Cybersecurity of Industrial Systems: Applicative Filtering and Generation of Attack Scenarios

Sécurité des systèmes industriels : filtrage applicatif et recherche de scénarios d'attaques

Maxime Puys

Ph.D Advisors: Marie-Laure Potet and Jean-Louis Roch

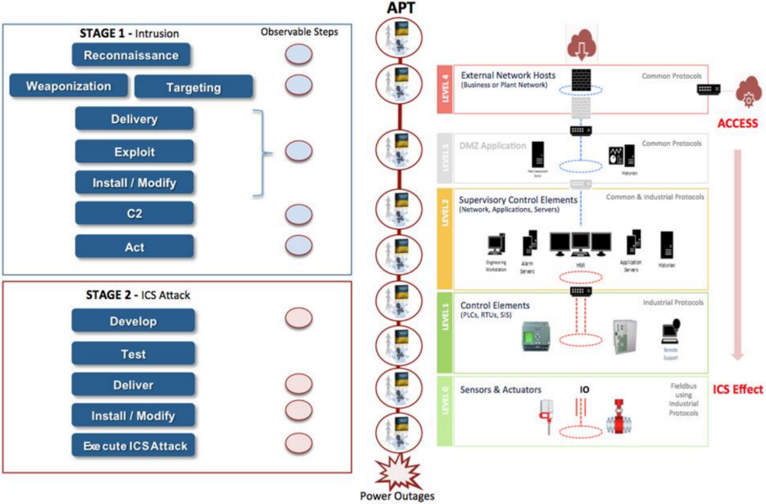
VERIMAG, Université Grenoble Alpes / Grenoble-INP, France

February 5^{th}, 2018

This thesis has been funded by project PIA ARAMIS (P3342-146798).

Blackout in Ukrain [LAC16]

- Occurred on Dec. 23rd, 2015, lasted up to 6 hours.
- Approximately 225,000 customers impacted.



Advanced Persistent Threats and Industrial Systems

Definition (Wikipedia)

Set of **stealthy and continuous** computer hacking processes, often orchestrated by a person or persons **targeting a specific entity**.

- Critical infrastructures:
  - ⇒ Potentially important damages.
- Less aware of cybersecurity risks:
  - ⇒ Easier initial compromising, less defences.
- Legacy and proprietary (often customized) components:
  - ⇒ Wider attack surface.

Protection becoming a priority for governments

- **Laws** to ensure security (*Opérateurs d'Importance Vitale*).
- Documents from government agencies (e.g.: ANSSI in France).

Challenges for Industrial Systems Cybersecurity

Recently Targeted by Cyberattacks

Historically **isolated** from networks:

⇒ Secure by design.

Properties to be Ensured Differ from IT Systems

Industrial systems require mainly:

- Availability, integrity, authentication, **dependability**.
- No focus on confidentiality.

⇒ Security verification tools not always adapted.

Need to Combine Safety and Security

- Safety = Protection against **identified/natural difficulties**.
- Security = Protection against **malicious adversaries**.
  - ⇒ Independent, opposite, complementary [PC10].

Thesis Goal

Wide cyberattack surface:

- Vectors: social engineering, **networks**, mobile devices, softwares, etc.
- In case of networks, possible targeted OSI layers: physical, ..., **security, applicative**.

Thesis Goal

Wide cyberattack surface:

- Vectors: social engineering, **networks**, mobile devices, softwares, etc.
- In case of networks, possible targeted OSI layers: physical, ..., **security, applicative**.

Goal

Uncover or block **applicative** network attacks mainly exploiting communication protocol weaknesses.

Thesis Goal

Wide cyberattack surface:

- Vectors: social engineering, **networks**, mobile devices, softwares, etc.
- In case of networks, possible targeted OSI layers: physical, ..., **security, applicative**.

Goal

Uncover or block applicative network attacks mainly exploiting **communication protocol weaknesses**.

Thesis Goal

Wide cyberattack surface:

- Vectors: social engineering, **networks**, mobile devices, softwares, etc.
- In case of networks, possible targeted OSI layers: physical, ..., **security, applicative**.

Goal

Uncover or block applicative network attacks mainly exploiting **communication protocol weaknesses**.

$\Rightarrow$ Provide risk and vulnerability analyzes combining safety and security.

$\Rightarrow$ Provide verifications relying on formal methods.

Industrial Systems (ICS) Composition 1/2

SCADA

PLC

Process

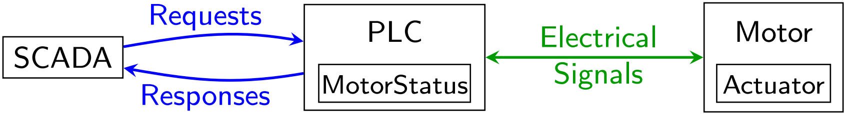
SCADA: Supervisory Control And Data Acquisition, controls and monitors the process.

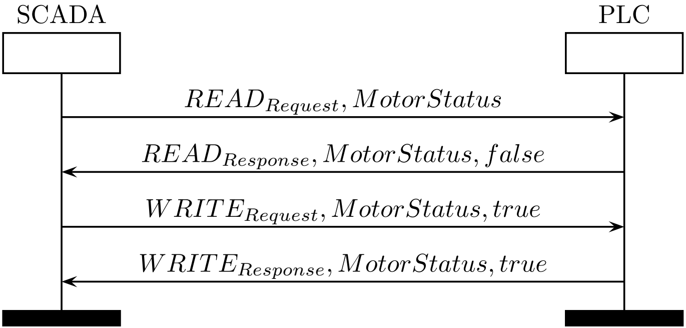
PLC: Programmable Logic Controller, interprets SCADA orders for the process.

Process: Actual industrial process managed by the system.

Industrial Systems (ICS) Composition 2/2

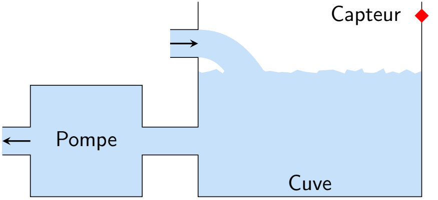
- Variables on PLC synchronized with process.
- Protocols used are specific (e.g., MODBUS, OPC-UA).





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.



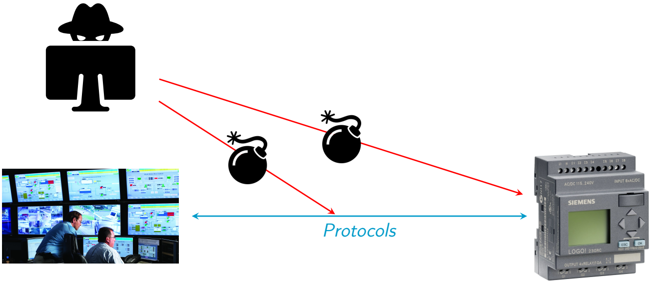
In our context, at least 3 vulnerabilities:

