Smartphone based Authentication & Authorization Protocol for Smart Physical Access Control System (SPACS)

By

Faisal Karim Bhutta
2010-NUST-MSCCS-02

Supervisor
Dr. Abdul Ghafoor
Department of Computing

A thesis submitted in partial fulfillment of the requirements for the degree of Masters in Computer and Communication Security (MS CCS)

In
School of Electrical Engineering and Computer Science,
National University of Sciences and Technology (NUST),
Islamabad, Pakistan.

(January 2014)

Approval

It is certified that the contents and form of the thesis entitled "*Smartphone based Authentication & Authorization Protocol for SPACS*" submitted by **Faisal Karim Bhutta** has been found satisfactory for the requirement of the degree.

Advisor: Dr. Abdul Ghafoor

Signature: ________________
Date: ________________

Committee Member 1: Dr. Awais Shibli

Signature: ________________
Date: ________________

Committee Member 2: Dr. Zahid Anwar

Signature: ________________
Date: ________________

Committee Member 3: Dr. Mureed Hussain

Signature: ________________
Date: ________________

I dedicate this work to my Father& Mother for their love and support.

Certificate of Originality

I hereby declare that the research work titled "*Smartphone based Authentication & Authorization protocol for SPACS*" is my own work to the best of my knowledge. It contains no materials previously written or published by any other person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST or any other education institute, except where due acknowledgment, is made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST or elsewhere, is explicitly acknowledged in the thesis.

I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the projects design and conception or in style, presentation and linguistic is acknowledged. I also verified the originality of contents through plagiarism software.

Author Name: FAISAL KARIM BHUTTA

Signature: ____________________

Acknowledgment

I am thankful to the faculty of SEECS for their guidance, cooperation and counseling throughout my research work. I would like to express my deep gratitude to my thesis research advisor Dr. Abdul Ghafoor (NUST) for his support, vision, mentoring and guidance. I have benefited from his useful assistance, feedback and critique. I am also thankful to my committee members Dr. Mureed Hussian (NESCOM) and Dr. Awais Shibli (NUST) for their support, availability and useful input in my research work. I am thankful to Dr. Zahid Anwar (NUST) for guiding me in formal verification and journal writing.

I extend my gratitude to my class mate and friends at KTH-SEECS AIS Lab (NUST) for sharing feedback and knowledge. In particular, I wish to thank Miss Shizra Sultan and Mr. Waleed Asif for assisting me in implementation of designed protocol. I am thankful to Miss Rahat Masood for sharing her knowledge of thesis writing. Lastly, I am grateful to my family for their support and love.

Abstract

Nowadays smartphone is becoming multi-purpose device because it has more processing power at affordable cost. The trend of using smartphone for business, banking and everyday tasks has attracted research community to address security issues in smartphone applications and their communication with external systems. Due to their wide acceptability in community public, it is becoming trivial to use smartphone as an authenticating device for banking applications and access control management systems. Current legacy solutions used for Physical Access Control System (PACS) are combination of software and hardware to control the access of users to physical resources (rooms, offices, buildings etc). Most of them are using biometric or smart card as an identity token. The associated cost and limited freedom to customize these solutions to organizational needs open research areas for smartphone researchers to use them in PACS. In our research, architecture for PACS along with security protocol for smartphone is designed that is used for identity verification, authentication and authorization in PACS.

The designed authentication protocol is an extension of two-factor authentication protocol described in FIPS-196 standard. Furthermore, the usage of symmetric key cryptography provides an efficient solution to achieve confidentiality of messages exchanged between components of designed PACS. In order to ensure the presence of the legitimate user in the premises, the system uses a pass-code feature that is only valid for one time usage. Hence using designed protocol, user's smartphone can act as authenticator in the system. Since, the solution may be used by the non-technical persons so it is designed that it should be user friendly and require minimum efforts for configuration of security parameters. In order to validate the security of designed protocol, automated protocol verification tool Scyther is used. After validation, it is verified that our security protocol resists against Man-in-the-Middle, replay and attacks on confidentiality of user's credentials.

Table of Contents

1. Introduction & Motivation.. 1

1.1 Introduction 1

1.2 Motivation 4

1.3 Problem Statement 4

1.4 Research Methodology... 5

1.5 Contribution 5

1.6 Thesis Organization... 6

2. Background & Literature Review.. 8

2.1 Related Work.. 9

2.1.1 Physical Access Control based on QRCodes 9

2.1.2 A Pragmatic Online Authentication using Smart Card 10

2.1.3 Strong Mobile Authentication 12

2.2 Related Work Analysis... 13

3. Smart Physical Access Control System (SPACS) Architecture............................ 14

3.1 Security Requirements 15

3.2 Architecture Overview 16

3.2.1 Identity Management System 17

3.2.2 SPACS Authentication & Authorization Server 17

3.3.3 Certification Authority 18

3.3.4 Electronic Door Key Panel 18

3.3.5 SPACS Application for Smartphone 19

3.4 SPACS Workflow Sequence.. 19

3.5 SPACS Architecture Features 21

3.6 Areas of Application 22

4. SPACS Authentication & Authorization Protocol ... 23

4.1 Design Goals ... 24

4.2 Authentication & Authorization Protocol ... 24

4.2.1 Authentication Phase ... 25

4.2.2 Session Key Exchange Phase ... 28

4.2.3 Door Access Request Phase ... 29

5. Formal Verification of SPACS Authentication & Authorization Protocol ... 32

5.1 Formal Analysis of Protocol ... 32

5.1.1 Model Description ... 33

5.1.2 Security Properties Specification ... 34

5.1.3 Formal Verification ... 35

5.2 Scyther SPDL Script for Protocol Verification ... 39

6. Conclusion ... 42

6.1 Future Work ... 43

Bibliography ... 44

List of Abbreviations

Abbreviations	**Descriptions**
PACS	Physical Access Control System
SPACS	Smart Physical Access Control System
IDMS	Identity Management System
RFID	Radio Frequency Identification
PIN	Personal Identification Number
XACML	eXtensible Access Control Markup Language
SAML	Security Assertion Markup Language
PKI	Public Key Infrastructure
PDP	Policy Decision Point

List of Figures

2.1 Physical Access Control based on QRCodes [3] 9

2.2 Strong Mobile Authentication Scheme [13] 13

3.1 Components of the SPACS Architecture and communication between Components 17

3.2 Smartphone Application showing One Time Pass Code 18

3.3 SPACS Components Message Exchange Sequence 20

3.4 SPACS Architecture Diagram for Deployment 21

4.1 SPACS Authentication & Authorization Protocol Phases 23

4.2 Smartphone Authentication Token Verification 26

4.3 Server’s Authentication Response Verification 27

4.4 Server’s Session Key Message Verification 28

4.5 SPACS Electronic Door Access Panel 30

4.6 SPACS Authentication & Authorization Protocol 31

5.1 Scyther Generated Verification Results 37

5.2 Scyther Generated Attack Tree 38

List of Tables

5.1 Message Types ………………….. 35

Chapter 1

Introduction and Motivation

"Hell, there are no rules here - we're trying to accomplish something"
Thomas Edison

1.1 Introduction

The security and safety of property and resources is the top most priority of any organization. For this purpose, organizations are keen interested to use state of the art Physical Access Control System (PACS). PACS are integral part of secure building design and are extensively used to safeguard against adversary. One of the common techniques is to use door locks to restrict access to only authorized users. PACS can be implemented using a wide range of technologies and protocols, from simple user name/password to more advanced and sophisticated solutions such as face recognition based PACS. The common methodology used in all PACS is that the building resources should be accessible either through scrutiny by security guards or by deploying physical control systems. But in such systems, the reliability and security is dependent on the procedure used to proof the physical identity of the user.

A lot of research work has been done in this area. Some of designed solutions are based on password, USB key, smart card, biometric and/or One Time Password based authentication [1]. After analyzing current approaches it is found that most of the designed systems for PACS provide adequate level of identity verification and authorization services. But some of them are based on proprietary standards and specifically designed for specific organizations so they are not scalable and interoperable with other systems. It is also observed that some of them do not consider communication level security. Also some of PACS are dependent on specific hardware and are designed for specific organizations. So the dependency on hardware also brings challenges to meet the requirements of physical security credentials. Therefore, in current situations it is needed to rethink about the Physical Access Control System using smartphone which provides strong authentication, standard authorization mechanism and secure communication. The system must be easy to use and must handle security credentials transparently.

New emerging trends have changed the thinking of society and as technology is driving at a rapid pace, smartphones are gaining popularity in general public. They are no more considered as luxury but a must carry device of person's attire. They not only bring comfort and conveniences in ordinary person's life but also are used for accessing business applications, banking accounts, online transactions, shopping online, social networking and file sharing. With the bulk emergence of free and cheap smartphone applications, these devices are no more considered as communicating devices. People are interested in using their smartphone as digital keys for their cars and houses. Due to the cost reduction and increasing computational power, these devices also have the capability to be used for authentication and authorization in accessing physical or logical resources. It is observed that currently the legacy systems for PACS rely heavily on RFID cards, Smart Cards or biometric devices. If different vendor readers or access control machines are installed in premises, then it is difficult and cumbersome task for security officials to upgrade the whole Access Control System. On public areas, people are reluctant to provide their biometric credentials, passwords/PIN or Smart Card as they think their credentials may be misused.

In this thesis, smartphone based PACS architecture and Authentication and Authorization protocol for Smart Physical Access Control System (SPACS) is presented. SPACS is a Physical Access Control System that uses smartphone as an authentication device in alternative to Smart Cards or RFID cards. It has added features of remote administration for security managers and allows them to revokes access rights of targeted user from a centralized system.

