Bridging Dolev-Yao Adversaries and Control Systems with Time-Sensitive Channels

Bogdan Groza and Marius Minea

Politehnica University of Timișoara and Institute e-Austria Timișoara, Romania

September 23, 2013

Formal verification tools (one side of our work)

Commonly employed for automatic protocol analysis

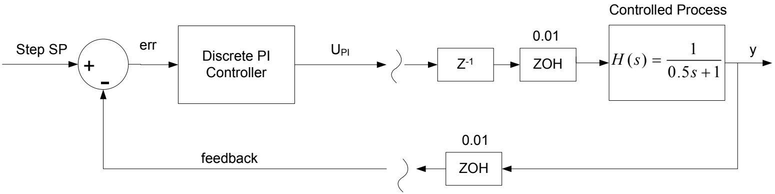
Outstanding results in showing protocol insecurity (for protocols assumed to be secure), e.g., Needham-Schroeder KE (Lowe'95)

Advantages compared to manual analysis:

- easy to use with limited security/cryptography expertise
- less prone to errors
- can deal with larger and complex systems

Control systems (the other side of our work)

Control systems regulate the behaviour of other systems (called plants) usually by means of a feed-back loop



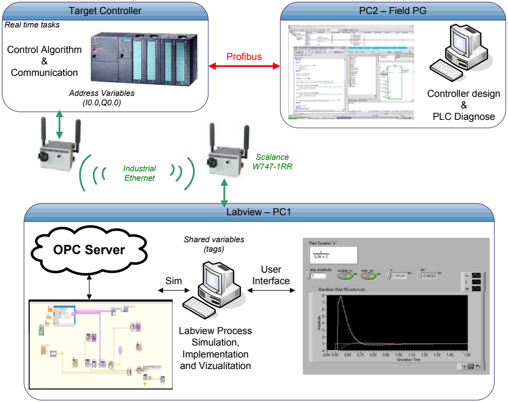
Discrete time systems, expressed as recurrent equations, are commonly used as abstractions for both controllers and plants

$$DTS(\mathbb{X}, \mathbb{U}, \mathbb{Y}, \mathcal{F}, \mathcal{G}) : \begin{cases} x(t+1) = \mathcal{F}(x(t), u(t)) \\ y(t) = \mathcal{G}(x(t), u(t)) \end{cases}$$

Relevance by practical scenario (previous work)

No doubts that control systems are increasingly exposed to cyber-attacks

One relevant target: WiFi (used in industrial networks if laying cables is difficult, e.g., moving objects: cranes, carousels)



SCALANCE (Siemens routers) Security Features

WPA2 WiFi Security and HTTPS Web Security (SSL/TLS)

Includes state-of-the-art cryptography: RSA, ECC, AES, etc.

Attack surfaces

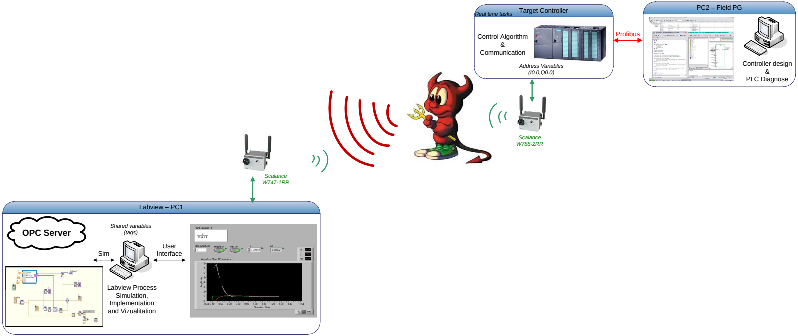
Two main targets:

- communication channel - allows manipulation of commands and responses from the control system and controlled process (WPA2)
- configuration interface - allows full control over the access points and clients (SSL/TLS)

State-of-the art, but is the system secure?

Attacking wireless communication

The easiest attack: cut down communication



Need a wireless signal jammer ?