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

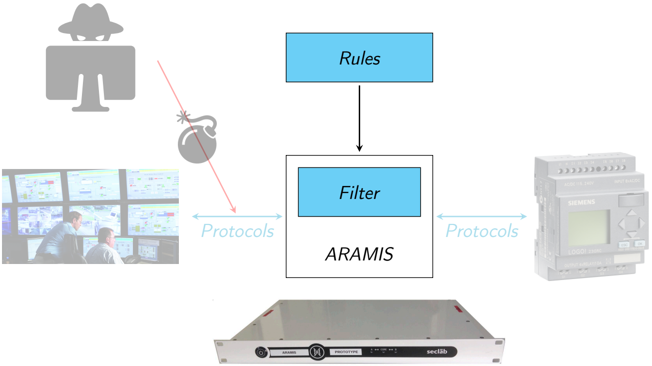
Overview of the Thesis

Protocols

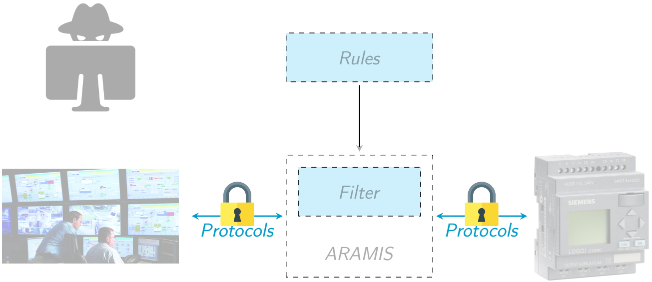
Overview of the Thesis



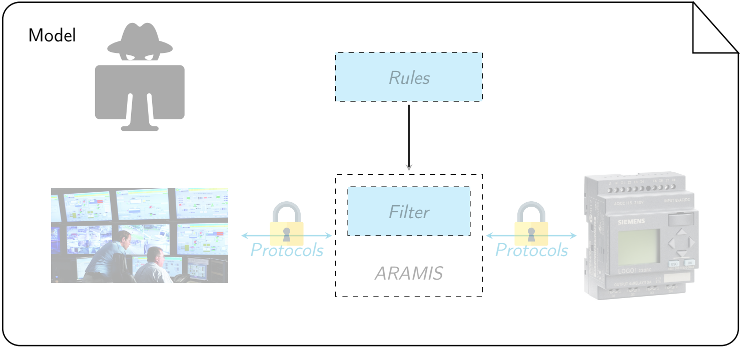
Overview of the Thesis: 1 – Applicative Filtering



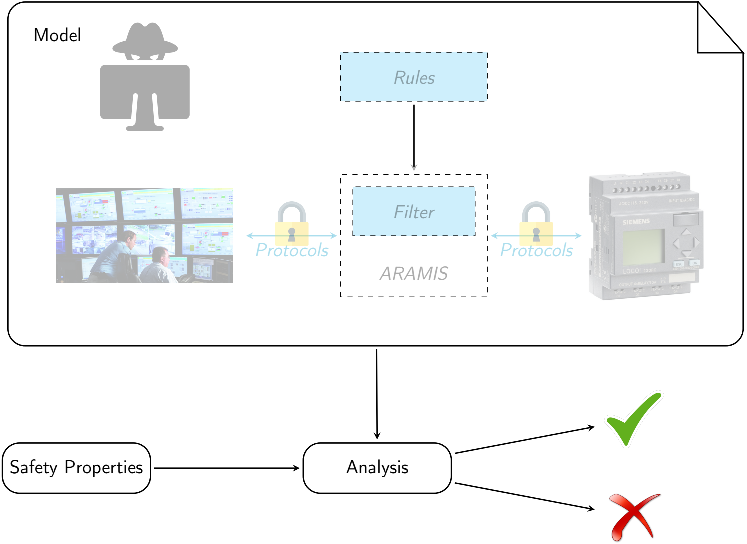
Overview of the Thesis: 2 – Protocol Verification



Overview of the Thesis: 3 – Attack Scenarios Generation



Overview of the Thesis: 3 – Attack Scenarios Generation



Contributions

Applicative Filtering for Industrial Systems

- Define and embed applicative filtering for industrial systems.

Formal Verification of Industrial Protocols

- Analysis of two sub-protocols of OPC-UA and integrity properties.

A^2SPICS: Attack Scenarios Generation

- Global approach to analyze safety properties in presence of attackers.
- Experimentations with multiple classes of verification tools.

Contributions

Applicative Filtering for Industrial Systems

- Define and embed applicative filtering for industrial systems.

Formal Verification of Industrial Protocols

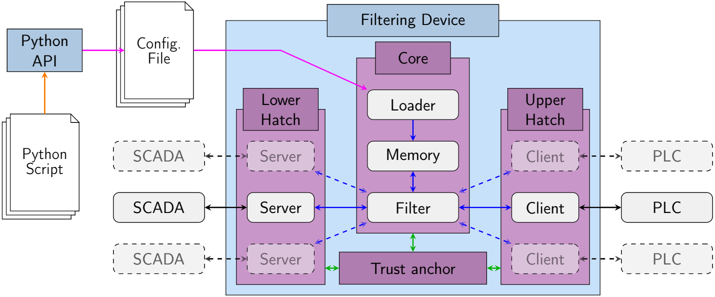
- Analysis of two sub-protocols of OPC-UA and integrity properties.

A^2SPICS: Attack Scenarios Generation

- Global Approach to analyze safety properties in presence of attackers.
- Experimentations with multiple classes of verification tools.

Applicative Filtering Device

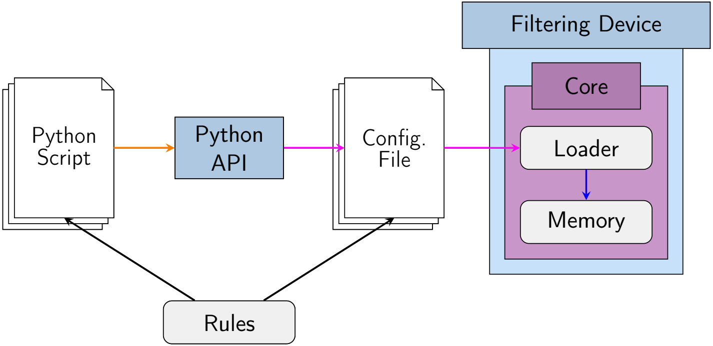
PIA project lead by Atos Worldgrid, supervised by ANSSI.
Objective: A transparent device to disrupt and **filter industrial flows**.



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

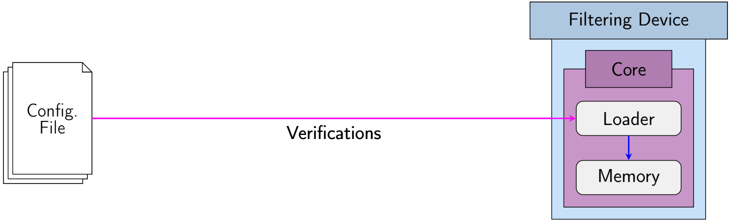
Rules Configuration

- Design of a language to specify rules.
- Filter acts as interpreter.
- Several requirements on functionalities, performances, security.



Rules Verifications

- Verifications on configuration file on loading:
  - Rules consistency.
  - Filter storage space (rules and process state).
  - Worst-case processing time for a message.



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).

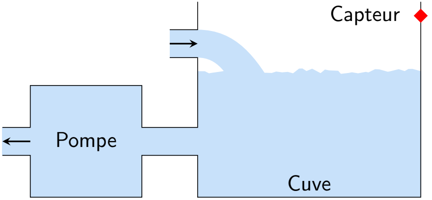
Case studies on real life examples:

- 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

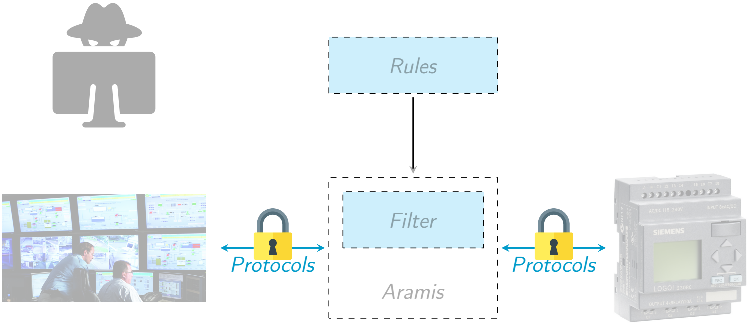
- **Vulnerability 1:** Absence of safety mechanism to avoid the spill.