Designed SPACS architecture comprises of five components: *Identity Management*

System, SPACS Authentication and Authorization Server, Certification Authority, Electronic Door Panel and *SPACS Application* for smartphone. The communication between smartphone and SPACS Authentication server is done through designed protocol which is an extension of FIPS-196 challenge/response protocol and supports smartphone based two-factor authentication. The extended features of protocol are (1) It verifies users from IDMS, (2) verifies certificates online and (3) ensures mutual authentication of smartphone and authentication server. The authentication protocol provides mechanism for registration, secure exchange of session keys and formal authentication process. It has been designed, implemented and integrated with SPACS system. For authorization, XACML standard with extended features are adopted so it enables designed protocol to use the concept of single-sign-on and SAML authorization protocols. The user while accessing physical door generates authentication request from his smartphone application which is verified by SPACS Authentication and Authorization server and a one-time passcode is transmitted securely to the smartphone application. After which it is keyed in by the user using electronic door panel to pass through the door. The key features of designed architecture and protocol are as follows:

1. Extension of FIPS-196 challenge/response protocol
2. Two-factor authentication using PIN and smartphone as security tokens.
3. Provides Remote Administration for Security Managers
4. Provides a simple, efficient, secure Authentication and Authorization protocol
5. Minimizes security management complexities of Physical Access Control System for security professionals.
6. Provides a qualitative based solution through the use of well-known security standards.
7. Provides an alternative against Smart Cards, biometric devices for large scale organization by cutting costs of device procurement, maintenance and reissuance.
8. No additional hardware requirement like card readers or biometric devices.
9. Utilization of existing network infrastructure.
10. Scalable and Interoperable architecture.

The main application areas of our designed architecture and protocol may be software houses, data centers, educational institutes, baffle gate/ parking area, non-military sensitive areas and commercial organizations. The designed protocol is verified and analyzed using threat modeling techniques. This model is further verified from automated protocol analyzer Scyther [2] whose results validate that the protocol resists against replay attack, Man-in-the-Middle attack and retain confidentiality of credentials during wireless transmission.

1.2 Motivation

Traditionally Access Control is restricted to logical access where user is granted access to logical resources after authorization using single factor authentication. These access grants are provided by the system after deciding whether the user has valid credentials e.g., user name and password. While most PACS are implemented using PIN codes, Smart Cards or biometric devices but due to wide acceptability of smartphone among masses it is possible to use them as authenticating device in physical access control.

Innovation in the field of smartphone is drastically changing people's lifestyle and is fulfilling their cyber needs through smartphone applications. Smartphones are now not limited to phones calls and messaging but now are used as digital keys for accessing logical and physical resources. Due to wide acceptability among general public, high computational power and decreasing cost, they are becoming attractive to use as authenticating device in PACS in place of Smart Cards and biometric devices.

The motivation of the thesis is to demonstrate the use of smartphone in Physical Access Control System. Because smartphone is a relatively new invention, not many physical access control applications have been developed. So, designing an architecture using smartphone for physical access control is a challenging task but interesting experience for my thesis.

1.3 Problem Statement

Nowadays security is a major concern and ensuring proper security to secure area is a challenging and difficult job. Different access control mechanism like Smart Card, RFID cards, PIN and biometric verification have already been proposed and are deployed commercially. Due to the increasing computational power and low cost, smartphones are becoming technology drive changer. They are emerging as a replacement for physical keys and can be a suitable authenticating device in PACS. Most legacy PACS are based on single factor authentication. Therefore, it is needed to have a

PACS that provide enhanced security and strong authentication using smartphone as authenticating device, hence deriving our problem statement as:

To design an architecture and communication protocol for Physical Access Control System which supports smartphone based two-factor authentication and authorization services using standard security features.

1.4 Research Methodology

The main purpose of this thesis is to describe in detail SPACS architecture and communication protocol using smartphone as authenticator and in particular, formally verifying designed communication protocol using automated security verification tool. The methodology involved defining real world problem of Physical Access Control System with respect to modern times, followed by extensive literature review of current proposed Physical Access Control architectures and methods. Furthermore, the designed architecture for Smart Physical Access Control System (SPACS) is derived using standard security features and a communication protocol is devised to secure communication between smartphone and other SPACS components. The designed SPACS system is developed using JAVA and Objective C as a proof of concept. Lastly, to check reliability and integrity of communication protocol, it is tested empirically using automated security protocol verification tool Scyther [2].

1.5 Contribution

In this thesis, we set out to design Smart Physical Access Control System (SPACS) architecture and communication protocol using standard security features and existing knowledge of physical access control techniques. Subsequently, using the designed architecture and protocol, we did develop a smartphone application and SPACS Authentication and Authorization server as a proof of concept.

In accordance with problem statement, we set some research objectives and the corresponding contributions.

- *Objective 1*: To devise architecture for Physical Access Control System using smartphone as authenticating device.

  Contribution: Designed architecture for Smart Physical Access Control

System that authenticates and authorizes legitimate user to access secure areas using his smartphone and one-time passcode.

- *Objective 2*: To devise Authentication & Authorization communication protocol for SPACS architecture.

  Contribution: Designed a communication protocol which is an extension of FIPS-196 challenge/response protocol. The designed protocol provides mutual authentication between smartphone and SPACS authentication and authorization server. It also ensures confidentiality, integrity and non-repudiation of messages transmitted between server and smartphone using standard security features.

- *Objective 3*: To verify and validate security and reliability of communication protocol using an automated security protocol analyzer.

  Contribution: Verification of secrecy and integrity claims for communication protocol using an automated protocol analyzer, Scyther. It is observed that the designed communication protocol resist against Man-in-the-Middle, replay attack and retains confidentiality.

More importantly, it is observed that users usually don't share their smartphone with their colleagues or acquaintances. In case of theft, they report it instantly which is beneficial in minimizing attack window for adversary. Also smartphone is a personal device and user's credentials are securely saved so there are fewer chances of copying and misusing them. Furthermore, the use of Public Key Infrastructure in designed architecture ensures non-repudiation.

1.6 Thesis Organization

This thesis consists of six chapters. In Chapter 2, literature review and critical analysis for user authentication and authorization schemes through mobile phone and smart card are presented. In Chapter 3, architecture for SPACS is presented in detail. In Chapter 4, Authentication and Authorization protocol for smartphone to exchange messages securely between smartphone and SPACS server is presented. In Chapter 5, formal verification results using automated protocol verification tool is discussed in detail. In last Chapter, conclusion and future work of the research is presented.

Chapter 2

Background and Literature Review

"Because ideas have to be original only with regard to their adaptation to the problem at hand, I am always extremely interested in how others have used them"
Thomas Edison

Security concern is becoming a major concern for enterprises, military installations, government buildings, data center, educational institutes and health care. To secure critical assets and resources companies are adhering to different international security standards. Most commonly used authentication of user to access critical resources is password. These passwords come with great security concerns. They can be guessed by intruders using different guessing techniques like brute force, dictionary attack, eavesdropping, offline guessing, social engineering etc. Along with that users use same passwords for multiple accounts, write them on sticky notes or computers and use weak passwords. Several password techniques have been proposed to make it strong password but practically they are not user friendly and difficult for user to remember. Two-factor authentication has been proposed to make it difficult for hacker to get access to critical resources. Two-factor authentication makes it difficult to hack by using devices like smart card, token generator or smartphone.

2.1 Related Work

In network security various authentication and authorization protocol are designed and implemented but many of them are related to specific application or considered conventional environment.

2.1.1 Physical Access Control based on QRCodes

In [3] the author designed a physical access control based on QR (Quick Response) Codes. Their designed system consists of three entities; ORS (OTP RSA Server), OQG (OTP QRCode Generator) and ORC (OTP RSA Clients). The encrypted OTP (One Time Password) QRCode is generated by OQG application installed on user's mobile phone. It is sent as *Multimedia Messaging Service* (MMS) to the ORS for authentication and authorization through carrier network or general network. The user further shows MMS from his mobile phone as authenticating token to the camera attached with ORC Client to request service which is verified from ORS server. In case of successful authentication and authorization, the door lock is released and user is allowed to open the door. Their architecture is shown in the Figure 2.1.

Critical Analysis: In their paper, authors have not elaborated details about authentication and authorization mechanism. Also they have not defined mechanism of generation of encrypted OTP QR Codes. Furthermore, the cost of provisioning client system with attached camera on every door is making their designed system unviable.

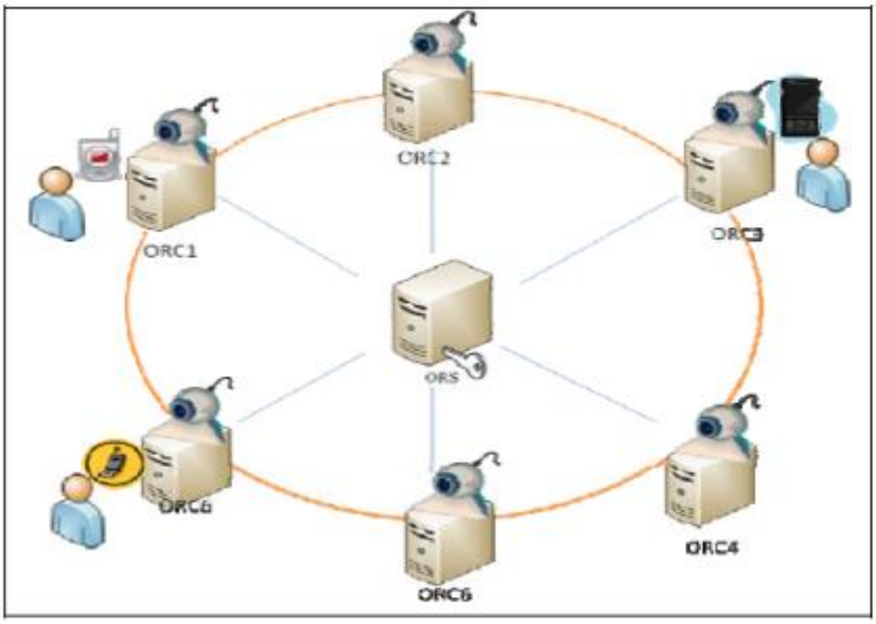


Fig 2.1 Physical Access Control based on QRCodes [3]

In [4], the authors described IdM (Identity Management) Card approach in which authentication and authorization statements are generated by IdM card after local verification of users using PIN. Their approach uses different signature keys and attributes to access different services through service provider. Their concept is based on EU project, SWIFT [5]. It divides IdM roles as Authorization Provider, Identity Aggregator, Attribute Provider and Service Provider. Another token based access control protocol is described in [6]. It uses NFC enabled smartphone as a security token. Their designed scheme allows users to partially delegate their rights to other smartphone users without involvement of token issuer. They also presented multi-level security architecture to protect underlying security credentials of their protocol.

