UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

Douglas Simões Silva

FORMAL VERIFICATION AND ACCESS CONTROL APPROACH OF AN IOT PROTOCOL

Florianópolis

Douglas Simões Silva

FORMAL VERIFICATION AND ACCESS CONTROL APPROACH OF AN IOT PROTOCOL

Dissertação de Mestrado submetido à Ciências da Computação para a obtenção do Grau de Mestre em Ciências da Computação.
Orientador: Prof. Dr. Jean Everson Martina
Universidade Federal de Santa Catarina
Coorientador: Prof. Dr. Ricardo Alexandre Reinaldo de Moraes

Florianópolis

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Silva, Douglas Simões
Formal Verification and Access Control Approach of an IoT Protocol / Douglas Simões Silva ; orientador, Jean Everson Martina; coorientador, Ricardo Alexandre Reinaldo de Moraes - SC, 2017.
80 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós Graduação em Ciência da Computação, Florianópolis, 2017.

Inclui referências.

1. Ciência da Computação. 2. Internet of Things. 3. Security Protocol Analysis. 4. Context-Aware. 5. Access Control. I. Everson Martina, Jean . II. Alexandre Reinaldo de Moraes, Ricardo . III. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Ciência da Computação. IV. Título.

Douglas Simões Silva

FORMAL VERIFICATION AND ACCESS CONTROL APPROACH OF AN IOT PROTOCOL

Este Dissertação de Mestrado foi julgado aprovado para a obtenção do Título de "Mestre em Ciências da Computação", e aprovado em sua forma final pela Ciências da Computação.

Florianópolis, May de 2017
.

Profa. Dra. Carina Friederich Dorneles
Universidade Federal de Santa Catarina
Coordenador do Curso

Prof. Dr. Jean Everson Martina
Universidade Federal de Santa Catarina
Orientador

Prof. Dr. Ricardo Alexandre Reinaldo de Moraes
Coorientador

Banca Examinadora:

Prof. Dr.
Ricardo Felipe Custódio
Universidade Federal de Santa Catarina

Profa. Dra.
Rebecca Montanari
University of Bologna

Profa. Dra.
Carla Merkle Westphall
Universidade Federal de Santa

Agradeço a minha família que me deu toda a educação necessária para que eu pudesse aproveitar todas as boas oportunidades na vida.

RESUMO

Protocolos de Segurança estão na nossa rotina diária e exemplos disto são compras utilizando o cartão de crédito, eleição eletrônica, redes sem fio e etc. O primeiro objetivo deste trabalho é a verificação formal dos aspectos de segurança de um protocolo voltado para *Wireless Sensor Networks* (WSN). O *Trustful Space-Time Protocol* (TSTP) engloba a maioria das características necessárias para aplicações WSN como por exemplo controle de acesso, roteamento geográfico de pacotes, estimativa de localização, *clock* precisamente sincronizado, canais de comunicação segura e um esquema de distribuição de chaves entre o *gateway* e os sensores. Após a análise formal do protocolo de distribuição de chaves do TSTP usando Proverif, nós encontramos duas falhas de segurança: uma relacionada ao componente de sincronização de tempo e outra relacionada ao método *mac-then-encrypt* empregado. Com as falhas encontradas nós propómos uma versão melhorada do protocolo de distribuição de chaves. O segundo objetivo é criar um esquema de controle de acesso sensível ao contexto para dispositivos Internet de Coisas(IoC) usando TSTP como canal de comunicação. O esquema da política foi projetado para um cenário Smart Campus e seu contexto. Aproveitamos os recursos do TSTP para adicionar dados de tempo e espaço como contexto para o nosso modelo. Após o desenho do modelo de política, descrevemos seu modelo simbólico e fizemos uma análise formal para ter certeza de que os valores das propriedades de contexto não foram adulterados.

Palavras-chave: Redes de sensores sem fio, internet das coisas, verificação formal, consciência do contexto ,controle de acesso

ABSTRACT

Security protocols are included in our every day routine. A few examples are credit card purchases, e-voting, wireless networks, etc. The first goal of this dissertation is the formal verification of the security aspects of a cross-layer, application-oriented communication protocol for Wireless Sensor Networks (WSN). The Trustful Space-Time Protocol (TSTP) encompasses a majority of features recurrently needed by WSN applications like medium access control, geographic routing, location estimation, precise time synchronization, secure communication channels and a key distribution scheme between sensors and the sink. After the security protocol analysis of TSTP's key distribution protocol using ProVerif we were able to find two security flaws: one related to the time synchronization component and another being a bad approach related to a mac-then-encrypt method employed. With our findings we propose an improved version of the key distribution protocol. The second goal is to create a context-aware access control scheme for Internet of Things(IoT) devices using TSTP as a communication channel. The policy's scheme was designed for a Smart Campus scenario and its context. We take advantage of TSTP's features to add time and space data as context for our model too. After the design of the policy model, we described its symbolic model and we did a formal analysis to be sure that the context properties values were not tampered.

Keywords: Wireless Sensor Networks, Internet of Things, Formal Verification, Security Protocol Analysis, Context-Aware ,Access Control

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

WSN	Wireless Sensor Networks
TSTP	Trustful Space-Time Protocol
IoT	Internet of Things
GPS	Global Positioning System
MAC	Message Authentication Code
SHA-1	Secure Hash Algorithm 1
ECC	Elliptic Curve Cryptography
LDUAS	Lighweight Dynamic User Authentication Scheme
LEAP	Localized Encryption and Authentication Protocol
LAS	Lightweight Authentication Scheme
NLPSF	Node Level Security Policy Framework
MitM	Man in the middle attack
EAKEP	Efficient Authenticated Key Establishment Protocol
CA	Certification Authority
MUBA	Multiuser Broadcast Authentication scheme
SPTP	Speculative Precision Time Protocol
HECOPs	Heuristic Environmental Consideration Over Position
RSSI	Received Signal Strength Indication
OTP	One-Time Password
AES	Advanced Encryption Standard
IETF	Internet Engineering Task Force
DMTF	Distributed Management Task Force
PDP	Policy Decision Points
PEP	Policy Enforcement Point
ECDH	Elliptic Curve Diffie-Hellman
MtE	MAC-then-Encrypt
EtM	Encrypt-then-MAC
TTP	Trustable Third Party

SUMÁRIO

1 INTRODUCTION

According to Statista website(NEWS, 2016) in 2016 the number of connected devices worldwide was already more than twenty two billion. It is also expected that this number will double in the next 4 years. This data indicates that the next big step in Internet expansion will be the wide integration of smart objects(COUNCIL, 2008b). The infrastructure of these smart things will comprise hardware, software, services and the communication channel between all of these(COMMUNITIES, 2008).

The new facet of the Internet will not be the people, but our everyday "things"(COMMUNITIES, 2008) . In this scenario the things will be everywhere, sensing our surroundings and interacting with the physical world to make our lives easier. These things will be communicating using wireless networks, making the existence of extensive Wireless Sensors and actuator Networks (WSNs) a reality. The omnipresence of such WSNs is what we will call the Internet of Things (IoT).

The IoT concept can be described as the presence of smart-objects spread all over the place with a unique address scheme (ATZORI; IERA; MORABITO, 2010). These things generally are able to communicate with neighboring nodes to achieve an arbitrary goal. The main aspect of the IoT concept is the high rate of adoption by people in their day to day routine. It is expected to be mostly noticeable in health care, assisted living, home automation, manufacturing, and many other fields. The problem that comes with the massive adoption of such "things"is that the number of possible threats rises proportionally to the number of devices on the market. The security risks present on the interactions of this everyday object could do more damage than the Internet has done to date.

Iot infrastructure can be divided into three layers(ZHAO; GE, 2013), the first layer, which works with comprehensive perception and is responsible for the data gathering anytime and anywhere, is called the perception layer. The second layer works with the network and guarantees data is updated safely and in real time or with a reliable transmission. The third layer is related to the application involved and works with pre-processing a data package before sending it to any other node. The main focuses of this dissertation lay in Layers 2 and 3.

The devices we are considering in this work are low cost sensors connected to a wireless sensor network spread on the environment making part of an IoT. This type of network could use the common

TCP/IP standard, but as pointed out by (XIAO; WANG; YANG, 2006) the abstract characteristic of TCP/IP networks is incomplete and inefficient for the security in WSNs. Another approach for constructing WSNs is the use of more concise communication protocol design based on a cross-layer approach.

A crosslayer-design for multi-hop wireless networks shares parameters in each layer preventing multi-layer attacks(KHAN; LOO; DIN, 2009). The main advantage a cross-layer approach has is that it generates less routing overhead and less exchanges of acknowledgement packets than the standard TCP/IP based networks. These advantages facilitate handling the important WSN features such as high energy efficiency, connectivity and hardware limitations that make environment sensing harder. The next step of the WSN is to integrate heterogeneous communication technologies in order to become more related to the Internet of Things(IoT).

According to Kevin Ashton(ASHTON, 2009) the Internet of Things shows potential to change the world just like the Internet did. Back in 2009, all the Internet content, around fifty petabytes, was mainly captured by human beings. The problem is that people are not as accurate as a computer and they have limits. They can't stay focused on some environmental situation twenty-four hours a day, seven days a week. Therefore, we need to push forward sensor technology so that computers can have a better awareness of the world, and all this collected data can become strategic information. The need for IoT security is clear.

IoT security is no longer just a future problem. It is a problem right now according to Cisco studies(CISCO, 2015). These studies indicate that between 37% and 47% of enterprises all over the world are planning to deploy IoT infrastructures and that between 29% and 38% of the same sample deployed IoT infrastructures over the last three years. There are different elements required to enable IoT solutions like Wi-Fi, real-time location tracking, GPS Tracking, security sensors, etc.

The IoT applications are susceptible to internal and external security issues like malwares, malicious software, network attacks and external hackers. As stated by Cisco, 28% to 47% of enterprise organizations have experienced security breaches in their deployed infrastructures. The numbers of IoT breaches has been increasing exponentially over the last two years and has reached a state where security must be a functional requirement for all applications that will run on this technology.

The study of IoT security is paramount to the creation of a

safer environment for us to live in. We need the creation of tailored communication protocols that envisage, among other things, security by design. In this sense we have seen the creation and deployment of cross-layer communication protocols with embedded security features. A missing part of this puzzle is the verification of the guarantees claimed by the protocol designers and their implementations.

1.1 MOTIVATION

Our motivations lay in the fact that most of the IoT design is being done without security in mind and that security is a feature that ends up being built in after the “thing"is done. There is a lot of space for things to go wrong as already discussed in the previous section. Even if security is embedded at the time of design, it is not always straight forward that the security will follow throughout the lifetime of the device.

With that in mind, our motivation is to demonstrate the feasibility of formally verifying an IoT security protocol that claims a secure design. We would like at first to use state of the art formal verification techniques to verify the key distribution scheme for a cross-layer IoT communication protocol. This verification can detect flaws or corroborate the key features of such protocol. Thorough study of such protocol can further develop its goals and identify the security scenarios where these goals can be claimed.

As a second part of our work we would like to give these things the ability to deploy an access control scheme that is based on context. The capability of deciding on infrastructure security based on contexts is something we take for granted but that is not usually available for our devices. In this sense we would like to extend the cross-layer security protocol of choice so that it can have a security by context Access control policy and that this policy is formally verified using state of the art formal verification tool.

1.2 JUSTIFICATION

The TCP/IP model has a blackbox design that hides internal information about the functioning of each layer(FU et al., 2014). Applying this model to IoT has been confirmed to be a big mistake(MIYOSHI; SUGANO; MURATA, 2002) (BALAKRISHNAN; KATZ, 1998), because so-

metimes it is more valuable to use the layer's hidden data to diagnosis network problems than to just disconnect it and try to reconnect again. One solution is the adoption of Cross-Layer Designs, which are able to evaluate the relationship between layers. These designs have a wide reconfiguration capability, too.

There are a lot of works on the Cross-Layer Designs area and some of them try to broaden IoT requirements(DUNKELS; ÖSTERLIND; HE, 2007), but generally focus on improving packet routing, MAC and QoS. Only a few works look to security needs. These designs usually maintain the modularity present on the original layered TCP/IP standard. Features that most of the works do not cover are geo-location applications and the characteristics involving this segment, such as spatial localization, time synchronization, geographic routing and security.

Most of the protocols that are available for IoT today only focus on simpler security goals and usually only achieve secrecy and authentication. Due to the intrinsic physical nature of the sensors and actuators that make up part of an IoT environment, it is expected that these things are able to make decisions not only made by observations but by context. In this sense we believe that adding context aware access control strategies to IoT security protocols can make these devices more usable and prepared for the real necessities that surround people.

We have chosen for this work the Trustful Space-Time Protocol (TSTP) (RESNER; FROHLICH, 2015). This protocol was readily available as the focus of this study of a close-by group and brings all the characteristics that we were aiming at formally verifying . The TSP brings a coupled key exchange and authentication scheme that was never verified. It also has a powerful implementation(RESNER; FROHLICH, 2015) and reasonable deployment in daily scenarios around the UFSC Campus in Brazil (RESNER; FRÖHLICH, 2015). These characteristics make the protocol an ideal candidate for us to verify and extend.

1.2.1 Objectives

Our objectives were to choose a communication protocol for IoT environments focused on applications with security requirements. We then formally verified the protocol to see if it could be trusted. We then designed and formalized a context-aware access control policy to guarantee secure operations.

1.2.2 Specific Objectives

- Describe TSTP's key bootstrapping protocol architecture using a formal language;
- Describe TSTP's time synchronization protocol architecture using a formal language;
- Verify the protocol's description into Proverif;
- Gather the policy requirements of UFSC's University Restaurant;
- Represent the policy scenario with all the features needed;
- Define the Requestor and Resource Contexts, Protection and Active Contexts for the policy based on Proteus Context Model;
- Describe the policy using a formal language;
- Verify the policy's description into Proverif;

1.2.3 Publications

The following publications are results of our research work:

1. We published the article "Formal Verification of a Cross-Layer, Trustful Space-Time Protocol for Wireless Sensor Networks"(SILVA et al., 2016a) at ICISS 2016;
2. We will submit the article "A Context-Aware Access Control Scheme for the Internet of Things"for ARES 2017

This dissertation includes direct text from the two publications. These text were re-arranged to avoid repetitions and to make sense in this dissertation format.