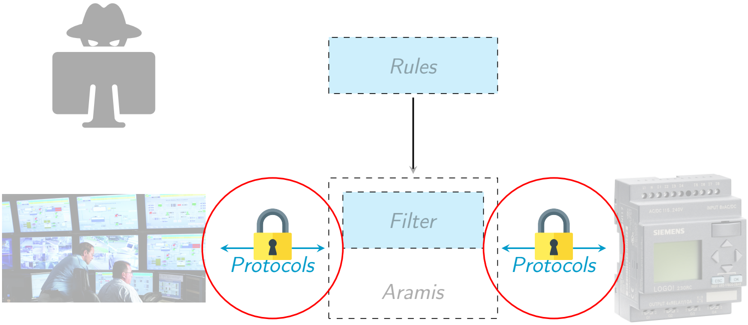

```
rule = filter.Filter(chan, pumpState, filtre.Service.WRITE)
rule.addSubRule(
    condition=filter.And(
        filter.Equal(captor.currentValue, 1),
        filter.Equal(filter.NewValue(), 0)
    ),
    thenActions=filter.Reject("Tank full!")
)
```

Formal Verification
of Industrial Protocols

Overview of the Thesis



Overview of the Thesis



Cryptographic Protocols Verification

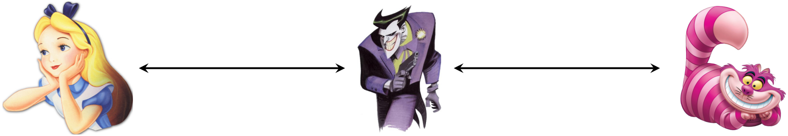
In Maroochy Shire attack, protocols provided no security against attackers:

$\Rightarrow$ Even when providing security feature, crucial to assess security.

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

- **Formally** verify the protocol **in presence of attacker** (Dolev-Yao).
- Check secrecy and authentication properties.

$\Rightarrow$ Not currently applied to industrial protocols.



Related Works on Analysis 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)

Related Works on Analysis 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)

Motivations on Studying OPC-UA Security

- Recent (2006), up to state-of-the-art, ongoing development.
- Probably **next standard** for industrial communications:
  - Designed by a consortium of key stakeholders.

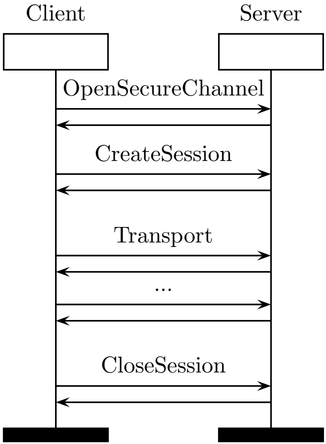
Official specifications: 1000 pages:

- Several terms redefined afterward.
- Highly context dependent.
  - $\Rightarrow$ Unclear on the use of some **security features**.

Idea: Models from the specifications.

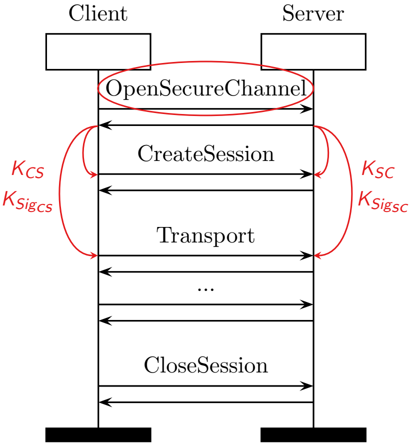
Details on OPC-UA

- **Handshake** protocol followed by **transport** protocol.
- Handshake composed of two sub-protocols.
- Expected security properties different for handshake and transport.



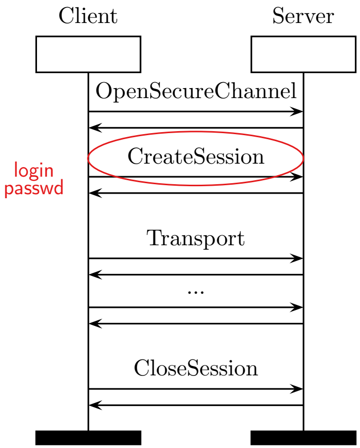
Details on OPC-UA

- **Handshake** protocol followed by **transport** protocol.
- Handshake composed of two sub-protocols.
- Secrecy of keys, authentication of agents.



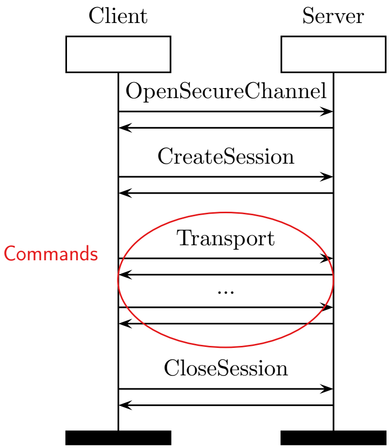
Details on OPC-UA

- **Handshake** protocol followed by **transport** protocol.
- Handshake composed of two sub-protocols.
- Secrecy and authentication on password.



Details on OPC-UA

- **Handshake** protocol followed by **transport** protocol.
- Handshake composed of two sub-protocols.
- Flow integrity of the commands.



OPC-UA Handshake Analysis

Two attacks found when security features are absent

Reuse of cryptographic signatures, password leaked.
Results communicated to OPC Foundation (specifications later clarified).

Challenges

Three possible security modes.
Combination of secure protocols may not be secure.

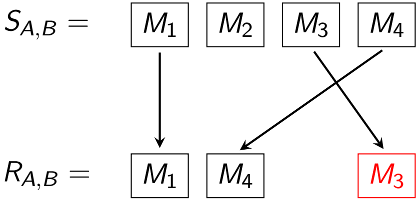
Modeling credentials with ProVerif

verifyCreds(pk(S), Login(pk(C)),Passwd(sk(C), pk(S))) = true.
User policy for password in models.

[Safecomp'16] M. Puys, M.-L. Potet, and P. Lafourcade, 2016.

OPC-UA Transport Analysis

Model properties required by (not limited to) industrial systems.



Check inclusion between $S_{A,B}$ and $R_{A,B}$:

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

[Secrypt'17] J. Dreier, M. Puys, M.-L. Potet, P. Lafourcade, and J.-L. Roch, 2017.

Flow Integrity Properties

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

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

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. »

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

Key Takeaways on Flow Integrity

Verification of MODBUS and OPC-UA

Protocol	MODBUS	[FCMT09]	[HEK13]	OPC-UA
Vulnerability	UNSAFE	UNSAFE	SAFE	SAFE

Challenges

In real life, machine integers are bounded and **wrap over**.
If so, all protocols are vulnerable.

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

$$R_{A,B} =$$

Key Takeaways on Flow Integrity

Verification of MODBUS and OPC-UA

Protocol	MODBUS	[FCMT09]	[HEK13]	OPC-UA
Vulnerability	UNSAFE	UNSAFE	SAFE	SAFE

Challenges

In real life, machine integers are bounded and **wrap over**.
If so, all protocols are vulnerable.

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

$R_{A,B} =$ | M_5 seq=1 |

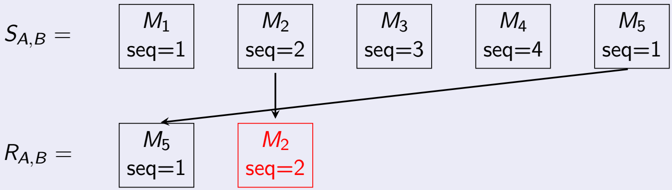
Key Takeaways on Flow Integrity

Verification of MODBUS and OPC-UA

Protocol	MODBUS	[FCMT09]	[HEK13]	OPC-UA
Vulnerability	UNSAFE	UNSAFE	SAFE	SAFE

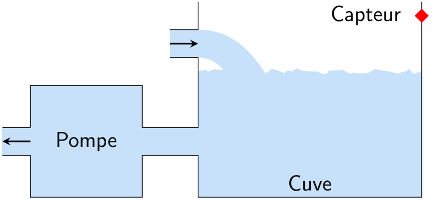
Challenges

In real life, machine integers are bounded and **wrap over**.
If so, all protocols are vulnerable.



Back to the Common Thread: Maroochy Shire

- **Vulnerability 2:** Absence of authentication mechanism in communication protocols.

