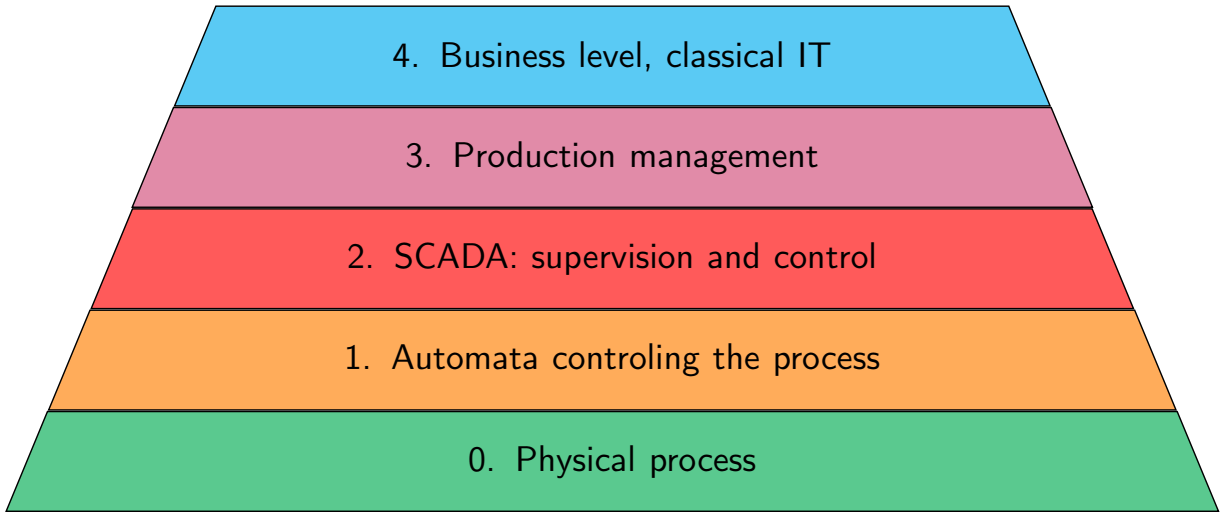
- SCADA : Supervisory Control and Data Acquisition
- Critical industrial infrastructures: energy, water, oil, gas

Hot topic : cybersecurity

- 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 Protocols

- Allow industrial devices to communicate.
- Must guarantee security properties such as:
  - Authentication
  - Integrity
  - (Secrecy when dealing with customer data).
  - (Non-repudiation)



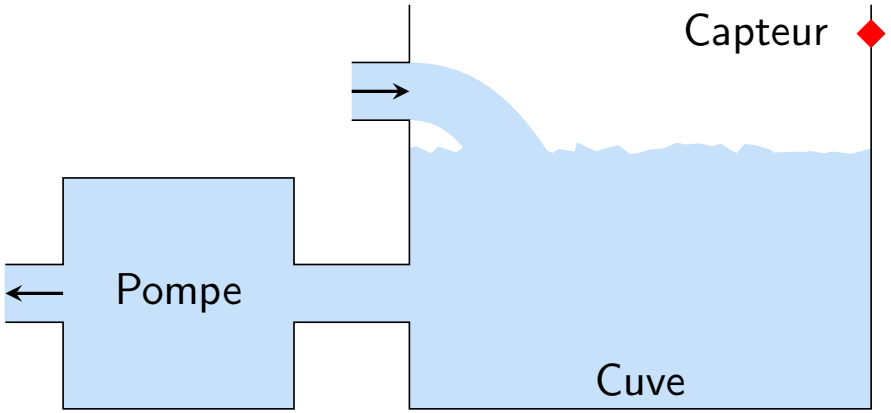
[Wil91] Theodore J Williams. *A reference model for computer integrated manufacturing (cim): A description from the viewpoint of industrial automation: Prepared by cim reference model committee international purdue workshop on industrial computer systems*, Instrument Society of America, 1991.

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.

A Common Thread: Maroochy Shire

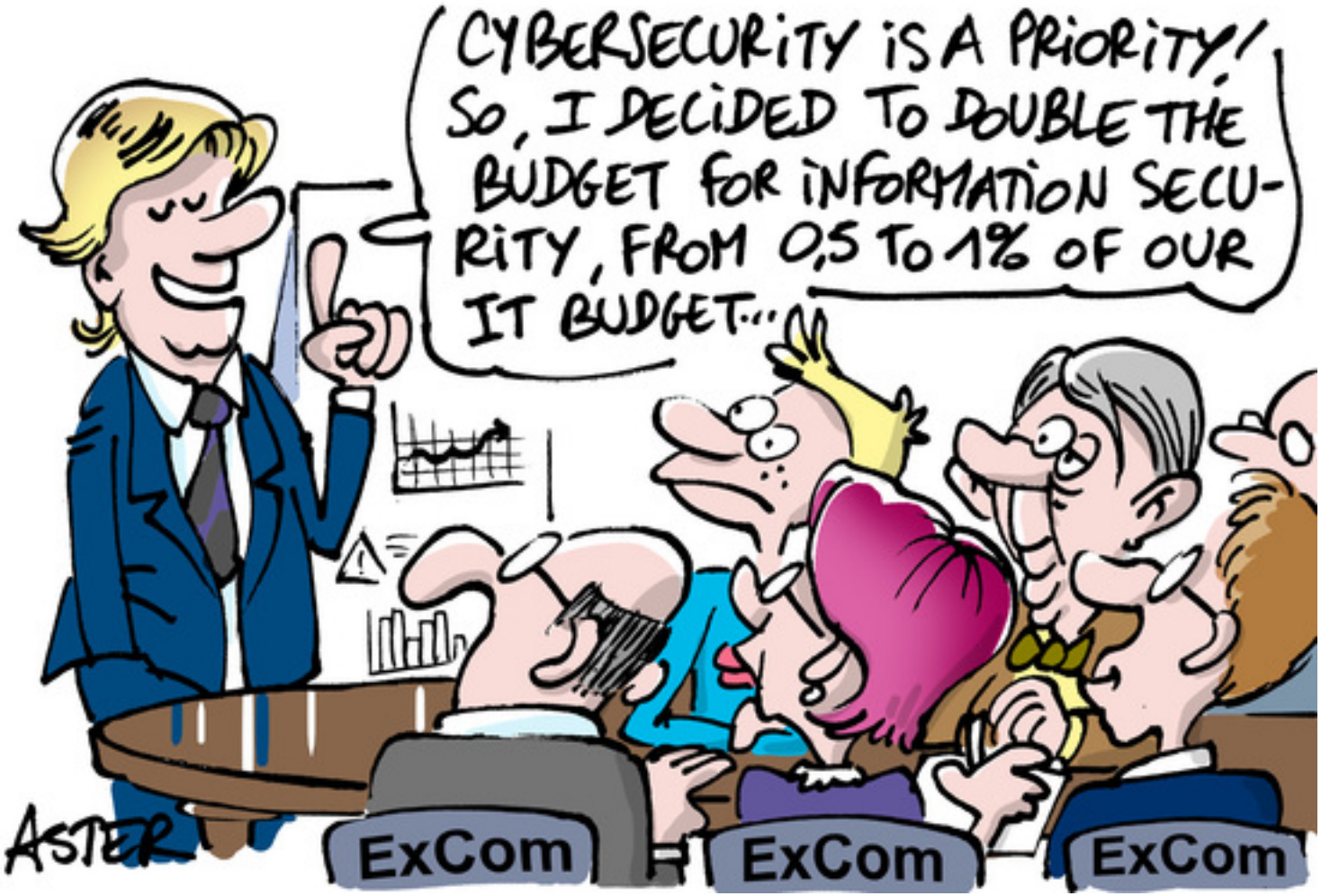
- Real attack occurring in 2000 in Australia.
- An insider spills $\sim$ 1M litters of raw sewage into nature.
- Attack over several months.



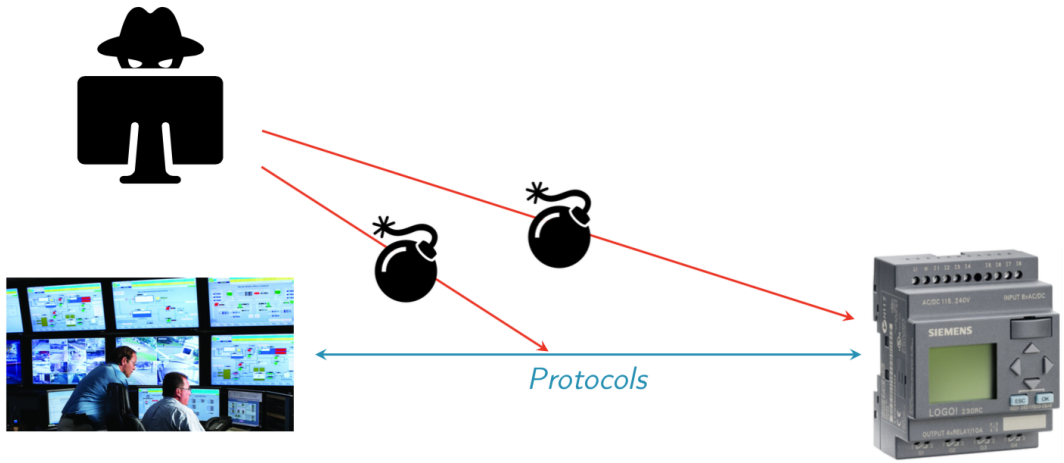
In our context, at least 3 vulnerabilities:

- **Vulnerability 1:** Absence of **authentication mechanism** in communication protocols.
- **Vulnerability 2:** Absence of **safety mechanism** to avoid the spill.
- **Vulnerability 3:** Absence of **prevision** of attacks.