1.2.4 Methodology

The methodology we used to achieve our goals was:

1. Review related work: this step aims to gather theoretical knowledge about the Internet of Things and the protocols related to this segment. A bibliography research about Protocols for Internet of Things, Protocol Analysis and Context-Awareness will be performed.

2. Review of the state-of-art: in this step we analysed co-related works found on the literature. For the first goal we analysed the surveys involving the IoT, Cross-layer Protocols approach for wired and wireless networks and for the second goal we analysed chapters of a book about access control policies for IoT environments and access control surveys. After that we were able to understand more about our research questions and this step made the path for achieving these goals easier.

3. Describe the TSTP using Proverif : this step was very important for the formal analysis. In this step we did iterations of formal description, trying to get the description as representative of TSTP's protocol as we can.

4. Analysis of Proverif Output : in this step we run TSTP's formal description on the Proverif tool. The output of the tool was very helpful to better represent the protocol and after that we were able to find the security breach.

5. Fix Protocol breaches: in this step we used our experience in security to guide us to a better solution to the found breaches. The first sub-step was to make changes to the protocol to fit our possible solution and test it again on the tool. The second sub-step was to change some structures within the protocol, to follow literature advice.

6. Draw a conclusion : this step was meaningful to understand what we could achieve after the goal achievement.

7. Representation of Access Control Policies: in this step we picked the most representative access control policy to represent using a formal language. After some representation on pen and paper, we started to formally describe the policies.

8. Policy Analysis: in this step we test the formal description against an attacker. We were able to see that it is hard to verify the policy's behavior with the known tools. The output of this analysis was that the sensitive data used on the policy remained secret.

1.2.5 Contributions

Our first contribution is a formal verification of the security aspects of the Trustful Space-Time Protocol, an application-oriented,

cross-layer WSN protocol. We specified and verified TSTP's time synchronization and key bootstrapping protocols(RESNER; FRÖHLICH, 2015) in the automatic cryptographic protocol verification tool ProVerif.

From the automatic analysis, we were able to find a subtle attack that would allow an attacker to effectively impersonate the gateway to a sensor node and read all (assumed) private and authenticated data it sends. We propose two possible punctual alterations in the protocol to counter-measure this security flaw, preventing this attack. Furthermore, we proposed another topical change not detected as a flaw by ProVerif, but that would increase security: a change of a MAC-then-Encrypt method employed to Encrypt-then-MAC.

Our second contribution so far is that we have selected our case study scenario and modelled it using a web ontology language to understand the necessary features. We described our access control context models based on Proteus(BOTTAZZI; MONTANARI; TONINELLI, 2007) models that are prepared for the context-aware variability needed. These descriptions will help us to formally represent the policies in a proper way so that we can analyse it against policy violations and conflicts using a theorem prover.

1.3 STRUCTURE OF THIS WORK

This thesis is organized in 5 chapters. Chapter 1 presented a contextualization of this work's scope. We selected our research goals, the generic and specific objectives, we showed our publications and the methodology followed for this thesis.

Chapter 2 shows a theoretical background about IoT Protocols, the Trustful Space-Time Protocol(TSTP), Formal Analysis and Access Control in IoT environments.

Chapter 3 is the corrected version of our first published article.The Chapter 4 is the corrected version of our second submitted article. We choose to remove some parts of the articles to reduce background content repetition, however it will have context repetition between the article and the thesis.

In Chapter 5 we show our conclusions, and we relate our two goals with the objectives. In this chapter we share our vision of future work, too.

2 CONTEXTUALIZATION AND LITERATURE REVIEW

In this chapter we will be reviewing the background information needed for the reader to understand our propositions, as well as to locate our work within the current state of the art of the area.

This section is a compilation from two papers we wrote with the results. We used this section to condense the background information and relate the work for the two topics in this dissertation.

2.1 IOT CONTEXTUALIZATION

Internet access by devices reached more than a billion users in 2010 and it is expected that by 2020 fifty billion gadgets will be accessing the Web(CERP-IOT, 2010). An incredible revolution was observed in the way that interconnected devices generate information by sensing environment variables(GUBBI et al., 2013). This environmental sensing requires new approaches and new techniques to manage the myriad of information flows generated by the Internet of Things.

In our work we will consider that the objects or things connected are sensors or actuators. All this hardware is connected using wireless communication technology, forming a Wireless Sensor Network(WSN). This kind of network could use the standard TCP/IP model(FU et al., 2014) as the standard "Internet of People" does. This would provide standard data encapsulation, but this model does not take into account particular WSN characteristics, such as resource-constrained devices, ad-hoc connectivity, the need for high energy efficiency, notions of the environment, and mainly the security needs inherent to objects interacting with the physical world. These open problems have been motivating researchers to propose different protocol designs. It has been shown that cross-layer designs are a great alternative for optimizing WSN and wireless networks in general(MENDES; RODRIGUES, 2011), fulfilling the mentioned requirements. Cross-Layer designs are built so that each layer's parameters can be shared directly with other layers to reach optimization goals, breaking the traditional black-box, stacked model.

As defined by Atzori et al.(ATZORI; IERA; MORABITO, 2010), IoT is a network of interconnected things on a world-wide scale. These things have a unique address and communicate with each other using

standardized WSN protocols. Aligned with the definition of this research area, we have the challenges that come with it, like the integration of things, computable resource restrictions, connectivity, energy efficiency, security and operation. Hence, there are many particularities about the inception of this area that have not been solved yet.

The Trustful Space-time Protocol(TSTP)(RESNER; FROHLICH, 2015) we chose to analyze is an application-oriented, cross-layer protocol with synchronized time, spatial localization and distribution of sink-node keys. This protocol intimately integrates multiple components that share data. TSTP was motivated by the problems observed in WSN and IoT, and aims to deliver directly to the application a data-centric API, trustfulness, geo-referencing, space-time synchronization and energy efficiency at the communication level. We chose this protocol because of its capacity to establish keys after deployment. This characteristic brings a specification and verification challenge.

We use the ProVerif(BLANCHET et al., 2010) tool to analyze the security aspects of the TSTP. ProVerif has already demonstrated valuable results for many cryptographic protocols in the literature(BLANCHET; ABADI; FOURNET, 2005). This tool models attackers and protocols using applied pi-calculus to describe message exchange. These descriptions are then converted to Horn clauses and are proven mechanically using a First-Order Logic theorem prover.

2.2 LITERATURE REVIEW AND RELATED WORK

The design of an authentication WSN/IoT protocol involves many critical decisions. We will explain the building blocks needed (§2.2.1) and two approaches that in our opinion outsource protocol security responsibilities. The first one is related to protocol assumptions (§2.2.2). The second one is about the protocol trust relations (§2.2.3).

2.2.1 IoT Cryptographic Algorithms

The selection of the right security algorithm to be used with WSN/IoT communications is not a trivial task, mainly because of the constraints related to code and data size, processing power, and energy consumption(WANG; ATTEBURY; RAMAMURTHY, 2006). Encryption algorithms are classically classified using symmetric or asymmetric cryptography(FAQUIH; KADAM; SAQUIB, 2015). Although this seems trivial,

it is important to discuss the impact of such choices when dealing with WSN/IoT protocol designs.

2.2.1.1 Symmetric Key Cryptography

This kind of cryptography uses the same key for encryption and decryption. These cryptographic operations are usually relatively simple and efficient. RC4, RC5, IDEA, are examples of symmetric cryptography algorithms.

Because WSN devices must usually manage constrained resources, the simplicity of symmetric cryptography algorithms is very desirable. The security scheme chosen must not significantly affect the performance of the network(KIRUTHIKA; EZHILARASIE; UMAMAKESWARI, 2015). The selection of algorithms for encryption and decryption to be employed takes into account parameters such as the size of the operands, modes of operation, and key expansion. The type of cipher depends mainly on how large the volume of data is to be processed through the network. Stream ciphers deal better with large amounts of information, while for smaller traffic, block ciphers may be a suitable option.

In this sense it is not only important to choose a symmetric algorithm, but to choose it based on the context of the data being collected within the WSN/IoT deployment scenario. A wise decision is to choose symmetric algorithms that can leverage good hardware support from the base platform of the nodes.

2.2.1.2 Asymmetric Key Cryptography

This cryptographic concept works with pairs of keys: everything that one key encrypts the other decrypts and vice-versa. This method yields more robust cryptographic schemes, but many authors agree that the required data size, code size, processing time and power consumption generally make this kind of cryptography impractical for WSN/IoT deployment scenarios (WANG; ATTEBURY; RAMAMURTHY, 2006).

Nevertheless, with the advent of Elliptic Curve Cryptography (ECC) schemes, it has been shown that it is possible to achieve a good trade off between all these factors and the resulting level of security provided(WANG; ATTEBURY; RAMAMURTHY, 2006). Although ECC operations are costly, they achieve the same level of security as

other asymmetric schemes such as RSA using smaller keys, resulting in smaller overhead and power consumption(WANDER et al., 2005).

Some special deployment scenarios, especially when the identity of peers must be attested to the whole network, require the use of public key cryptography. It is important to note that these algorithms usually do not come with efficient hardware implementations and they usually are heavy multiplication based. With ECC we swap integer modular exponentiation by logical operations representing addition and multiplication over finite fields.

2.2.2 Pre-established Information Protocols

Wireless Sensor Networks are usually ad-hoc networks. The topology is not fixed or pre-determined, and nodes may enter or leave the network during its lifetime. Such networks require a dynamic key assignment, that will use sensors' data for key generation and establishment. Thus, the use of pre-distributed information as a strong parameter on key generation is not advised by literature(RAJESWARI; SEENIVASAGAM, 2016). The generation of life-long keys at fabrication time is also not recommended. When doing that, the security root will stay centered on the institution or company that injected this key material into the sensor. Hence if there is a data leak, all those who consider this information trustful are potentially compromised. To avoid that, we prefer to use unique data available only to the sensor and the sink to provide a key establishment scheme. This strategy we call pre-established information protocols.

In the literature, several authentication protocols for WSN have been proposed with the premise of pre-distributed key material. The Lightweight Dynamic User Authentication Scheme (LDUAS)(WONG et al., 2006) relies on a previously-defined user name, password and time period to establish key material. The sensor authenticates to the sink with this information and the key material is set out of that. So if any of this data is weak in terms of entropy, or if the attacker gets this info somehow, the protocol is insecure.

The Localized Encryption and Authentication Protocol (LEAP) provides multiple mechanisms involving keys, which are capable of providing confidentiality and authentication(ZHU; SETIA; JAJODIA, 2006) to messages. This protocol has individual keys shared with the base station, a pair of keys that can be shared with other Wireless Sensor Networks, a key to exchange information with neighboring nodes and

a group key. However all of these mechanisms are mainly based on a pre-distributed key to create a secure communication channel and to generate these multiple keys. If someone discovers that key, the attacker will have knowledge about the other keys too. This protocols lacks a property called forward-secrecy.

The Lightweight Authentication Scheme (LAS) for WSN only uses an HMAC hash function and a set of encryption algorithms using symmetric ciphers to provide confidentiality and authenticity to the messages exchanged(DELGADO-MOHATAR; FÚSTER-SABATER; SIERRA, 2011). This protocol is divided in three phases, starting with the pre-distribution of keys, explicitly specifying that the manufacturer must insert a master symmetric key at the time of sensor manufacturing. Therefore the trust is not only in the manufacturer but also in the manufacturing process, and even in the employee who carries out the procedures, because a forgery in any one of these steps can compromise any network installed with this scheme. The other two phases are inherently dependent of the first one, making it the main point of failure for the protocol.

The Node Level Security Policy Framework (NLPSF) uses node information for authentication and group keys based on identity-based cryptography(CLAYCOMB; SHIN, 2011). The protocol's initialization is divided into four parts, the second of which being the initialization of the sensor node with a data set. This data set contains public information representing the group to which this sensor belongs, characterizing a static group assignment. The data set also contains an identity-based key. Hence this approach not only relies on the integrity of pre-deployment information, but also relies on a fixed and pre-defined network topology, which is generally not suitable to ad-hoc and mobile wireless networks.

Security protocols based on pre-established static information for the seeding or derivation of keys for nodes are inherently corruptible at the time of production of the node. More over, basing the new keys derived later on the pre-established keys will also lack forward-secrecy of this key material. Nevertheless, a lot of people opt for such protocols because they are easy to design and easy to deploy, even if not delivering ultimate security.

2.2.3 Trusted Third Party Protocols

The literature contains numerous WSN protocols for authentication using digital certificates, public key cryptography and shared key cryptography. The main characteristics of these protocols are that they rely on the presence of a trusted third party to be the dealer of the protocol. These protocols can prevent man in the middle attack(MitM), because they can identify the identity of the two sides of communication. However, these checks may impair the speed of packet switches or create a bottleneck on the side of the institution responsible for checking.

The Efficient Authenticated Key Establishment Protocol (EAKEP)(VIJAYAKUMAR; VIJAYALAKSHMI, 2008) uses elliptic curve cryptography (ECC), which is recognized in the literature by providing the desired level of security with smaller keys, low computational complexity and high-speed cryptographic operations. This protocol uses digital certificates to protect itself from impersonation attacks. Therefore all nodes must be connected to the Certification Authority (CA) for testify the identity of this sensor and every message exchanged needs communication between base station and CA.

The Multiuser Broadcast Authentication scheme (MUBA)(REN et al., 2009) proposes four methods based on public key cryptography to provide different benefits and respond to different constraints. Nonetheless there are several methods of implementation for this authentication scheme. All of them still need an authority (trusted third party) to certify sensors and answer for their identity. However this protocol has its importance in literature, because it is not common to present various certification schemes focused on IoT.

The design of trusted third party protocols in the WSN/IoT context is not usual. Mostly because it will inherently yield a more complex deployment scenario, including the new type of peer. Another big issue with these protocols is that they usually rely on public-key cryptography, which is very costly to the constrained environment of the sensors.