2.1.2 A Pragmatic Online Authentication using Smart Card

The authors [7] of this paper propose a framework for accessing smart card based web applications. In their framework, the web browser asked user to enter PIN which is verified from the smart card inserted into the reader attached to the computer. A challenge is sent from the authentication server to the web browser that redirects it to smart card where it is signed by user’s private key and sent back to the authentication server for verification along with user’s digital certificate. At the authentication server end, the digital certificate and signed value is verified and in case of successful verification user is granted access to smart card based web applications. They used challenge-response protocol using X.509 certificate and private key for authentication. The private key is stored in user smart card which is used to generate signature of challenge received from authentication server. They also mentioned that the designed solution is implemented as an extension for browser.

Ilias Daradimos, Konstantinos Papadopoulos, Ilias Stavrakas, Maria Kaitsa, Theophanis Kontogiannis and Dimos Triantis [8] present a PACS that utilizes existing computer and networking infrastructure for its operations. In their paper, existing network and computer infrastructure is used that increased reliability and scalability while communication utilizes TCP/IP data network and Access Control Management is done through a web based application. The major advantage of using existing network infrastructure and resources is to minimize the implementation and installation cost and to achieve larger scalability.

In [9] the article evaluates the applications and benefits of contactless credentials for access control stored on mobile phones. These credentials can act as virtual keys and can be used to lock and unlock doors. When stored within a mobile handset, these contactless credentials provide more benefits over existing contactless access control systems. The strongest advantage is the instant delivery of credential to a person's handset. Security managers can manage all users from a central access control system and instantly grant appropriate level of access to employees and visitors. These mobile contactless credentials can be used to support real time traceability for offline and online doors in case of particular credential being used to gain access at a particular time.

In [10] Public Key Infrastructure (PKI) can be used at the door to determine a card's authenticity in order to upgrade existing PACS and to achieve FIPS 201 and SP 800-116 compliant. A random challenge is sent to the smart card and is verified and signed using public private keys. These challenges can correctly be responded only by legitimate card. In the article they presented three basic configurations for PKI at the door. Firstly, the panel generates challenge and passes it to the smart card through reader. The response from the smart card is processed at the panel and all cryptographic processing and certificate revocation status are performed and cached at the panel. As panels are designed to operate offline, no extra equipment or board is required. Secondly, PACS server generates a challenge and passes it to the smart card through the reader. The smart card generates a response and passes it to the PACS server through reader that verifies the response and directs door locks to be released. Thirdly, an additional board or controller generates a challenge and pass it to the smart card through reader. The smart card generates a response and passes it to the controller through reader that verifies the response. It sends card identifier to the access control panel in case of successful verification. This configuration operates independently of PACS server for longer period of time but requires separate controller and wiring. Currently physical cards are the primary mode of getting access to high assurance areas, however NFC (*Near Field Communication*) is becoming standard in smartphones, hence making phone both a contactless credential and a reader at the same time. They require special readers to be installed in premises or public areas and offer an additional option for commercial markets but they incur an additional cost.

In [11] the authors have analyzed the existing mobile banking transaction protocols with their vulnerabilities. Based on their findings, they proposed a protocol based on two-factor authentication mechanism that resists against common security threats. The authors have selected a Bluetooth communication mode which establishes security functionalities before the actual meaningful communication. Their authentication protocol is based on *Something You Have* and *Something You Know* paradigm where credentials such as passwords/usernames or certificates are provided by non-mobile terminals. A pre shared key already stored on mobile and non-mobile terminals is used for mutual authentication between them without revealing the key. A challenge is sent from the bank to the user who replies it using his username, password or certificate to ensure his reality after user profile creation at the bank end. After ensuring the provided credentials are real, remote terminal send a nonce to the fixed terminal which is encrypted using pre shared secret at the fixed terminal along with terminal ids of both. The encrypted message is decrypted at the remote terminal using pre shared secret and a tag is calculated at the remote terminal and transmitted to the fixed terminal. The fixed terminal calculates tag after receiving message and compares it for user authentication. The authors have not formally verified their developed protocol.

Yoshiaki, Yoichi and Masanori [12] proposed a scheme based on finger print to grant access to terminal or to an application. They mitigated the risk of storing the biometric templates remotely and locally and proposed to store a signed copy of the biometric template on the locked terminal. The integrity of user supplies fingerprint data is checked against stored template on the locked terminal using the user's public key and is matched with the decrypted for validation. On successful verification, terminal is unlocked and protected application is started. They suggested user's private key to be stored on user's smart card and must only be known to Certification Authority or user alone.

2.1.3 Strong Mobile Authentication

Hassinen, Marko and Konstantin Hypponen [13] propose a scheme for strong mobile authentication by using *Short Messaging Service* (SMS). In their scheme message is encrypted using public key of receiver. Hash of this encrypted message is generated and concatenated with the timestamp of sender's mobile phone. It is further signed by the private key of sender stored securely on mobile SIM. The encrypted message along with signed value is sent to the receiver using SMS. At the receiver's end, encrypted message and signed value is extracted. Hash of encrypted message is generated and used for comparison with the received hash after decryption of signed value with the public key of sender. If both hashes match and time stamp verified, encrypted message is decrypted using private key of receiver stored securely in his SIM.

They claimed that their scheme provides authentication, confidentiality and non-repudiation. In their scheme, operations performed using private keys are performed by SIM while operations performed using public keys are done by mobile phone. Their scheme is shown in the Figure 2.2.

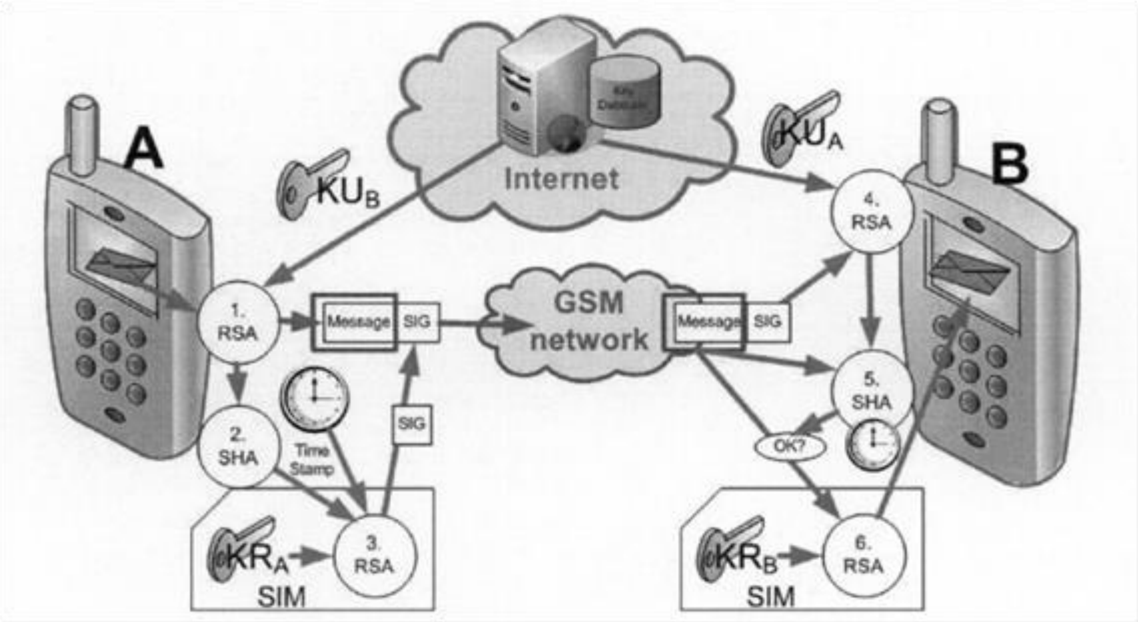


Fig 2.2 Strong Mobile Authentication Scheme [13]

Critical Analysis: Their scheme requires special SIMs (additional cost) and is prone to replay attack in case clock of both mobile phones are not synchronized.

2.2 Related Work Analysis

After analyzing current approaches we found that the most of PACS provide adequate level of identity verification and authorization services in conventional systems. Some of them are based on proprietary standards and specifically designed for specific environments so they are not cost effective or interoperable with other systems. We also observed that many of them are based on single factor authentication and some of them do not consider communication level security. Therefore, it is needed to design architecture for PACS using smartphone as authenticating device that provide strong authentication using two-factor authentication. Also it must provide standard authorization mechanism, secure communication, easy to use and transparent handling of security credentials.

Chapter 3

Smart Physical Access Control System (SPACS) Architecture

"I never did anything by accident, nor did any of my inventions come by accident; they came by work."
Thomas Edison

Physical Access Control System (PACS) has become most concerning issue for enterprises to secure their physical and logical assets. Physical security and protection of critical assets is the most important need of the time. Access control mechanisms to protect rooms, buildings, warehouse, and military installations have been around since the commercial deployment of access control devices like door locks, keypads, smart cards and biometric devices but these deployed mechanisms are either insecure, costly or have limited scalability. Also as the technology advances it become a nightmare for a medium sized enterprise to cope with new security threats and to upgrade their devices and at last will have to bear additional cost of replacement or reissuance. There is a limited freedom to change underlying security architecture and IT team has to limit their focus for finding threat countermeasure to the defined architectural boundaries. There is need to provide flexible authentication mechanism in terms of confidentiality, integrity and scalability through smartphones acting as smart software token for accessing PACS.

Most people are reluctant to provide biometric credentials like fingerprint, retina scan, hand geometry and face recognition due to privacy, social reasons or standard laws. We are going to propose architecture for PACS that will help minimize damages in case of ambush on building by terrorists and will protect precious lives and make it difficult for terrorists to break security in a limited time.

