MODELLING SECURITY IN CYBER-PHYSICAL SYSTEMS

Mike Burmester, Florida State University
Emmanouil Magkos & Vassilis Chrissikopoulos, Ionian University

Sixth IFIP International Conference on Critical Infrastructure Protection
March 19, 2012
1. Motivation
   - SALT II (crypto for command & control)
   - The Russia – Ukraine dispute (crypto verification & validation)

2. A model for CPS
   - Yet another model! (holistic security)

3. A threat model for provable security …
   - To specify what the hackers can …

4. An application
   - Crypto for treaty verification (privacy)
1. The Strategic Arms Limitation Treaty SALT II between the US and the Soviet Union (1977--1979) sought to curtail the number of strategic delivery vehicles (ICBM) to 2,250 on both sides.

2. This would involve installing tamper resistant sensor control units in all ICBM silos to detect the presence of missiles.

3. Both sides would have access to this information, but to no other information regarding the silos, particularly the location of the responding silos (need-to-know precondition).
4. The sensor control units were to be used to verify the number of deployed ICBMs.

5. Appropriate cryptographic protocols that support privacy, integrity and authentication had to be designed.

6. Cryptography for verifying adherence to treaties is particularly challenging when precondition on need-to-know.
Motivation – SALT II

7. Gus Simmons, Sandia National Labs

- Crypto for command & control operations
- Major concern: information hiding, & covert channels (aka subliminal channels)
- This started a new direction of research in cryptology

8. SALT II was never ratified …
Several disputes between Naftogaz Ukrainy and the Russian gas supplier Gazprom involving gas supplies, prices and debts, threatened the gas supplies to the European Union from 2005 to 2010.

The dispute peaked in 2006 when Russia cut off all supplies to Ukraine.

Russia provides approximately one quarter of the natural gas consumed in the EU, and 80% of this is piped across Ukraine to reach the EU.
The Russia-Ukraine natural gas dispute

- Need cryptography for *verifying adherence* to agreed natural gas flows.
- Ukraine must be able to verify that it gets the agreed allocation.
- *Covert channels*: Ukraine may be able to find out information about the flows by using covert channels (e.g., payments made by EU customers).
- This is not a serious issue in this case: what Russia wants is that, *knowledge about the gas network flows does not leak from the network itself.*
The security issues of Cyber-Physical Systems are not new …

… and there is a lot of work that is relevant for today’s needs.
A model for Cyber-Physical Systems

Timed automata with faults

Parties involved

- Those specified by the system
  - e.g., the operators.
- The adversary
  - an entity that controls all parties that do not adhere to the system specifications.
- Nature
  - Nature’s faults are determined by a stochastic process that takes into account the reliability of physical components.
The reliability of a component is the probability that it fails between time $t$ and $t + \delta t$.

If the failure rate $\lambda$ is constant then we use the exponential probability density function $f(x) = \lambda e^{-\lambda x}$.

The probability of failure is $\int_{t_{i-1}}^{t_i} \lambda e^{-\lambda x} = e^{-\lambda t_{i-1}} - e^{-\lambda t_i}$.

The probability distribution $F$ of failure of a CPS must take into account the probabilities of all components.
Nature uses a *time schedule* $\tau$ to instantiate events.

$\tau$ is a monotonic increasing discrete valued sequence of reals: $\tau = t_1, t_2, \ldots$.

For real time applications $\tau$ must be linked to actual time.

Actions in response to events are instantiated at time $t_i$. 

A formal model for CPS

A timed automaton $\mathcal{A}$ with faults:

A tuple $(\tau, A, Q, q_0, D, \mathcal{F})$

- $A$ is the set of actions and includes an action $\bot$ that is invoked only by Nature
- $Q$ are the system states
  - partitioned into $Q_s, Q_c, Q_t$
- $D \subseteq Q \times Q \times A \times 2^C \times T(C)$, the transition function
- $\mathcal{F}$ is the distribution of faults
A timed execution of the automaton $\mathcal{A}$

$$q_0 \xrightarrow{t_1,a_1} q_1 \xrightarrow{t_2,a_2} \ldots \xrightarrow{t_{i-1}, \ldots} q_{i-1} \xrightarrow{t_i, \perp} q_i \ldots$$

A path that starts at $q$ and ends at $q'$ is said to link these states.