In section 2.3 we will be describing the Trustful Space-Time Protocol (TSTP) (RESNER; FROHLICH, 2015) security features. This protocol called our attention because it deals with communication and key establishment using the very efficient schemes of symmetric cryptography, coupled with once in a life-time use of asymmetric cryptography for master secret establishment. It does not rely on trusted third parties, and by using a synchronized time seed to generate sessions keys,

it provides very strong forward-secrecy properties.

2.3 THE TSTP PROTOCOL

The Trustful Space-Time Protocol (TSTP) (RESNER; FROHLICH, 2015) is an application-oriented, cross-layer communication solution for WSN and IoT, ranging from the application layer to the link layer. TSTP handles geographic information inherent to the network (such as time and space) as much as possible, including its key generation protocol. TSTP defines a key generation protocol between sensor nodes and a central node (*gateway*, or *sink*).

WSN devices communicate through wireless technology, allowing any radio interface configured at the same frequency band to monitor or participate in communications – which is very convenient for attackers. In order to avoid attacks, a secure infrastructure must provide the principles of *confidentiality*, *authenticity* and *integrity*(SUO et al., 2012). TSTP provides these principles as well as temporality, while not requiring a trustable third party. It relies on unique sensor IDs, precisely synchronized clocks, and time and place of deployment as information shared between gateway and sensor.

Although we are mostly interested in the security verification of the key distribution part of TSTP, the next subsections present some key components of the cross-layer protocols that are used in the setting of establishing key material. We will present the time synchronization scheme (§2.3.1), address and positioning scheme (§2.3.2) and the key distribution scheme itself (§2.3.3).

2.3.1 Time Synchronization

TSTP's Speculative Precision Time Protocol keeps clocks in the network synchronized with sub-microsecond precision(RESNER; FRöHLICH; WANNER, 2016). TSTP has two non-exclusive modes of time synchronization: speculative and explicit. Speculative synchronization happens every time a node receives a TSTP message. Since TSTP defines the MAC component that controls directly the physical layer, and since fine-grained, MAC-level time stamps are pigtailed in every TSTP message, a receiver of any message can determine its clock offset in relation to the sender without the exchange of any extra message.

With the reception of at least two messages from the same sen-

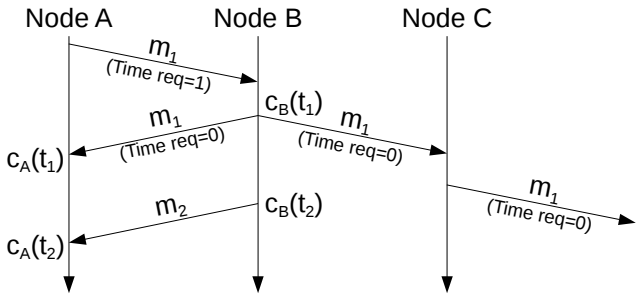

Fig. 1 – TSTP explicit time synchronization.

der, receivers are also able to estimate their frequency error with high accuracy(RESNER; FRöHLICH; WANNER, 2016). Since clocks in sensor nodes drift from each other over time (even if once synchronized), speculative synchronization is carried out for every received message, and its accuracy is proportional to the amount of traffic in the area of the network in which a given node is located.

The second synchronization mode is used when a node can't afford to wait for eventual messages to synchronize with a given precision, and consists of the transmission of an explicit "Time request" message. This message is replied to by a neighbouring node twice, so that the requestor node can extract the two time stamps necessary for synchronization. Figure 1 illustrates the explicit mode where Node A requests synchronization with message m_1, which is replied twice by Node B. In this case, m_1 is not only a Time request, but a message destined to a node to the right, and so it is forwarded normally through Node C. $c_N(t_i)$ represents a read on the local clock of node N at physical time t_i.

2.3.2 Addressing and Positioning

TSTP's location estimation is also done passively on every message that a node overhears. The position estimation algorithm is based on the Heuristic Environmental Consideration Over Position (HECOPs)(REGHELIN; FRÖHLICH, 2006), which uses multilateration and Received Signal Strength Indication (RSSI) measurements.

To boost accuracy, HECOPS introduces confidence values and heuristics to estimate environmental effects on the radio signal, effectively boosting the estimation's accuracy. Figure 2 depicts the "devia-

tion" heuristic: when two highly confident nodes (e.g. nodes equipped with GPS) detect that the RSSI estimation between them is off, they inform neighbor nodes about this offset, so that they can apply it to their own estimations. In this figure Node A and B are anchors and detect that their estimated distance via RSSI is wrong by a coefficient dev_{AB}. They broadcast this information so that C can apply the same coefficient to its estimations (representing that the area inside the triangle is under environmental interference).

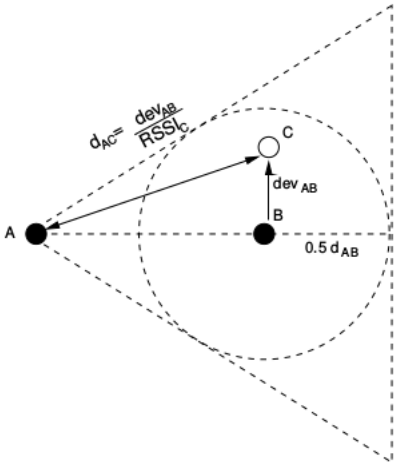

Fig. 2 – HECOP's deviation detection.

In TSTP, every message carries the geographic coordinates of the sender node, such that any node overhearing the network for long enough may harvest enough information to estimate its own coordinates without the injection of any extra message. Furthermore, estimation is done continuously, and its accuracy tends to get better with each new message overheard.

It is important to note that addressing of nodes within the TSTP cross-layer protocols is based on their actual position in space. This makes positioning important to our evaluation because this is how nodes are addressed in the WSN setting.

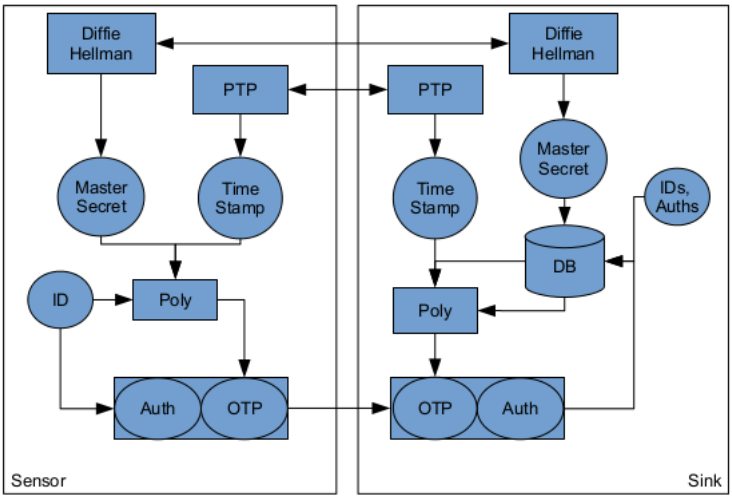

Fig. 3 – Overview of interactions between blocks of TSTP's key establishment protocol.

2.3.3 Security

The key bootstrapping protocol can be explained by these sequence of events. First it happends a mutual authentication and key establishment, with an ECDH agreement, between one sensor and the gateway. After that the next step is a One-Time Password(OTP) operation using Poly1305. The OTP is sent with the id encrypted by the id, that is basically the Auth operation. For authentication purposes, the sink do a database fetch seeking for the correspondant ID, received as Auth on the last step. When that operation matches with the stored identifiers on the database, the key generated by both sides has a confirmation that this key was only shared with the real node and the real sink. After this steps every message exchanged is encrypted using a new OTP, together with a MAC of the message.

TSTP's key bootstrapping protocol's architecture is illustrated in Figure 3. It involves the Speculative Precision Time Protocol (Section 2.3.1) to precisely synchronize clocks; The Heuristic Environmental Consideration Over Position for addressing the nodes (Section 2.3.2); Elliptic Curve Diffie-Hellman to establish strong asymmetric key pairs;

AES for lightweight encryption/decryption of messages; Poly1305-AES (BERNSTEIN, 2005a) and unique sensor node IDs for authentication via One-Time Password (OTP);

The Poly1305 is highly indicated by the research area. There is a proof of security level that we can achieve using this algorithm (BERNSTEIN, 2005b). Some of the user responsabilities are defined by Bernestein to maintain the security level. For TSTP the identifier used on each ID should be unique and the key generated for each device should be unpredictable, we are using Diffie-Hellman so this conditions are not a problem for TSTP. The last constraint is that this algorithm needs a parameter that is used just one time, this is why we used timestamps.

Figure 4 details the operations carried out by the protocol. Operation 1 is the part responsible for the choice of Diffie-Hellman parameters. The G parameter is the base point of an elliptic curve. The p variable is a prime that defines F_p, the prime field in which the protocol operates. n represents the order of the group used, being proportional to p, and its size will be the size of the keys that will be generated. The IDs correspond to sensor unique identifiers large enough to be considered secure.

Operations 2 and 3 are related to the size of the key and its generation. The private key (K_s) is a random integer less than n and the public key (P_s) is derived from the multiplication of the private key value and the base point G. Operation 4 generates a hash value of the sensor's ID, which is also known to the gateway.

At step 5, the two interested parties' clocks are synchronized. The protocol is agnostic to the exact synchronization method used. After that the sensor stays waiting for the message of operation 6 (DH Request) that will come from the gateway with its public key P_g at the window of time defined for that sensor's deployment. When the sensor receives this request, it will send a DH Response message (7) with its own public key P_s.

The master secret K_{sg} is calculated with the multiplication of the public key sent through the network and the private key of the sensor in operation 8. In operations 9, 10, 11, the sensor prepares a One-Time Password with the Poly1305-AES algorithm(BERNSTEIN, 2005a) using three inputs: the sensor's ID, the master secret just established and the current time. The purpose of this request is to tie the master secret to a sensor ID, effectively ensuring to the gateway that K_{sg} was established with a trusted sensor node. The time stamp included protects this message from replay attacks.

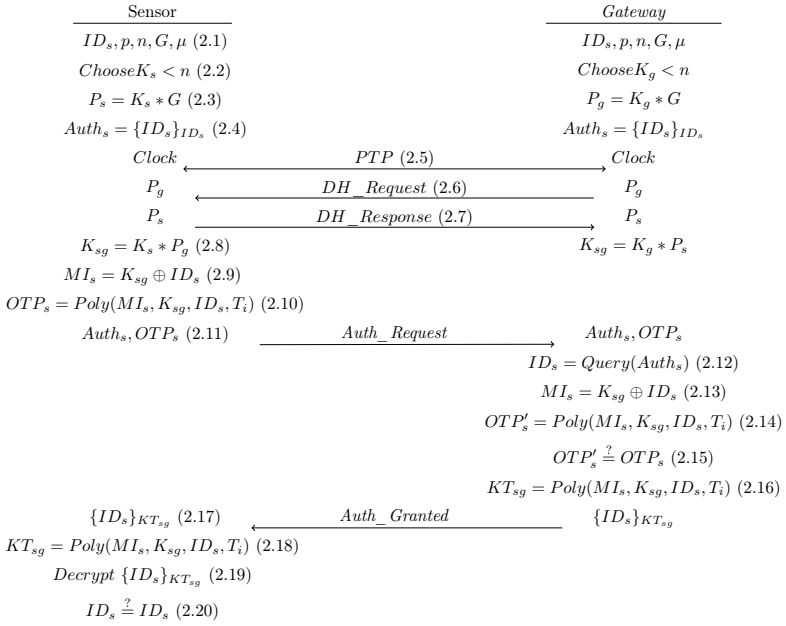

Fig. 4 – Protocol Operations.

Upon reception of the Auth Request message, the gateway tests all the data calculated by the sensor (operations 12 to 15) and then sends back a confirmation to the sensor if it passes. In operation 12 the gateway verifies in its database if there is an ID that can decrypt Auths, and the result of this decryption is its own ID. At step 13 another information is calculated on the sink side by using the xor of the master secret and the ID found in the last step. The next two steps show that the Auth request information was sent from a valid sensor to the gateway. After that, in operation 16, the sink asserts that the authentication was successful sending a confirmation message.

The message contains the sensor's ID encrypted with a disposable key, which is derived from the master secret and the ID itself at operation 17. In the next operation the sensor generates a key with its own parameters. It then tries to decrypt the Auth Granted message at operation 19 and finds its own ID in the last operation. This way, it has evidence that K_{sg} was in fact shared with a party that knows the ID, assuming only the gateway this far. At this point, the parties have

synchronized clocks and shared an authenticated master secret K_{sg}.

The master secret is not used directly as an encryption key. A disposable key is generated each time a message is sent, just as in operations 8 to 10, which is used for encryption. Figure 5 depicts the process of sending secure messages.

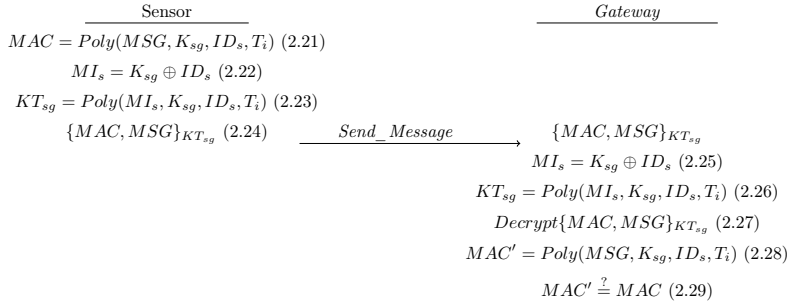

Fig. 5 – Sending a confidential, authenticated, and timed message.

2.4 ACCESS CONTROL IN IOT ENVIRONMENTS

The efforts of the security research area have been directed toward the design, development and improvement of lightweight security mechanisms. The optimizations are focused at the physical, the network or the application levels, or concentrated into account constrained resource availability of smart objects. However, there are important characteristics of IoT that have been forgotten. For example, heterogeneity, dynamicity and extreme variability of operating conditions. These features raise security management issues, associated with the need to dynamically adapt the exploited security mechanisms and enforced policies to achieve the best dynamic trade off between overhead/costs and security levels. Adaptation is a crucial open problem to address for security management in IoT environments.