How to asset industrial system integrity?



How to asset industrial system integrity?



- On line : eg firewall, stateful monitoring and filtering.
- Off line : formal verification.

Formal Verification

- Crucial for industrial systems due to:
  1. Their interactions with physical world.
  2. Their really long lifetime and difficulty to patch.

$\Rightarrow$ Better check the protocol beforehand to save time and money.

Table of Contents

Table of Contents

Cryptographic Protocols Verification 1/2

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**.

Cryptographic Protocols Verification 1/2

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

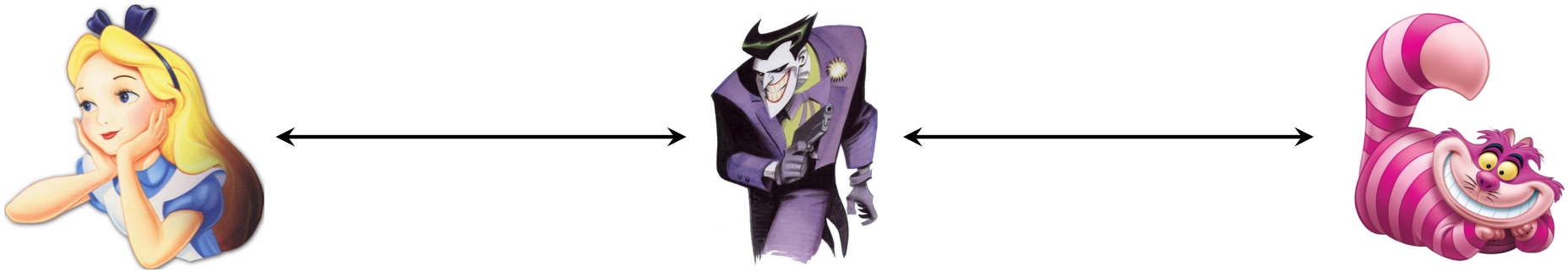
1. $A \rightarrow I : \{A, N_A\}_{KI}$
2. $A \leftarrow I : \{N_A, N_B\}_{KA}$
3. $A \rightarrow I : \{N_B\}_{KI}$

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

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.
- Used to prove IT protocols (TLS, SSH).
- Verified properties: secret, authentication, observational equivalence



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 a ciphertext and the key, he can deduce the plaintext.

Related Works on industrial protocol

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
[PPL16]	2016	OPC-UA	Formal (ProVerif)
[DPPLR17]	2017	MODBUS, OPC-UA	Formal(Tamarin)

J. Dreier, M. Puys, M.-L. Potet, P. Lafourcade, and J.-L. Roch. *Formally verifying flow integrity properties in industrial systems*. SECRYPT'17, 2017.

- Formalized properties for industrial systems
- Implemented them in the Tamarin prover
- Tested on 2 real industrial protocols and academic works

Table of Contents

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 $\textbf{\textit{set}}(R_{A,B}) \subseteq \textbf{\textit{set}}(S_{A,B})$.

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

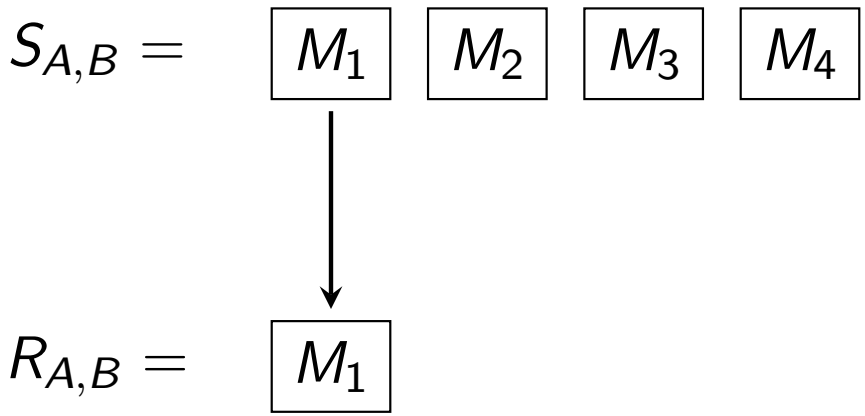
$$R_{A,B} =$$

Non-Injective Message Authenticity (NIMA)

Property

« All messages received have been sent. »

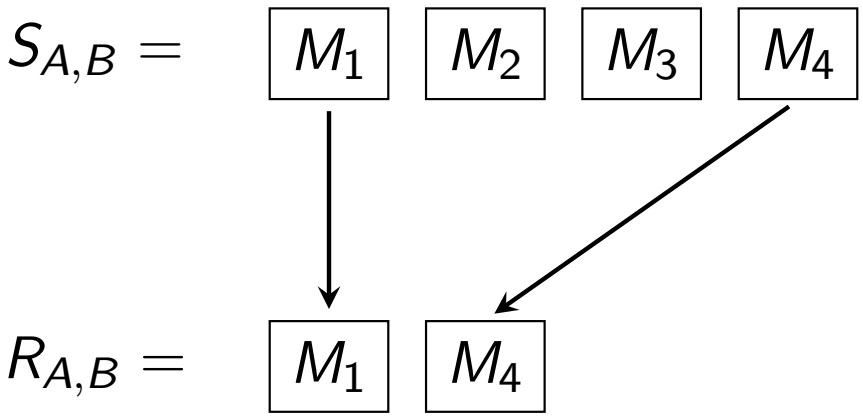
A protocol ensures Non-Injective Message Authenticity (NIMA) between sender A and receiver B if $\textbf{\textit{set}}(R_{A,B}) \subseteq \textbf{\textit{set}}(S_{A,B})$.



Non-Injective Message Authenticity (NIMA)

Property

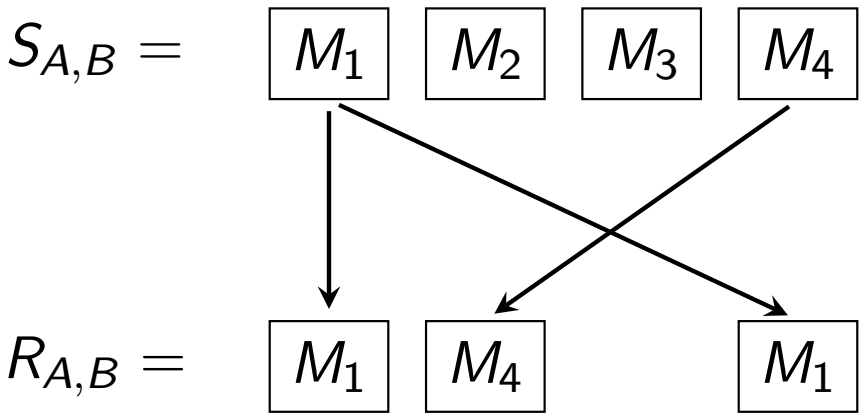
« All messages received have been sent. »
A protocol ensures Non-Injective Message Authenticity (NIMA) between sender A and receiver B if $\mathbf{set}(R_{A,B}) \subseteq \mathbf{set}(S_{A,B})$.



Non-Injective Message Authenticity (NIMA)

Property

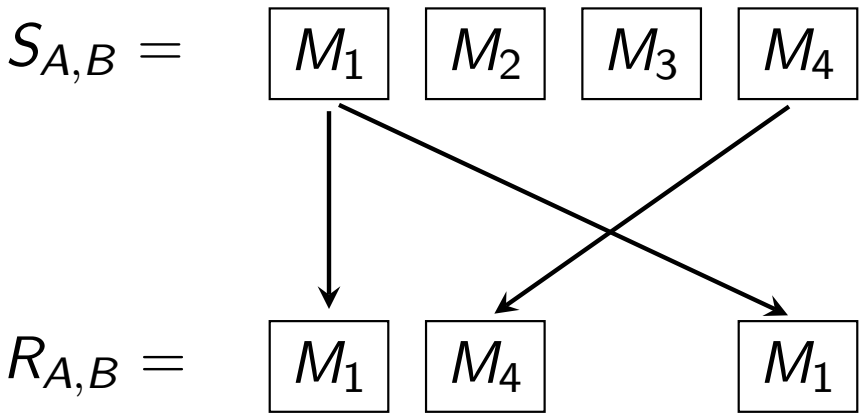
« All messages received have been sent. »
A protocol ensures Non-Injective Message Authenticity (NIMA) between sender A and receiver B if $\textbf{set}(R_{A,B}) \subseteq \textbf{set}(S_{A,B})$.



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 $\textbf{set}(R_{A,B}) \subseteq \textbf{set}(S_{A,B})$.