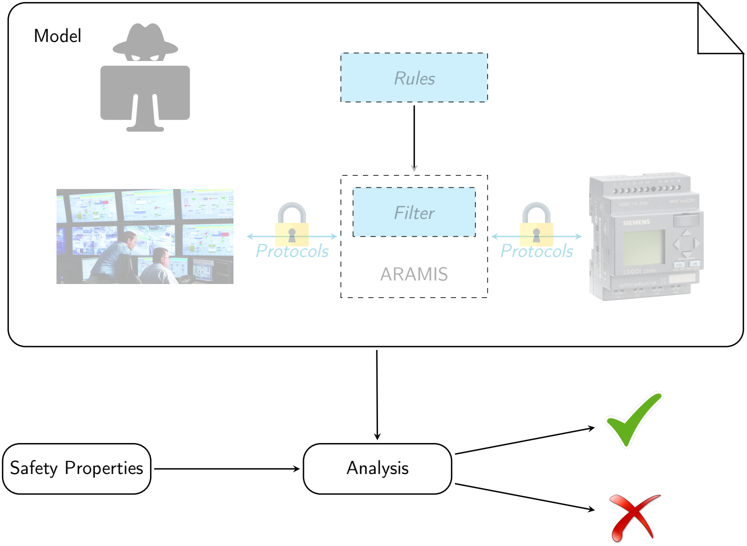


Methodology to catch properties required by industrial protocols.
Proofs of security for OPC-UA:

$\Rightarrow$ Provides authentication and integrity.

A^2SPICS: Attack Scenarios Generation

Overview of the Thesis



Idea

Effects of Maroochy Shire attack lasted several months, meaning no prevision of attacks.

A^2SPICS

- Analyze **safety properties in presence of attackers.**

  ⇒ Find **applicative attacks** on industrial systems.

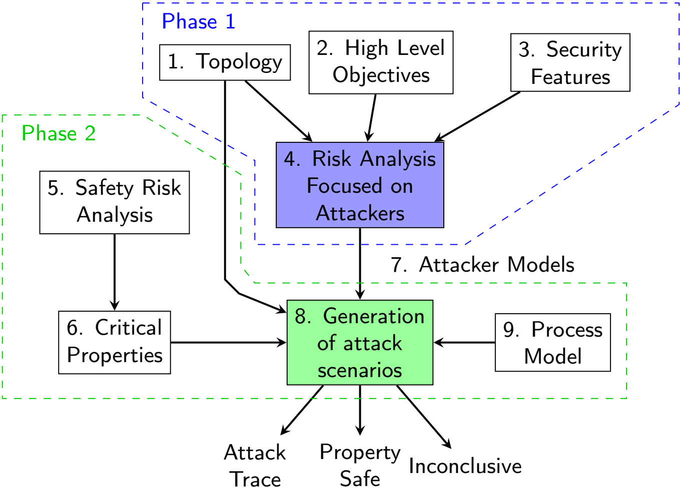
Tailored Attackers

- Attackers resulting of risk analyzes and protocol verification.

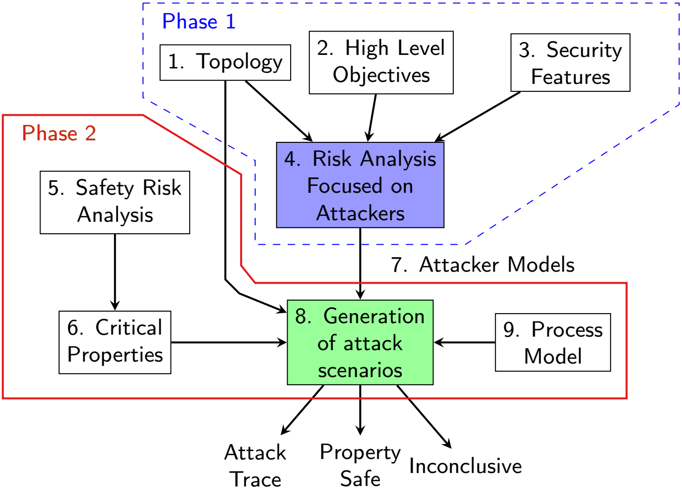
  ⇒ Apply only useful countermeasures.

[FPS'17] M. Puys, M.-L. Potet, and A. Khaled, 2016.

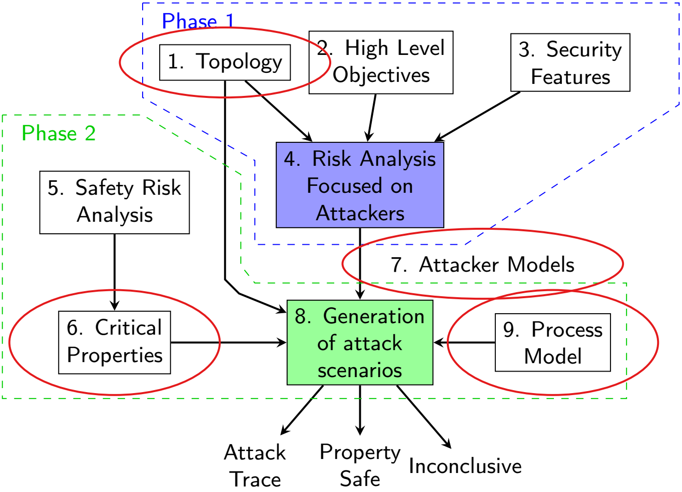
The A²SPICS Approach



The A^2SPICS Approach



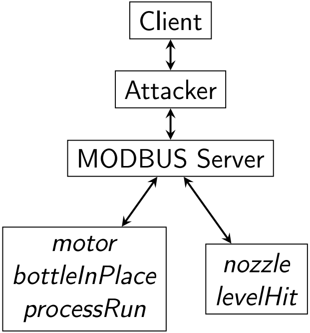
The A²SPICS Approach

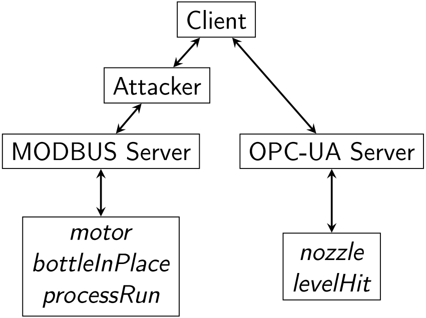


Topologies

Network topology of the system:

- **Communication channels** between components;
- **Position** of attackers.
  - ⇒ Impact on the variables they can attack.



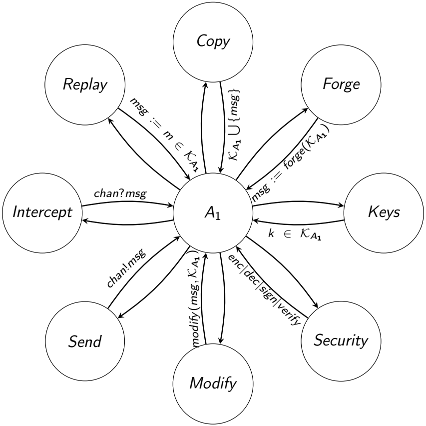


Attackers 1/2

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

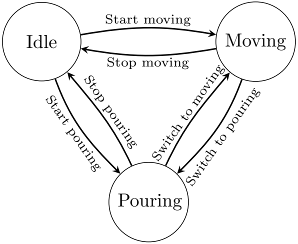


Four attackers shows as examples:

- A_1 = close to Dolev-Yao;
- A_2, A_3 and A_4 are subsets of A_1.
- In the global approach, **attacker models depending on risk analyzes**.

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

Behaviors and Safety Properties



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	$processRun = false$	$motor := false \wedge nozzle := false$

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: $A\Box\neg(nozzle = true \text{ and } motor = true)$

Instrumentation using Different Tools

Implementation of A^2SPICS using 3 different tools:

- UPPAAL, Uppsala University, Aalborg University [YPD94], 1994:
  - Model-checker.
  - Mainly designed for timed automata.

    $\Rightarrow$ **Safety** oriented verification tool.

- ProVerif, Inria [Bla01], 2001:
  - Protocol verification tool.