From now on, software architectures design focused on IoT environments, should be prepared for the heterogeneity of smart objects and networks, and should provide service provision for many different applications and users on the most diverse contexts(PREUVENEERS; BERBERS, 2008). Context awareness is important to these infrastructures because it will allow service customization with minimal human

intervention (PERERA et al., 2014). Analysing and understanding relevant context data, regarding the user, will be fundamental to originate a new range of smart entertainment and business applications totally user-friendly(DEY, 2001).

2.4.1 Literature Review

The management of access control in IoT environments addresses complex characteristics like their heterogeneity and the high dynamicity of the smart objects where we can apply policy-based approaches. The policy management research area was initially focused on large-scale problems such as enterprise-wide and Internet-wide systems(SLOMAN; LUPU, 2002)(BOUTABA; AIB, 2007). In these kind of applications, policies have been extracted from observations of network administration tasks, such as security, recovery, configuration, quality of service, etc. The policies are defined as means to regulate the behavior of system components in a dynamic way, without changing any part of the code and sometimes without any warning system to the components being governed. Therefore, making adjustments in policies can update or even change the imposed constraints in order to help the system adapt itself to environmental conditions.

According to Bradshaw(BRADSHAW; MONTANARI; USZOK, 2014), there are a number of strategies to define policies and policy languages. The novel policy languages, in IoT environments, should be able to ensure a proper level of expressiveness, because when the system has a wide range of policy requirements it will certainly need a high level of expressiveness while being efficient at the same time. The enforcement of access control policies can be applied with different approaches, too, relying on the limitations of the IoT devices on the network as they may not have enough computational power to implement complex access control mechanisms(ROMAN; ZHOU; LOPEZ, 2013). There is a standard for policy management frameworks designed by IETF and DMTF. This pattern is not aimed for IoT, but it should work if the deployment requirements match the pattern and if the efficient allocation of Policy Decision Points(PDP) and Policy Enforcement Point(PEP) are distributed following the standard, then any change on that jeopardizes the performance and scalability of the system.

3 FORMAL VERIFICATION OF A CROSS-LAYER, TRUSTFUL SPACE-TIME PROTOCOL FOR WIRELESS SENSOR NETWORKS

3.1 PROTOCOL SPECIFICATION AND FORMALIZATION

In this section, we specify the Trustful Space-Time Protocol (TSTP) (RESNER; FROHLICH, 2015) using ProVerif (§3.1.1). Out of that Specification we conduct a protocol verification assisted by the tool (§3.1.2).

3.1.1 ProVerif Protocol Specification

Our specification effort was focused on the description and evaluation of the security components of TSTP. We specified the secure key establishment protocols and the later communication process using the shared keys generated in this first phase.

We initialize our ProVerif specification informing that the channel where TSTP data pass through is an open channel, vulnerable to tampering, eavesdropping and other threats. This is a standard procedure that puts the protocol on the setting of a reasonable threat model for where it will actually be executed.

We then define a unique ID for every sensor and that its size is at least 128 bits, because it needs a good entropy as well as being hard to guess. After that we define a timestamp value, which is a parameter of the one-time-password that avoids the generation of the same key twice.

The parameters of Elliptic Curve Diffie-Hellman (ECDH) functions are represented by G, x and y exponents. These parameters establish how hard it will be for the attacker to solve the discrete logarithm problem for elliptic curves. The security of key derivation is proportional to the size of elliptic curve keys. We considered that the ECDH key has at least 256 bits and with that assumption we are able to derive a 128 bit key, that will be used with AES to encrypt and decrypt messages.

3.1.2 ProVerif Protocol Verification

We were able to find a security problem related to the time synchronization part of the protocol, as well as a possibility of improvement in the message authentication scheme. We explain next the main steps that the ProVerif tool was guided through to find the security flaw. In the end of this chapter we evaluate the advantages that our proposals bring to TSTP.

Listing 3.1 – Sensor's Process

```
let OTP_t0 = Poly(xor(masterSecret, id), masterSecret, id, t0) in
event sensor_request_auth(id);
out(c, (Auth(id), OTP_t0));
```

As shown in the listing above, the problem was found on the first sensor's time synchronization. The flaw is related to the Auth Request message. At operation 11 (Figure 4), two pieces of data are sent: the first one is the ID encrypted by itself and the second part is an unencrypted One-Time Password that is crucial for the attack found.

Within the Proverif specification we were able to find a way for attackers to impersonate the gateway and send seemingly legitimate messages to the sensor. To do this, the attacker must follow three basic steps:

1. Synchronize its clock to the network;
2. Store an Auth Request message from a given sensor s containing OTP_s, as well as the precise time t where it was sent;
3. Wait for an Auth Granted response from the gateway to s;
4. Manipulate the clock synchronization algorithm to make the clock of sensor s become t again;
5. Send a message to s encrypted with OTP_s (stored at step 2), or read the messages that s sends.

Steps 1 and 4 are generally possible since TSTP does not secure clock synchronization. In fact, it is not trivial to do so because TSTP relies on synchronized clocks to provide security in the first place. Step 2 is possible since OTP_s goes through the network as plain text. It is wrong, however, to assume that all plain data that passes through the

network can cause an easy to see problem: even ProVerif took some steps to discover the subtle flaw.

An attacker can thus successfully impersonate the gateway to a sensor node. It cannot, however, impersonate a sensor node to the gateway if the gateway's clock is the reference clock for the time synchronization protocol (i.e. the network synchronizes to the gateway's clock, and not vice-versa).

Evaluating the protocol we can see another issue that happens the moment that a message is sent, involving the order of the Message Authentication Code (MAC) and encryption operations. This was not actually found by the use of Proverif, but as a result of the reasoning involved in the specification and verification efforts.

This order is important, with each ordering covering different properties(BELLARE; NAMPREMPRE, 2008). There are two possible orderings:

- The MAC-then-Encrypt (MtE) ordering consists of first generating a MAC from the plaintext and then encrypting the result. This provides plaintext integrity and does not leak any information about the plaintext (because it is encrypted), but does not provide any integrity on the ciphertext;

- The Encrypt-then-MAC (EtM) order consists of first encrypting the message, and then generating a MAC from the ciphertext. This provides integrity of both ciphertext and plaintext, and does not leak any information on the plaintext, assuming the output of the cipher appears random and there is no output from the receiving end indicating whether the MAC was valid or not(BELLARE; NAMPREMPRE, 2008).

Therefore EtM ensures that you only read valid messages. With this method, if one modifies the ciphertext or tries to extend it, that will result in an invalid MAC. EtM also protects against the padding oracle attack(PATERSON; YAU, 2004), because decryption of messages with invalid MACs are prevented.

3.2 PROTOCOL RE-DESIGN AND PROPOSED SOLUTIONS

We propose two solutions to this problem. The first one is a change in the time synchronization algorithm. TSTP is built on top of an IEEE 802.15.4 physical layer. The 802.15.4 standard dictates that the oscillator used as clock by radio hardware must have an accuracy of

at least ±40ppm. Some WSN systems based on this standard (such as Texas Instruments' CC2538 SoC[1]) propagate this accuracy to the system clock, so that any node in the network, once synchronized, knows for every further synchronization operation the upper bound $\bar{\phi}$ of its drift:

$$\bar{\phi} = \alpha \times (t_i - t_s)$$

where α is the oscillator's accuracy (e.g. 40ppm), t_i is the current time and t_s is the time at which the last synchronization happened. Any synchronization that attempts to adjust the clock by an amount δ, such that $|\delta| > |\bar{\phi}|$, can be detected as violating the physical limit given by the oscillator's accuracy, and therefore rejected. This method drastically limits the attacker's ability to manipulate the sensor's clock, preventing step 4 of the detected attack.

The second solution is to adapt the security protocol itself: TSTP could use the same messages and data, but with one more instance of encryption. The OTP that is packaged within the Auth Request message could, instead of being transmitted as plain text, be encrypted by the master secret K_{sg} that is already established between sensor and gateway. The rest of the protocol would be carried out identically, since the gateway already has to try every K_{sg} pending authentication in steps 12-15 (Figure 4). This security measure would turn the mentioned attack infeasible because we assume that the attacker is not able to break ECDH in a reasonable time to find K_{sg} and therefore decrypt the Auth Request message to find OTP_s. Figure 6 shows the proposed changes.

To use the EtM pattern we will only have to modify the send message part of the protocol. We will do operations 22 and 23 (Figure 4) before the generation of MAC tag (operation 21 of Figure 4), after that we will encrypt the message, generate the key to do the encryption and then we will generate the tag.

Operation 30 (Figure 6) impacts the security level and the sensor's power consumption. With our improvement the attacker is incapable of impersonating the gateway. However, the sensor has to encrypt one more data packet at key establishment. Considering that this will happen one time for each sensor, this impact is amortized by the rise of the security level. Operation 31 does not affect the protocol, because its execution is on the gateway side where power consumption is not a problem.

[1] www.ti.com/CC2538

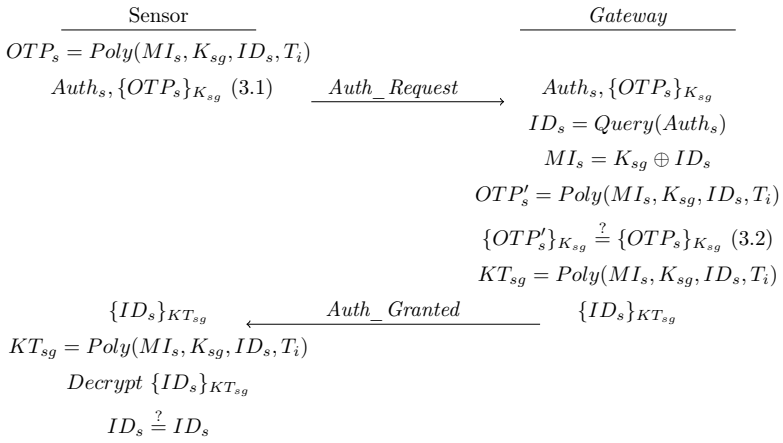

Fig. 6 – Proposed changes to the key establishment protocol.

3.3 CONCLUDING REMARKS

In this work, we provided a formal verification of the security aspects of the Trustful Space-Time Protocol, an application-oriented, cross-layer WSN protocol. We specified and verified TSTP's time synchronization and key bootstrapping protocols(RESNER; FRÖHLICH, 2015) in the automatic cryptographic protocol verification tool ProVerif.

From the automatic analysis, we were able to find a subtle attack that would allow an attacker to effectively impersonate the gateway to a sensor node and read all (assumed) private and authenticated data it sends. We propose two possible punctual alterations in the protocol to counter-measure this security flaw, preventing this attack.

Furthermore, we propose another topical change not detected as a flaw by ProVerif, but that would increase security: the change of a MAC-then-Encrypt method employed to Encrypt-then-MAC.

In summary we could verify that TSTP uses unique IDs as well as time and space as implicitly shared information between the parties during key establishment. Because the protocol does not derive keys exclusively from the ID, security holds even if they are deployed with low care (RESNER; FRÖHLICH, 2015). In TSTP, an attacker must not only find a correct ID, but deploy a malicious node at the right time

and place(RESNER; FROHLICH, 2015). Moreover, ID leakage does not compromise any keys already established.

The TSTP uses a pre-distributed identifier on each node, but not all the security relies on this information(RESNER; FROHLICH, 2015). The protocol specifies a secured minimum size for that data, making brute force attacks harder to perform. There is a step where the sink will be loaded with sensor ids information. This step will allow the base station to test if the node that is trying to authenticate has a valid id or not.

The TSTP does not need a Trustable Third Party(TTP) and certificates to identify who is sending the message. The protocol is not susceptible to full MitM, because the attacker cannot impersonate a sensor. To perform a full MitM the attacker needs to impersonate the gateway to the sensor and then they will exchange a key with each other. After that the attacker needs to impersonate a sensor, too, and start to send messages to the base station. However, he can not impersonate a sensor because he does not know a valid identifier.

As future work, we intend to design an extension for TSTP that would allow for key distribution within groups. Such groups should be space-time constrained so that the key establishment and the surrounding protocol would use a mechanism of geo-encryption.

4 A CONTEXT-AWARE ACCESS CONTROL SCHEME FOR THE INTERNET OF THINGS

4.1 INTRODUCTION

According to Prognostics (CERP-IOT, 2010) prediction, fifty to one hundred billion devices will have Internet access close by 2020. By 2025 it is expected (COUNCIL, 2008a) that Internet-connected nodes will be in human's every day things, such as food packages, furniture, paper documents and etc, leading to network of always/intermittently connected objects in a world-wide scale, i.e., the Internet of Things (IoT).

One of the major obstacles to wider uptake of the IoT in the real world is security. IoT security challenges are many and with some elements of technical originality if compared with other deployment environments and application domains. The threats that may affect IoT environments are various and at various levels (physical, application, communication, ..), ranging from attacks to the confidentiality, integrity, and authenticity of communication channels to denial of service, identity fabrication, malicious access to objects, and unauthorized control over IoT entities.

Up to now, related security research efforts have been mostly directed towards the design and development of lightweight security mechanisms and optimizations at the physical/network/application levels, by mainly taking into account constrained resource availability of smart objects. However, the nature of IoT (heterogeneity, dynamicity, and extreme variability/unpredictability of operating conditions) raises other relevant security management issues, associated with the need to dynamically adapt the exploited security mechanisms and enforced policies to achieve the best dynamic trade-off between overhead/costs and security levels. Adaptation is a crucial open aspect to address for security management in IoT environments.