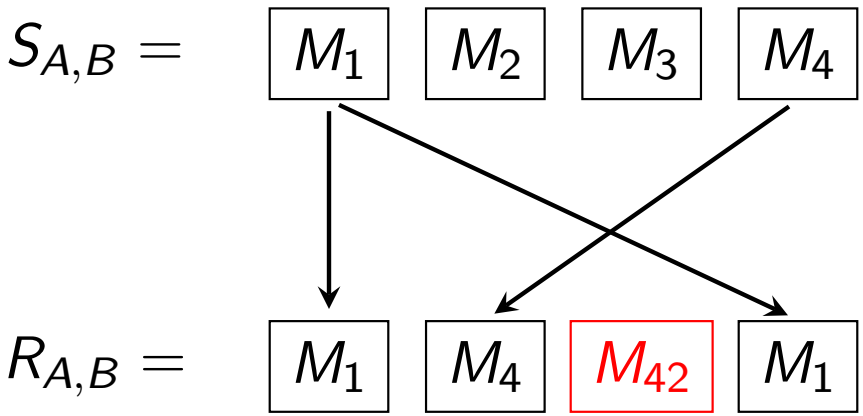


✓ NIMA verified

Non-Injective Message Authenticity (NIMA)

Property

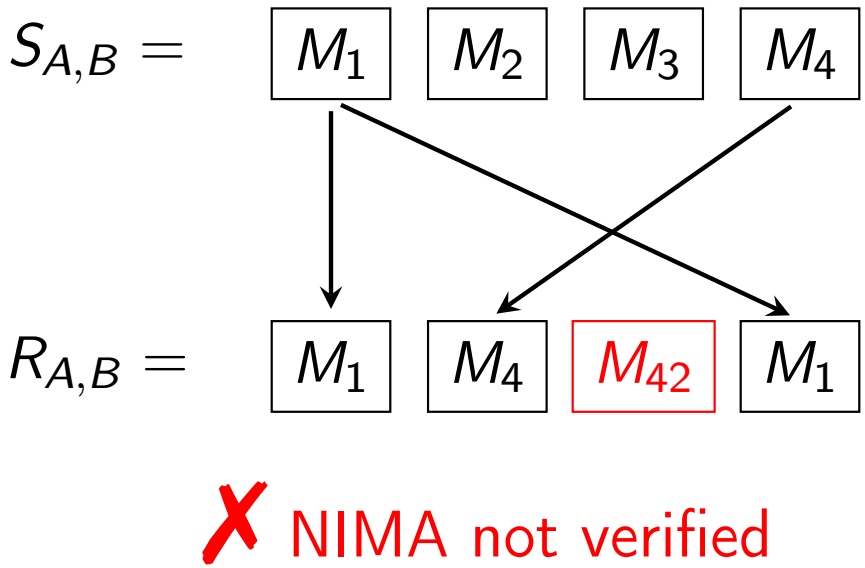
« **All messages received have been sent.** »
A protocol ensures Non-Injective Message Authenticity (NIMA) between sender A and receiver B if $\textbf{set}(R_{A,B}) \subseteq \textbf{set}(S_{A,B})$.



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

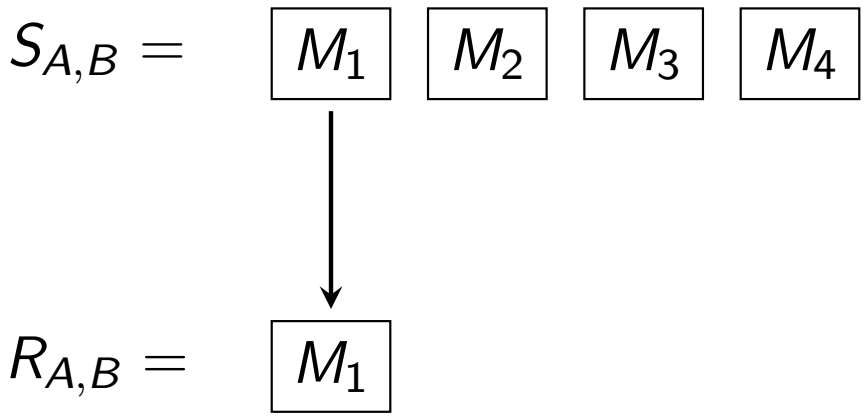
$$S_{A,B} = \boxed{M_1}\ \boxed{M_2}\ \boxed{M_3}\ \boxed{M_4}$$

$$R_{A,B} =$$

Injective Message Authenticity (IMA)

Property

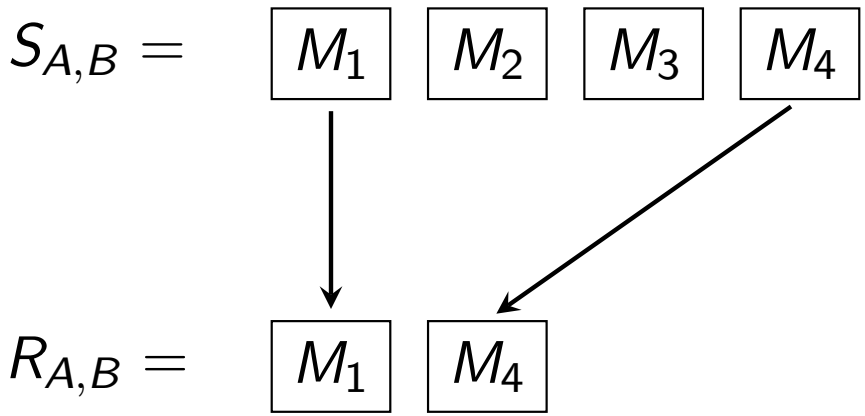
« **All messages received n times have been sent n times.** »
A protocol ensures Injective Message Authenticity (IMA) between sender A and receiver B if $\textbf{\textit{multiset}}(R_{A,B}) \subseteq \textbf{\textit{multiset}}(S_{A,B})$.



Injective Message Authenticity (IMA)

Property

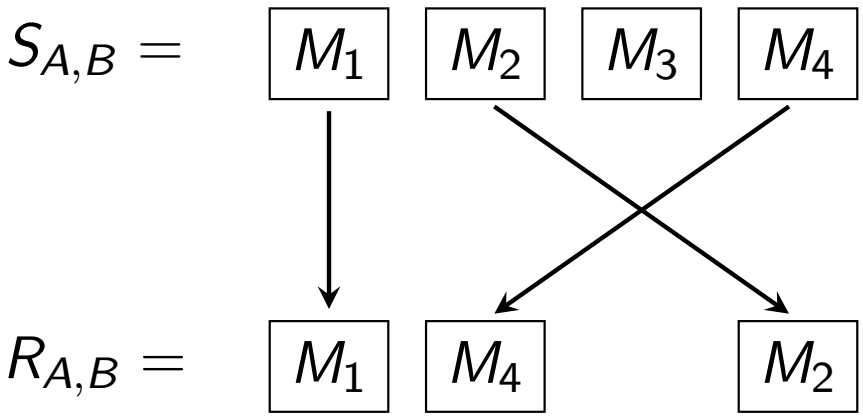
« All messages received n times have been sent n times. »
A protocol ensures Injective Message Authenticity (IMA) between sender A and receiver B if $\textbf{multiset}(R_{A,B}) \subseteq \textbf{multiset}(S_{A,B})$.



Injective Message Authenticity (IMA)

Property

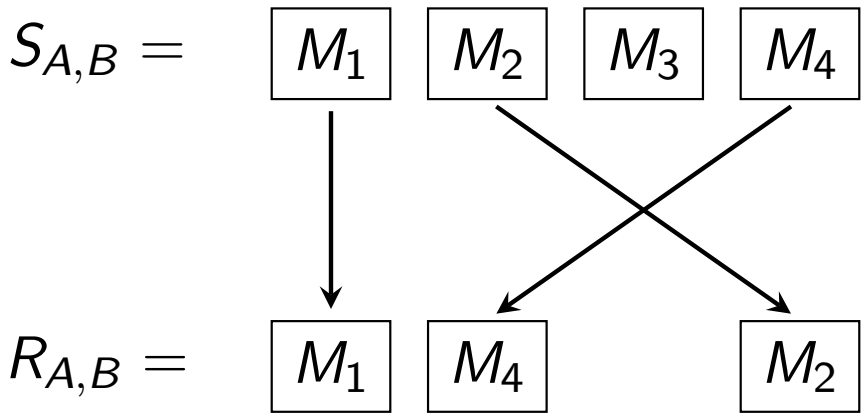
« **All messages received** n **times have been sent** n **times.** »
A protocol ensures Injective Message Authenticity (IMA) between sender A and receiver B if $\textbf{\textit{multiset}}(R_{A,B}) \subseteq \textbf{\textit{multiset}}(S_{A,B})$.



Injective Message Authenticity (IMA)

Property

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

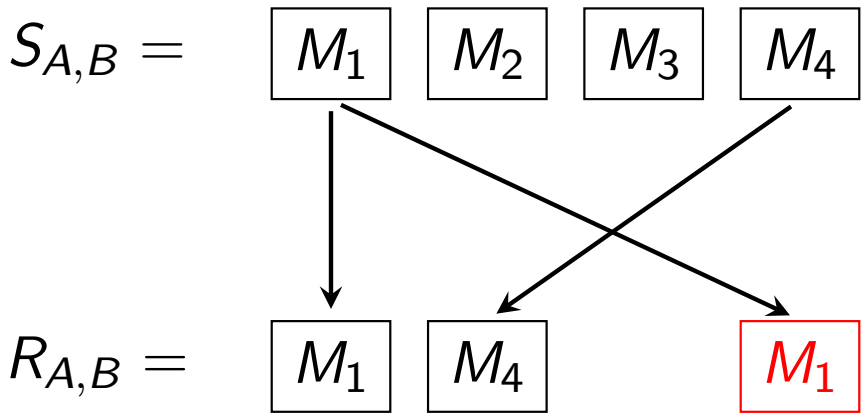


✓ IMA verified

Injective Message Authenticity (IMA)

Property

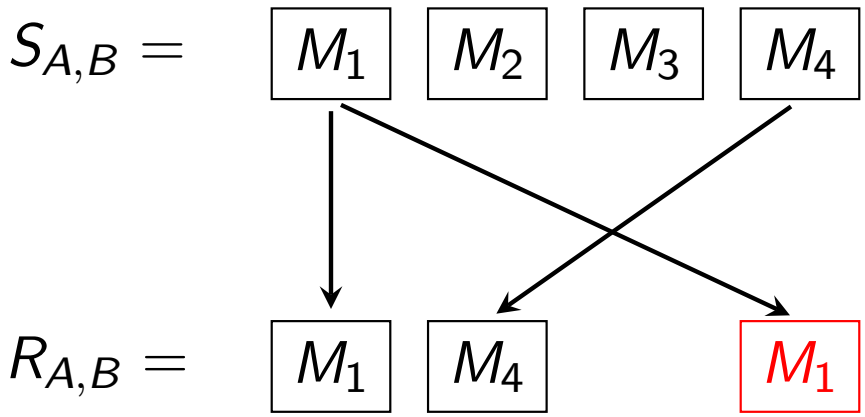
« **All messages received** n **times have been sent** n **times.** »
A protocol ensures Injective Message Authenticity (IMA) between sender A and receiver B if $\textbf{\textit{multiset}}(R_{A,B}) \subseteq \textbf{\textit{multiset}}(S_{A,B})$.