No jammer needed - just use the 802.11 standard

Deauthentication packets force the STA to disassociate from AP

”Deauthentication shall not be refused by either party”
- IEEE 802.11 (2007)

A print-screen from the 802.11 standard

The complete set of IEEE 802.11 architectural services are as follows:

a) Authentication
b) Association
c) Deauthentication
d) Disassociation

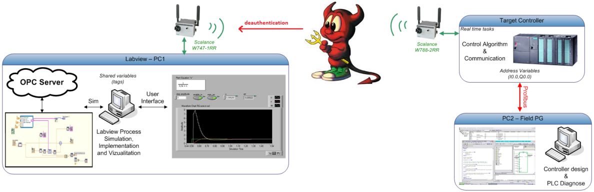
. . .

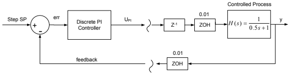
Clone AP MAC address then use *Aircrack-ng* to generate the deauthentication packets

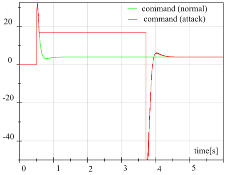
```
sudo aireplay-ng -0 0 -a 00:0E:8C:BF:25:78 -c 00:0E:8C:BC:2D:60 mon0
```

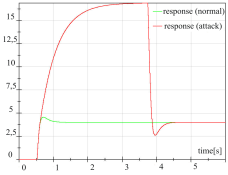
Effect

Last command preserved by the controller and process response increases rapidly (abnormal behavior)





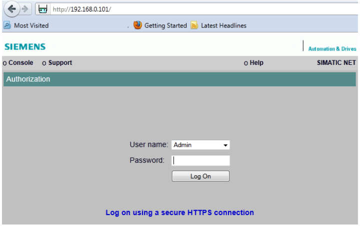




Important: the attack used mere standard specifications to manipulate time-sensitive goals (introduce delays) and subvert the output

Attacking the configuration interface

Protection by SSL/TLS - bullet proof?



Step 1: find how authentication works

No obfuscation of the JavaScript Code

$\Rightarrow$ authentication protocol obvious

Weak password-based protocol

1. $C \rightarrow AP$: *request*
2. $AP \rightarrow C$: N_{AP}
3. $C \rightarrow AP$: $C, MD5(C, pw_C, N_{AP}), N_{AP}$

No nonce from the client side – dictionary attacks

Fortunately runs under SSL/TLS if HTTPS is used

Remark 1: HTTP still works while HTTPS is locked

Inject a wrong SSL/TLS packet
$\Rightarrow$ HTTPS locks but HTTP still works

(same could be done by flooding with HTTPS requests)

Bug or feature?

Security implication: users can be tempted to log on HTTP

Step 2: determine user to login over HTTP
User enters password over HTTP $\Rightarrow$ intercept response

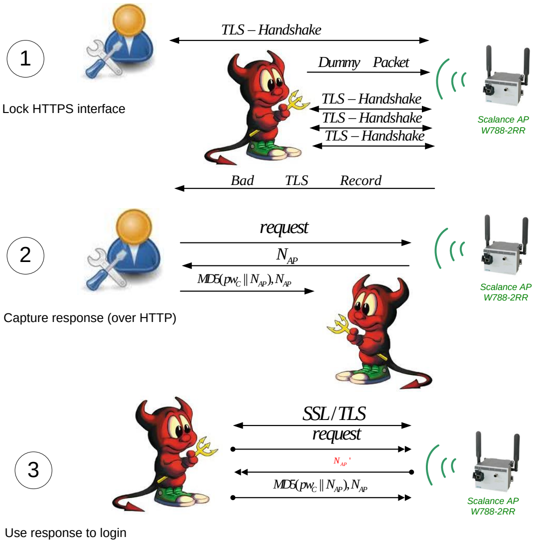
Remark 2: bug in the authentication protocol

Previous responses can be reused under HTTPS

Step 3: Send the response over HTTPS

1. $Adv(C) \rightarrow AP$: *request*
2. $AP \rightarrow Adv(C)$: N_{AP}
3. $Adv(C) \rightarrow AP$: $C, MD5(C, pw_C, N'_{AP}), N'_{AP}$

Attack summary



Problems and potential solutions

Attacks due to:

