Formal Security Assessment of Modbus Protocol

Roberto Nardone[†], **Ricardo J. Rodríguez**[‡,§], Stefano Marrone[§]

roberto.nardone@unina.it, rjrodriguez@ieee.org, stefano.marrone@unina2.it

Universidad Zaragoza

[†]Univ. di Napoli Federico II [‡]Universidad de Zaragoza [§]Seconda Università di Napoli

December 05, 2016

11th International Conference for Internet Technology and Secured Transactions

Barcelona (Spain)

Agenda

Universidad Zaragoza
1542

Agenda

Universidad Zaragoza 1542

Introduction

Critical Infrastructures

- **Large, complex heterogeneous systems**
  - Example: water treatment, power distribution, or logistics
- Our day-to-day basis depends on
- Disruptions of services may provoke issues, from economical to personal damages
  - Different origins: from man made to unexpected acts of nature
  - **Intended and unintended**

Universidad Zaragoza 1542

Introduction

- **Automated and distributed control systems rapidly adopt information and communication technology solutions**
  - Support operation and monitoring of industrial and critical processes

Problem: legacy devices running routable protocols

- Isolated components at the beginning: not any more
- New attack surface: communication protocol

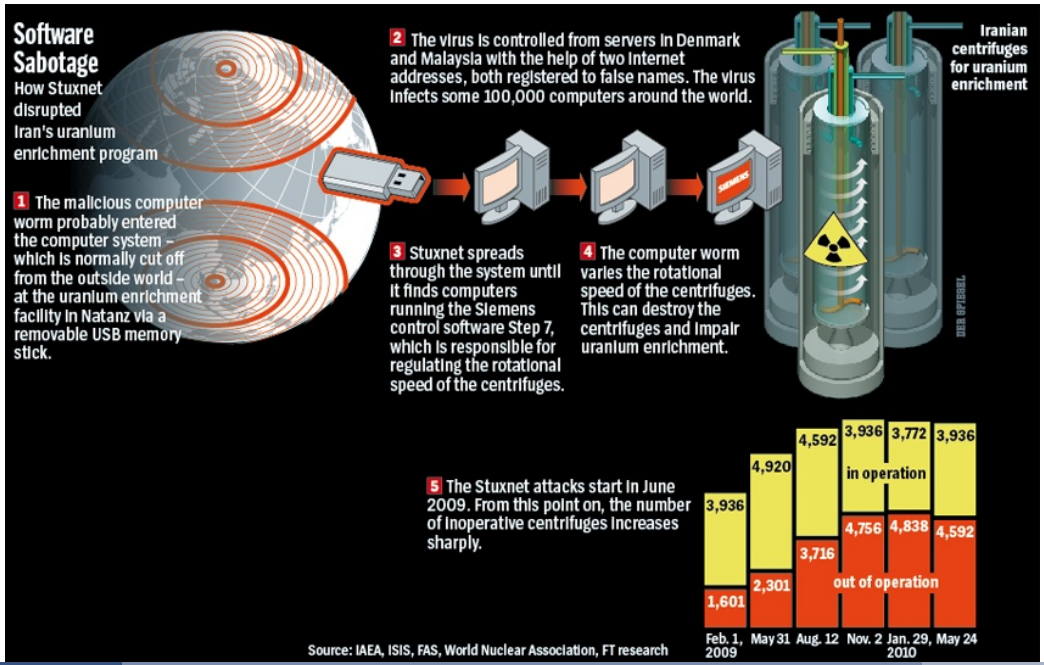
Fact: attacks to critical infrastructures are increasing

Universidad Zaragoza 1542

Introduction

Advanced Persistent Threats examples

- `Operation Aurora`: attributed to China, in 2010 a lot of companies from different domains (such as Google, Yahoo, Morgan Stanley, or Dow Chemicals) were attacked
- `Stuxnet`: attributed to US-Israel and discovered in 2010, affected to Siemens PLCs of SCADA networks in Iran nuclear facilities
- Others: `GhostNet`, `Duqu`, `Flame`, `...`