    $\Rightarrow$ **Security** oriented verification tool.
  - Relying on π-calculus and Horn clauses.

- Tamarin, ETH Zurich, Loria, Oxford Univeristy [SMCB12], 2012:
  - Protocol verification tool.

    $\Rightarrow$ **Security** oriented verification tool.
  - Relying on Maude-NPA rewriting tool.
  - Fine modeling of temporal properties.

Limitations and Difficulties

UPPAAL: Attacker behavior is too wide and deep

- Number of actions per attack is bounded (configurable, classical limitation of model-checking).

ProVerif: Very tedious state modeling

- Requires resilient channels, value enumeration, etc.

Tamarin: Impossible state modeling

- Backward search loops if behaviors have cycles.

Key Takeaways

- UPPAAL: Enhance attacker model.
- Protocol Verification Tools: Not adapted at the moment.

Related Works

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

Ref.	Type	Focus	Process model	Probabilistic	Automated
[BFM04]	Model	Atk	No	No	No
[MBFB06]	Model	Atk	No	Yes	No
[PGR08]	Model	Atk	No	Yes	No
[TML10]	Model	Atk	No	Yes	Yes
[KBL15]	Model	Atk	No	Yes	Yes
[RT17]	Model	Atk,Goal	Yes	No	Yes
A^2SPICS	Model	Atk,Goal	Yes	No	Yes

- [RT17] 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**.

Related Works

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

Ref.	Type	Focus	Process model	Probabilistic	Automated
[BFM04]	**Model**	**Atk**	No	No	No
[MBFB06]	**Model**	**Atk**	No	Yes	No
[PGR08]	**Model**	**Atk**	No	Yes	No
[TML10]	**Model**	**Atk**	No	Yes	Yes
[KBL15]	**Model**	**Atk**	No	Yes	Yes
[RT17]	Model	Atk,Goal	Yes	No	Yes
A^2SPICS	Model	Atk,Goal	Yes	No	Yes

- [RT17] 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**.

Related Works

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

Ref.	Type	Focus	Process model	Probabilistic	Automated
[BFM04]	Model	Atk	No	No	No
[MBFB06]	Model	Atk	No	Yes	No
[PGR08]	Model	Atk	No	Yes	No
[TML10]	Model	Atk	No	Yes	Yes
[KBL15]	Model	Atk	No	Yes	Yes
[RT17]	Model	Atk,Goal	Yes	No	Yes
A^2SPICS	Model	Atk,Goal	Yes	No	Yes

- [RT17] 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**.

Related Works

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

Ref.	Type	Focus	Process model	Probabilistic	Automated
[BFM04]	Model	Atk	No	No	No
[MBFB06]	Model	Atk	No	Yes	No
[PGR08]	Model	Atk	No	Yes	No
[TML10]	Model	Atk	No	Yes	Yes
[KBL15]	Model	Atk	No	Yes	Yes
[RT17]	Model	Atk,Goal	Yes	No	Yes
A^2SPICS	Model	Atk,Goal	Yes	No	Yes

- [RT17] 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**.

Related Works

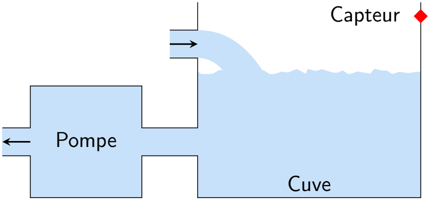
- Survey on assessment of security in industrial system ([PCB13, KPCBH15, CBB^+15]).
- Comparison criteria from [KPCBH15, CBB^+15]:

Ref.	Type	Focus	Process model	Probabilistic	Automated
[BFM04]	Model	Atk	No	No	No
[MBFB06]	Model	Atk	No	Yes	No
[PGR08]	Model	Atk	No	Yes	No
[TML10]	Model	Atk	No	Yes	Yes
[KBL15]	Model	Atk	No	Yes	Yes
[RT17]	Model	Atk,Goal	Yes	No	Yes
A^2SPICS	Model	Atk,Goal	Yes	No	Yes

- [RT17] rely on Cl-Atse (protocol verification tool):
  - Dolev-Yao intruder $\Rightarrow$ less precise control on attacker capacities.
- A^2SPICS aims at modeling **attackers resulting on risk analysis**.

Back to the Common Thread: Maroochy Shire

- **Vulnerability 3:** Absence of prevision of attacks.



A^{2}SPICS allows to discover possible attack scenarios:

$\Rightarrow$ Counter-measures could have been installed.

Conclusion

Contributions & Perspectives: Applicative Filtering

Applicative Filtering for Industrial Systems

- Define and embed applicative filtering for industrial systems.

Hot Topic

- Segregation and filters are among most required security measures.

Handle expressiveness of recent protocols

- Method calls: simulate if method call violates rules ?
- Custom data structures ?
- Notifications

Contributions & Perspectives: Protocol Verification

Formal Verification of Industrial Protocols

- Analysis of two sub-protocols of OPC-UA and integrity properties.

Protocol Encapsulation

- E.g.: MODBUS through OPC-UA, shared keys, parts not encapsulated, etc.

Observational Equivalence

- Currently used for e-vote protocol, interesting for customer data.

Contributions & Perspectives: Attack Scenarios

A^{2}SPICS: Attack Scenarios Generation

- Global Approach to analyze safety properties in presence of attackers.
- Experimentations with multiple classes of verification tools.

Refine Model

- Attacker capacities depending on safety properties (only generate useful messages.)

Enhance Model

- Attacker collusions, resilience properties.

Deeper Combining Contributions

Joint Use of Contributions

- Test filtering device using A^2SPICS method.
- Include protocol modeling in A^2SPICS method.

Transversal View of Cybersecurity

- Focus on multiple linked security mechanisms:
  $\Rightarrow$ Idea of **defence in depth**.

Conclusion

Thanks for your attention!

Applicative Filtering for Industrial Systems:

- B. Badrignans, V. Danjean, J.-G. Dumas, P. Elbaz-Vincent, S. Machenaud, J.-B. Orfila, F. Pebay-Peyroula, F. Pebay-Peyroula, M.-L. Potet, M. Puys, J.-L. Richier, and J.-L. Roch. Security Architecture for Embedded Point-to-Points Splitting Protocols. WCICSS'17, 2017.
- M. Puys, J.-L. Roch, and M.-L. Potet. Domain specific stateful filtering with worst-case bandwidth. CRITIS'16, 2016

Industrial Protocol Verification:

- J. Dreier, M. Puys, M.-L. Potet, P. Lafourcade, and J.-L. Roch. Formally verifying flow integrity properties in industrial systems. SECRYPT'17, 2017. Best Student Paper Award.
- M. Puys, M.-L. Potet, and P. Lafourcade. Formal analysis of security properties on the OPC-UA SCADA protocol. SAFECOMP'16, 2016.

A^{2}SPICS– Attack Scenarios Generation:

- M. Puys, M.-L. Potet, and A. Khaled. Generation of applicative attacks scenarios against industrial systems. FPS'17, 2017.
- M. Puys, M.-L. Potet, and J.-L. Roch. Génération systématique de scénarios d'attaques contre des systèmes industriels. AFADL'16, 2016

Other topics:

- J.-G. Dumas, P. Lafourcade, J.-B. Orfila, and M. Puys. Dual protocols for private multi-party matrix multiplication and trust computations. Computers & Security, 2017.
- J.-G. Dumas, P. Lafourcade, J.-B. Orfila, and M. Puys. Private multi-party matrix multiplication and trust computations. SECRYPT'16, 2016. Best Paper Award.
- P. Lafourcade and M. Puys. Performance evaluations of cryptographic protocols verification tools dealing with algebraic properties. FPS'15, 2015.

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–.

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.

References II

- 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.
- 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.
- 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 III

- Jérémie Guiochet and David Powell, *Etude et analyse de différents dispositifs externes de sécurité-innocuité de type safety bag*, Ph.D. thesis, LAAS-CNRS, 2005.
- 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 IV

- 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.
- Robert M Lee, Michael J Assante, and Tim Conway, *Analysis of the cyber attack on the ukrainian power grid*, SANS Industrial Control Systems (2016).
- 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.