To achieve adaptation, an emerging trend is the adoption of context-awareness as the guiding principle for supporting adaptive IoT security management. For example, based on context, such as available battery, CPU power, type of network connectivity, communication performance indicators (e.g., bandwidth, latency, packet loss), application- and environment-specific risks, and trust level of interworking components, it should be possible to decide which authentication and authorization mechanisms to apply to IoT objects and to drive authentication

and access-control decisions. In particular, access control for IoT environments can greatly benefit from a context-aware approach. Location (i.e., checking from where users/devices are accessing the services offered by a smart object, either locally or remotely) and other context information (such as user profiling data, operating environment conditions, and IoT device status) are crucial elements of access control policies in IoT scenarios.

In context-aware access control models, permissions are typically associated to specific context: when an entity/object operates in a specific context, it automatically acquires the ability to perform the set of permitted actions in the current context. Context-aware policies can be created and managed without direct reference to potentially numerous subjects and objects, but only by predetermining which subject/object context attributes have to be considered for granting/denying access to resources. Context-aware policies avoid the need for explicit authorizations to be directly assigned to individual subjects prior to an access request, thus paving the way to easier dynamic adaptation of access control policies.

However, the deployment of context-aware access control models in real cases has to face several complexity factors, ranging from the adoption of proper policy definition languages to the design/development of suited context management and policy enforcement support. Among the various issues, one main problem that needs to be addressed is to verify how violations to context information impact on policy applicability. Being policies strongly dependent on context information, it is crucial to control which attacks can comprise context information leading to policy abuse. Formal analysis is needed to ensure that every context conditions that are present in policy specifications and that govern policy applicability of policies are correct.

This paper addresses this issue by applying the ProVerif (BLANCHET et al., 2010) tool to verify the symbolic model of the context-aware access control model, called Proteus (BOTTAZZI; MONTANARI; TONINELLI, 2007). In the literature (BLANCHET; ABADI; FOURNET, 2005) ProVerif is known as a automatic cryptographic protocol verifier. This tool can handle a lot of cryptographic primitives and it accepts unbounded sessions with a myriad of messages. It models attackers and protocols based on pi-calculus and these models are then converted to Horn clauses, that can be proved mechanically using a First-Order Logic theorem prover.

Our main contributions in this work are twofold. First we motivate the need for a context-aware access control model within a Smart

Campus scenario where food at the restaurant may be subsided based on other actions from the students, such as having classes on that day or being above a certain level of attendance, and describe several examples of needed context-aware access control policies. The depicted scenario is pictorial and directed to our real campus scenario at our University, but it can be easily adapted to other ones with reasonable simplicity. Our second contribution is the formal specification and verification of an adaptive security policy based on Proteus using Proverif. This specification and verification strategy verifies the common security goals involved in the security policy, such as, confidentiality of message exchanges, as well as guaranteeing that no leak of sensitive information occurs. We can also demonstrate with our formalisation then non-interference of context-changes on the achievement of security goals. This work on formal verification of the security policy can help us to understand it better and to check that it is complete and secure.

In the next section of this paper we will see a Section (§4.2) on Novel Access Control Models for the IoT and Challenges for Verification Tools were we cover the necessary related and parallel work to oursa as well as give some motivation for our work. On Section 4.3 we describe our pictorial Smart Campus scenario where the dynamic context-aware access control policy is deployed and how it influences on the IoT device's behaviour. Section 4.4 brings use cases and security policy descriptions to the scenarios presented on Section 4.3. On Section 4.5 we describe our specification and formalisation strategy for verifying our security policy goals using Proverif. And finally, on Section 4.6 we draw some concluding remarks as well as we summarise our finding and propose some future work on the field of IoT security policies and their formal verification strategies.

4.2 NOVEL ACCESS CONTROL MODELS FOR THE IOT AND CHALLENGES FOR VERIFICATION TOOLS

Heterogeneity, resource constraints, the size, and the high dynamicity of IoT environments complicate the engineering and the management of access control. Embedded devices need to be low power and may have limited connectivity. The size of IoT ecosystems require solutions capable of scaling properly with limited overhead and the high degree of IoT dynamicity requires adaptive security solutions.

IoT is often characterized by high mobility of nodes across different physical/administration domains; IoT devices experience continu-

ous changing, not only of connectivity conditions and network topologies, but also in their collaborating peers. Collaborating devices cannot be statically pre-determined and pre-identified. In such dynamic scenarios, a security challenge is how to decide whom to trust in the plethora of opportunistically encountered entities and how to govern access control to the resources of collaborating entities. In addition, whereas traditional distributed systems rely on a relatively static characterization of the operating conditions and accessible resources, where changes are relatively small, rare, or predictable, the continuous modification of the visibility and availability of collaborating smart objects is the rule in the IoT.

Novel access control models are needed to properly handle the new management issues posed by IoT characteristics. A still open debate in the field covers the issue of which access control model can be considered appropriate for the IoT with only few proposals been put forward. Among them a very few rely on role-based and attribute-based models whereas most solutions adopt capability-based models (GUSMEROLI; PICCIONE; ROTONDI, 2013). A novel recently emerging trend is to follow a context-aware policy-based management approach to separate subject identities along with their set of attributes from the set of privileges needed to operate on resources. Coupling context-awareness with policy-based management can address the security adaptation requirements of IoT ecosystems.

In a context-aware access control approach subjects acquire set of privileges depending on the context where they operate, not only on the basis of their identities, or roles, or only static attributes. In addition, applying the policy-based management approach to IoT environments can bring the benefits commonly recognised in traditional Internet systems (SLOMAN; LUPU, 2002; BOUTABA; AIB, 2007).

Policies are defined as means to regulate the behavior of system components in a dynamic way, without changing any part of the code and sometimes without any warning system to the components being governed. Therefore making adjustments in policies can update or even change the imposed constraints in order to help the system to adapt itself to environmental conditions. According to Bradshaw (BRADSHAW; MONTANARI; USZOK, 2014), there are a number of different strategies to define policy languages. The novel policy languages, in IoT environments, should be able to ensure a proper level of expressiveness, because when the system has a wide range of policy requirements it will certainly need a high level of expressiveness being efficient at the same time.

The scenarios where we will deploy our context-aware access control policy will involve money payments for meals as well as attendance control at university level. Both of these requirements are too sensitive for using standard strategies such as observations and failures. We need a formal verification approach to guarantee the soundness of the policy before deployment. We also have a requirement for having the policy to dynamically adapt itself (values of meal) based on other contexts captured by our smart campus infrastructure.

There are not tools designed specific to perform the access control policies formal verification, but there are verification tools for protocol analysis that can be used for this purpose. These tools help the design of security protocols to be less error prone inserting them on promiscuous environment. This insertion make it possible to apply access control policies on the worst contexts scenarios and it shows how secure the policy is. We looked at three automatic tools Tamarin, TA4SP and Proverif. Tamarin seems to be slower than Proverif for secrecy and authentication purposes, because of its modelling granularity that implies in a more complex analysis(LAFOURCADE; PUYS, 2015). The TA4SP tool has problems to deal with exclusive-or properties and its efficiency is slower than Proverif too. Therefore for this work we choose to work with Proverif knowing that it is not a tool made for access control policy verification, but it is capable of analyse the policy execution as a protocol.

4.3 SMART CAMPUS SCENARIOS

In our university it was detected that the students were having serious attendance problems in the classroom but the same thing could no be said for the university restaurant. The university restaurant is maintained by university administration and it has a meal subsidy for students. This subsidy can reduce the meal cost to less than U$0.50 per meal.

The two main problems we have then are: the low attendance for classes, and the misuse of university cards for access to the subsided restaurant. The university then decided to add two new access control systems using an existing Smart Campus strategy, which are a in-classroom racket for controlling attendance and rackets for access control and charge at the university restaurant.

Nowadays they operate independently only allowing for behavioural data collection from the users. Our intention is to devise an

adaptive context-aware access control policy that would allow for dynamically controlling the subsidy for the restaurant based on other campus activities, such as class attendance. This policy must be subject of a formal verification strategy because it will involve the money charge for the restaurant, as well as to minimize the impact on the user in case of policy conflicts.

4.3.1 Use Cases Description

For the sake of simplicity we can consider two students presents in the use cases Alice and Bob. Alice is a student that has classes in the morning until middle-day and right after lunch time. Hence she has to have lunch really fast, because she had to attend classes in the morning, after that she has to queue to have lunch at the university restaurant and she has to attend classes in the afternoon too. Bob is a student that only has classes in the morning and he usually eats at the university restaurant. Alice and Bob have very different approaches for attending lectures. Alice goes to all the lectures and do not skip any, she may even skip lunch to be able to get to class 5 minutes before the starting time. Bob intermittently misses some classes. He often arrives late and leaves early. Bob never skip a cheap lunch at the university restaurant.

4.3.2 Requirements and Design Guidelines

To properly describe the scenarios later on, the university administration has given us a set of requirements and design guidelines for the new context-aware access control scheme. These were not our choices, but administrative guidelines given to us. They are:

1. The university should be able to limit the the subsidy given on student meals provided by the restaurant service based on different contexts.

2. We need a way to determine presence of the student inside the classroom and use this in our context.

3. The students should have an identification card for their authentication within university access control systems, as well as, to carry a money transaction with the university restaurant.

4. The policy in place should assure fairness and only be deployed with the correct assurances that no student will be charged more than specified on university regulations.

5. There should be not obvious way to subvert the policy either to get better attendance ratio, as well as getting more subsidy.

Although we are only demonstrating in this work the class attendance context, the actual university scenario is much more complex. It includes the ability of the student not being able to use his card twice per meal service, avoiding the card being lent to other. It also may change the charge based on other IoT collected data, such as parking use, day of the week and weather conditions. We are leaving some of the complex IoT integration behind and only focusing on demonstrating the feasibility of our context aware model approach.

All the hardware for policy deployment was already in place. The restaurant rackets and the classroom rackets are using the an IoT communication protocol and deal with mifare cards, as well as they take photos of the users at each card touch so that fraud can be detected. University is aware of mifare cards attacks(GARCIA et al., 2008), but the legacy of 50.000 issued cards puts its change out of scope for now.

The main tasks given to us were to devise the access control policy and to formally verify it against security goals.

4.3.3 University Restaurant Case

On subsection (4.3.4) we will describe a scenario of a University restaurant that serves subsided meals for students and is controlled under a Smart Campus infrastructure based on IoT motes. This restaurant serves more than 11.000 meal a day and is one of the main social policies drivers for our university.

We then present some other ideas in the Smart Campus domains (§4.3) where we can see that IoT and our proposed Context-Aware Access control Scheme can be applied. These scenarios allows for the application of some direct social policies within the university as well as to understand how the campus behaves as a whole.

4.3.4 Access Control to a University Restaurant in a Smart Campus Setting

We chosen a setting we have available at our university which is a University Restaurant that serves subsided food to students and staff under certain conditions. The subsidy varies from 100% to 0% depending rules established by the university.

4.3.4.1 User Groups

Students that have classes everyday all day long are allowed the highest subsidy (90%). These students need to be enrolled on lectures during the morning and the afternoon or the afternoon and the night since the restaurant servers lunch and dinner. This group of students is considered not to have the right conditions to pursue other meal elsewhere. It is important to say regarding this group of students that enrollment does not mean attendance, what would actually be the a fairer criteria the university would like to use.

We also have students that are enrolled on lectures everyday only one period a day. These students are allowed a 75% subsidy. This group has the same motivation for the subsidy as the previous. Also enrollment does not mean attendance. This group has also a peculiarity that they may be enrolled only partially on the period, meaning they may start mid-morning or mid-afternoon. In this late case, they should carry the subsidy for the the dinner instead of lunch. The other meal has 60% subsidy.

We have another group of students that are enrolled less than four days a week. This group of students are allowed a 60% subsidy on the days they have lectures and 30% on the other days. The same rules regarding enrollment versus attendance are present here. There is no differentiation on whether the student is enrolled all day or part time.

We have a fourth group of students which are those considered economically vulnerable that are allowed a 100% subsidy on all their meals, on the rule they keep attendance above 75%. In fact the 75% attendance rule is valid for all the four groups, meaning the student must be attending lectures on a regular basis to have the right for the subsidy. Drop outs with valid enrollment, or students on any sort of leave are not allowed to have access to the restaurant.

Another class of users for the University restaurant are adminis-

trative and academic staff. This group of users is allowed a single meal a day with a 70% subsidy. There is not other rule regarding this group of users.

Finally we have the group of temporary contractors and general visitors or users of the university community services. This group of users are not allowed any subsidy, meaning they have to pay full price. It is important to notice that full price in the University restaurant is very attractive, since the university aims no profit on those meals. This group only need to be registered with the University Restaurant where that access conditions will be assessed by the managerial staff of the restaurant and will be set for this group.

There are also some system wide rules for accessing the University restaurant that are also encoded on our access control policies: Only one subsided meal is allowed per user per meal time, avoiding that the user lends his card to someone else. The security officers that monitors the access rackets need to perform identity matching for the ownership of the card that allows for access.

4.3.4.2 IoT Infrastructure in Place

The university restaurant has a physical access and credit control system which is integrated to the other ICT systems using an wireless sensor network (WSN) setting. This WSN setting is composed by a series of IoT motes that communicate between themselves and an Internet connected sink to establish the enforcement of the access control and charging rules we described above. In this work we consider IoT objects acting either as sensors or actuators and communicating via wireless communication technologies, thus constituting a Wireless Sensor Network (WSN).

The access control system is mainly based on the usage of MiFare (GARCIA et al., 2008) cards from NXP. To add credit to these cards the users should go to an on-line system and request the adding of the amount they want by purchasing it on-line. These credit will be added to their card on their next interaction with the system.

Although we know the security issues of MiFare cards (GARCIA et al., 2008), this was part of a legacy system, and all the precaution were taken to avoid attacks. These precaution include the diversification of card keys using a Key Derivation Function for the card, the usage of application specific keys for read-only and read-and-write access privileges and the use of back-office processing to validate the credits system.

This is standard industry procedure when using MiFare cards.