Injective Message Authenticity (IMA)

Property

« All messages received n times have been sent n times. »
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 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** $S_{A,B}$.

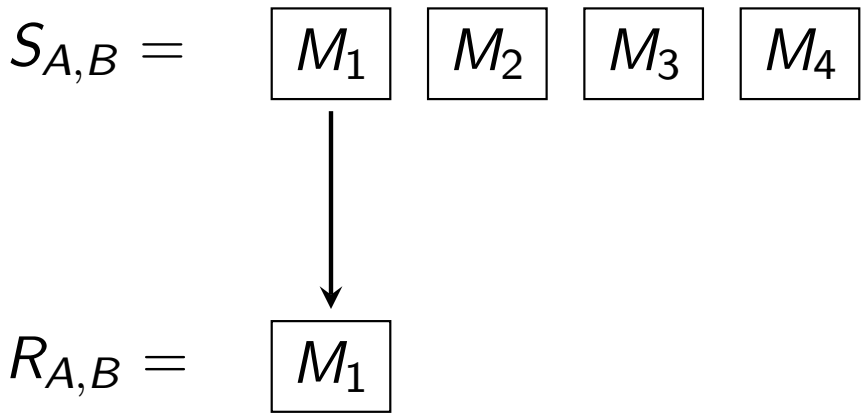
$$S_{A,B} = \boxed{M_1}\ \boxed{M_2}\ \boxed{M_3}\ \boxed{M_4}$$

$$R_{A,B} =$$

Flow Authenticity (FA)

Property

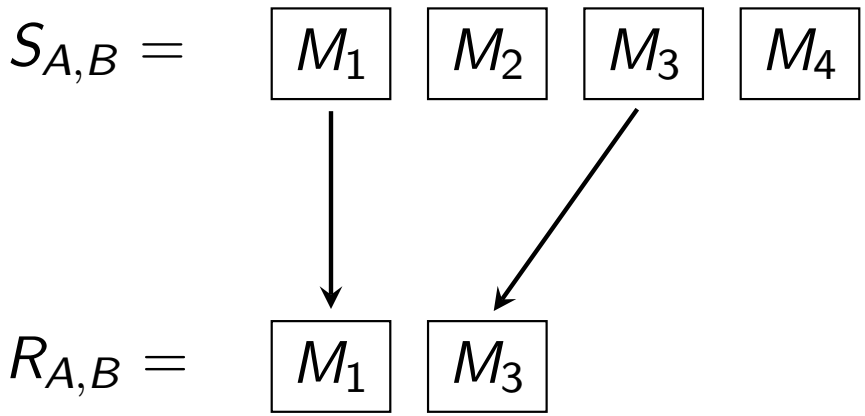
« **All messages are received in the order they have been sent.** »
A protocol ensures Flow Authenticity (FA) between sender A and receiver B if $R_{A,B}$ **is a subchain of** $S_{A,B}$.



Flow Authenticity (FA)

Property

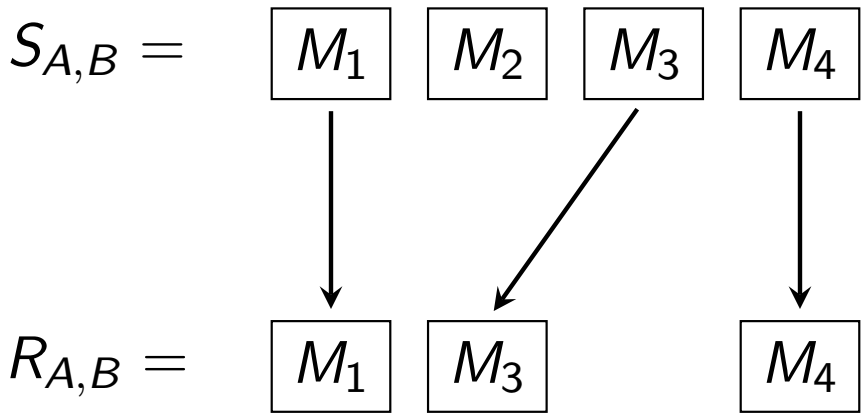
« **All messages are received in the order they have been sent.** »
A protocol ensures Flow Authenticity (FA) between sender A and receiver B if $R_{A,B}$ **is a subchain of** $S_{A,B}$.



Flow Authenticity (FA)

Property

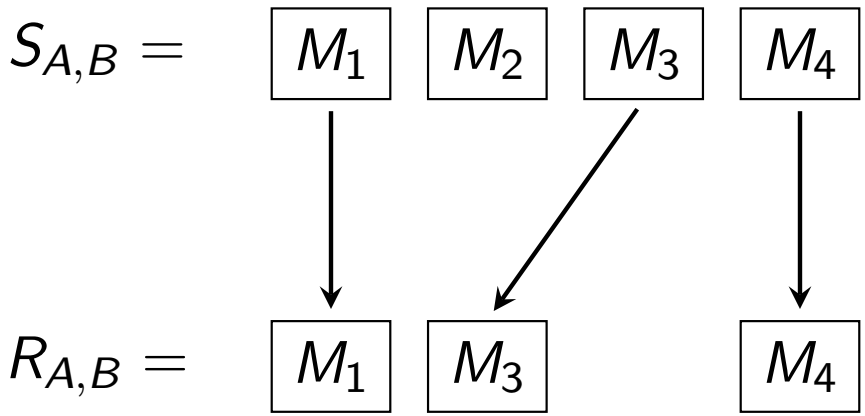
« **All messages are received in the order they have been sent.** »
A protocol ensures Flow Authenticity (FA) between sender A and receiver B if $R_{A,B}$ **is a subchain of** $S_{A,B}$.



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** $S_{A,B}$.

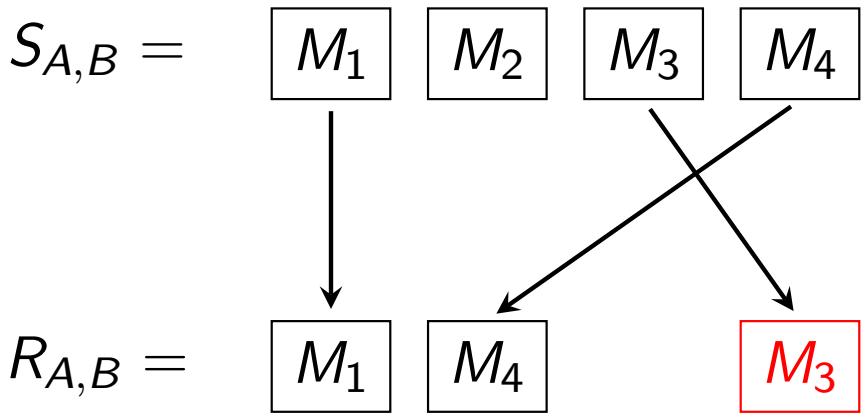


✓ FA verified

Flow Authenticity (FA)

Property

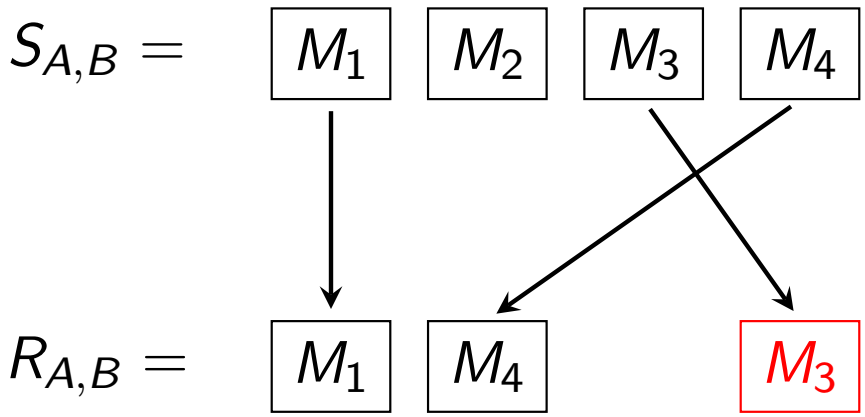
« **All messages are received in the order they have been sent.** »
A protocol ensures Flow Authenticity (FA) between sender A and receiver B if $R_{A,B}$ **is a subchain of** $S_{A,B}$.