References V

- 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 VI

- 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 VII

- 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.
- B. Schmidt, S. Meier, C. Cremers, and D. Basin, *Automated analysis of diffie-hellman protocols and advanced security properties*, Computer Security Foundations Symposium (CSF), 2012 IEEE 25th, June 2012, pp. 78–94.
- 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.

References VIII

- 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.
- 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).
- Wang Yi, Paul Pettersson, and Mats Daniels, *Automatic Verification of Real-Time Communicating Systems By Constraint-Solving*, Proc. of the 7*th* Int. Conf. on Formal Description Techniques (Dieter Hogrefe and Stefan Leue, eds.), North–Holland, 1994, pp. 223–238.

TOC

Industrial Systems are Ubiquitous

Electricity

Water Treatment

Chemistry

Industrial Systems are Ubiquitous

Electricity

Water Treatment

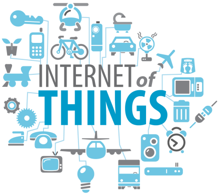
Chemistry

Food Production

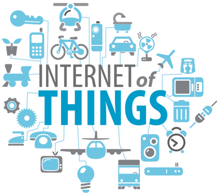
Transportation

Healthcare

Industrial Internet of Things



Industrial Internet of Things



Rio Tinto Mine, Australia

Oil Platform, North Sea

« Smart » Buildings

Autonomous Industrial Systems

Purdue Model

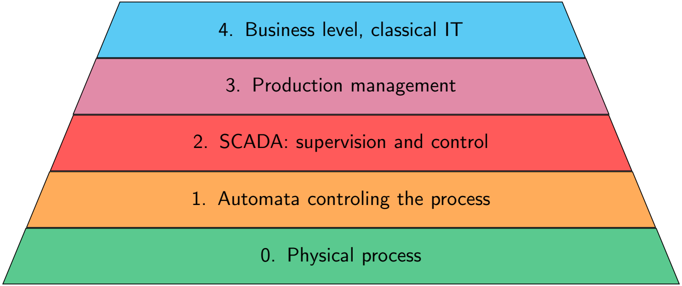


Figure : Purdue model [Wil91]

Norms and Guides on Industrial Systems Security

Generic

ISA-99/IEC-62443 (2007, 2013), ENISA (2011), ISO-27019 (2013), IEC-62541 (2015), etc.

Government Agency

CPNI (2008), BSI (2009), NIST (2011), ANSSI (2012), etc.

Domain Specific

Oil/Gaz (AGA, 2006), Electricity (IEC-62351, 2007]), Nuclear (IEC-62645, 2008), Air Traffic (CSFI, 2015), Railways (RSSB, 2016), etc.

Key Takeaways

$\Rightarrow$ Lots of documents, mainly released since 2006. Balanced partition between industry and governments, often in collaboration.

Properties to Ensure

For the process

Availability: System keeps running.

Integrity: Preservation of the coherence of a data over time.

Authenticity: An entity is who he/she pretends.

Non-repudiation: One cannot deny its actions.

Dependability: Domain specific properties.

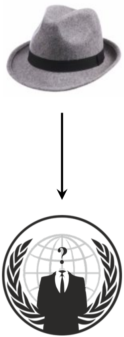
For customer data

Confidentiality: Only authorized entities can access designated data.

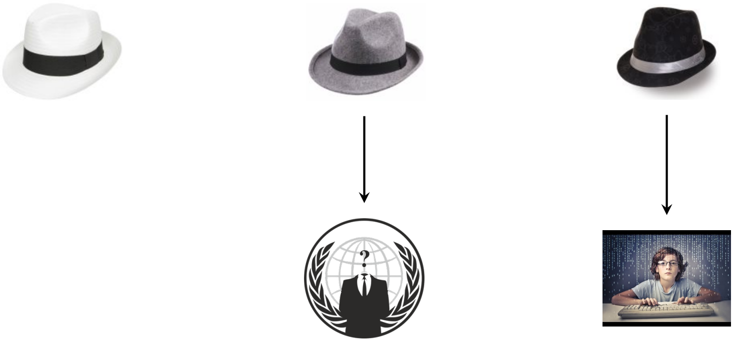
Anonymity: Prevent linking a data with its owner.