Introduction

Modbus protocol: main characteristics

Characteristics

- Top layer of OSI model
- Designed by Modicon in 1979 to operate using a RS232 port
  - Widely deployed and adopted as *de facto* standard
  - Now, it operates over different links (serial buses, routable networks over TCP/IP, or intercommunicated buses
    - Each device has a unique identifier
- **Request/response (master/slave) protocol**

Universidad Zaragoza 1542

Introduction

Modbus protocol: communication issues and security

- Type of commands:
  - Read/write commands
  - Device identification
  - Diagnostic
- Command format:
  - Destination, code function, and sub-code function (may be none)
- **Communication flow**:
  - Master device initiates a command query containing destination device
  - Destination device performs the function requested by the master, and replies

What about security?

- **No authentication**
- **No encryption at any form**
- **No integrity**

Introduction

Contribution

- **Security analysis assessment of the Modbus protocol using formal models**
  - From a high-level formalism based on hierarchical state machines, we derive a Promela model to apply model checking
- **Formal verification of the weaknesses of Modbus protocol** (in particular, to man-in-the-middle attacks)
- A **formal framework to evaluate solutions against security concerns** is provided

Universidad Zaragoza 1542

Agenda

Universidad Zaragoza
1542

Related work

Security of Modbus protocol

- **Classification of attacks** (Huitsing et al., 2008)
  - Modbus protocol specification, vendor implementations, control system assets
  - 60 attack identified, including spoofing, replay, and flooding attacks
- **Impact of malicious traffic injection** (Kobayashi et al., 2009)
- **Vulnerable to spoofing** confirmed by Bathia et al., 2014
  - Two detection algorithms proposed: anomaly and signature
- **Intrusion detection in network layer** (Goldenberg and Wool, 2013)

Defenses proposed

- **Modbus protocol enhancement** (Fovino et al., 2009)
  - Accounts for **authentication, non-repudiation, and replay protection**
  - **Overhead in performance and packet size**

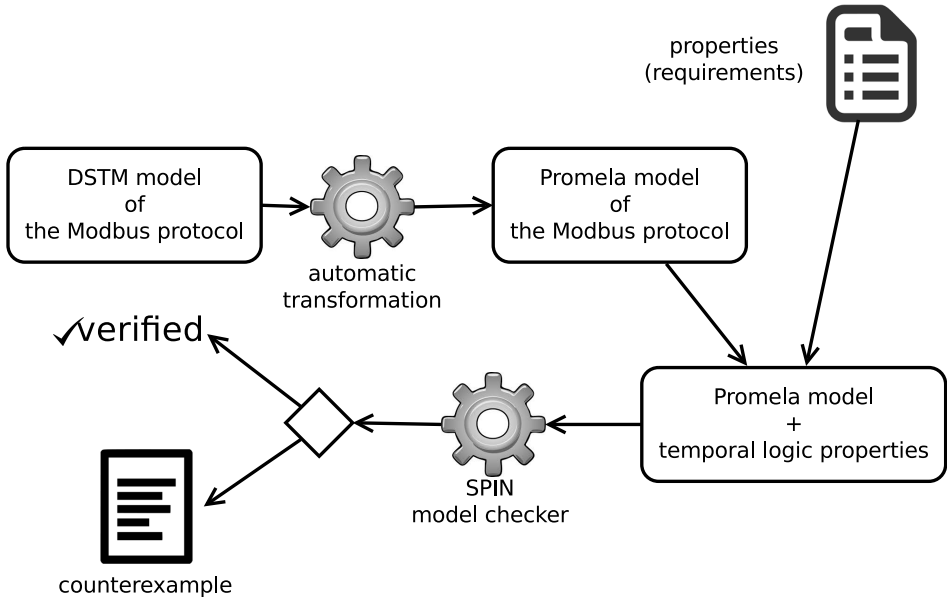
Zaragoza 1542

Agenda

Universidad Zaragoza 1542

General Approach

From high-level syntax to low-level syntax



Model transformation

- **Dynamic StaTe Machine** (DSTM): extension of hierarchical state machines
- **Promela notation analyzable by SPIN**
  - Promela model **enriched with temporal logic formulas derived from the set of requirements to assess**
  - **Automatic counterexample generation** when properties are violated