Flow Authenticity (FA)

Property

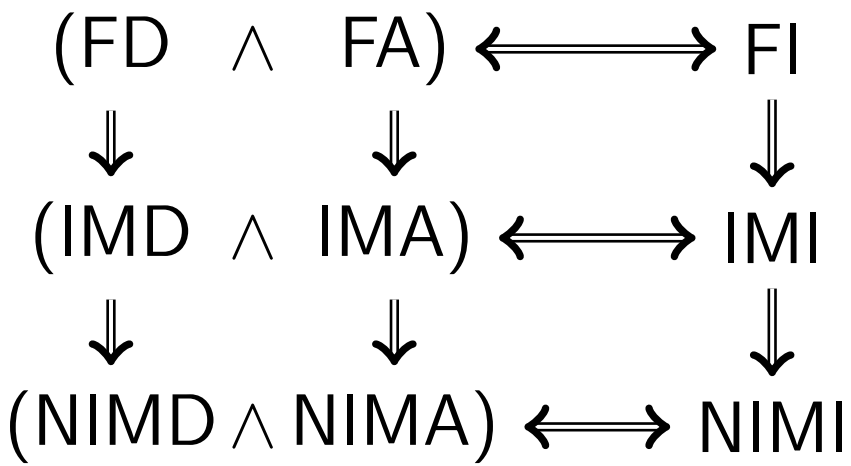
« All messages are received in the order they have been sent. »
A protocol ensures Flow Authenticity (FA) between sender A and receiver B if $R_{A,B}$ **is a subchain of** $S_{A,B}$.



✗ FA not verified

Flow integrity properties and relations

Suffix: **A**=Authenticity ; **D**=Delivery ; **I** = Integrity.

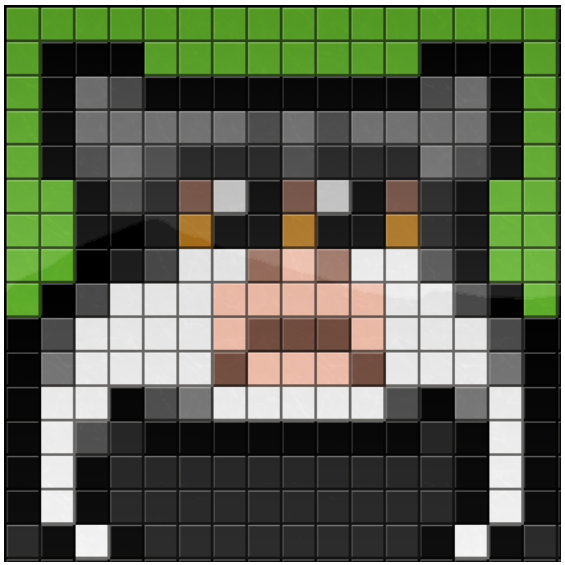


[DPPLR17] Relationships: $A \Rightarrow B$ if a protocol ensuring A also ensures B.

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

Table of Contents

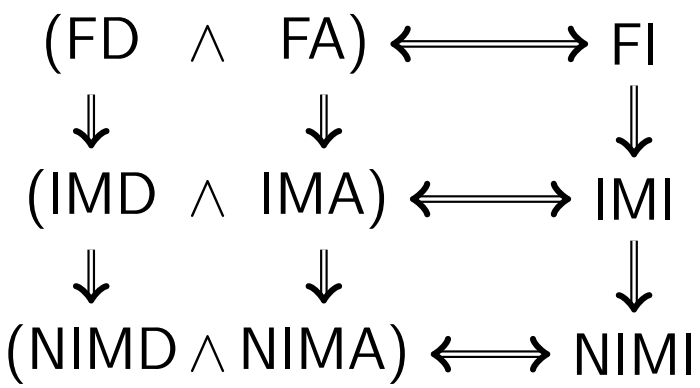
Tamarin Prover



- 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`

Flow Integrity Properties in Tamarin



Implementation in collaboration with developers of Tamarin:

- Models for **sequences numbers** (i.e.: counters) and **resilient channels**.

Property FA (Flow Authenticity)

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

$$\forall i, j : time, A, B : agent, m, m_2 : msg.($$
$$\quad Received(A, B, m)@i \wedge Received(A, B, m_2)@j \wedge i \lessdot j$$
$$) \Rightarrow (\exists k, l : time.$$
$$\quad Sent(A, B, m)@k \wedge Sent(A, B, m_2)@l \wedge k \lessdot l$$
$$)$$

Application to Industrial Protocols

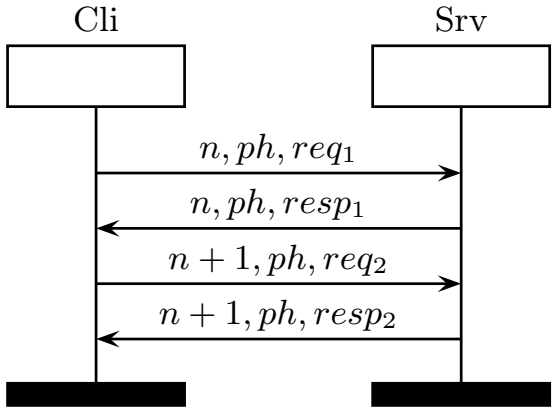
MODBUS (1979)

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

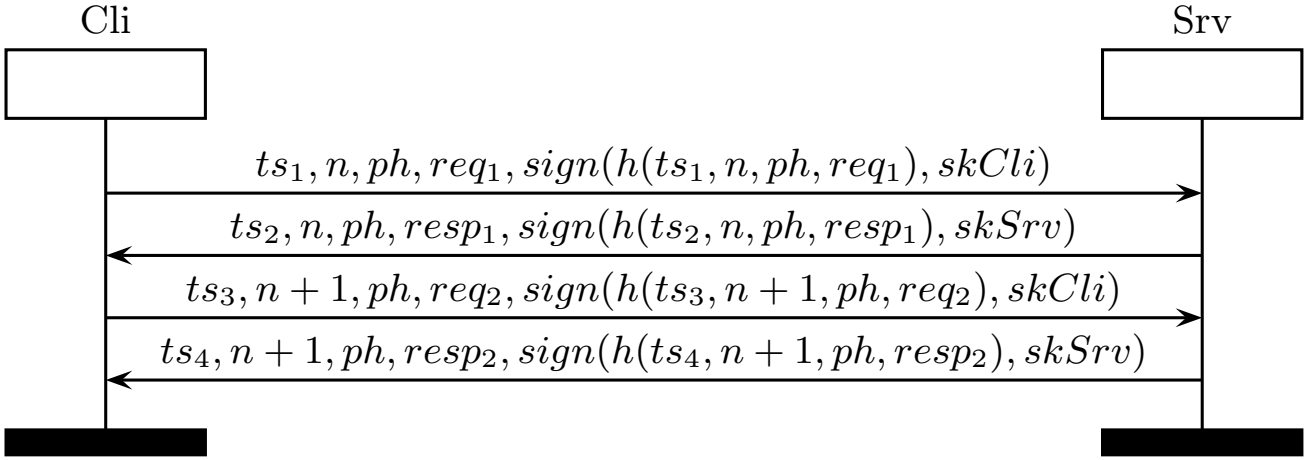
OPC-UA (2006)

- Security layer: OPC-UA SecureConversation (similar to TLS).
- Next standard for industry (consortium of key stakeholders)
- Currently developed and maintained (1000 pages of specification)
- Three security modes:
  - None, Sign, SignAndEncrypt.

MODBUS

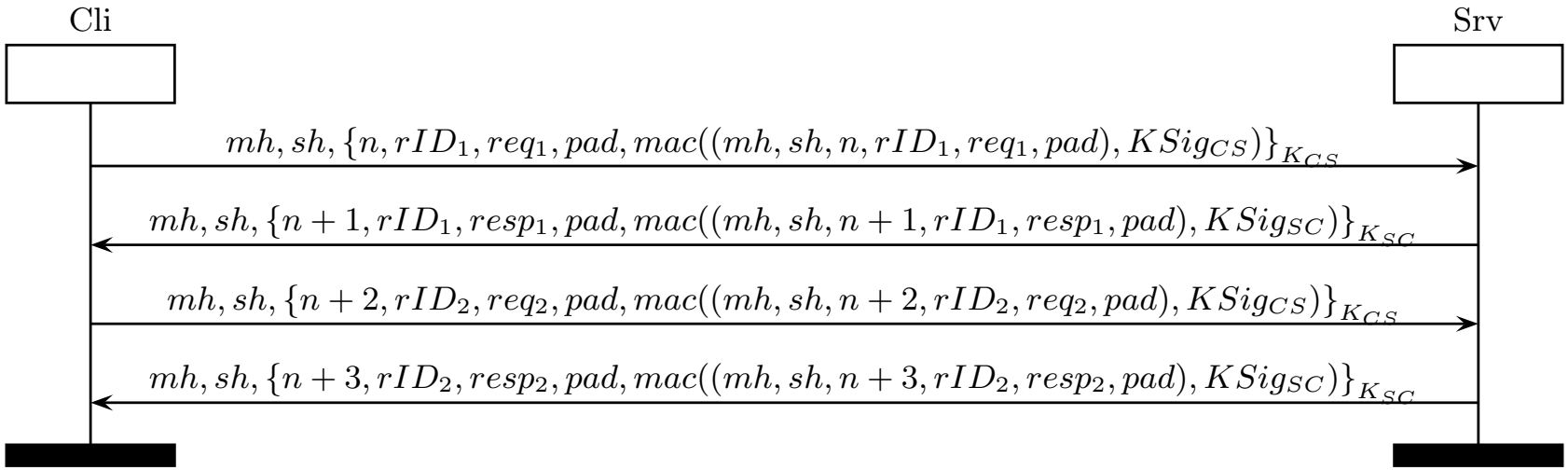


Textbook MODBUS [MOD04]



Secure MODBUS from [FCMT09]