Different Attackers

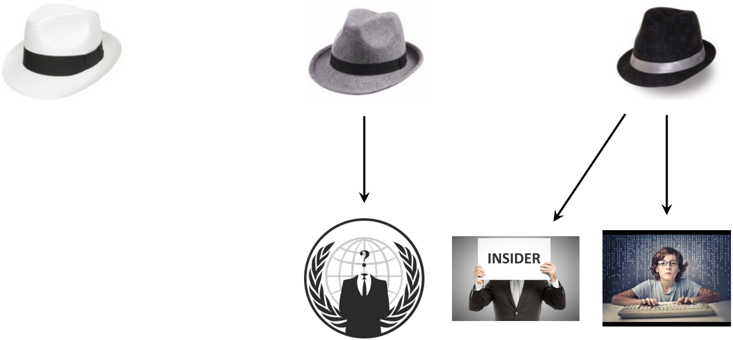
Different Attackers



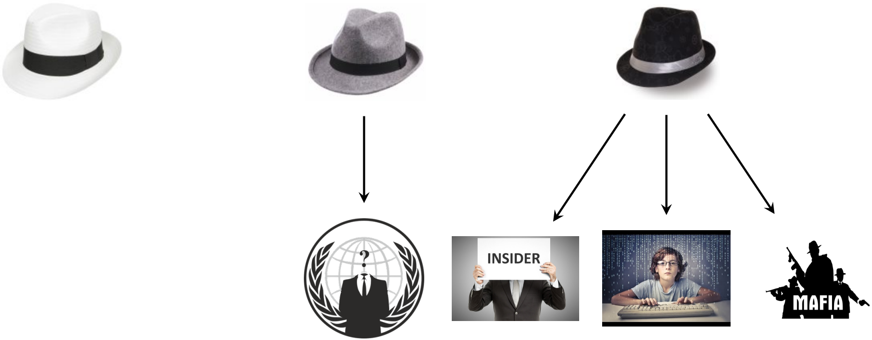
Different Attackers



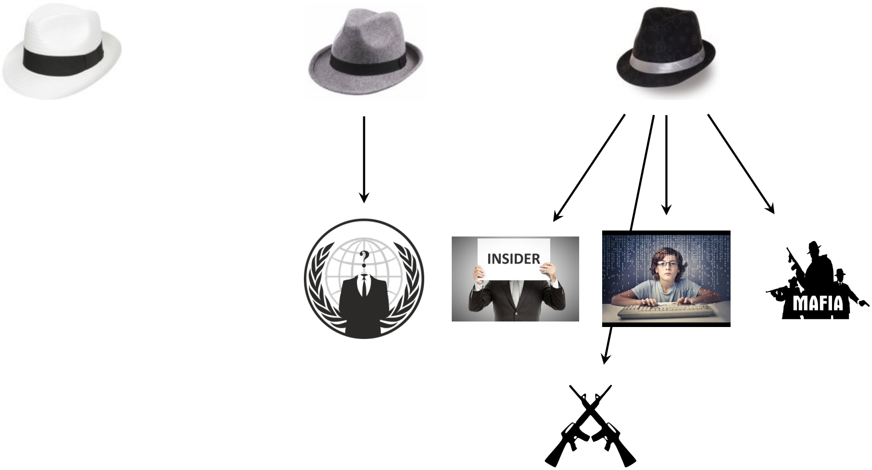
Different Attackers



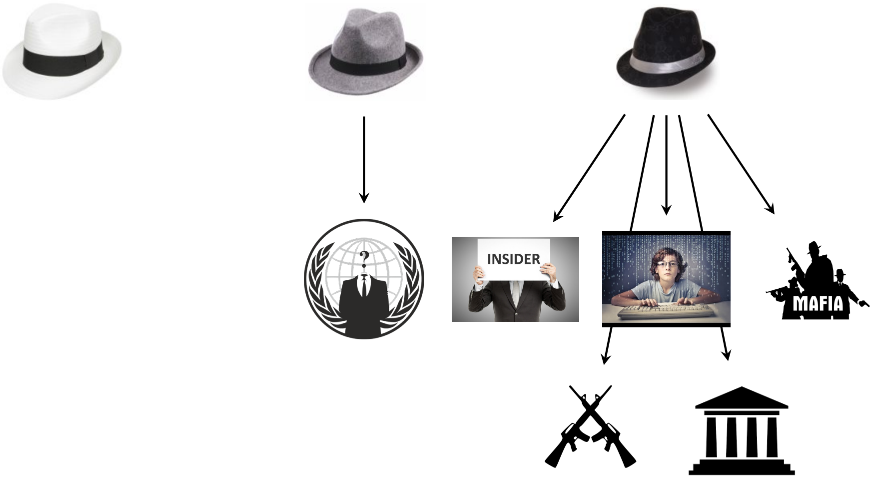
Different Attackers



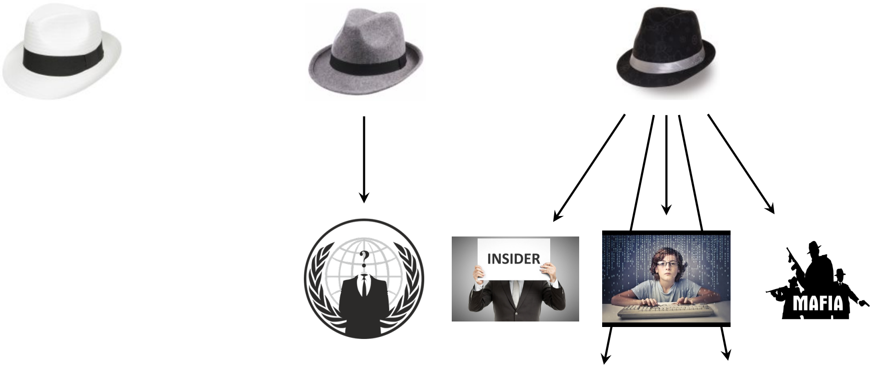
Different Attackers

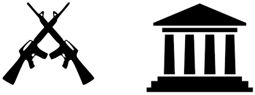


Different Attackers



Different Attackers





Various Profiles

Different objectives, skills, money and human resources.

Open Challenges

Availability Requirement

- Rising concern with IOT (DynDNS attack, 2016).
- Also a requirement for IT systems.
- Yet among most important requirements for industrial systems.

Software Updates/Patches

- Applying patches often requires to stop/reboot system.
- How to ensure backwards compatibility.
  - Much more easier for IT systems (e.g.: virtualization).

Skill Transfers from Academia to Industry

- Strong bonds with industry through ARAMIS.
- Also thanks to projects PEPS CNRS ASSI and ASTRID SACADE.

Worst-Case Bandwidth

Both conditions and actions have to be processed in constant time:

- Conditions are O(1) boolean predicates.
- Actions are : (i) Block or transmit the message, (ii) Log information, (iii) Update a local variable,

Thus processing one command only depends on the number of rules:

- For all predicates P, worst case processing time T of a message is $T = \sum \tau_i n_i$
- With τ_i the processing time of predicate P_i
- And n_i number of occurrences of predicate P_i

In practice, as only relevant rules are tested for a message.
Worst-case happens for an accepted message.

Open Secure Channel Sub-Protocol

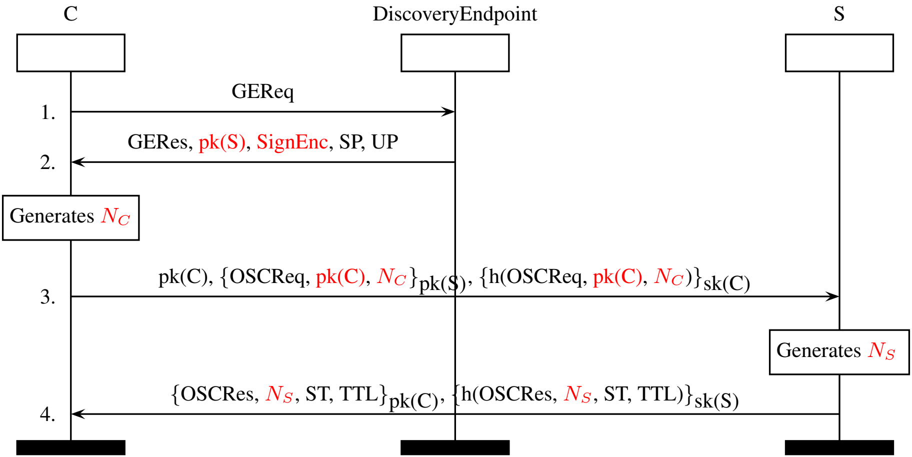


Figure : OPC-UA OpenSecureChannel

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.

Attack on Authentication on N_C in SignAndEncrypt

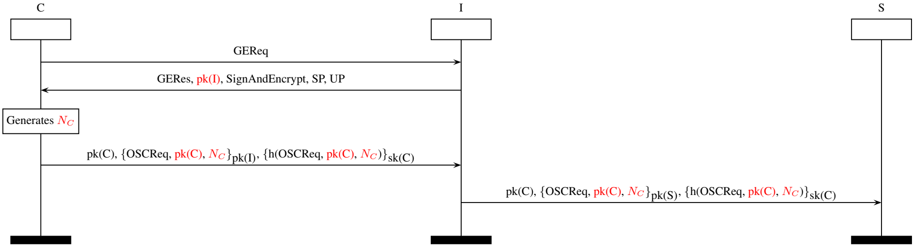


Figure : Attack on OPC-UA OpenSecureChannel

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

Create Session Sub-Protocol

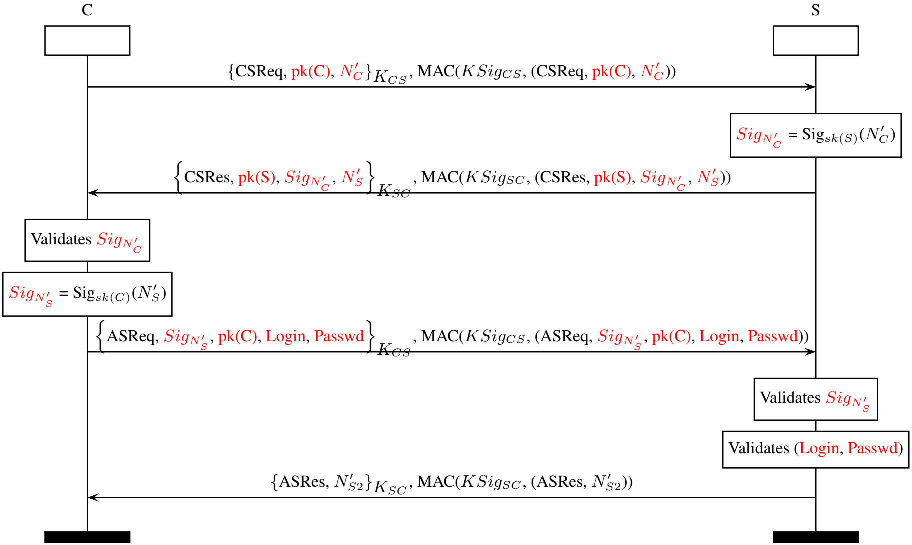


Figure : OPC-UA CreateSession

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

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

$$R_{A,B} =$$

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

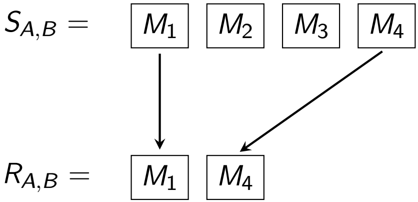
$$S_{A,B} = \quad \boxed{M_1}\ \boxed{M_2}\ \boxed{M_3}\ \boxed{M_4}$$