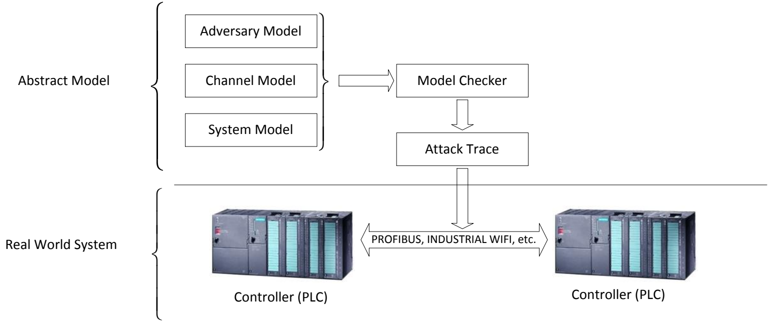
- obscure specifications in standards: de-authentication
- strange engineering decision: HTTP works when HTTPS locked
- erroneous implementation: reuses responses

The attacks can be circumvented if the system is formally analyzed before releasing it in the real world

Challenges and goals

Main challenge: bind formal verification tools (work with transition systems and symbolic terms) with control systems expressed as discrete time systems (defined by recurrent equations on real numbers)

Goals: find attack traces and (future work) test them on real-world industrial networks (e.g., penetration testing)



The ASLan language

Supported by the tools of AVANTSSAR and SPaCIoS projects

Used to define protocol actions via transitions

1. $A \rightarrow B : A$
2. $B \rightarrow A : N_B$
3. $A \rightarrow B : N_A, H(k_{AB}, N_A, N_B, A)$
4. $B \rightarrow A : H(k_{AB}, N_A)$

```
state_A(A,ID,1,B,Kab,H,
        Dummy_Na,Dummy_Nb)
.iknows(Nb)
=[exists Na]⇒
state_A(A,ID,2,B,Kab,H,Na,Nb)
.iknows(pair(Na,apply(H, pair(Kab,
        pair(Na, pair(Nb, A))))))
```

`iknows`: communication mediated by intruder

`exists`: generates fresh values

`state`: set of ground terms

`transition`: removes terms on LHS, adds terms on RHS (iknows is persistent)

Time-sensitive properties

Having principals A_{i_0} and A_{i_1} of a protocol specification, we can formally define:

- uniqueness, messages accepted only once, i.e.,

$$\mathrm{recv}_{i_b}(m, i_{\neg b}, t_1) \wedge \mathrm{recv}_{i_b}(m, i_{\neg b}, t_2) \Rightarrow t_1 = t_2, \forall b \in \{0, 1\}$$

- ordering, order of messages at sender is the same as at receiver, i.e.,

$$\mathrm{recv}_{i_b}(m_1, i_{\neg b}, t_1) \wedge \mathrm{recv}_{i_b}(m_2, i_{\neg b}, t_2) \wedge t_1 < t_2$$

$$\Rightarrow \mathrm{sndtime}_{i_{\neg b}}(m_1) < \mathrm{sndtime}_{i_{\neg b}}(m_2), \forall b \in \{0, 1\}$$

- δ-bounded lifespan, messages not accepted no later than some delay δ, i.e.,

$$\mathrm{recv}_{i_b}(m, i_{\neg b}, t) \Rightarrow t \leq \mathrm{sndtime}_{i_{\neg b}}(m) + \delta, \forall b \in \{0, 1\}$$

Next step?

We can reason about protocol properties and time-related goals but control-systems are still out of reach ... symbolic terms vs. real valued functions

Δ-grain abstractions

Used to make the state-space model approachable with our symbolic verification tools

Definition

$DTS^\natural(\mathbb{X}^\natural, \mathbb{U}^\natural, \mathbb{Y}^\natural, \mathcal{F}^\natural, \mathcal{G}^\natural)$ is a Δ-grain abstraction of $DTS(\mathbb{X}, \mathbb{U}, \mathbb{Y}, \mathcal{F}, \mathcal{G})$ under relations $\mathcal{R}_x$, $\mathcal{R}_u$, $\mathcal{R}_y$ if