OPC-UA



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

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

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

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

Attack on FA with bounded counters (modulo 4)

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

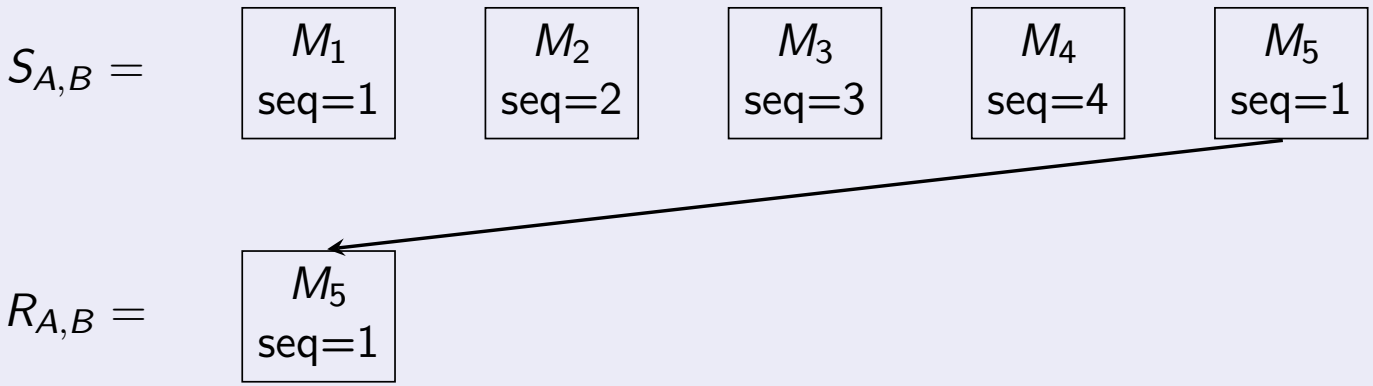
$R_{A,B} =$

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

Attack on FA with bounded counters (modulo 4)

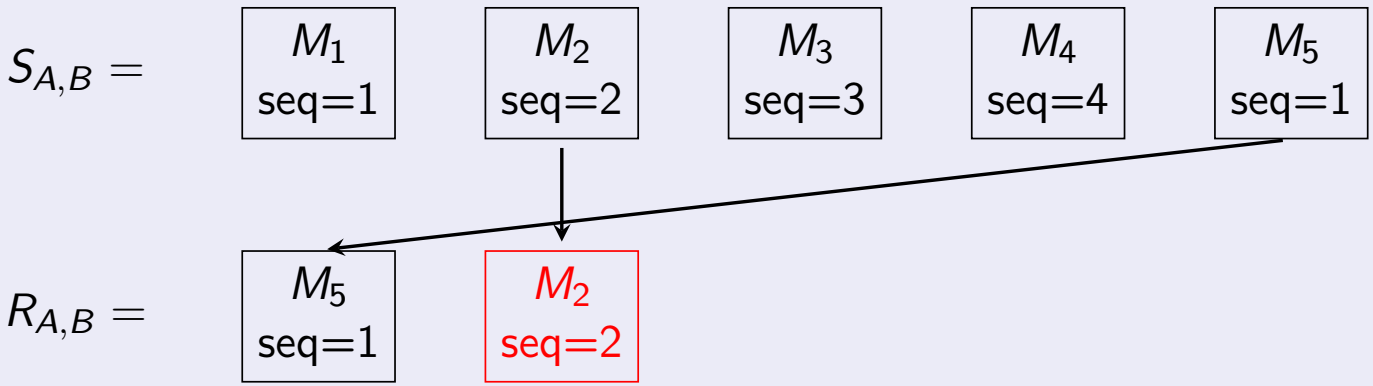


Results on OPC-UA with bounded counters

- 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

Attack on FA with bounded counters (modulo 4)

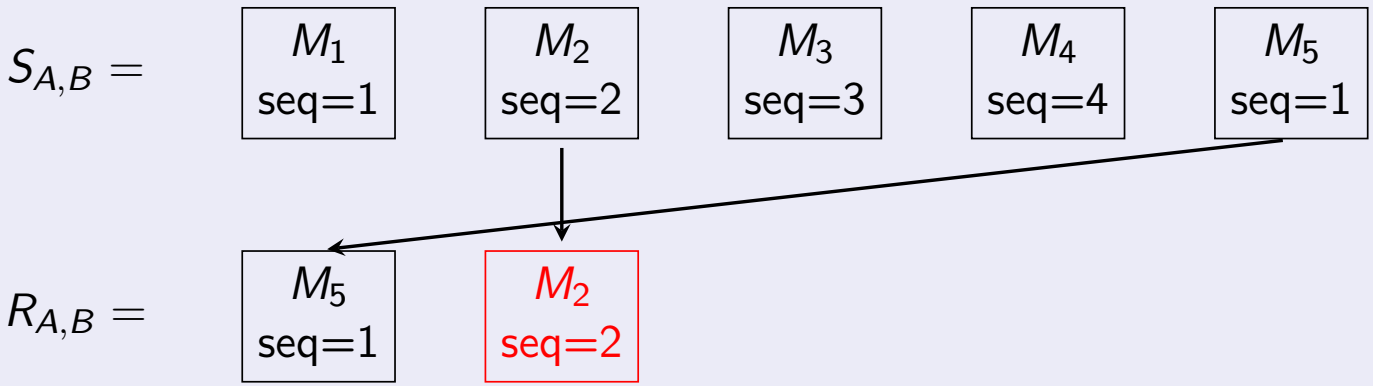


Results on OPC-UA with bounded counters

- 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

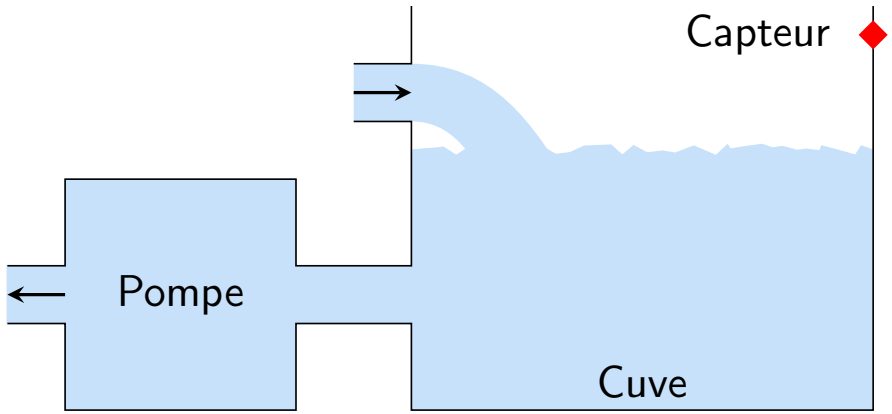
Attack on FA with bounded counters (modulo 4)



- Paper [DPPLR17] coined by OPC Fundation (that develops OPCUA):
  - interactions to understand attacks;
  - exchanges on the evaluation of CVSS score
- to appear: erratum on standard clarifying recommandation.
  - In practice, OPC-UA renegociates keys when sequence numbers wrap.

Back to the Common Thread: Maroochy Shire

- **Vulnerability 1:** Absence of authentication mechanism in communication protocols.

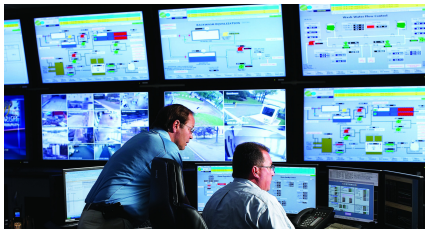


Methodology to catch properties required by industrial protocols.
Proofs of security for OPC-UA:

$\Rightarrow$ Provides authentication and integrity.