General Approach

Dynamic StaTe Machine formalism (1)

Reasons

- Formal (textual and graphical) syntax semantics of both structural elements and annotations over transition (i.e., triggers, conditions, and actions)
- External messages are non-deterministically generated
- Automatic transformation from DSTM to Promela exists

- **Extension of hierarchical state machines**: novel semantics for fork and join
  - Removes the constraint of branches of a control flow exiting a fork must always be merged by a join element
- **Dynamic and recursive instantiation of machines**
- **Preemptive termination**
- **Passing parameters (to the machine) at instantiation time**

General Approach

Dynamic StaTe Machine formalism (2)

Description

- **Collection of parametric machines, channels (internal and external), and variables**
- **Definition of own datatypes allowed**
- **Datatypes**: basic, compound (records), multi-types (collection)
- **Channels**
  - *Internal*: entirely managed by the specified state machines; buffered with predefined length. Messages are consumed when reading
  - *External*: unbuffered. Message is valid during a single step and can be only read without removal

Universidad Zaragoza 1542

General Approach

Dynamic StaTe Machine formalism (3)

Workflow execution

- At starting of each step, a new message is present over external channels
  - **Generated in the previous step**
  - (or) **Randomly generated over possibilities given by the datatype**

- **Machine components**: vertices (nodes) and transitions
  - Initial, entering, and exiting nodes
- **Boxes represent single or multiple machine instantiation**
  - Needs machine parameters

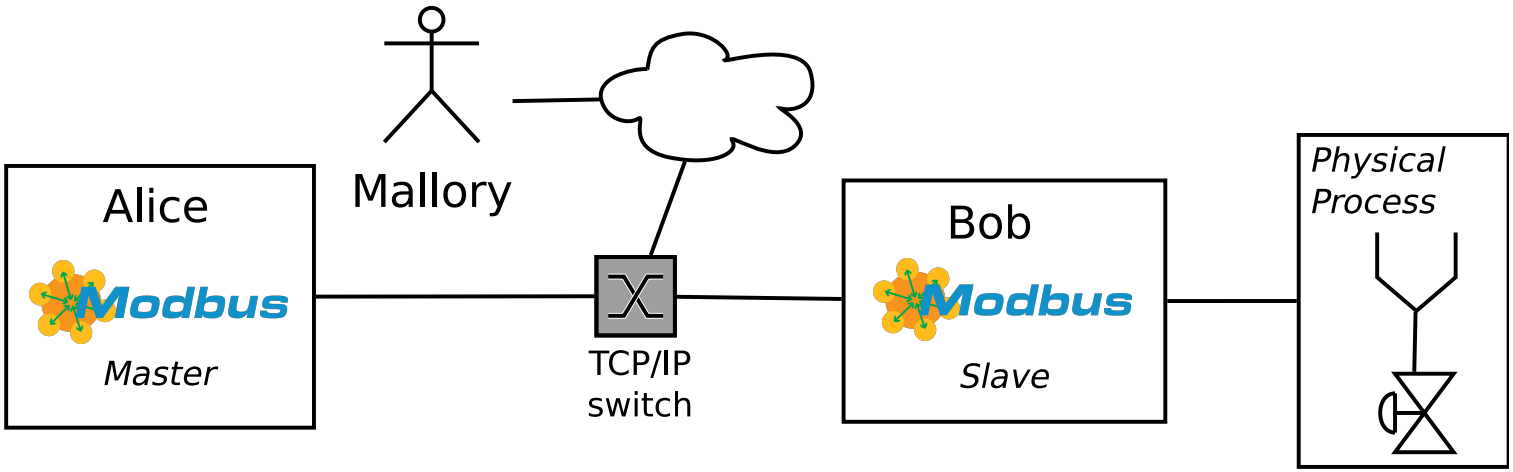
Universidad Zaragoza 1542

Agenda

Universidad Zaragoza
1542

Security Analysis of Modbus Protocol

Reference scenario



Question here:

Can Alice receive messages that indicate a value of the physical process equal to one while the process is indeed stuck at zero?

