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.

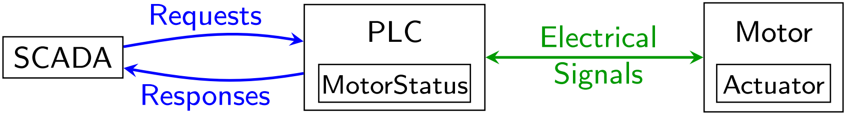
Industrial Systems 1/2

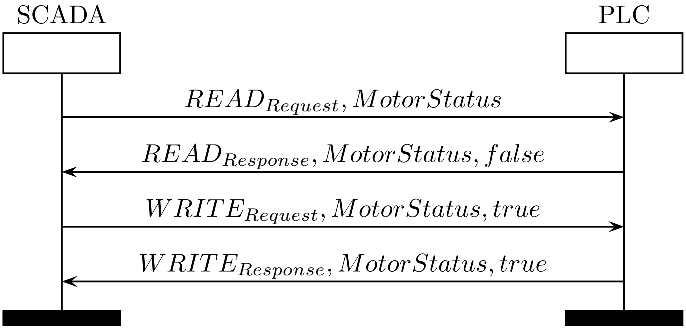
Hot topic

- 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

1. Introduction
2. Formal Verification of Industrial Protocols
   - Formal Verification of OPC-UA handshake
   - Flow Integrity Properties
3. Generation of Attack Scenarios

Table of Contents

1 Introduction

2 Formal Verification of Industrial Protocols
- Formal Verification of OPC-UA handshake
- Flow Integrity Properties

3 Generation of Attack Scenarios

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

1 Introduction

2 Formal Verification of Industrial Protocols
- Formal Verification of OPC-UA handshake
- Flow Integrity Properties

3 Generation of Attack Scenarios

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.

- Published in SAFECOMP'16, Trondheim, Norway.

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

1 Introduction

2 Formal Verification of Industrial Protocols
- Formal Verification of OPC-UA handshake
- Flow Integrity Properties

3 Generation of Attack Scenarios

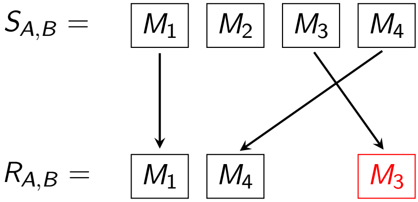
Contributions

⇒ Main Objective: add properties adapted to industrial systems in automatic verification tools.

- Published in SECRYPT'17, Madrid, Spain.

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

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

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.* »

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

Key Takeaways on Flow Integrity

- 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

1. Introduction
2. Formal Verification of Industrial Protocols
   - Formal Verification of OPC-UA handshake
   - Flow Integrity Properties
3. Generation of Attack Scenarios

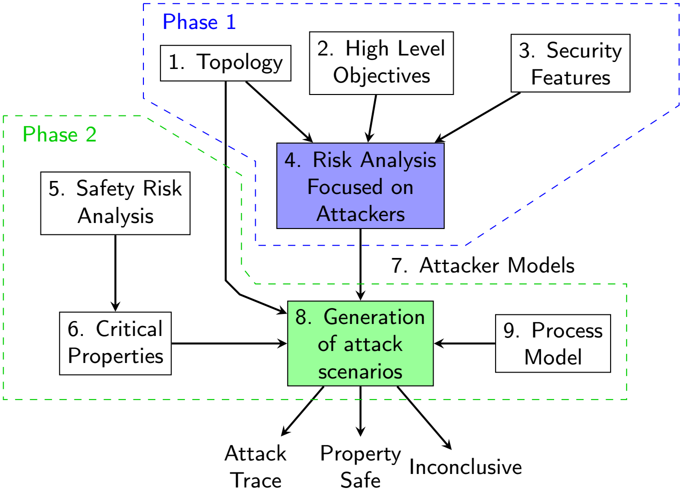
Idea & Contributions

- 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.
- Published in FPS'17, Nancy, France.

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

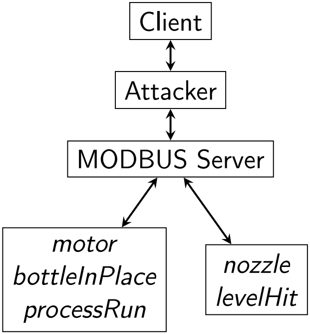


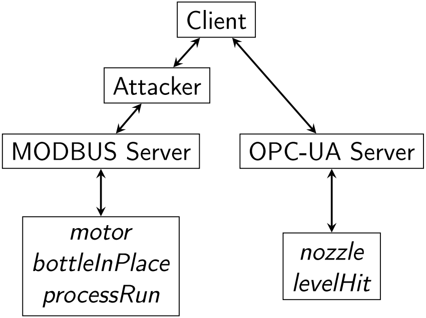
Phase 1 presented at AFADL/MTV2/MFDL 2016 in Besançon.

Topologies

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

- Communication channels between components;
- Position of attackers.



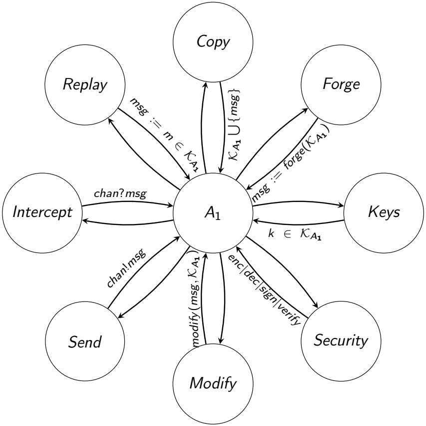


Attackers 1/2

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

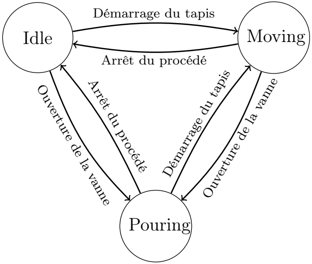


Four attackers:

- 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*). $A\square \neg(nozzle = true \text{ and } bottleInPlace = false)$
- Φ_2: $A\square \neg(motor = true \text{ and } levelHit = false)$
- Φ_3: $A\square \neg(nozzle = true \text{ and } motor = true)$

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

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.

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.

References IV

- 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.
- 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.

References V

- 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.
- 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 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.
- 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.
- 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).