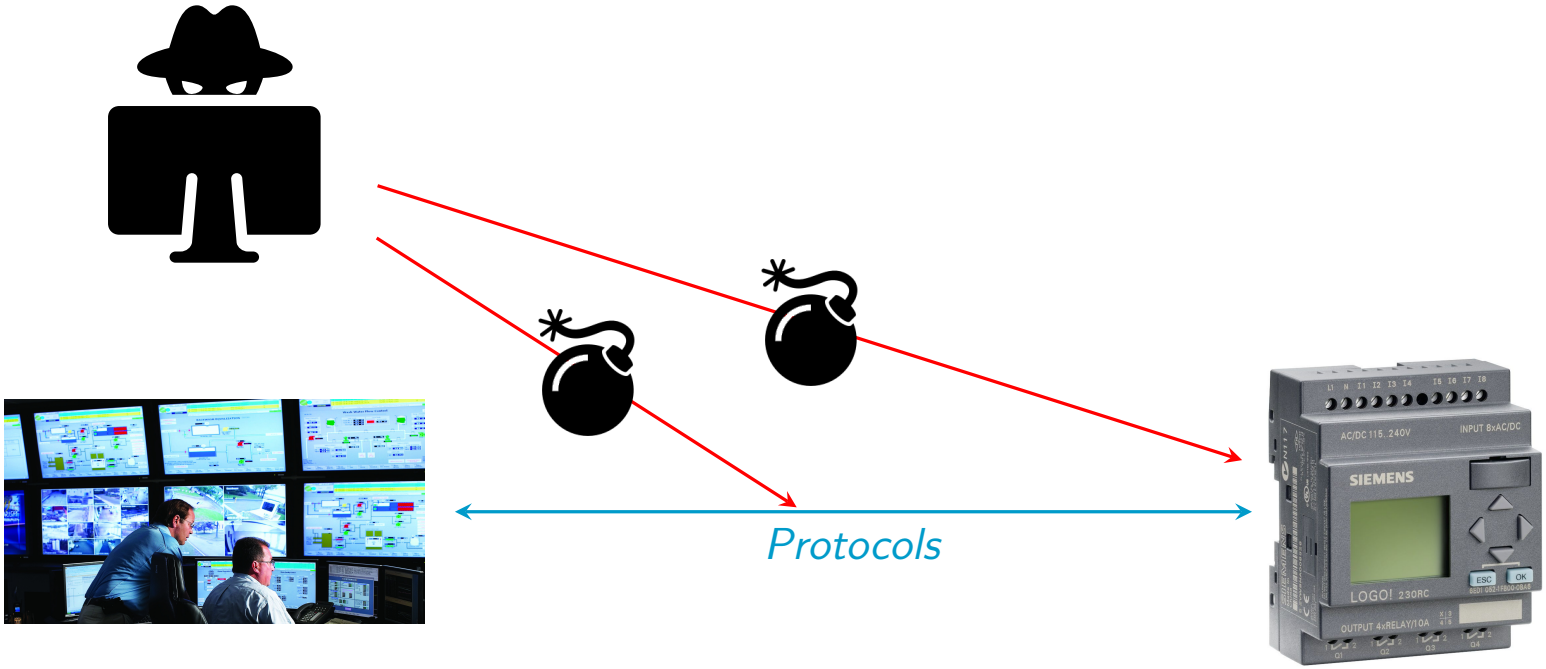
Table of Contents

Content Integrity by Applicative Filtering

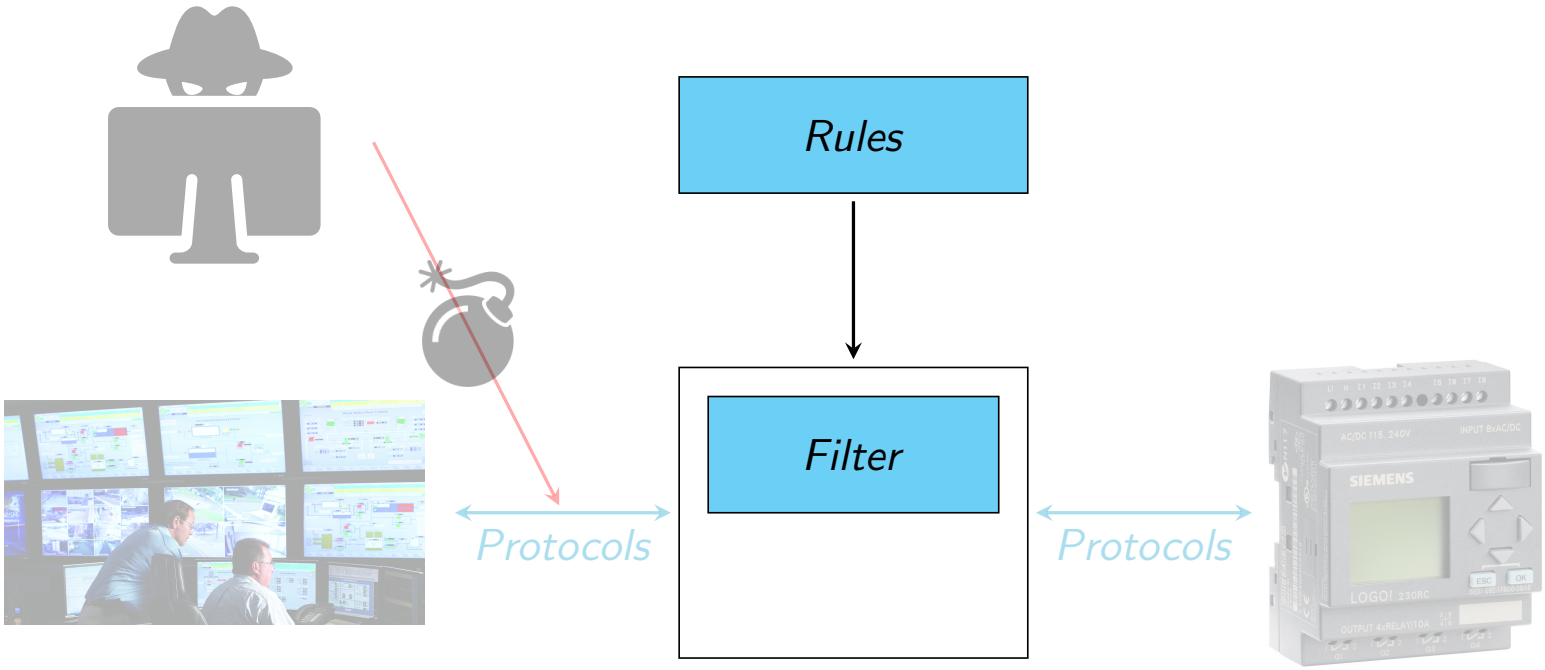


Protocols

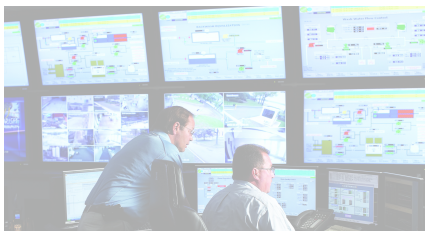
Content Integrity by Applicative Filtering

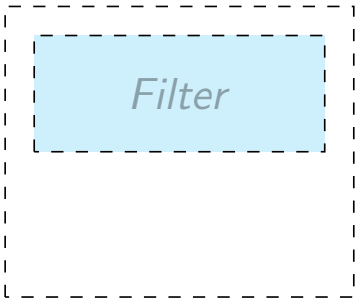


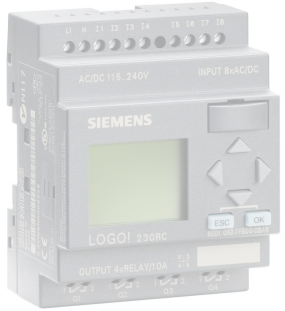
Content Integrity by Applicative Filtering



Content Integrity by Applicative Filtering





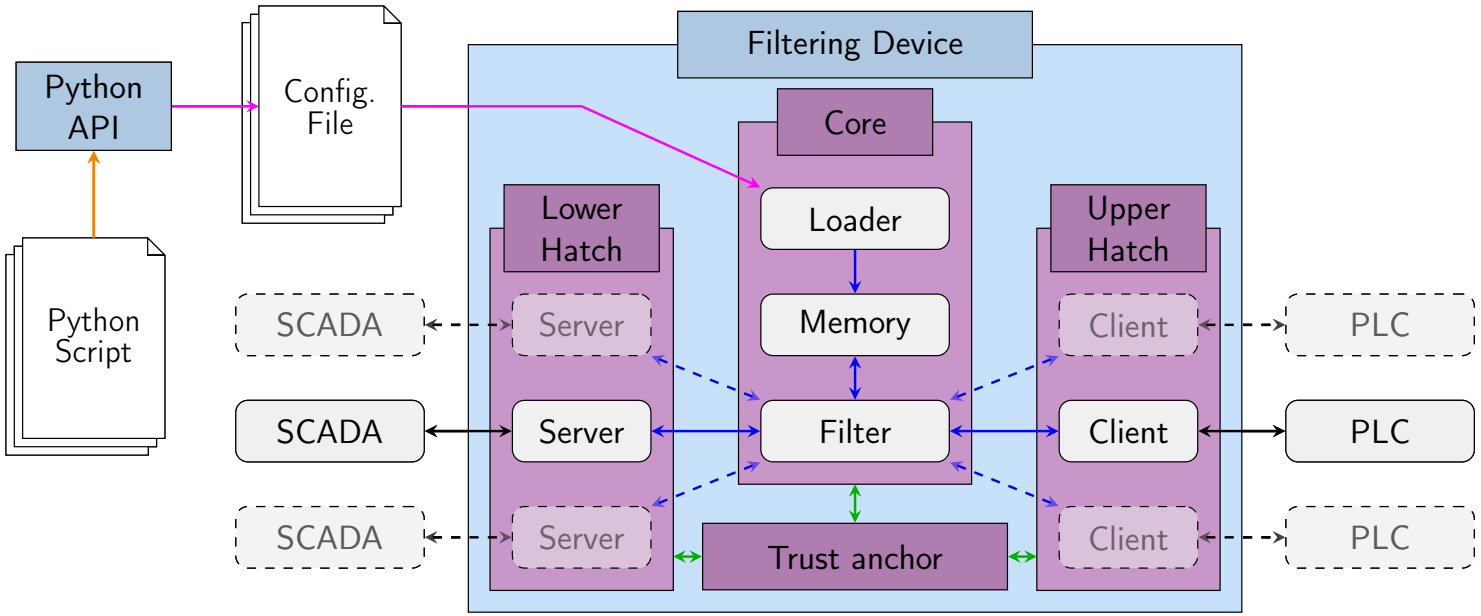


ARAMIS : Applicative Filtering Device

France PIA project lead by Atos Worldgrid, supervised by ANSSI.

Partners: Atos, CEA, Seclab, University Grenoble Alpes

Objective: A transparent device to disrupt and **filter industrial flows**.



[WCICSS'17] B. Badrignans *et al.* Security Architecture for Embedded Point-to-Points Splitting Protocols, 2017.

Rules Example

Stateless rules (e.g.: access control, permissions, values written).

Domain specific **stateful** rules:

- Temporal rules (e.g.: not receive more than 1 command per minute).
- Global process state (e.g.: pump must not be stopped if tank is full).