(i) $\forall x \in \mathbb{X}, y \in \mathbb{Y}, u \in \mathbb{U}$ there exist $x^\natural \in \mathbb{X}^\natural, y^\natural \in \mathbb{Y}^\natural, u^\natural \in \mathbb{U}^\natural$ with $(x, x^\natural) \in \mathcal{R}_x$, $(y, y^\natural) \in \mathcal{R}_y$ and $(u, u^\natural) \in \mathcal{R}_u$,

(ii) $\forall x_0 \in \mathbb{X}, u_0 \in \mathbb{U}$, $x_0^\natural \in \mathbb{X}^\natural$, $u_0^\natural \in \mathbb{U}^\natural$ with $(x_0, x_0^\natural) \in \mathcal{R}_x$, $(u_0, u_0^\natural) \in \mathcal{R}_u$ there exists $0 < k \leq \Delta$ such that $(x(k), x^\natural(1)) \in \mathcal{R}_x$ and $(y(k), y^\natural(1)) \in \mathcal{R}_y$ if the input is constant for k steps, i.e., $u(i) = u_0, i \in [0, k-1]$, and

(iii) for any $k' < k$, $(x(k'), x^\natural(0)) \in \mathcal{R}_x$ and $(y(k'), y^\natural(0)) \in \mathcal{R}_y$ (all intermediary states and outputs have the same abstraction).

Completeness results

Two relevant proofs for the the correctness of the approach:

- Proposition 2, realizability of the abstract trajectory, shows that for any trajectory of the abstract system there exists a trajectory of the real system
- Proposition 3, couplability of abstractions, shows that for any two coupleable abstractions (controller-plant ensembles) there exists a trajectory of the coupled real system

Adversary goals

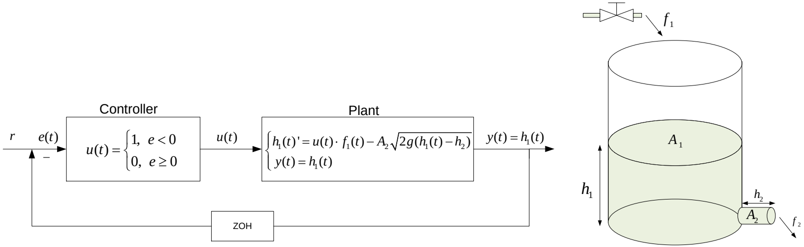
Definition (λ-step subversion)

Let the execution $\rho^\natural = (\sigma^0, \mathrm{X}^0, \mathrm{U}^0) \xrightarrow{r^1, s^1} (\sigma^1, \mathrm{X}^1, \mathrm{U}^1) \ldots \xrightarrow{r^t, s^t} (\sigma^t, \mathrm{X}^t, \mathrm{U}^t)$ $\rho^\natural = \sigma^0 \xrightarrow{r^1, s^1} \sigma^1 \ldots \xrightarrow{r^t, s^t} \sigma^t$ and let $\mathcal{G}_{adv} = \{\mathrm{X}^0_{adv}, \mathrm{X}^1_{adv}, ..., \mathrm{X}^\lambda_{adv}\}$ (defined over λ transitions). The adversary can perform a λ-step subversion w.r.t. $\mathcal{G}_{adv}$ over the control system if the states in the goal of the adversary hold during all of the last λ steps of the execution, i.e., $\forall i \in [0, \lambda) : X^{t-i} = X^{\lambda - i}_{adv}$.

Note: adversary actions are standard Dolev-Yao capabilities, i.e., he can intercept, modify and send messages at will, he can perform crypto-operation only if he has the corresponding keys (cryptography is perfect)

Example: a flow-control system

Scenario: simple *on/off* controller that regulates the water level inside the tank



Adversary's goal: subvert the water-level at will by using Dolev-Yao abilities (includes tampering with time-sensitive goals freshness, ordering and lifespan)

Δ-grain abstraction of the flow-control system

Associate states to symbolic operators that change at steps where controller and intruder behaviour requires changes