Innovation in the field of smartphone is drastically changing and these devices are becoming more and more powerful in terms of computational power, memory and secure operating systems so in long term it will be easier for any organization to change their access control architecture and choice of underlying programming language implementation. Also it will be feasible for them to let users smartphone act as their identity for different access services rather than issuing smart cards or registering each user on biometric devices and to upgrade code dynamically to cope with new security threats.

3.1 Security Requirements

Besides the obvious benefits of using smartphone as authenticating device in PACS, there are several security and privacy issues related to the users. Due to wireless nature of communication between smartphone and Physical Access Control System, message alteration or malfunctioning of the system may compromise security and may deteriorate access control matrix or even worse, be the cause of unauthorized user access to highly secure places. Before designing architecture for physical access control using smartphone, it is required to ensure the security of the system and its ability to provide reliable authentication and authorization to legitimate users. Following requirements need to be fulfilled in order to achieve reliable security.

Authentication: It is the foremost requirement of any PACS architecture. If PACS are based on *something you have* based authentication mechanism then only mutually authenticated parties are allowed to communicate with each other. Specifically, in our designed architecture user's smartphone and the Authentication Server must be mutually authenticated. It is to be ensured that Authentication Server only send responses to legitimate users after implementing an access control mechanism. Moreover, authenticity and integrity of the response messages must be retained during transmission. To achieve above mentioned security goal, the Authentication Server has to securely authenticate users and prevent impersonators from misleading legitimate users from exposing their secure credentials.

Access Control: The PACS architecture must implement an access control mechanism so that only legitimate users are allowed to open physical doors and enter secure areas after their successful identity verification and access grants verified, while access requests reported from unauthorized users should be discarded. In this way, only legitimate users can pass through doors.

Message Integrity: The PACS architecture must ensure that messages exchanged between smartphone and Authentication server is not altered by adversary. It is necessary to guarantee that each message in the network is received unchanged.

Accountability: In case of misuse of security credentials, there may be provision in PACS architecture to disclose the identity of a user and instant revoking of access rights granted against doors. Because of the criticality, initiator must take responsibility of the message sent by him. It is necessary to prevent fake access requests, which may lead to malfunctioning of the system. It is necessary to provide a mechanism for easy revocation of a user’s access to the system.

3.2 ARCHITECTURE OVERVIEW

The goal of the proposed architecture is to provide a two-factor authentication and authorization system for physical access control using smartphone as authenticating device. The components of the designed architecture are based on the principle of separation of concerns and separation of duties. It is also kept in mind while designing architecture to deprive any unauthorized entity to have the ability to access both authentication and identity management system. Furthermore, the communication protocol used for authentication is an extension of FIPS-196.

The components of our designed system are shown in Figure 3.1. These components are Identity Management System, SPACS Authentication and Authorization Server, Certification Authority, Electronic Door Panel and SPACS Application for smartphone.

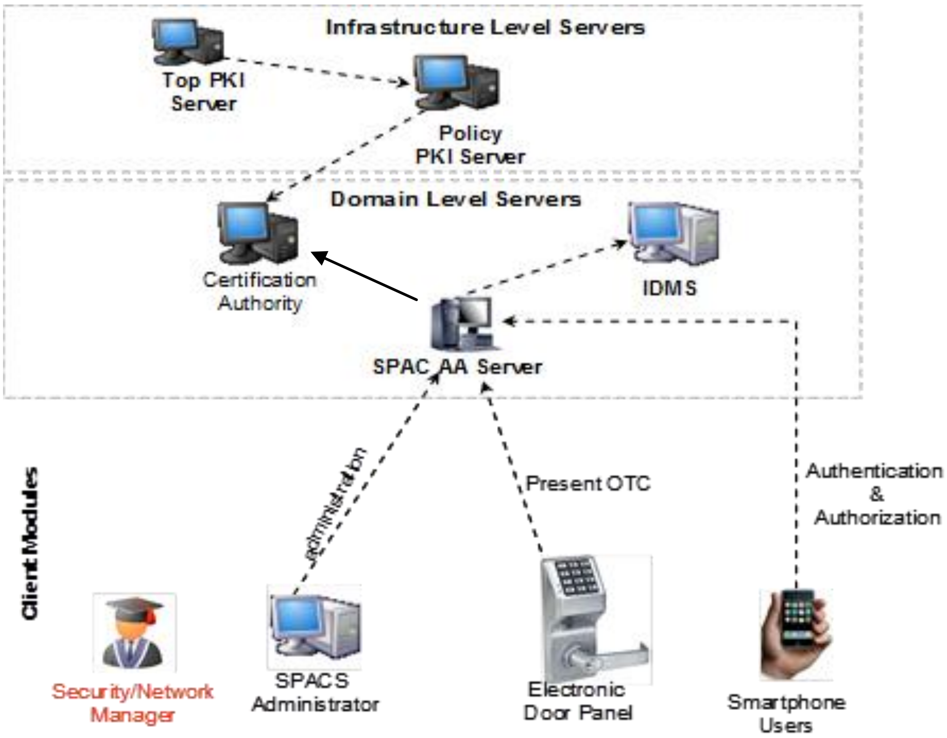


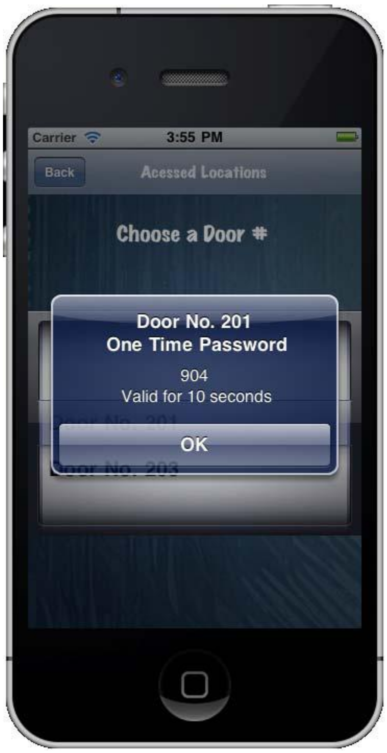
Fig. 3.1: Components of the SPACS Architecture and communication between the components

3.2.1 Identity Management System (IDMS): This server manages the information about users and resources. All users in an organization must be registered in IDMS prior to use the SPACS. Each user is registered with the system through SPACS Client application. The user downloads SPACS client application on his smartphone and initiates registration request from his smartphone. The user is registered with the system but no access grant for physical resources is allocated. The security manager grants access to designated physical resources to the user after verification from administration and activates his account to use the system.

3.2.2 SPACS Authentication and Authorization Server: This server is responsible to authenticate and authorize the user using our designed protocol. During the execution of protocol, this server verifies the user's against IDMS and also verifies user's certificate from local Certification Authority. The authorization part of this component facilitates the Security Administrator to define the access level of each user. All these access control parameters are stored in XACML based policy files which are further used by the same component to evaluate the authorization request so this entity can also act as a Policy Decision Point (PDP) in the system. After successful authentication and authorization, the system will generate random One Time Passcode and transmits it back to the user smartphone as shown in Figure 3.2. It will also maintain One Time Passcode issued against each electronic door panel in its database and transmits success signal to release door lock after successful verification of passcode. It also ensures validity of passcode and discards expired passcodes (i.e., 10 seconds).

3.2.3 Certification Authority: This is a standard certification authority which distributes X.509 certificates to all the components of the system. Each user when registered and verified by security administrator gets a user certificate from this authority. This certificate is kept on user's smartphone encrypted by Advanced Encryption Standard (AES) and secured by PIN. This certification authority is also responsible for verification of user's certificate and maintaining certification revocation list. It verifies validity of user certificate and transmits validity message to SPACS Authentication and Authorization server.

3.2.4 Electronic Door Panel: This device physically exists with each door and have secure channel (SSL) with SPACS Authentication and Authorization Server. Each door panel has a unique identifier and is registered with IDMS. It receives One Time Passcode from user through key pad and then sends to the SPACS Server using secure channel. It releases door lock after getting success message from SPACS Authentication and Authorization server.



Door No. 201
One Time Password
904
Valid for 10 seconds

Fig. 3.2: Smartphone Application showing One Time Pass Code

3.2.5 SPACS Application for smartphone: This application is downloadable to the users and only registered users can activate it after providing security credentials to the SPACS. It has user friendly interface and transparently manages security credentials on user's smartphone. It transmits IMEI (*International Mobile Equipment Identity*) along with user information to the SPACS Authentication and Authorization server during Registration Phase. The application is secured by a PIN and to execute application, user is required to enter PIN in order for the secure communication to take place between user smartphone and SPACS Authentication and Authorization server. The application displays all granted doors to the user in drop down order and when clicked for access request, it displays One Time Pass Code as shown in Figure 3.2. It uses our designed protocol for secure communication for authentication and authorization. This application also stores security credentials in a file that is encrypted and protected using the user's PIN.

All the above mentioned components use our designed protocol in order to perform authentication, exchange of session keys and authorization. We already described that each user must be registered in the IDMS and have acquired a certificate from Certification Authority.

3.3 SPACS Workflow Sequence

The workflow of messages exchanged between different components of SPACS is shown in Figure 3.3. It is assumed that user is already registered with the system and his private key is securely stored in user's smartphone.

1. User initiates door access request using his SPACS application for smartphone already installed and configured on his smartphone.
2. SPACS application for smartphone sends an authentication request to SPACS Authentication Server.
3. SPACS Authentication Server verifies user's certificate from Certification Authority and sends an error message in case of revoked or expired certificate.
4. After verification of user certificate, IMEI of user smartphone is checked from Identity Management System and an error message is sent if it is invalid or found in black listed smartphones. The authentication messages exchanged between smartphone and SPACS server are digitally signed using user's certificate and SPACS server certificate to achieve confidentiality and mutual authentication.