At each of the entrances for the university restaurant there are a series of access rackets that are used to physically control the access to the restaurant. To these access rackets the user must present his MiFare card, which hold the information to start the access control decision procedure. These access rackets are instrumented with IoT motes that execute a variation of the TSTP protocol(RESNER; FRÖHLICH, 2015). The Trustful Space-time Protocol(TSTP) the university choose to work with is an application-oriented, cross-layer communication protocol with accurate time synchronization, heuristic spatial localization and distribution of sink-node keys. TSTP was motivated by the needs observed in WSN and IoT, and its target to deliver directly to the application a data-centric API, trustfulness, geo-referencing, space-time synchronization and energy efficiency at communication level. The main choice for this protocol is because it is capable of establishing security keys after deployment of the system and it comes with a formal specification using Proverif (SILVA et al., 2016b).

These smart access rackets operate using a ad-hoc communication scheme based on the zig-bee protocol and forward information towards a sink that is interconnected with the on-line university systems for credit management and for attendance. For resilience of the whole systems, each smart access racket also holds a local partial implementation of the access control policy so that in the event of loss of communication, either within the ad-hoc network, or the university systems, the racket can operated independently and fairly allow for the users to enter the restaurant. The infrastructure is shown on Figure 7.

It is important to note that some rules apply dynamically to the system and that information is first managed locally. For example the rule regarding one subsided meal per meal time can be determined within the IoT network events if the connection with the university systems is down. This requires a special encoding in terms of access control rules.

Another important requirement fulfilled by this IoT infrastructure is that it needs to provide timeliness for the operations, since we provide the security officer now with a screen where the photos of the users are shown when the card is read. In this sense he does not need to randomly choose some users to check identity, since it is automated and can be done for everybody.

We also have in our deployed Context-Aware Access Control a scheme to collect data thought a series on IoT connected terminals (also rackets) where students register their attendance. With the attendance

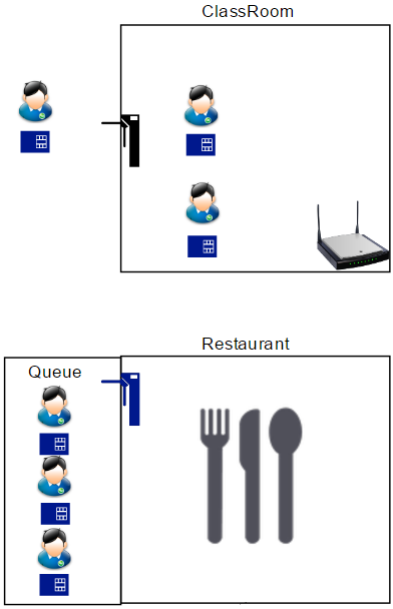

Fig. 7 – Smart Campus scenario.

data collected we were able to establish some interesting context-aware access restrictions. The attendance registration terminal are on the student entrance door of some lecture rooms and is used for the student to register his/her entrance and exit to get the minimum attendance requirement for our courses.

The class attendance is now one scenario in place, but the university is studying others. Some other data collection points that can increase or decrease subsidy for the university restaurant may be the use of car parking space (decrease the subsidy), or using carpool parking spaces (increasing subsidy).

4.4 USE CASES AND POLICY DESCRIPTION

In this section we briefly lay down the use cases for our context-aware access control policy and work over the policy description before doing the formalisation in the next section.

4.4.1 Use Cases

The use cases described down below were important to built our access control policy. With this info we could split what should have in each context element. Therefore the sentences are:

- A student that pass his smartcard on the racket in classes entrance and after the authentication the racket can confirm that his position inside the classroom was confirmed for at least 30 minutes, confirming class attendance.
- A student that attended to classes in the period(morning or afternoon) and he has classes all day long, he is allowed to eat(lunch AND dinner) in university restaurant with 90% subsidy.
- A student that attended to classes in the period(morning or afternoon) and he has classes only in one period, he is allowed to eat(lunch OR dinner) in university restaurant with 75% subsidy.
- A student that attended to classes in a period(morning or afternoon) and he has classes less than four days a week, he is allowed to eat(lunch or dinner) in university restaurant with 60% subsidy on classes days.
- A student that attended to classes in the period(morning or afternoon) and he has classes less than four days a week, he is allowed to eat(lunch or dinner) in university restaurant with 30% subsidy on days without classes.
- A student that is economic vulnerable has the right to eat(lunch or dinner) in university restaurant with 100% subsidy.
- Administrative and academic staff users are allowed to eat a single meal in university restaurant with 70% subsidy.
- A group of temporary contractors and general visitors or users of the university community services are not allowed to any subsidy, they have to pay full price to to eat(lunch or dinner) in university restaurant.

Gathering the information of this use cases we can accomplish access control properties and the policy description. We did not implemented every piece of content because the tool used for analyse our

policy model is limitant on expressivity, but it can be used for unbounded sessions. Therefore the next step was to specify our context elements and after that do the formal analysis.

4.4.2 Policy Description

The policy model description resulted from the use cases's abstraction and we had to make a few changes on that model targetting the formal analysis. Firstly we abstracted the required roles, considering that students, academic staff and administrative staff should be differentiated. Secondly we thought about the IoT devices present on our symbolic model environment, this part we could follow two approaches, one way was to create a myriad of representational devices or we could stick with the most representative ones. The tool used to do the policy verification it is powerfull, but has its own limits, so we had to built a description thinking about that too.

We differentiated the roles using identifiers and we generalize them to just one process. We applied the University diversification operation to generate identifiers that discriminate roles in our model. We could have one process per role, but this would make the formal analysis part take much more time and computational power. Therefore we generalize the attributes and data behind a role to build a requestor process that encompasses everything.

We started our symbolic model with two required IoT devices that are the class racket and the restaurant racket. The Class Racket was required to characterize one of the contexts that our context-aware system should judge. The Restaurant Racket represents the IoT device responsible for charge the requestor with less or more subsidy. However after the implementation of this two devices as two different processes we found similarities between these things and other IoT devices, for example if we need to describe a gadget that will count how many litters of water each student is drinking, we could represent with the same description of class racket process just changing a few things.

We ended up having four things as context that are the attendance, the requestor's campus, amount of credit and the role. The attendance is verified by the class racket and it could be read by restaurant racket. The campus and role identifiers are granted on the moment that the student's card is issued. The credit could be acquired at restaurant's racket and after the subsidy analysis it could be subtracted according to University scenario rules. So now that the policy

features were listed and it has a symbolic model better abstracted, we could specify it using the Proteus model and do the formal analysis.

4.5 POLICY SPECIFICATION AND FORMALIZATION

We split the Smart Campus Model into several properties, we will explain how we abstracted that and how TSTP helped us. There are properties involving user location, class and work attendance and authentication devices. We shrunk the Smart Campus model in a certain way that made the formal model easier and clear to be understood by Proverif. The obstacles on gathering precisely time and location were reduced by using TSTP features.

4.5.1 Access Control Policy Specification

To fully understand the solution to be proposed, let us briefly recall some characteristics of the underlying Proteus model (BOTTAZZI; MONTANARI; TONINELLI, 2007). Proteus is a semantic context-aware policy model that is centered around the concept of context as the key element of policy specifications, where context means any characterizing information about controlled system entities and about their surrounding world. Contexts act as intermediaries between entities and the set of operations that they can perform on resources. In the Proteus model a policy is a rule that defines, for each context, how to operate on its associated resources. In particular, policies can be viewed as one-to-one associations between contexts and allowed actions. Contexts are associated with the resources to be controlled and represent all and only those conditions that enable access to resources. Hereinafter we call protection contexts the contexts that allow operations on resources. In particular, a protection context consists of all the characterizing data and meta-data that is considered relevant for access control, logically organized in parts that describe the state/properties of the resource to protect (resource context), the characteristics/properties of the entities requesting access to the resource (requestor context), such as their roles, identities or security credentials, and time conditions, such as the interval time allowed for operating on the resource (time context).

We subdivided requestor context into three main context elements that are physical, computing and user. The first one is concerned about the things physically related to the requestor, like time and

space. This part was made it easier by the properties already available with TSTP. The second one refers to the devices that a user has with him or that he will activate. The third one characterize a person, like the correspondent identifiers and personal information about the subject. In this work the Protection Context and the Resource Context will be important too, the protection element is triggered when the requestor context properties matches the resource element constraints.

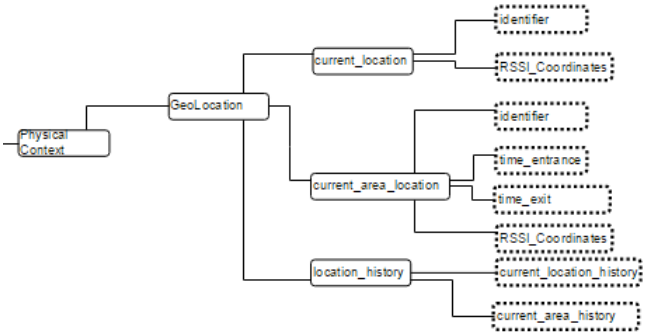

Fig. 8 – Physical Context Specification

The specification of our context-aware access control policy starts with the Physical Context shown on Figure 8. This context element is composed by Geo-location properties, that involves not only location but time too. The concepts within this element are represented in Proverif by campusID identifier and it is verified on every operation that the requestor done. A practical example could be if some student is in a different campus trying to get lunch, he can't get the greater subsidy even attending to class, because in the policy context a requestor has major discounts only on the campus that issued his IDs.

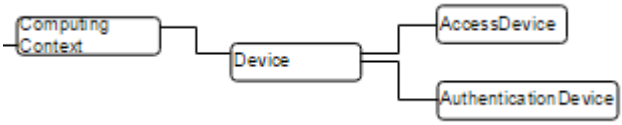

Fig. 9 – Computing Context Specification

The second specification part was about the Computing context and is shown on Figure 9. This element was represented at the sym-

bolic model as a smart card that has credits on it. The smart card and restaurant tickets are represented by creditsAtCard variable. This variable can only be written or read by restaurant's racket otherwise any other interested entity could change credit quantities. We imposed that to make our policy centralized within the restaurant racket, there are other kind of data written by others entities, but the restaurant racket can still see and update this datum.

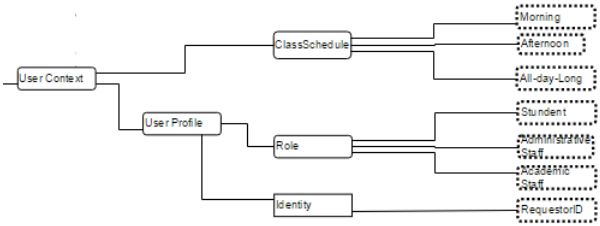

Fig. 10 – User Context Specification

The third specification part refers to the User context as shown in Figure 10. We implemented this element with almost fully representation of all sub-elements shown in picture 10. We did that because in an access control scheme the user or in our case the requestor will respect or disrespect every single policy rule. This is the element that will have different contexts and it will impact in our context-related in how good is our policy context-awareness.

Our protection context element can be described joining the relevant variables for trigger restaurant's subsidy that is our resource element. The first one protects our application and the University against resource waste. The second one is only enjoyed if the set of requestor attributes follow the protection context rules. The rules are explained with the formalization of our access control policy. We will give a brief explanation about how does the proverif code works.

4.5.2 Proverif Code

We are assuming the reader is not proficient in Proverif, so we bring a brief explanation of its operational semantics. For those versed on applied pi-calculus and Proverif syntax, this section can be skipped.

In Proverif we can create types, execute all logic operations, and

we can execute some arithmetic operations too. All these operations can send results through public and private channels. We are also allowed to subdivide the process execution into smaller ones to get the granularity we want. We should define queries for verifying if a secret data remains secret and other properties we want to verify. The user can define how the flow of events should happen and how the events can be co-related.

All the types and constants necessary for the formal verification should be created at the begining of the script. For defining a type using proverif language the user needs to just write "*type X*"being x the name of this type. To declare a constant the user should inform explicitly the type, for example "*const y: X*"meaning that y is a constant of type X. To declare variables the syntax is "*free name:Type*", but if you want to define this variable as secret you should use "*free name:Type[private]*"for example. After types declarations of functions can be created.

A funcion in Proverif should have parameters and a return value. A function that accepts a string and key as a parameter, and it returns string can be declared as "*fun funname(string,key): string*". For calling functions you just need the name of the function and the data required as parameter. There are some special functions that do not need to be declared as input(in) and output(out) functions. When needed by the user to pass information through channels these functions can be called as "*in(channel,data)*". However sometimes you need to do attributions and “if"statements to keep the flow of your application as intended.

If and else statements are declared in the same way as in other programming languages. Proverif allows the user to test multiple logic operations with boolean algebra. The variable attribution follow the same rules plus a prefix "*let*", for example "*let variable = value*". There is another way that is attributing random value to a variable like "*new value:int*". The flow of the executions for the specification is controlled by events.

Events exists to control the flow of the specification and queries inform to Proverif what should be tested. If for some reason the security protocol or security policy you are testing needs to follow a specific sequence of events, you can chain them with "*event(one) ==>event(two)*". A query can be declared as "*query attacker(data)*" meaning that, all the effort done by Proverif attacker should target to obtain "data". After these explanations we can easily explain our implementation's decisions at policy formalisation.

4.5.3 Threat Modelling

Proverif being mainly used in cryptographic protocol verification assumes a threat model base on the Dolev-Yao specification (DOLEV; YAO, 1983). The Dolev-Yao attacker has full control of the setting where he is in. He is able to learn everything that is exchanged between other peers, being also capable of intercepting, forging and blocking the delivery of any communication. The attacker can also play as an insider to gain knowledge in one run and leverage that in another. The Dolev-Yao attacker is only constrained by cryptography and he can not guess random numbers.

In our smart campus setting this attacker is probably the ona that would be able to derive the most of the attacks that could be conceived, including standard man-in-the-middle and oracles attacks. So it makes sense to verify our context aware polices against it. Moreover, it is the advisable to demonstrate that no leak of private information happens, and well as not interference from an attacker can occur in the security policy setting. This will help us to verify the security of the context-aware policy.

4.5.4 Access Control Policy Formalization

We focused our specification on getting the best policy representation and we prioritised relevant responses for the formalisation. We evaluated in each step if the formal description was as close as possible to our policy primitives. However some results could not be proved in Proverif, so we had to manage to demonstrate that the right operations happened on the right time showing a good response without the actual result value.