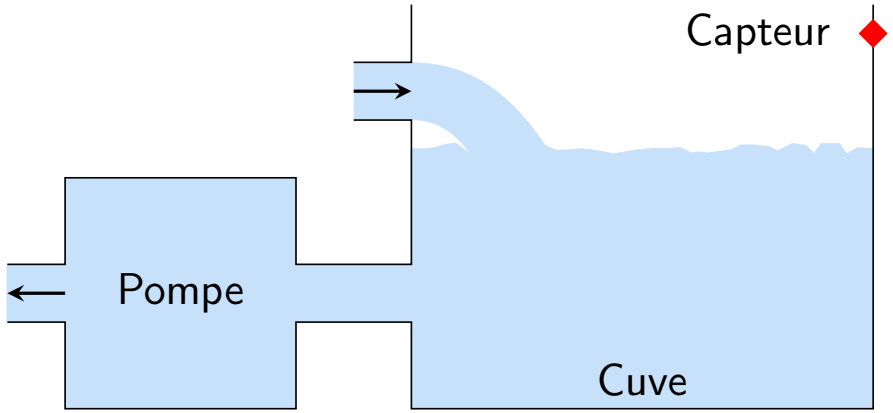
Case studies on real life examples:

- Demonstration of a prototype showed to ANSSI.

[CRITIS'16] M. Puys, J.-L. Roch, and M.-L. Potet. Domain specific stateful filtering with worst-case bandwidth, 2016.

Back to the Common Thread: Maroochy Shire

- **Vulnerability 2:** Absence of safety mechanism to avoid the spill.




```
rule = filter.Filter(chan, pumpState, filtre.Service.W
rule.addSubRule(
    condition=filter.And(
        filter.Equal(captor.currentValue, 1),
        filter.Equal(filter.NewValue(), 0)
    ),
    thenActions=filter.Reject("Tank full!")
)
```

Conclusion and Perspectives

- Industrial protocols need security proofs
  - Integrity is critical
- Flow integrity : formal verificaton
  - OPCUA protocol with Tamarin
- Content integrity : on-line verification
  - Both stateless and stateful verifications
- Perspective: Process integrity
  ↪verification that commands have been performed
  - Secure by Design (isolated system)
  - Secure by Proof of Results (eg interactive proof)
  - Secure by Proof of Consensus (eg blockchain)

Conclusion and Perspectives

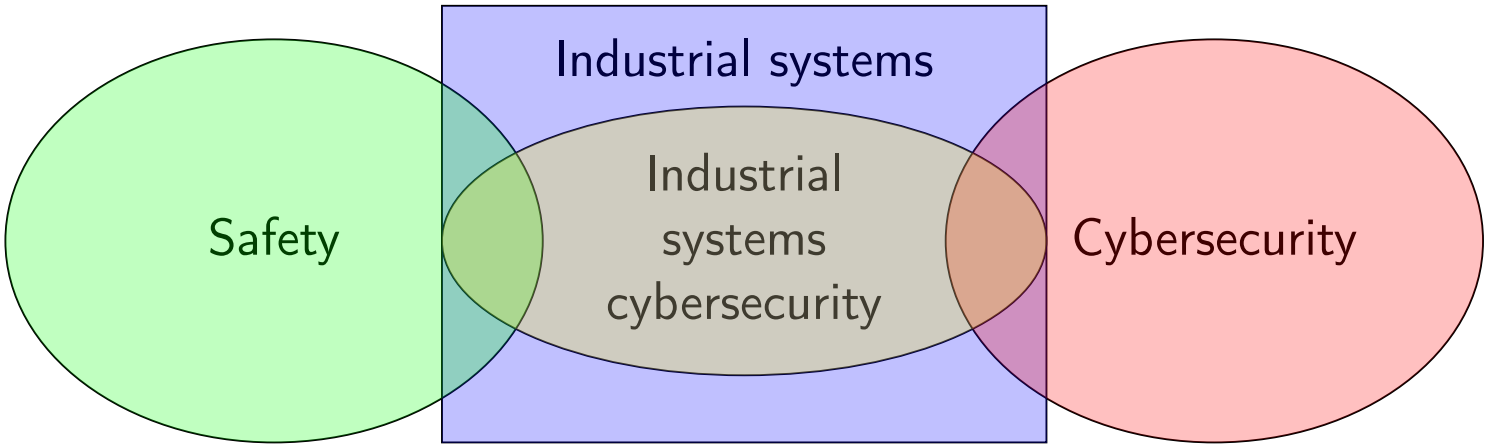
Thanks for your attention!

Jean-Louis Roch
Jean-Louis.Roch@grenoble-inp.fr

Disambiguation

Security concepts

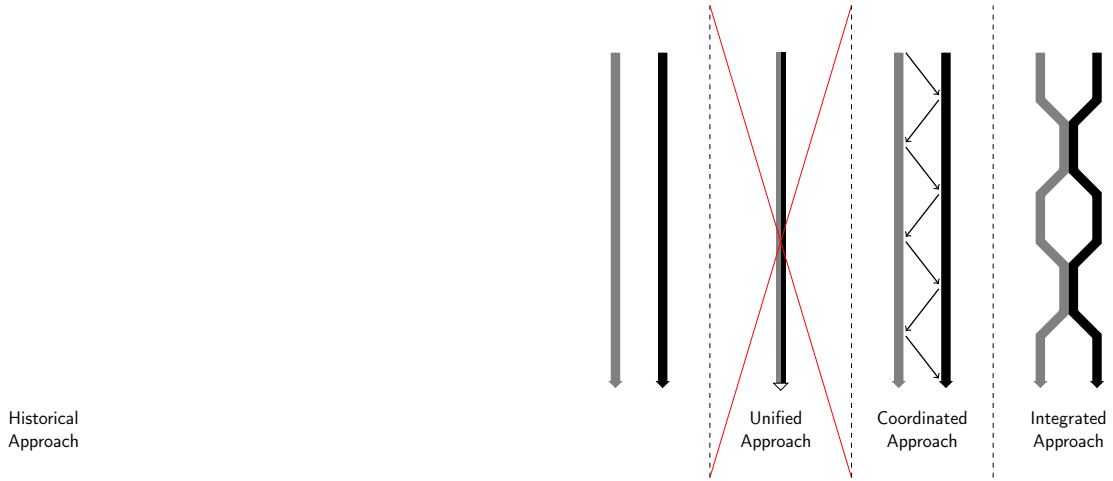
- Safety = Protection against identified/natural difficulties.
  - Historic industrial concern.
- Cybersecurity = Protection against malicious adversaries.
  - Often called Security.



Relations among security concepts

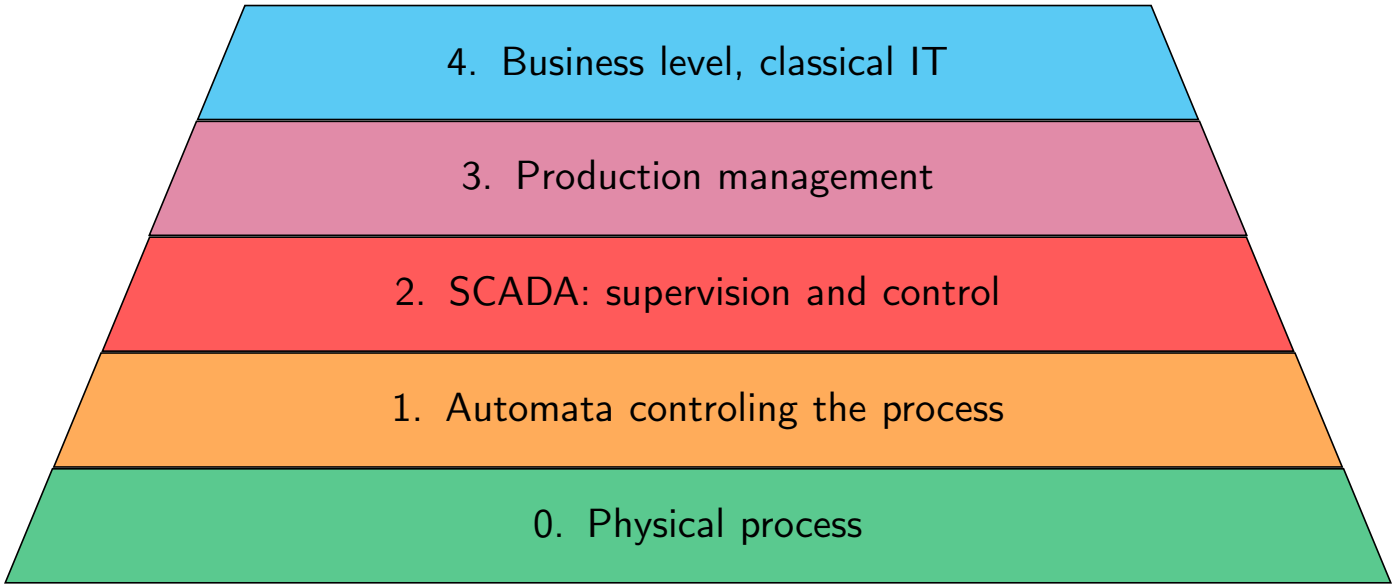
- Ludovic Pietre-Cambacedes' thesis: On the relationships between safety and security, Telecom ParisTech and EDF, 2010.

Safety and Security



How to link safety and security [PC10]

Purdue Model



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.

References I

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

References II

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).

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.

References III

- IEC-62541, *OPC Unified Architecture*, International Electrotechnical Commission, August 2015.
- MODBUS, *MODBUS IDA, MODBUS messaging on TCP/IP implementation guide v1.0a*, 2004.
- 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.

References IV


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.


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 V


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).