5. After successful mutual authentication, a session for user is created and a session key is transmitted to the smartphone for further communication.
6. SPACS smartphone application displays authorized doors on smartphone which is selected by user for door access request.
7. SPACS Authentication Server verifies authorization against door request from IDMS and sends an error message in case of unsuccessful authorization.
8. SPACS server will generate a One Time Passcode on successful authorization and sends an encrypted passcode to smartphone using session key.
9. The user enters One Time Passcode through electronic door key panel attached to the door that is verified from SPACS server.
10. The door lock is released by electronic door panel and user is allowed to pass through after successful verification.

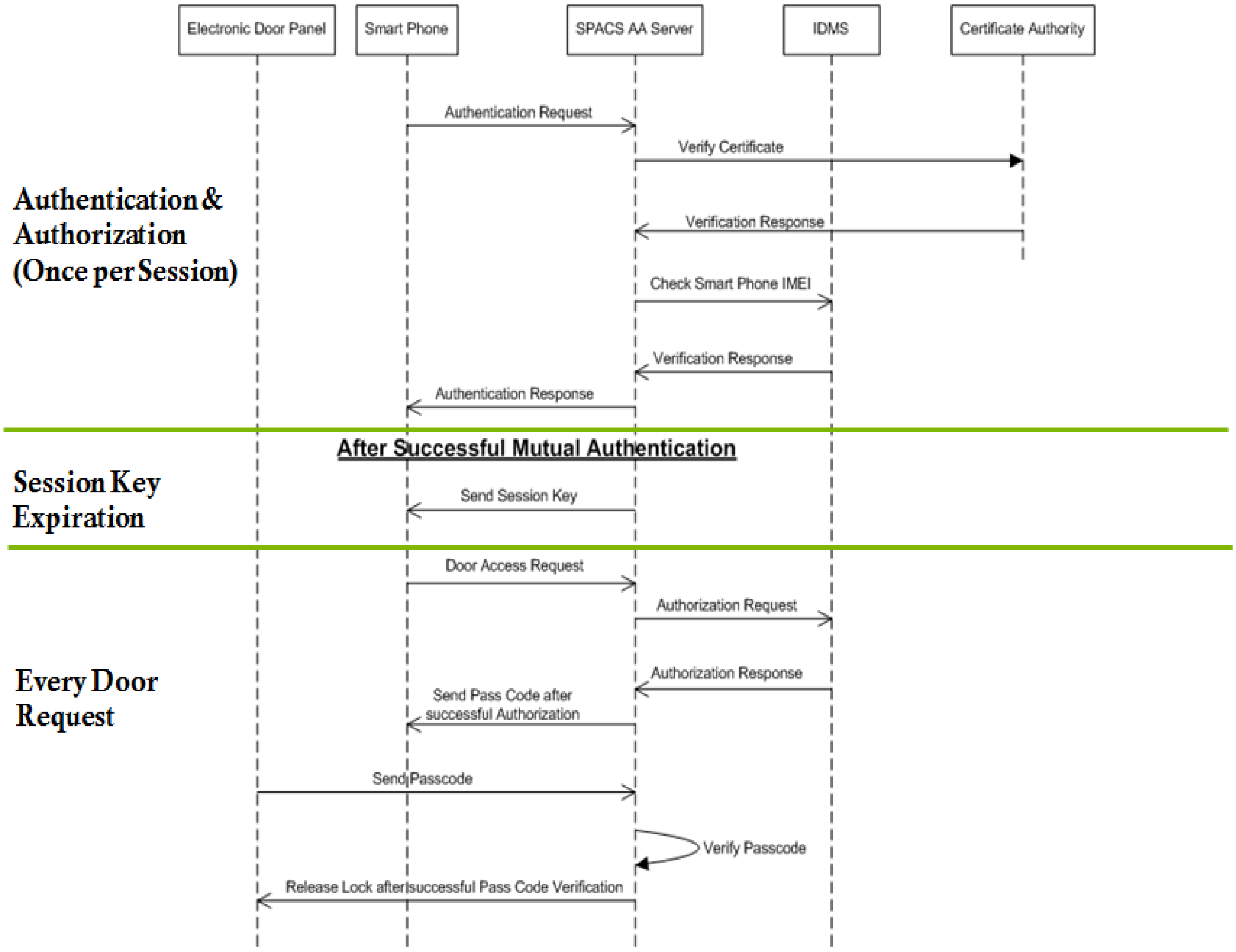


Fig. 3.3: SPACS Message Exchange Sequence

3.4 SPACS Architecture Features

The main features of our designed SPACS Architecture are as follows:

1. Provides Remote Administration for Security Managers
2. Provides a simple, efficient, secure authentication and authorization protocol
3. Minimizes security management complexities of Physical Access Control System for security professionals.
4. Provides a qualitative based solution through the use of well-known security standards.
5. Provides an alternative against smart cards, biometric devices for large scale organization by cutting costs of devices, maintenance and reissuance.
6. No additional hardware like card readers, biometric devices required, hence no special training required for security staff.
7. Two Factors authentication using PIN and Smartphone as security tokens.
8. Utilization of existing network infrastructure.
9. Extension of FIPS-196 Challenge response protocol in authentication and authorization protocol
10. Designed architecture is scalable and interoperable.

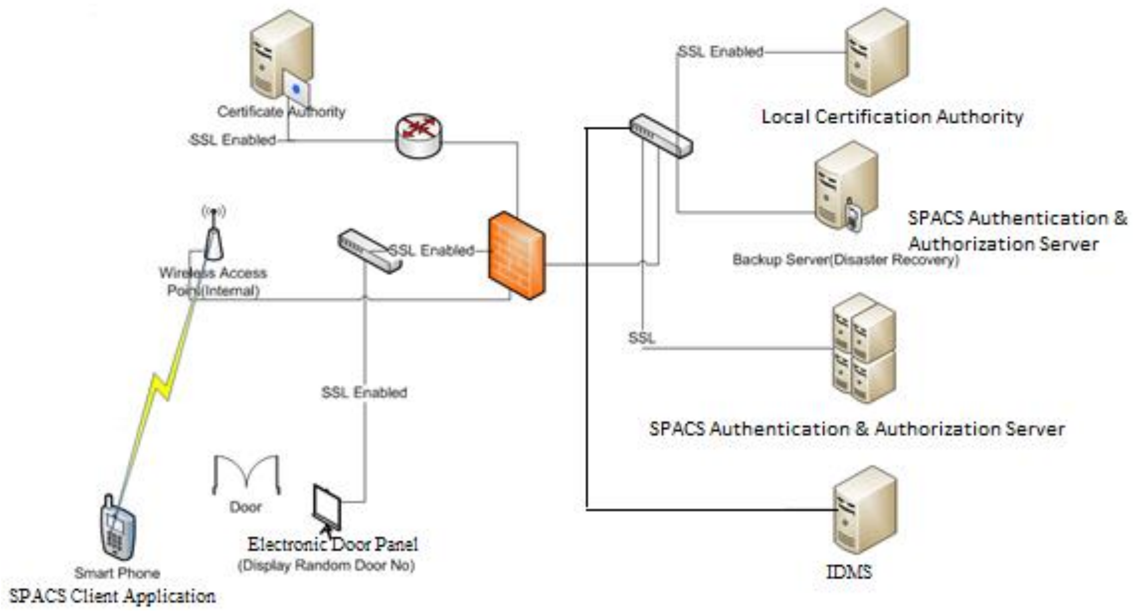

Certificate Authority
SSL Enabled
SSL Enabled
Local Certification Authority
Wireless Access Point(Internal)
SSL Enabled
SPACS Authentication & Authorization Server
Backup Server(Disaster Recovery)
SSL
SSL Enabled
SPACS Authentication & Authorization Server
Door
Electronic Door Panel
(Display Random Door No)
Smart Phone
SPACS Client Application
IDMS

Fig. 3.4: SPACS Architecture Diagram for Deployment

3.5 Areas of Application

Software Houses: Successful implementation of proposed architecture will facilitate software houses to focus on integrating additional security features for PACS instead of sorting out user devices secure communication problems.

Security Research Community: Research Community can further carry out research in PACS with respect to smartphones. Security researchers can not only analyze and explore the designed architecture and authentication and authorization protocol but also utilize in many different applications for controlling user access to subways and airports parking areas. Our research work will give new directions to researchers in terms of authorization and confidentiality for PACS.

Data Centers: Today physical security of majority of data centers are controlled by biometric devices. By using our designed architecture, user does not have to keep card/security tokens. There will be no need to install card readers or biometric devices in large data centers hence saving costs, maintenance and management.

Commercial Organizations: Organizations can adopt the designed architecture to control access to rooms, parking areas, lifts, meeting rooms and record rooms to their employees using already available network infrastructure. There will be no need to spend for card readers, biometric devices, magnetic card printers or smart card printers. They can also fully trust on security of their physical assets as designed protocol will provide security, reliability and confidentiality.

Educational Institutes: Universities can also be another area of interest for the practical deployment of designed architecture. By using faculty smartphone as their authenticating devices they can restrict user's access to only desired physical locations in the university.

Military Non Sensitive Areas: The designed architecture can also be used as another security layer for physical access to military barracks, workshops or training areas.

Baffle Gate/ Parking Area: The designed architecture can also be used to control access of users passing baffle gates in subways or public parks where one has to pay entry fee. Furthermore it can also be used in paid parking plazas.