Universidad Zaragoza

Security Analysis of Modbus Protocol

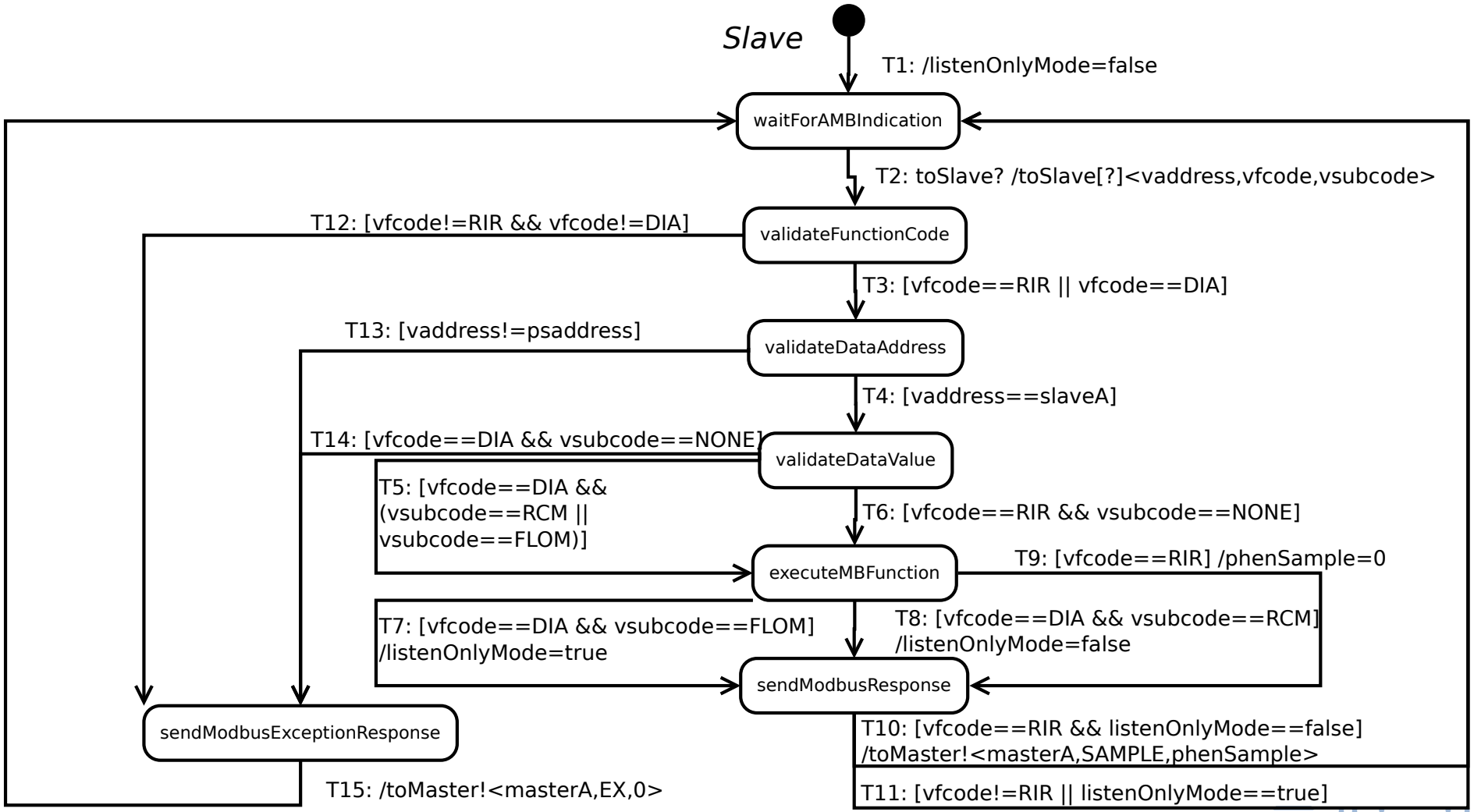
Datatypes and variables

```
//enumerations
Enum address {slaveA, masterA};
Enum fcode {RIR, DIA};
Enum subcode {NONE, RCM, FLOM};
Enum answer {EXCEPTION, SAMPLE};
//structures
Struct toMasterMsg {address, answer, Int};
Struct toSlaveMsg {address, fcode, subcode};
//channels
Chn external toMaster of toMasterMsg;
Chn external toSlave of toSlaveMsg;
//master's variables
Int sampleToMaster;
//slave's variables
address vaddress;
fcode vfcode;
subcode vsubcode;
Bool listenOnlyMode;
Int phenSample;
```