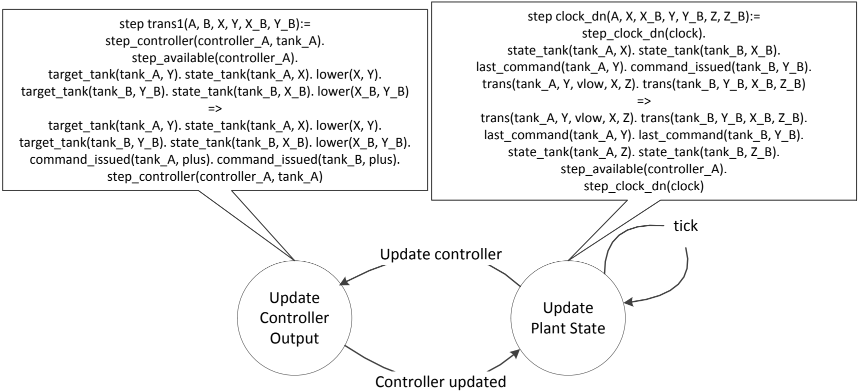
Abstraction sets	Relations between abstract and real values
$\mathbb{U}^\natural = \{off, on\}$	$\forall x \in [0,5) : (x, vlow^-) \in \mathcal{R}_x, \forall x \in [5,10) : (x, vlow^+) \in \mathcal{R}_x$
$\mathbb{X}^\natural = \mathbb{Y}^\natural = \{vlow^-,$	$\forall x \in [10,15) : (x, low^-) \in \mathcal{R}_x, \forall x \in [15,20) : (x, low^+) \in \mathcal{R}_x$
$vlow^+, low^-, low^+,$	$\forall x \in [20,25) : (x, med^-) \in \mathcal{R}_x, \forall x \in [25,30) : (x, med^+) \in \mathcal{R}_x$
$med^-, med^+, high^-,$	$\forall x \in [30,35) : (x, high^-) \in \mathcal{R}_x, \forall x \in [35,40) : (x, high^+) \in \mathcal{R}_x$
$high^+, vhigh^-, vhigh^+\}$	$\forall x \in [40,45) : (x, vhigh^+) \in \mathcal{R}_x, \forall x \in [45,50) : (x, vhigh^+) \in \mathcal{R}_x$
	$(0, off) \in \mathcal{R}_u, (1, on) \in \mathcal{R}_u$

Table : Abstraction sets and relations with the real system

Define state evolution based on precedence operators, e.g.,
$\mathcal{G}_{\mathcal{P}}^\natural(off, y^\natural(n-1)) = prec(y^\natural(n-1))$ and
$\mathcal{G}_{\mathcal{P}}^\natural(on, y^\natural(n-1)) = succ(y^\natural(n-1))$

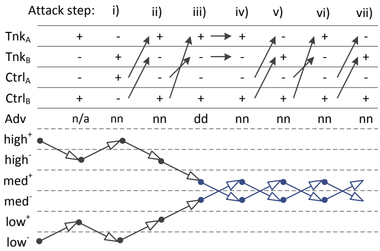
Defining the system

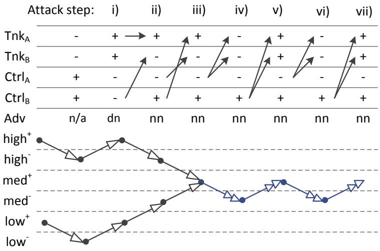
Now the control system can be described in the formal language (ASLan) via state-transitions



Attack traces reported by the CL-Atse model-checker

Adversary’s abilities are successfully used subvert the water-level at will: *delays* (step i), *redirection* (step ii), *replay* (steps iii) to vii)





Conclusions

Assuring time-related security goals (freshness, timeliness, life-span) in the presence of Dolev-Yao adversaries is critical for control systems

Δ-grain abstractions provide a workable model to tackle control systems properties in the framework of protocol analysis (via formal verification tools)

Experimental results: practical scenarios are within reach

Future work: more complex practical scenarios/protocols

SPaCIoS Secure Provision and Consumption in the Internet of Services, Project no. 257876, FP7-ICT-2009-5, 1.4: Trustworthy ICT 01.10.2010 - 31.01.2013