Chapter 4

SPACS Authentication and Authorization Protocol

"The only real valuable thing is intuition."
Albert Einstein

Communication between Smartphone and SPACS AA server in our designed architecture for Smart Physical Access Control System defined in Chapter 3 uses our designed Authentication and Authorization Protocol. The designed protocol uses two common authentication techniques; (1) using X.509 certificate and challenge-response extending FIPS-196 and (2) One-Time Passcode (OTP). The designed protocol completes its functions in the following three phases:

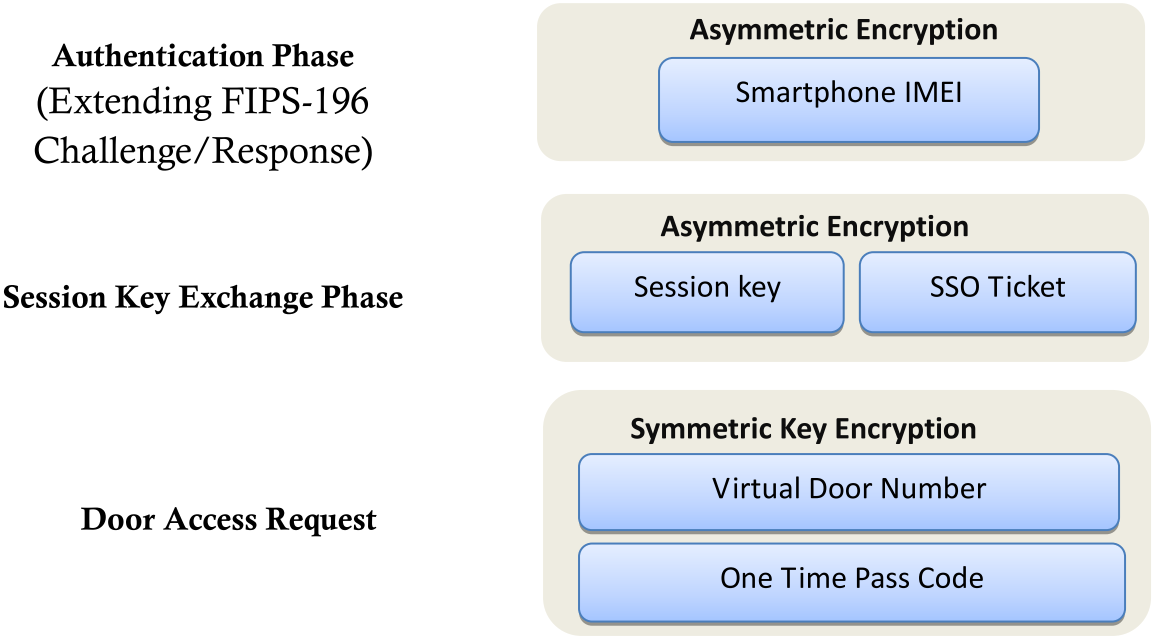


Fig 4.1 SPACS Authentication & Authorization Protocol Phases

4.1 Design Goals

Secrecy: Our first and foremost design consideration is to make sure that the designed protocol must retain confidentiality of secret data trasmitted from smartphone to SPACS server and vice versa. In case of active attacks by adversary, the messages captured must be hard to decipher and difficult to read contents. To achieve this we have used both assymetric and symmetric cryptographic primitives in our designed protocol. We have also made use of Public Key Infrastructure (PKI) in our designed SPACS architecture.

Message Integrity*:* Our next design goal is to make sure that integrity of message contents be retained during transmission. In case of message modification by adversary, the designed protocol must have the ability to discard modified messages. To achieve this we have used cryptographic hashes with timestamps in our designed protocol.

Mutual Authentication: Our next design goal is to make sure that protocol must provide mutual authentication between smartphone and SPACS server. The protocol must discard all fake smartphone or SPACS Server. To achieve this we have used nonces and extended challenge response FIPS-196 protocol. We have also used PKI infrastructure for the verification of digital certificates used for signing in our designed protocol.

Non Repudiation: Our next design goal is to make sure that participating smartphone and SPACS server must own responsibility of messages exchanged between them. To achieve this we have used digital certifcates verified by Certification Authority by the protocol.

4.2 Authentication and Authorization Protocol

Registration: The legitimate user is registered with the Identity Management System through SPACS Client application for Smartphone. During Registration, his/her SmartPhone IMEI (International Mobile Equipement Identity) along with user information is sent to the SPACS AA Server that is stored in the database of IDMS. The security administrator activates the user account and sets his/her physical door access priveleges. A user digital certificate is issued from Certification Authority and is sent to the user's smartphone which is stored securely encrypted with PIN by SPACS Smartphone Client Application.

The designed protocol is an extension of FIPS-196 Challenge/Response authentication technique and is based on methodology of two-factor authentication. The protocol is invoked by the user through smartphone application by first providing PIN *(something you know)* that decrypts secured user's digital certificate stored on smartphone *(something you have)* for further securing of credentials exchanged between Smartphone and SPACS server.

4.2.1 Authentication Phase

This phase is an extension of FIPS-196 Challenge/Response protocol. It is responsible for mutual authentication between Smartphone and SPACS AA Server. In this phase, the user initiates the mutual authentication process using SPACS Client application for Smartphone with SPACS server. It generates a client nonce and forms a message which contains nonce (CN), smartphone timestamp (CT), and hash of client nonce, smartphone IMEI and timestamp as shown in Eq.1. Smartphone Application encrypts the hash value with user's private key and forms a token containing Client nonce (CN) as challenge, Client Time (CT) and signed hash as shown in Eq.1 before transmitting to the SPACS server.

Token AB1: $\mathbf{[CN|\ CT\ |(RSA_{Encrypt}\ \{H\ (CN||IMEI\ ||CT)\},\ K_{priv}\ )]}$ (1)

SPACS server after successfully receiving the authentication message follows the following sequence as shown in Figure 4.2.

1. It checks time clock synchronization and discards all messages received after predefined time slot.
2. It verifies the user's certificate from Certification Authority.
3. It verifies its identity from IDMS by retrieving IMEI form IDMS database. It also checks IMEI validity from black listed and blocked smartphone's IMEI maintained in IDMS database.
4. It extracts Client nonce (CN), Client Timestamp (CT) and signed hash from received message.
5. It generates hash of Client nonce, IMEI and Client Time.

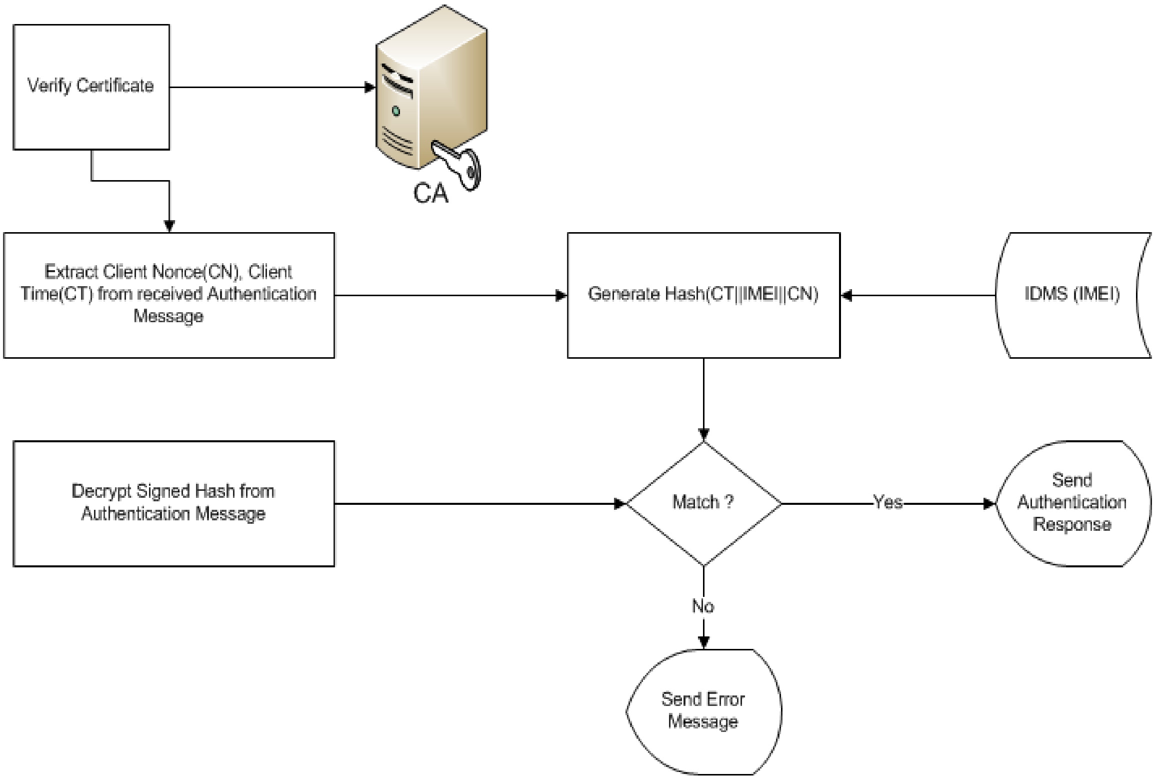

Verify Certificate
CA
Extract Client Nonce(CN), Client Time(CT) from received Authentication Message
Generate Hash(CT||IMEI||CN)
IDMS (IMEI)
Decrypt Signed Hash from Authentication Message
Match ?
Yes
Send Authentication Response
No
Send Error Message

Fig 4.2: Smartphone Authentication Token Verification

6. It decrypts signed hash and compares generated hash with received hash.
7. On successful verification, it sends authentication response as defined in Eq.2.

If authentication message sent from smartphone is successfully verified, SPACS server generates server nonce (SN) and forms a message which server nonce (SN), server timestamp (ST), and hash of server nonce, smartphone IMEI, client nonce (CN) as response and server timestamp as shown in Eq.2. SPACS server encrypts the hash value with server's private key and forms a token as shown in Eq.2 before transmitting to the Smartphone application.

Token BA1: $\mathbf{[SN|\ ST\ |\ (RSA_{Encrypt}\ \{\ H\ (SN||IMEI\ ||CN||ST)\}\ KS_{priv}\ )]}$ (2)

Smartphone application after successfully receiving the authentication response follows the following sequence as shown in Figure 4.3.

1. It checks time clock synchronization and discards all messages received after predefined time slot.
2. It verifies the server's certificate from Certification Authority.
3. It extracts Server nonce (SN), Server Timestamp (ST) and signed hash from received message.
4. It generates hash of Server nonce, IMEI, Client nonce and Server Time.
5. It decrypts signed hash and compares generated hash with received hash.
6. After successful verification, the user sends a request to the server to exchange the session key.