The pair $(\tau, a)$ with $a = a_1 a_2 \ldots$ is called a run of $\mathcal{A}$.

Nature’s input $\perp$ takes priority over other inputs.
We make four assumptions regarding the system specifications:

1. For each period \((t_{i-1}, t_i]\) the system selects an action that either maintains the current state or transitions to safe state.

2. If Nature causes the system to transition to a critical state then the system will restore its state to safe in the next transition.

3. The only way for the system to transition from a safe state to a terminal state is via a critical state.

4. Nature will cause a transition to a terminal state at time \(t_i\) only if the states \(q_{i-k}, \ldots, q_i\) were critical, where \(k\) is a system parameter.

The security goal for the automaton \(A\) is to never enter a terminal state.
The language \( L \) over \( \mathcal{A} \) of timed words \((\tau, a)\) with errors is defined by:

1. Any state in \( Q \setminus Q_t \) repeats infinitely along any run of the timed word \((\tau, a)\),

2. Nature’s faults distribution \( \mathcal{F} \) determines when action \( a_i \) of \((\tau, a)\) is \( \bot \) (an error, nondeterministic behavior), and

3. Patterns of consecutive errors \( a_i = \bot \) of \((\tau, a)\) are shorter than \( k \).
A model for CPS

**Definition**

A CPS is resilient if there is a flow control protocol $\pi$ that on input a pair of safe states $(q, q')$ will output a timed safe path that links $q, q'$ with overwhelming probability.

**Consequently,**

If the initial state of a resilient CPS is safe then throughout the lifetime of the system its state will either be safe or critical.
This must capture the adversarial *intent* and the *scope* of an adversarial attack.

It must address those features of the system that the adversary may abuse and which may lead to system failure.

System failure can result from actions by Nature, the adversary or both.
We identify the specific features of the system that the adversary can exploit. These are the vulnerabilities identified by:

- the *security policies* of the system (e.g., the availability of services, the privacy of records, etc),
- *vulnerability assessments*, or
- *grey-box penetrating tests*. 
The vulnerabilities are specified by the function

\[ f : (\tau, Q) \rightarrow 2^V; (t_i, q_i) \rightarrow V_i \subseteq V, \]

The threat model of a CPS is modeled by the \textit{timed threat transition function}

\[ D_f(t_i): f(q_{i-1}) = V_i \xrightarrow{t_i,a} f(q_i) = V_i, \]

that specifies the priori/posteriori features of an adversarial attack.
Timed threat transition vulnerabilities of an automation $\mathcal{A}$

\[ f(t_{i-1}, q_{i-1}) \rightarrow V_{i-1} \rightarrow D(t_i) \rightarrow (t_i, q_i) \rightarrow V_i \]
Definition

An adversary restricted to the threat transition function $D_f(\tau)$ is called a $D_f(\tau)$-adversary.

A cyber-physical system is $D_f(\tau)$-tolerant if it is resilient, and if it operates as specified (by the policies of the system) in the presence of a $D_f(\tau)$-adversary.
$D_f(\tau)$-adversaries are \textit{temporal}.

$D_f(\tau)$-adversaries can exploit vulnerabilities that are not normally present, e.g., we may have $D_f(\tau) = \emptyset$ most of the time.

This captures insiders who can hack into a CPS at unexpected times, or while a catastrophic event takes place, zero-day attacks, etc.
A variant of the Russian-Ukraine dispute

In this model the natural gas grid has three branches in Ukraine:

- one for North EU (network B)
- one for South EU (network C), and
- one that supplies natural gas to Ukraine (network D).

Ukraine is allocated 10% of the gas flow through B and 5% of the flow through C.
The Russia-Ukraine natural gas grid with three sub-networks
The flows are automatically enforced by the Flow Controllers $FCS_B$, $FCS_C$ and $FCS_D$.

The security of the RU grid concerns

the privacy of flow $B$, flow $C$

the correctness of the allocation to Ukraine (flow $D$).