Universidad Zaragoza 1542

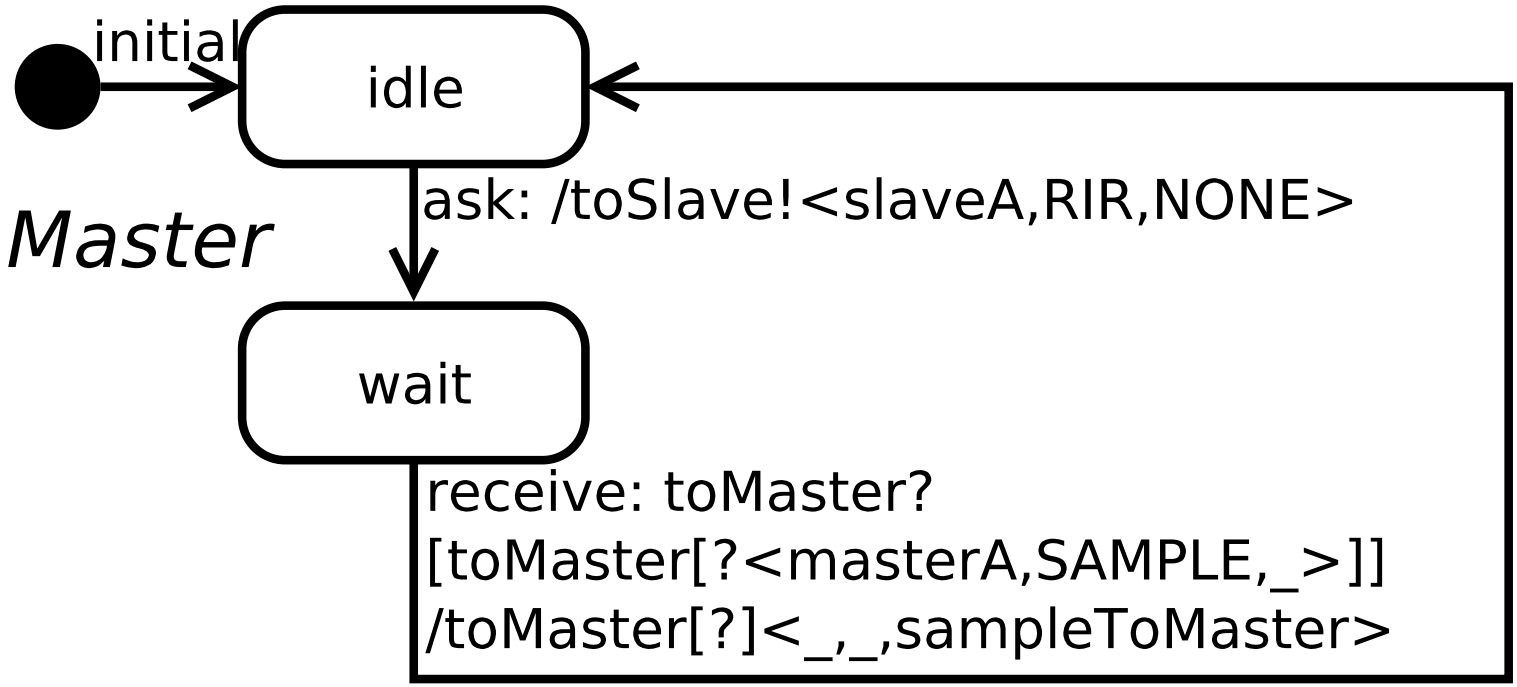
Security Analysis of Modbus Protocol

DSTM models of Modbus slave



Security Analysis of Modbus Protocol

DSTM models of Modbus master



Property definition and evaluation

"it is always true that, once `SM` is equal to 1, it still remains equals to 1"

$$AG((SM == 1) \Longrightarrow AG(SM == 1))$$

Security Analysis of Modbus Protocol

Counterexample generated by SPIN

```
M->A: <masterA,SAMPLE,0>
M->B: <slaveA,RIR,NONE>
M->A: <masterA,SAMPLE,0>
...
M->B: <slaveA,RIR,NONE>
M->A: <masterA,SAMPLE,0>
A->B
B->A
M->B: <slaveA,DIA,FLOM>
M->A: <masterA,SAMPLE,0>
A->B
M->A: <masterA,SAMPLE,0>
M->B: <slaveA,RIR,NONE>
M->A: <masterA,SAMPLE,0>
```

```
A->B
M->A: <masterA,SAMPLE,0>
M->B: <slaveA,RIR,NONE>
M->A: <masterA,SAMPLE,0>
A->B
M->A: <masterA,SAMPLE,0>
M->B: <slaveA,DIA,FLOM>
M->A: <masterA,SAMPLE,1>
A->B
M->A: <masterA,SAMPLE,0>
M->B: <slaveA,RIR,NONE>
M->A: <masterA,SAMPLE,1>
A->B
M->A: <masterA,SAMPLE,0>
M->B: <slaveA,RIR,NONE>
M->A: <masterA,SAMPLE,1>
...
```

Universidad Zaragoza 1542

Security Analysis of Modbus Protocol

Discussion

- **Plausibility of this scenario**: Modbus specification addresses diagnostic functions only in serial line and ModbusPlus communication
  - Exist solutions extending these functionalities also in scenarios which exploit the tunneling of serial communication over TCP/IP and TCP/IP Modbus gateways