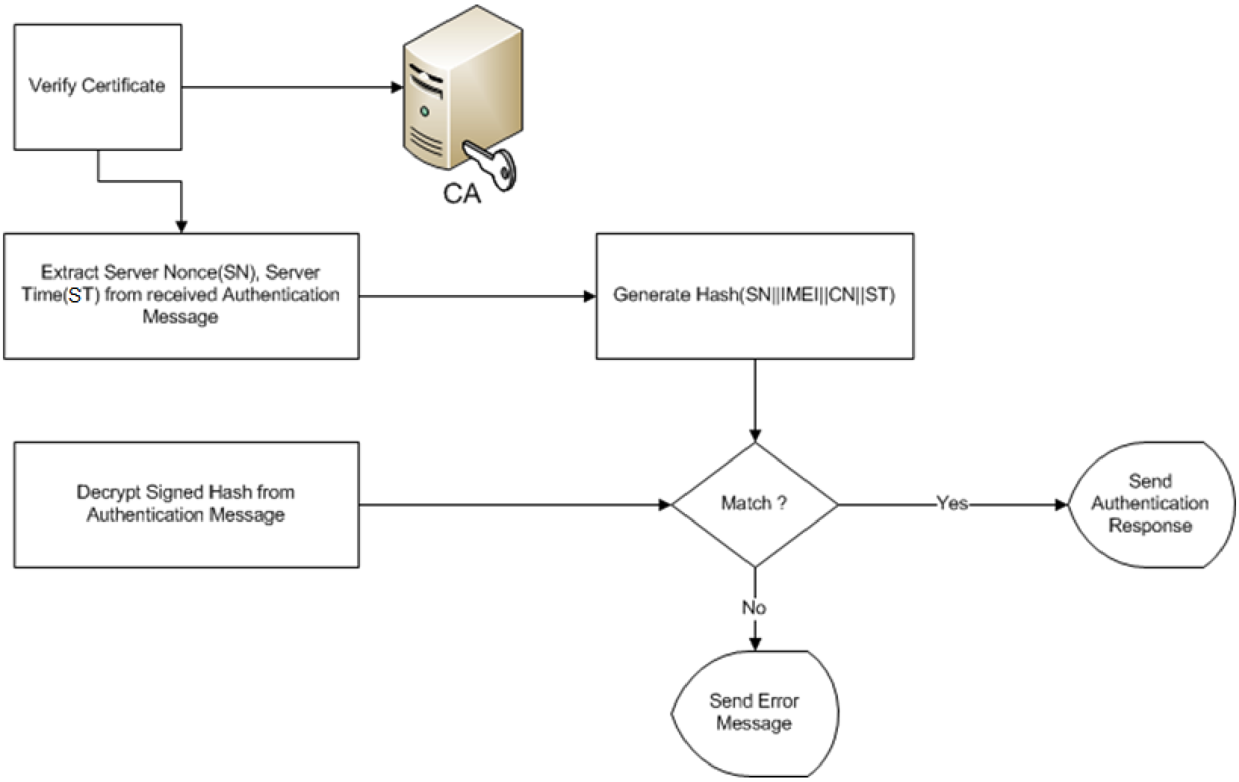

Verify Certificate
CA
Extract Server Nonce(SN), Server Time(ST) from received Authentication Message
Generate Hash(SN||IMEI||CN||ST)
Decrypt Signed Hash from Authentication Message
Match ?
Yes
Send Authentication Response
No
Send Error Message

Fig 4.3: Server's Authentication Response Verification

4.2.2 Session Key Exchange Phase

This phase is invoked by SPACS Smartphone Application after successful mutual authentication of smartphone and SPACS server in Authentication Phase. The SPACS server generates a random session key (K_s) and a Single-Sign-On (SSO) ticket (T`). This generated session key K_s is only valid for short span of time and may expire after some time. The smartphone application will have to reinitiate the Session Key Exchange Phase after expiry of K_s to get new session key. SPACS server maintains sessions of all connected smartphone users on the server and update newly issued session keys instantly. The SPACS server encrypts the K_s and T` by using the public key of smartphone's user and then digitally signing it with its own private key as shown in Eq.3. The SPACS sever sends the encrypted message to the SPACS smartphone application. This protocol phase is automatically invoked as the session expires and it retransmits new session key message to smartphone application.

Token BA2: $$\mathbf{RSA_{Encrypt}\ \{\ K_s\ ,\ ST,\ T',\ [RSA_{Encrypt}\ \{\ H(T'||\ K_s\ ||\ ST)\}\ KS_{priv}\ ]\}\ K_{pub}} \quad (3)$$

SPACS Smartphone application after successfully receiving the session key message follows the following sequence as shown in Figure 4.4.

1. It discards all session key messages received after predefined time slot.

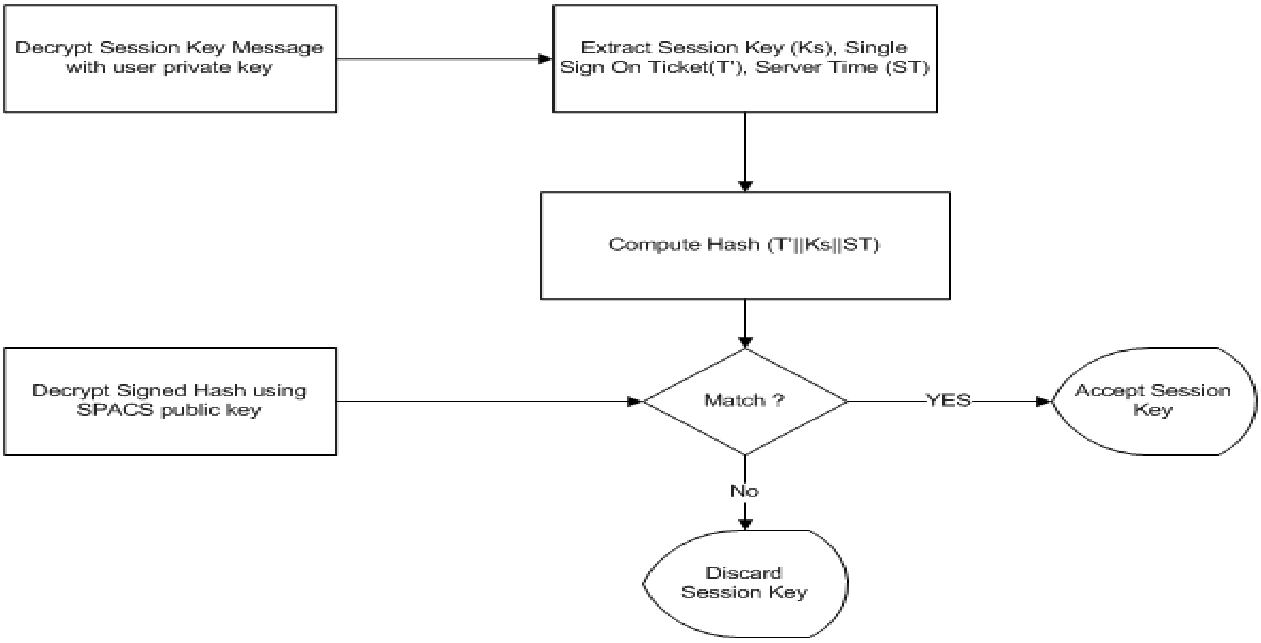

Decrypt Session Key Message with user private key
Extract Session Key (Ks), Single Sign On Ticket(T'), Server Time (ST)
Compute Hash (T'||Ks||ST)
Decrypt Signed Hash using SPACS public key
Match ?
YES
Accept Session Key
No
Discard Session Key

Fig 4.4: Server's Session Key Message Verification

2. It decrypts session key message using user's private key.
3. It extracts SSO ticket T', session key K_s and server time ST.
4. It computes hash of T', Ks and ST and compares it with received hash after decrypting it with SPACS server public key.
5. On successful validation, it accepts and stores session key on smartphone for further communication in Door Access Phase of protocol defined later in this chapter.

The SPACS Smartphone Client application will update newly generated session key after expiry of session and will discard old session key. It will also store SSO Ticket (T') that will be used to access logical resources currently not in scope of this research.

4.2.3 Door Access Request Phase

After successfully receiving session key by SPACS Smartphone Application, a list of all granted physical doors in Access Control System is transmitted to SPACS Smartphone Application and is populated. This phase is initiated after selection of physical door in SPACS Smartphone Application by user. In this phase, user's smartphone sends a physical access request to the SPACS Server. This request contains client time CT, hash (SSO Ticket T' and CT) and door number (DN). The request is formed according to the Eq.4 which is encrypted using the session key K_s.

Token AB2: **Encrypt [{CT| H(T`||CT)| DN}, K_s]** (4)

SPACS Smartphone application after successfully receiving the door access request message follows the following sequence.

1. It discards all door access request messages received after predefined time slot.
2. It decrypts received message using session key K_s.
3. It extracts client time CT and door number DN.

4. It generates hash of SSO Ticket T’ and client time CT and compares it with received hash.
5. It consults XACML policy file to determine access rights of user against requested door.
6. If he is authorized to enter in the specified door number (DN) then the system will generate One Time Passcode.
7. SPACS server stores One Time Passcode against door number in the database for further processing.
8. It generates access granted message to the SPACS Smartphone Application containing encrypted One Time Passcode and Server time ST as shown in Eq.5.

Token BA3: **Encrypt [{OneTimePassCode, ST}, K_S]** (5)

The mobile application will display the pass-code on the mobile screen. The validity period of this pass-code is for ten (10) seconds but it can be changed according to the requirement. The use enters the pass-code in the electronic door panel shown in Figure 4.5 which creates a secure session with SPACS server. It sends the pass-code to the SPACS server along with electronic door panel terminal id. The SPACS server receives the message and verifies it from the local database. If it is a valid pass-code then it permits to open the lock otherwise it denies.