System requirement

Ukraine must be able to verify that it gets its correct allocation of gas!
Adversarial behavior

**Intent**
1. Cause the system to malfunction
2. Learn the values of flows $\text{flow}_B$, $\text{flow}_C$
3. Violate correctness of $\text{flow}_D$

**Resilience**
If the FCS are tamper resistant and we have redundancy, then the first goal is thwarted.

**Privacy, integrity**
The second and third goals would certainly be of interest to an insider, such as Ukraine.

**Scope of malicious behavior**
Restricted because of the assumption that the FCS’s are tamper resistant.
The vulnerabilities of this model involve the flows:

\[ a = \text{flow}_A, \ b = \text{flow}_B, \ c = \text{flow}_C, \ d = \text{flow}_D. \]

In particular \( f \) (state) = \( (z_1, z_2, z_3, z_4, z_5, z_6) \) with

\[ z_1 = a, \quad z_2 = b, \quad z_3 = c, \quad z_4 = d, \]

\[ z_5 = a - b - c - d, \quad z_6 = 20d - 2b - c. \]
The following constraints on the normal, critical and danger levels of gas supplied to S-EU and N-EU apply:

- \( c_1 : 0 \leq z_1 < Y_1, \quad c'_1 : Y_1 \leq z_1 < Y'_1, \quad c''_1 : Y'_1 \leq z_1 \)
- \( c_2 : 0 \leq z_2 < Y_2, \quad c'_2 : Y_2 \leq z_2 < Y'_2, \quad c''_2 : Y'_2 \leq z_2 \)
- \( c_3 : 0 \leq z_3 < Y_3, \quad c'_3 : Y_3 \leq z_3 < Y'_3, \quad c''_3 : Y'_3 \leq z_3 \)
- \( c_4 : 0 \leq z_4 < Y_4, \quad c'_4 : Y_4 \leq z_4 < Y'_4, \quad c''_4 : Y'_4 \leq z_4 \)
- \( c_5 : 0 \leq z_5 < \varepsilon, \quad c'_5 : \varepsilon \leq z_5 \)
- \( c_6 : 0 \leq z_6 < \varepsilon, \quad c'_6 : \varepsilon \leq z_6 \)

where \( Y_i, Y'_i, \ i = 1,2,3, \) are system parameters with

\[
Y_2 + Y_3 + Y_4 \leq Y_1 < Y'_1 \leq Y'_2 + Y'_3 + Y'_4.
\]
Setting for a Flow Verification protocol

The Flow Verification protocol uses a family of multiplicative integer groups whose modulus is a "safe" prime $p$, that is

$$p = 2q+1,$$

where $q$ is a prime.

- Let $g \in \mathbb{Z}_p$ have order $q$ and $G_q$ be the group generated by $g$.
- We assume that $b$, $c$ are rounded to an integer value and that $2b + c \ll q$.
- The function $F : \mathbb{Z}_q \to G_q : x \to g^x$ is a one-way homomorphism ($F(x + y) = F(x) \cdot F(y)$).
A Flow Verification protocol

\[ \text{FCS}_B(b, t_b) \xrightarrow{g^{2bt_b}} \text{FCS}_C(c, t_c) \]

\[ \text{FCS}_B(b, t_b) \xrightarrow{g^{2bt_c}} \text{FCS}_C(c, t_c, s_c) \xrightarrow{g^{s_c, (2b+c)s_c}} \text{FCS}_D(d) \]

Verification \( \text{FCS}_D(d) \):
\[ g^{20ds_c} \overset{?}{=} g^{(2b+c)s_c} \]
The Flow Verification protocol

Threat model
The identified vulnerabilities \( V \) concern the communication channels of the Flow Controllers.

**Theorem 1.** Suppose that:

1. \( FCS_A, FCS_B, FCS_C \) and \( FCS_D \) are tamper resistant and that the flow control policies described earlier are adopted.

2. The communication channels linking \( FCS_A, FCS_B, FCS_C \) and \( FCS_D \) are reliable and authenticated.

Then the RU grid is resilient in the presence of a \( D_f(\tau) \)-adversary.

**Proof**
Resilience in the presence of a \( D_f(\tau) \)-adversary follows from the tamper resistance of the FCs and the fact that the communication channels are reliable and authenticated.
Theorem 2

If the RU grid tolerates a $D_f(t)$-adversary and,

1. There are no covert channels that leak the values of $b,c$.
2. The DDH assumption holds.

Then the Flow Verification protocol is correct and provides privacy for the flows $b,c$ against a $D_f(t)$-adversary who knows the flow $d$.

Proof

Correctness follows from: "$20d = 2b + c$".

Privacy of the communication channels is reduced to the Decisional Diffie-Hellman problem—see Reference.
Any questions?