In Proverif the first thing that should be done is the definition of communication channel and after that the definition of some auxiliary functions. We start specification on Proverif with the public channel where the messages will pass through, that can be tampered, it is susceptible to eavesdrop and other threats as explained in the previous section. The functions necessary for the entire proof are subdivided in Elliptic Curve Diffie-Hellman (ECDH), Symmetric Encryption and some conceptual arithmetic functions.

The Elliptic Curve Diffie-Hellman (ECDH) will be represented by G, x and y exponents and it is analogue to one the University uses in the real scenario. All these values define how hard it will be to solve

the discrete logarithm problem on the attacker side. We mainly use the ECDH to simulate a University process called diversification. This operation creates a unique key based on subject's card serial number. The security of this key derivation is proportional to the size of elliptic curve keys and these will be used to encrypt or decrypt data on our formal description.

We implemented the symmetric encryption and decrytion functions following the description on ProVerif's manual(BLANCHET; SMYTH; CHEVAL, 2015) and we added extra functions. These functions were applied with integer values and in a certain way with boolean values. It was also required to use a type converter, which is also described in the Proverif manual. Our specification of the algorithms and the converter is shown in Listing 4.5.4.

Listing 4.5.4-1 – Diffie-Hellman and Symmetric Encryption

```
(*****Diffie-Hellman*****)
        type G.
        const g : G [ data ] .
        fun exp (G, exponent ) : G.
        equation forall x:exponent, y:exponent ; exp(exp(g, x), y) = exp(exp(g, y), x).

        fun G_as_key(G) : key [data, typeConverter].

(*****Symmetric Encryption*****)

        fun senc(int, key) : bitstring.
        fun sdec(bitstring, key) : int.

        fun bsenc(bool, key) : bitstring.
        fun bsdec(bitstring, key) : bool.

        equation forall m : bool, k : key; bsdec(bsenc(m, k), k) = m.
        equation forall m : bitstring, k : key; bsenc(bsdec(m, k), k) = m.

        equation forall m : int, k : key; sdec(senc(m, k), k) = m.
        equation forall m : bitstring, k : key; senc(sdec(m, k), k) = m.
```

We then described the requestor process looking for to the credits that any requestor had and, if he attended or not to a necessary appointment which could be a class or work. The creditsAtCard is a value encrypted by the restaurant racket's key, so this variable can travel through the public channel without problems. The attendanceAtCard is a boolean value encrypted by classes racket's key, it will be further explained why this maintains the centralized approach at restaurant's racket. It is shown in Listing 4.5.4 how the first write in the two variables happens. It is important to remember that demonstrating the actual value of creditsAtCard is different from the new value it is enough to show that the variable was updated and the same works for attendance control.

Listing 4.5.4-2 – Requestor Process

```
let requestor(sk:exponent, id:ID, roleID:G,campusID:G)=
(*Requestor Key Generation*)
        phase 1;
        let pkRequestor = exp(g,sk) in
        out(c, pkRequestor);

(*First Credits Value*)
        phase 2;
        in(c,(creditsEncrypted:bitstring,id_req:ID, req_CampusID:G));
        if id_req = id && campusID = req_CampusID then
        let creditsAtCard = creditsEncrypted in

(*First Attendance Value*)
        phase 3;
        in(c,(attendanceValue:bitstring,id_areq:ID, att_CampusID:G));
        if id_areq = id && campusID = att_CampusID then
        let attendanceAtCard = attendanceValue in

(*Class Authentication*)
        phase 4;
        out(c,(creditsAtCard,attendanceAtCard,id,roleID,campusID));

(*Attendance Update*)
        phase 5;
        in(c,(newRequestorAttendance:bitstring, id_class:ID, classCampusID:G));
        if id_class = id && campusID = classCampusID then
        if attendanceAtCard <> newRequestorAttendance then
        event attendanceUpdate(id);

(*Restaurant Authentication*)
        phase 6;
        out(c,(creditsAtCard,attendanceAtCard,id,roleID,campusID));

(*Credits Update*)
        phase 7;
        in(c,(newRequestorCredits:bitstring, id_s:ID, reqCampusID:G));
        if id_s = id && campusID = reqCampusID then
        if creditsAtCard <> newRequestorCredits then
        event creditUpdate(id_s).
```

We also specified the attendance racket process keeping the centralized architecture approach. We made it capable of judging when the requestor attended or not the appointment required by the context-aware access control policy. The process starts with two exchange of ECDH keys, one to have a unique key to handle the requestor information and other to assure that every thing that was done by this racket can still be read by restaurant's racket. Therefore the context behind the attendance evaluation is that, if the requestor is a student and if it is on the right campus, then the attendance will be true, this can be observed at Listing 4.5.4. In our specification we just created rackets for classes, but it can be generalized to compose all the others appointments that the University is interested in monitory for the context-aware access control.

Listing 4.5.4-3 – Class Racket Process

```
let attendanceracket(sk:exponent,stuRoleID:G, campusID:G)=

(**Key Generation**)
        phase 1;
        let pkClassRacket = exp(g,sk) in
        in(c,(pkRequestor:G,idRequestor:ID));
        let classRacketMasterSecret = G_as_key(exp(pkRequestor, sk)) in

        phase 2;
        in(c,(pkOtherRacket:G,idOtherRacket:ID));
        let racketToRacketMasterSecret = G_as_key(exp(pkOtherRacket, sk)) in
        insert rackets_and_keys(racketToRacketMasterSecret,idOtherRacket);

        phase 3;
        new attendance:bool;
        let firstAttendance = bsenc(attendance,classRacketMasterSecret) in
        out(c,(firstAttendance,idRequestor,campusID));
(*Attendance Authentication*)
        phase 4;
        in(c,(creditsAtCard:bitstring,attendanceAtCard:bitstring,id_req:ID,roleID:G,req_CampusID:G));
        get authentications_and_keys(classRacketMasterSecret, =id_req) in
        event classAuth(id_req);


(*Attendance Update*)
        if roleID = stuRoleID && campusID = req_CampusID then
                let totalAttendance = true in
                let encryptedTotalAttendance = bsenc(totalAttendance,classRacketMasterSecret) in
                out(c,(encryptedTotalAttendance, id_req));
                event attendingClass(id_req)
        else
                let noneAttendance = false in
                let encryptedNoneAttendance = bsenc(noneAttendance,classRacketMasterSecret) in
                out(c,(encryptedNoneAttendance, id_req)).
```

We then started the specification of the restaurant racket process exchanging keys with all other entities . The end of this process shows our context aware concerns. We designed this procedure in a key-diversified fashion, since the racket should have one key to communicate with each related entity. After initializing the key values, we implemented the requestor authentication, searching for the correspondent key on racket database. With the right key this racket is able to decrypt the sensitive information and analyse whether the subsidy is total, partial or none. In this analysis the racket should seek for the attendance racket key, but in a separate database, so that the search for keys could be parallel. Therefore the critical constraints that characterize our policy as a context-aware access control policy are shown in Listing 4.5.4.

Listing 4.5.4-4 – University Restaurant Process

```
let ruracket(sk:exponent,stuRoleID:G,acdRoleID:G,admRoleID:G,campusID:G)=
(**Key Generation between Requestor & RuRacket**)
        phase 1;
        let pkRacket = exp(g,sk) in
        in(c,(pkRequestor:G,idRequestor:ID));
        let racketMasterSecret = G_as_key(exp(pkRequestor, sk)) in
        insert authentications_and_keys(racketMasterSecret,idRequestor);

(**Key Generation between RuRacket & ClassRacket**)
        phase 2;
        in(c,(pkOtherRacket:G,idRacket:ID));
        let racketToRacketMasterSecret = G_as_key(exp(pkOtherRacket, sk)) in
        insert rackets_and_keys(racketToRacketMasterSecret,idRacket);

(*First Requestor Credits Value*)
        phase 3;
        new value:int;
        let firstCredit = senc(value,racketMasterSecret) in
        out(c,(firstCredit,idRequestor,campusID));

(*Authentication to use tickets*)
        phase 4;
        in(c,(creditsAtCard:bitstring,attendanceAtCard:bitstring,id_req:ID,roleID:G,req_CampusID:G));
        get authentications_and_keys(racketAndRequestorMasterSecret,=id_req) in
        event restaurantAuth(id_req);
        let requestorCredits = sdec(creditsAtCard, racketAndRequestorMasterSecret) in


        get rackets_and_keys(classMasterSecret, =idRacket) in
        let attendance = bsdec(attendanceAtCard, classMasterSecret) in

(*Subsidy Analysis and Credit Update*)
        if attendance = true && campusID = req_CampusID then
                if roleID = stuRoleID then
                let updatedStuCredits = ticketSubtraction(requestorCredits) in
                event creditPayment(idRequestor);
                let encryptedStuCredits = senc(updatedStuCredits,racketAndRequestorMasterSecret) in
                out(c,(encryptedStuCredits, id_req));

                if roleID = acdRoleID then
                let updatedAcdCredits = ticketSubtraction(ticketSubtraction(requestorCredits)) in
                event creditPayment(idRequestor);
                let encryptedAcdCredits = senc(updatedAcdCredits,racketAndRequestorMasterSecret) in
                out(c,(encryptedAcdCredits, id_req));

                if roleID = admRoleID then
                let updatedAdmCredits = ticketSubtraction(ticketSubtraction(ticketSubtraction(requestorCredits))) in
                event creditPayment(idRequestor);
                let encryptedAdmCredits = senc(updatedAdmCredits,racketAndRequestorMasterSecret) in
                out(c,(encryptedAdmCredits, id_req))
        else
                let updatedCredits = ticketSubtraction(ticketSubtraction(ticketSubtraction(requestorCredits))) in
                event creditPayment(idRequestor);
                let encryptedCredits = senc(updatedCredits,racketAndRequestorMasterSecret) in
                out(c,(encryptedCredits, id_req)).
```

The last step showed in Listing4.5.4 was to execute all processes and to do the queries.

The Proverif tool needs to execute the process to understand how to perform the formal proof. Here we defined that each requestor has its own identifier and that information on a practical example would stay on a card. Each one of the roles uses an specific id too. This is important for differentiating the requestors on context analysis. The same idea for roles was applied for campus identifiers, this covers the geolocated information needed to unlock maximum discount. The exclamation mark before some entity means to Proverif that it should simulate the entity with unbounded number of examples.

Listing 4.5.4-5 – Process Execution

```
1         (*Requestor 1*)new skS1:exponent;
2         (*Requestor 2*)new skS2:exponent;
3         (*Requestor 3*)new skS3:exponent;
4         new skR:exponent;new skCR:exponent;
5
6         new skStu:exponent;new skAcd:exponent;new skAdm:exponent; new skCampusOne:exponent;new skCampusTwo:exponent;
7         let stuRoleID = exp(g,skStu) in
8         let acdRoleID = exp(g,skAcd) in
9         let admRoleID = exp(g,skAdm) in
10        let campusOneID = exp(g,skCampusOne) in
11        let campusTwoID = exp(g,skCampusTwo) in
12        ( !requestor(skS1,ID1,stuRoleID,campusOneID) | !requestor(skS2,ID2,acdRoleID,campusOneID) |
13         !requestor(skS3,ID3,admRoleID,campusOneID) | ruracket(skR,stuRoleID,acdRoleID,admRoleID, campusOneID) |
14         attendancracket(skCR,stuRoleID,campusOneID)) |
15        ( !requestor(skS1,ID1,stuRoleID,campusTwoID) | !requestor(skS2,ID2,acdRoleID,campusTwoID) |
16        !requestor(skS3,ID3,admRoleID,campusTwoID) | ruracket(skR,stuRoleID,acdRoleID,admRoleID, campusTwoID) |
17        attendancracket(skCR,stuRoleID,campusTwoID))
```

Regarding the queries specification, it is necessary to show that the credits present in each requestor card remains secret and we tied together the event of restaurant authentication with the event of University restaurant service. By doing this we assured that for each credit payment should have one requestor authentication at the restaurant.

4.5.5 Formal Verification

We gathered the most relevant outputs that come out of our process execution and we are able to see in Listing4.5.5.

The first result(lines 1-2) shows that the attacker can not change or see the credit value, so that protects University resources. In this sense we demonstrate that they keys do not leak from and the information can not be decrypted within the policy implementation. Moreover, we can assert that the information used to determine the context-aware access control decision is private and can not be tampered with by an attacker.

Listing 4.5.5-6 – Process Execution

```
1
2  Starting query not attacker_p7(creditsAtCard[])
3  RESULT not attacker_p7(creditsAtCard[]) is true.
4
5  Starting query event(creditPayment(idRequestor_206)) ==> event(restaurantAuth(idRequestor_206))
6  RESULT event(creditPayment(idRequestor_206)) ==> event(restaurantAuth(idRequestor_206)) is true.
7
8  RESULT Weak secret id is true (bad not derivable).
```

The second result(lines 4-5) is showing that every requestor tested on the numerous sessions had to authenticate with the restaurant racket before any payment or credits update. This authentication avoids a a man-in-the-middle attack, because no other entitie can pretend to be a racket.

This second result is important because by assuring the identities of the rackets within our context-aware access control policy we can assert the properties of geolocation required by the policy specifi-

cation. Asserting identities and avoiding MitM attacks will guarantee the enforcement of the location context with Proteus

The third result(line 7) is just to confirm that the identifiers used by roles, campus and requestors is weak but this does not make it derivable by guessing. We added this testing because we are required to use the mifare contactless cards and this low entropy is a known problem of these card.

4.6 CONCLUSION

In this work, we devised a context-aware access control policy, based on the Proteus model, that uses a Trustful Space-Time Protocol for communication between IoT devices. We described the interactions of a scenario based on smart campus setting. We specified the symbolic model of the scenario and verified it using an automatic protocol verification tool called Proverif.

Our access control policy covers a Smart campus scenario and it is prepared for IoT environments needs. We described all context elements for a resource context on University domain and it can be generalized for other resources. After conceiving the descriptions for the policy we were able to build a symbolic model. This model was formally proved on Proverif and its outputs demonstrated that our model keep sensitive personal information secret.