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

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

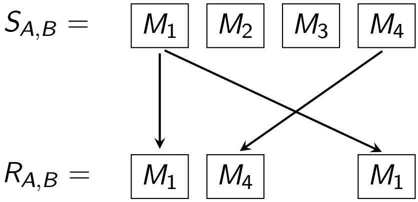


Non-Injective Message Authenticity (NIMA)

Property

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

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

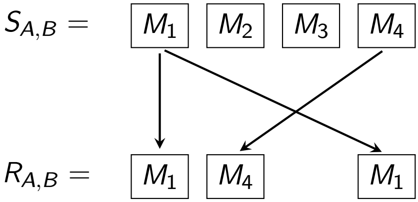


Non-Injective Message Authenticity (NIMA)

Property

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

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



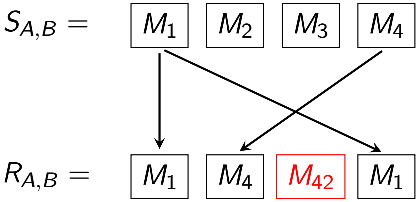
✓ NIMA verified

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



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

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

$$R_{A,B} = \boxed{M_1}\ \boxed{M_4}\ \boxed{M_{42}}\ \boxed{M_1}$$

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

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

$$R_{A,B} =$$

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

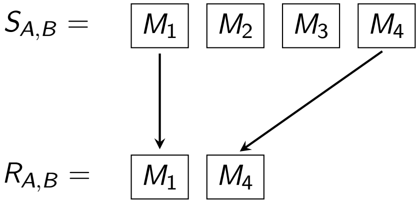
$$S_{A,B} = \quad \boxed{M_1}\ \boxed{M_2}\ \boxed{M_3}\ \boxed{M_4}$$

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

Injective Message Authenticity (IMA)

Property

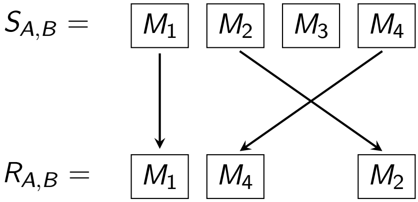
« **All messages received n times have been sent n times.** »
A protocol ensures Injective Message Authenticity (IMA) between sender A and receiver B if $\textbf{multiset}(R_{A,B}) \subseteq \textbf{multiset}(S_{A,B})$.



Injective Message Authenticity (IMA)

Property

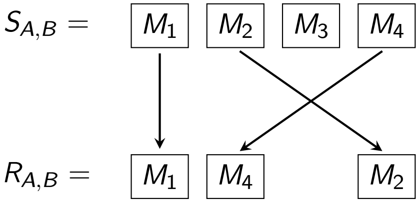
« **All messages received n times have been sent n times.** »
A protocol ensures Injective Message Authenticity (IMA) between sender A and receiver B if $\textbf{multiset}(R_{A,B}) \subseteq \textbf{multiset}(S_{A,B})$.



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

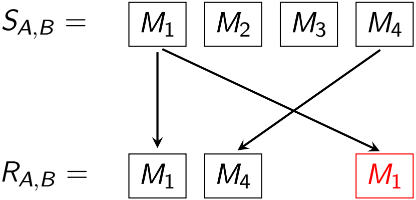


✓ IMA verified

Injective Message Authenticity (IMA)

Property

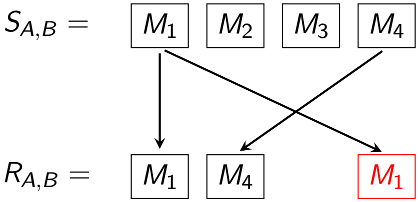
« **All messages received** n **times have been sent** n **times.** »
A protocol ensures Injective Message Authenticity (IMA) between sender A and receiver B if $\textbf{multiset}(R_{A,B}) \subseteq \textbf{multiset}(S_{A,B})$.



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



✗ 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 sub-chain 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 sub-chain of** $S_{A,B}$.

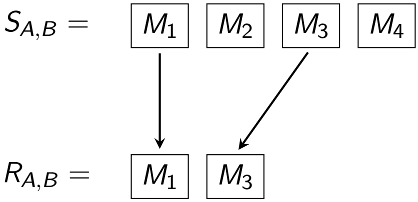
$$S_{A,B} = \quad \boxed{M_1}\ \boxed{M_2}\ \boxed{M_3}\ \boxed{M_4}$$

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

Flow Authenticity (FA)

Property

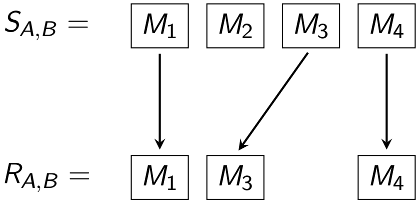
« **All messages are received in the order they have been sent.** »
A protocol ensures Flow Authenticity (FA) between sender A and receiver B if $R_{A,B}$ **is a sub-chain of** $S_{A,B}$.



Flow Authenticity (FA)

Property

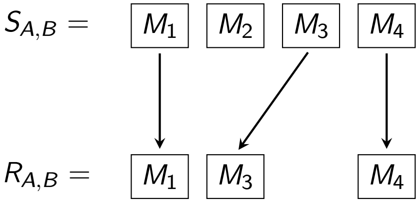
« **All messages are received in the order they have been sent.** »
A protocol ensures Flow Authenticity (FA) between sender A and receiver B if $R_{A,B}$ **is a sub-chain of** $S_{A,B}$.



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 sub-chain of** $S_{A,B}$.

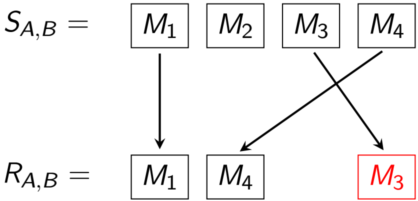


✓ FA verified

Flow Authenticity (FA)

Property

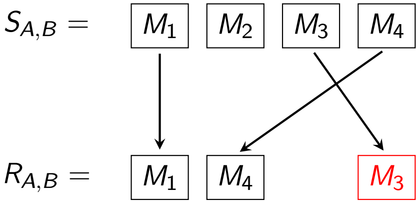
« **All messages are received in the order they have been sent.** »
A protocol ensures Flow Authenticity (FA) between sender A and receiver B if $R_{A,B}$ **is a sub-chain of** $S_{A,B}$.



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 sub-chain of** $S_{A,B}$.



✗ FA not verified

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 n times have been sent n times.* »

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

Flow Authenticity (FA)

Property

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

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

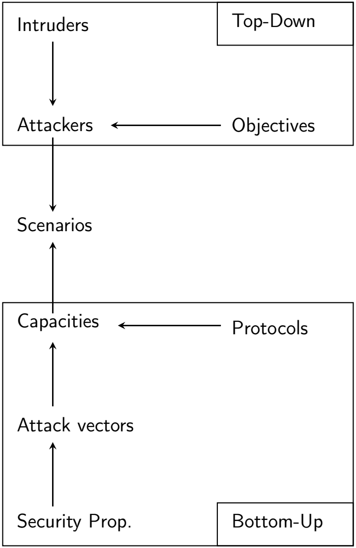
Resilient Channels

- 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).

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

Phase 1: Attacker Models

- Risk analysis focused on attackers.
- Based on:
  - **Topology** of the system;
  - Attacker **objectives**;
  - **Security properties** 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).



[AFADL'16] M. Puys, M.-L. Potet, and J.-L. Roch, 2016.

Top-Down Example

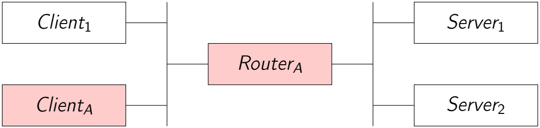


Figure : Infrastructure example

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 realization 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:

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

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.