- **Performance**: negligible time to generate counterexample
  - Only a subset of functionalities are modeled
  - Optimizing strategies are necessary: SPIN does not generate the shortest possible counterexample

Universidad Zaragoza 1542

Agenda

Universidad Zaragoza
1542

Conclusions

- Automated and distributed control systems built using legacy devices and running legacy protocols
- Rapidly evolving to routable networks, increasing attack surface
- Modbus is widely adopted in industrial control systems
  - **Not designed to operate in a hostile environment**

Contribution

- **Analyzed Modbus with formal models**
  - From a high-level formalism based on hierarchical state-machines to a model suitable for model-checking
  - **We proved the existence of man-in-the-middle attacks**

Future work

- **Complete the modeling of Modbus protocol**
- **Formally prove other attacks**

Formal Security Assessment of Modbus Protocol

Roberto Nardone[†], **Ricardo J. Rodríguez**[‡,§], Stefano Marrone[§]

roberto.nardone@unina.it, rjrodriguez@ieee.org, stefano.marrone@unina2.it

Universidad Zaragoza

[†]Univ. di Napoli Federico II [‡]Universidad de Zaragoza [§]Seconda Università di Napoli

December 05, 2016

11th International Conference for Internet Technology and Secured Transactions

Barcelona (Spain)