As future work, we plan to build more policies targeting the IoT environments heterogeneity, dynamicity, unpredictability. These policies could integrate a more generical and complex policy, that will give us better scenary context abstractions. We further could apply the entire Proteus Middleware Architecture, composed by Police Instalation Manager, Reasoning Core, Policy Enforcement Manager and the Context Manager.

5 FINAL REMARKS

In this work, we did a formal analysis of the Trustful Space-Time Protocol(TSTP) and we came up with a new access control policy that uses TSTP. We described TSTP's keybootstrapping and time synchronization protocols(RESNER; FRÖHLICH, 2015) using an automatic cryptographic protocol verifier tool named Proverif. We described interactions of a Smart Campus scenario and we used our knowledge and experience with Proverif to specify a symbolic model of that scenario. The protocol formally verified in our first contribution was used as communication protocol for the access control policy that was our second contribution.

The automatic analysis of the protocol found an attack involving the time synchronization part, that could result in impersonation attacks and the attacker could read any message passed on the secure channel. We proposed two specific modifications to the protocol as counter-measures for the security flaw. The first modification was to change the MAC-then-Encrypt approach for Encrypt-then-MAC; this was a problem that even Proverif did not detect and this change could increase the security. The second modification could be done changing the time synchronization part or making all the clocks involved on TSTP to follow the IEEE 802.15.4 standard, that has an advisable clock accuracy of at least $\pm$40ppm.

We were able to see that in TSTP an attacker might not only find a correct ID, but he could deploy a malicious node at the right time and place(RESNER; FROHLICH, 2015). This synchrony is needed because TSTP does not derive its key from just one pre-distributed information, but from a set of data with good entropy. Of course, as there is always a pre-distributed datum, it is still important to avoid a key derivation based on just one value. Therefore the TSTP protocol decisions and its features are the reasons behind our choice to use this protocol on our second work.

Our second goal achieved was a context-aware access control scheme for IoT environments. We created an access control policy for that scheme, covering a Smart Campus scenario. We specified all the required elements for a resource context on University domain, based on Proteus model(BOTTAZZI; MONTANARI; TONINELLI, 2007). After that we described a symbolic model according to the context elements requirements. Our last step was a formal analysis of the symbolic model, that represents the access control policy, assuring that the sensitive

data remained secret.

For future work, we could create different policies encompassing more heterogeneous IoT gadgets. We could extend the use of TSTP for different environments and maybe a computational model verification of the entire Smart Campus scenario's infrastructure. The policies could become more susceptible to changes, implementing the mechanism of policy re-evaluation and policy enforcement present in the Proteus description. As far as we know, the use of TSTP it is still just for academic purposes, but after the formal verification it should be tested inside other areas, like smart-buildings, smart-manufacturing and so on. The Smart Campus scenario was tested in a minimal scenario, so it would be very interesting if we could test the computational model of the hole infra, using CryptoVerif, another automatic verification tool.

REFERĘNCIAS

ASHTON, K. That 'internet of things' thing. *RFiD Journal*, v. 22, n. 7, p. 97–114, 2009.

ATZORI, L.; IERA, A.; MORABITO, G. The internet of things: A survey. *Computer networks*, Elsevier, v. 54, n. 15, p. 2787–2805, 2010.

BALAKRISHNAN, H.; KATZ, R. H. Explicit loss notification and wireless web performance. In: *Proc. IEEE Globecom*. [S.l.: s.n.], 1998. v. 98.

BELLARE, M.; NAMPREMPRE, C. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. *Journal of Cryptology*, Springer, v. 21, n. 4, p. 469–491, 2008.

BERNSTEIN, D. J. The poly1305-aes message-authentication code. In: *Proceedings of Fast Software Encryption*. Paris, France: [s.n.], 2005. p. 32–49.

BERNSTEIN, D. J. The poly1305-aes message-authentication code. In: SPRINGER. *International Workshop on Fast Software Encryption*. [S.l.], 2005. p. 32–49.

BLANCHET, B.; ABADI, M.; FOURNET, C. Automated verification of selected equivalences for security protocols. In: IEEE. *20th Annual IEEE Symposium on Logic in Computer Science (LICS'05)*. [S.l.], 2005. p. 331–340.

BLANCHET, B. et al. *Proverif: Cryptographic protocol verifier in the formal model*. 2010.

BLANCHET, B.; SMYTH, B.; CHEVAL, V. Proverif 1.90: Automatic cryptographic protocol verifier, user manual and tutorial. *URL: http://prosecco. gforge. inria. fr/personal/bblanche/proverif/manual. pdf*, 2015.

BOTTAZZI, D.; MONTANARI, R.; TONINELLI, A. Context-aware middleware for anytime, anywhere social networks. *IEEE Intelligent Systems*, IEEE, v. 22, n. 5, p. 23–32, 2007.

BOUTABA, R.; AIB, I. Policy-based management: A historical perspective. *Journal of Network and Systems Management*, Springer, v. 15, n. 4, p. 447–480, 2007.

BRADSHAW, J. M.; MONTANARI, R.; USZOK, A. Policy-based governance of complex distributed systems: What past trends can teach us about future requirements. In: *Adaptive, Dynamic, and Resilient Systems*. [S.l.]: Auerbach Publications, 2014. p. 259–284.

CERP-IOT, V. Challenges for realising the internet of things, no. *March. European Commission-Information Society and Media DG*, 2010.

CISCO, C. B. Security: The vital element of the internet of things. 2015.

CLAYCOMB, W. R.; SHIN, D. A novel node level security policy framework for wireless sensor networks. *Journal of Network and Computer Applications*, Elsevier, v. 34, n. 1, p. 418–428, 2011.

COMMUNITIES, C. of the E. *Early Challenges regarding the Internet of Things*. setember 2008. Commission Staff Document.

COUNCIL, N. Six technologies with potential impacts on us interests out to 2025. *Disruptive Civil Technologies 2008*, 2008.

COUNCIL, N. I. *Six Technologies With Potential Impacts on US Interests Out to 2025*. april 2008. Conference Report.

DELGADO-MOHATAR, O.; FÚSTER-SABATER, A.; SIERRA, J. M. A light-weight authentication scheme for wireless sensor networks. *Ad Hoc Networks*, Elsevier, v. 9, n. 5, p. 727–735, 2011.

DEY, A. K. Understanding and using context. *Personal and ubiquitous computing*, Springer-Verlag, v. 5, n. 1, p. 4–7, 2001.

DOLEV, D.; YAO, A. C. On the security of public key protocols. *IEEE Transactions on Information Theory*, v. 29, n. 2, p. 198–208, mar. 1983.

DUNKELS, A.; ÖSTERLIND, F.; HE, Z. An adaptive communication architecture for wireless sensor networks. In: ACM. *Proceedings of the 5th international conference on Embedded networked sensor systems*. [S.l.], 2007. p. 335–349.

FAQUIH, A.; KADAM, P.; SAQUIB, Z. Cryptographic techniques for wireless sensor networks: A survey. In: IEEE. *2015 IEEE Bombay Section Symposium (IBSS)*. [S.l.], 2015. p. 1–6.

FU, B. et al. A survey of cross-layer designs in wireless networks. *IEEE Communications Surveys & Tutorials*, IEEE, v. 16, n. 1, p. 110–126, 2014.

GARCIA, F. D. et al. Dismantling mifare classic. In: _____. *Computer Security - ESORICS 2008: 13th European Symposium on Research in Computer Security, Málaga, Spain, October 6-8, 2008. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. p. 97–114. ISBN 978-3-540-88313-5. <http://dx.doi.org/10.1007/978-3-540-88313-5_7>.

GUBBI, J. et al. Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, Elsevier, v. 29, n. 7, p. 1645–1660, 2013.

GUSMEROLI, S.; PICCIONE, S.; ROTONDI, D. A capability-based security approach to manage access control in the internet of things. *Mathematical and Computer Modelling*, v. 58, n. 5–6, p. 1189 – 1205, 2013. ISSN 0895-7177. The Measurement of Undesirable Outputs: Models Development and Empirical Analyses and Advances in mobile, ubiquitous and cognitive computing. <http://www.sciencedirect.com/science/article/pii/S089571771300054X>.

KHAN, S.; LOO, K. K.; DIN, Z. U. Cross layer design for routing and security in multi-hop wireless networks. *Journal of Information Assurance and Security*, v. 4, n. 2, p. 170–173, 2009.

KIRUTHIKA, B.; EZHILARASIE, R.; UMAMAKESWARI, A. Implementation of modified rc4 algorithm for wireless sensor networks on cc2431. *Indian Journal of Science and Technology*, v. 8, n. S9, p. 198–206, 2015.

LAFOURCADE, P.; PUYS, M. Performance evaluations of cryptographic protocols verification tools dealing with algebraic properties. In: SPRINGER. *International Symposium on Foundations and Practice of Security*. [S.l.], 2015. p. 137–155.

MENDES, L. D.; RODRIGUES, J. J. A survey on cross-layer solutions for wireless sensor networks. *Journal of Network and Computer Applications*, Elsevier, v. 34, n. 2, p. 523–534, 2011.

MIYOSHI, M.; SUGANO, M.; MURATA, M. Improving tcp performance for wireless cellular networks by adaptive fec combined with explicit loss notification. *IEICE transactions on communications*, The Institute of Electronics, Information and Communication Engineers, v. 85, n. 10, p. 2208–2213, 2002.

NEWS, H. *Internet of Things (IoT) number of connected devices worldwide from 2012 to 2020 (in billions)*. 2016. Https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/, Accessed: 2017-01-19.

PATERSON, K. G.; YAU, A. Padding oracle attacks on the iso cbc mode encryption standard. In: SPRINGER. *Cryptographers' Track at the RSA Conference*. [S.l.], 2004. p. 305–323.

PERERA, C. et al. Context aware computing for the internet of things: A survey. *IEEE Communications Surveys & Tutorials*, IEEE, v. 16, n. 1, p. 414–454, 2014.

PREUVENEERS, D.; BERBERS, Y. Internet of things: A context-awareness perspective. *The Internet of Things: From RFID to the Next-Generation Pervasive Networked Systems*, Auerbach, p. 287–307, 2008.

RAJESWARI, S. R.; SEENIVASAGAM, V. Comparative study on various authentication protocols in wireless sensor networks. *The Scientific World Journal*, Hindawi Publishing Corporation, v. 2016, 2016.

REGHELIN, R.; FRÖHLICH, A. A. A decentralized location system for sensor networks using cooperative calibration and heuristics. In: ACM. *Proceedings of the 9th ACM international symposium on Modeling analysis and simulation of wireless and mobile systems*. [S.l.], 2006. p. 139–146.

REN, K. et al. Multi-user broadcast authentication in wireless sensor networks. *IEEE Transactions on Vehicular Technology*, IEEE, v. 58, n. 8, p. 4554–4564, 2009.

RESNER, D.; FROHLICH, A. A. Design rationale of a cross-layer, trustful space-time protocol for wireless sensor networks. In: IEEE. *Emerging Technologies & Factory Automation (ETFA), 2015 IEEE 20th Conference on*. [S.l.], 2015. p. 1–8.

RESNER, D.; FRÖHLICH, A. A. Key establishment and trustful communication for the internet of things. *4th SENSORNETS*, 2015.

RESNER, D.; FRöHLICH, A. A.; WANNER, L. F. Speculative Precision Time Protocol: submicrosecond clock synchronization for the IoT. In: *21th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2016). To appear.* Berlin, Germany: [s.n.], 2016.

ROMAN, R.; ZHOU, J.; LOPEZ, J. On the features and challenges of security and privacy in distributed internet of things. *Computer Networks*, Elsevier, v. 57, n. 10, p. 2266–2279, 2013.

SILVA, D. S. et al. Formal verification of a cross-layer, trustful space-time protocol for wireless sensor networks. In: *Information Systems Security.* [S.l.]: Springer, 2016. p. 426–443.

SILVA, D. S. et al. Formal verification of a cross-layer, trustful space-time protocol for wireless sensor networks. In: _____. *Information Systems Security: 12th International Conference, ICISS 2016, Jaipur, India, December 16-20, 2016, Proceedings.* Cham: Springer International Publishing, 2016. p. 426–443. ISBN 978-3-319-49806-5. <http://dx.doi.org/10.1007/978-3-319-49806-5_23>.

SLOMAN, M.; LUPU, E. Security and management policy specification. *IEEE network*, IEEE, v. 16, n. 2, p. 10–19, 2002.

SUO, H. et al. Security in the internet of things: a review. In: IEEE. *Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on.* [S.l.], 2012. v. 3, p. 648–651.

VIJAYAKUMAR, P.; VIJAYALAKSHMI, V. Effective key establishment and authentication protocol for wireless sensor networks using elliptic curve cryptography. In: *Proceedings of the Conference on Mobile and Pervasive Computing (CoMPC'08).* [S.l.: s.n.], 2008.

WANDER, A. S. et al. Energy analysis of public-key cryptography for wireless sensor networks. In: IEEE. *Third IEEE international conference on pervasive computing and communications.* [S.l.], 2005. p. 324–328.

WANG, Y.; ATTEBURY, G.; RAMAMURTHY, B. A survey of security issues in wireless sensor networks. *IEEE Communications Surveys & Tutorials*, IEEE, v. 8, n. 2, p. 2–23, 2006.

WONG, K. H. et al. A dynamic user authentication scheme for wireless sensor networks. In: IEEE. *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06)*. [S.l.], 2006. v. 1, p. 8–pp.

XIAO, M.; WANG, X.; YANG, G. Cross-layer design for the security of wireless sensor networks. In: IEEE. *Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on*. [S.l.], 2006. v. 1, p. 104–108.

ZHAO, K.; GE, L. A survey on the internet of things security. In: IEEE. *Computational Intelligence and Security (CIS), 2013 9th International Conference on*. [S.l.], 2013. p. 663–667.

ZHU, S.; SETIA, S.; JAJODIA, S. Leap+: Efficient security mechanisms for large-scale distributed sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, ACM, v. 2, n. 4, p. 500–528, 2006.