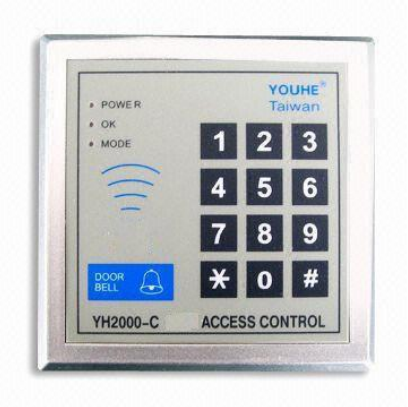
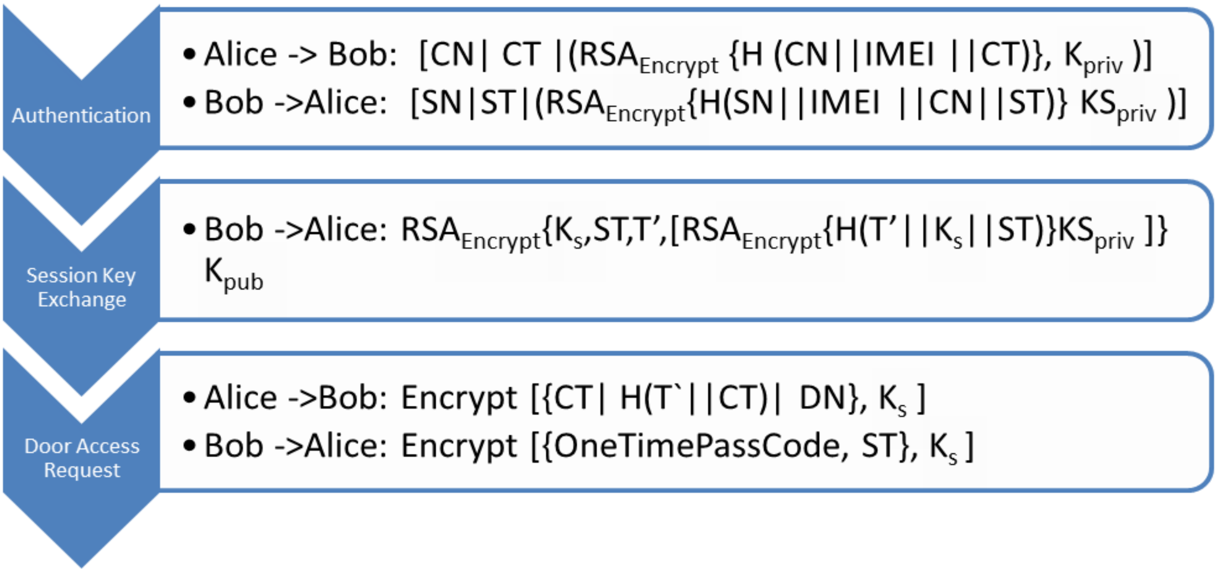


Fig 4.5: SPACS Electronic Door Access Panel

The complete SPACS Authentication and Authorization protocol blueprint is shown in Figure 4.6.



Authentication

- Alice -> Bob: [CN| CT |($\text{RSA}_{\text{Encrypt}}$ {H (CN||IMEI ||CT)}, K_{priv})]
- Bob ->Alice: [SN|ST|($\text{RSA}_{\text{Encrypt}}${H(SN||IMEI ||CN||ST)} KS_{priv})]

Session Key Exchange

- Bob ->Alice: $\text{RSA}_{\text{Encrypt}}${$K_s$,ST,T',[$\text{RSA}_{\text{Encrypt}}${H(T'||$K_s$||ST)}$KS_{\text{priv}}$]} K_{pub}

Door Access Request

- Alice ->Bob: Encrypt [{CT| H(T`||CT)| DN}, K_s]
- Bob ->Alice: Encrypt [{OneTimePassCode, ST}, K_s]

Fig 4.6: SPACS Authentication & Authorization Protocol

The implementation of designed SPACS architecture and communication protocol is done using JAVA coded SPACS AA Server and Objective C as a proof of concept. It is simulated using IPhone 4 emulator on Apple laptop connected with a SPACS AA server (Corei3 server with 4 GB RAM) through hub. The base OS of server is Windows 2003 Enterprise Server with X.509 Windows Certification Authority.

Chapter 5

Formal Verification of SPACS Authentication and Authorization Protocol

"Mathematics is the science which draws necessary conclusion"
Benjamin Peirce

We formally verified our protocol using automated security verification tool, Scyther [2]. Scyther is an automated security protocol checking tool that has been effective for analysis, verification and falsification of security protocols. It guarantees termination and verifies protocol with unbounded number of sessions. Also it is capable of verifying synchronization. In other words, synchronization is the transmission of messages sent received exactly as prescribed in the protocol description. This property is a strictly stronger than agreement for the standard intruder model and is used to detect preplay attacks.

5.1 Formal Analysis of Protocol

To analyze security protocol, Scyther uses backward symbolic state search technique that uses Arachne engine based on Athena method [17]. The operational semantics of Scyhter is based on *"Operational semantics of security protocol"* [18] defined by C. Cremers and S. Mauw while the attacks are found from broken claims through searching backward by Arachne engine. Scyther handles both multiple key structures and non-atomic keys and verifies authentication properties like synchronization.

5.1.1 Model Description

In our model, we have defined *Initiator* or *Responder* roles to elaborate behavior of the protocol. There are two communicating agents, Smartphone (SP) and SPACS AA Server (SS) in our system and each agent perform both roles. When an agent performs a role, it is called a run and agent's described security goals are checked by the execution of their runs. Each role specification consists of certain security claims as well as sequence of events describing *send* and *receive* messages described for agent in the protocol. An intruder or adversary may try to negate such security goals. His capabilities determine his strength in attacking protocol runs while agent may use cryptographic primitives to resist such attacks.

We focus on following security properties in this thesis: secrecy, secrecy of session key, authentication and secrecy of One Time Passcode. An adversary never gets knowledge of something claimed in the secrecy claim statement. The session key claim states that the session key must be secret and acts as session identifier and must be unique across all runs of the protocol for the same role. The notion of synchronization is used to validate authentication which states that corresponding *send* and *receive* messages are executed in the expected order. The claim that is tested for the role is denoted by *x* and *y* in the message. Following claims have been used in formal verification:

- **Claim (*x*, *Secret*, *y*)** states that intruder will never get knowledge of secret *y* claimed by agent performing the role *x*.
- **Claim (*x*, *Nisynch*)** states that the messages received by the agent *x* has been received by authenticated sender.

The components of security protocol model is described as *Protocol specification, Agent model, Communication model, Threat model, Cryptographic primitives* and *Security requirements*. The behavior of entire system is encoded in traces and every claim event is determined *may be true* or not by trace results.

The behavior of the roles in protocol is described by *Protocol specification* and a role is more often described as sequential list of events. The execution of roles of the protocol is described by *Agent model* and is usually based on assumption that honest agents show behavior in conjunction with protocol specification. The *Threat model* is a parameter of semantics based on Dolev-Yao network threat model [19] where communication network is under intruder control. The *Cryptographic primitives* are ideal mathematical constructs such as encryption and are used using black box technique in which adversary cannot get anything from encrypted message except if he succeeds to become aware of encryption key. The *Security requirements* are safety properties to ensure that something bad will never happen.

5.1.2 Security Properties Specification

Information Confidentiality: This claim is fulfilled if the *International Mobile Equipment Identity* (IMEI) which consists of 15 digits of Smartphone (SP) is not revealed to the adversary during transmission of mutual authentication messages between Smartphone and SPACS AA server. The formal definition is shown below:

Property 1: *claim (SPACS AA|SP, Secret, IMEI)*

Information Confidentiality of Session Key Exchange & Door Access Request Messages: The below mentioned claims are fulfilled if the user data exchanged between SPACS AA Server and Smartphone for Session Key Exchange and Door Access Request Phases are kept secret. All the messages exchanged between SPACS AA Server and Smartphone is called Msg. User data information (**α**) in Msg should remain secret. The formal definitions are shown below:

Property 2: $\forall_{\alpha \in \mathrm{Msg}}$ (*claim (SP, Secret, α)*)

Property 3: $\forall_{\alpha \in \mathrm{Msg}}$ (*claim (SPACS AA, Secret, α)*)

Secrecy of Session Key: If the secrecy of exchanged session key (*skey*) is ensured between SPACS AA Server and Smartphone, then this property is fulfilled. This claim is evaluated for concerned sessions between trusted agents. The formal definition is shown below:

Property 4: $\forall_{\mathrm{skey}}$ (*claim (SPACS AA|SP, Secret, skey)*)

Authenticity of Messages: It the messages exchanged between Smartphone and SPACS AA server are received from trusted agent and are received in synchronized order as defined in protocol specification, then this claim is fulfilled. The formal definition is as follows:

Property 5: (*claim (SPACS AA|SP, Nisych)*)

5.1.3 Formal Verification

Following table shows attributes used in the messages in our formal model. The results obtained from Scyther are shown in Figure 5.1.

Message Element	Description
Na	Smartphone's Nonce
Nb	SPACS AA's Nonce
SPCert	Smartphone's Certificate
SPACSCert	SPACS AA Server's Certificate
skey	Session Key
T'	SSO Ticket
Tc	Smartphone's Timestamp
Ts	SPACS AA Server's Timestamp
dno	Door identifier
OTP	One Time Passcode
Sig(x)	x's RSA signature over attributes of message
{x}pk(y)	x is encrypted with the public key of y
H(x)	One way Cryptographic Hash of x
{x}y	x is encrypted symmetrically by y

Table 5.1 : Message Types

The formal definition of *mutual authentication* scenario in our protocol (Authentication Phase) is described as below:

- *SP→SPACS AA : Na,Tc, Sig(H(Na,IMEI,Tc))*
- *SPACS AA→SP : Nb,Ts,Sig(H(Na,IMEI,Nb,Ts))*

The above mentioned model is challenged using scyther tool against following requirements:

1. *Property 1:* The property is proved and confidentiality of IMEI is ensured. The nonce has been used for identification of legitimate communicating server and smartphone. Public Key Infrastructure (PKI) is used to issue and validate certificates and to handle forgery attacks and *Man-in-the-Middle* attack. Cryptographic Hash has been used for message integrity and timestamp is used to ensure synchronization. No attack tree has been generated by Scyther to show possible attack.

The formal definition of *Session Key Exchange* and *Door Access Request* scenario in our protocol is described as below:

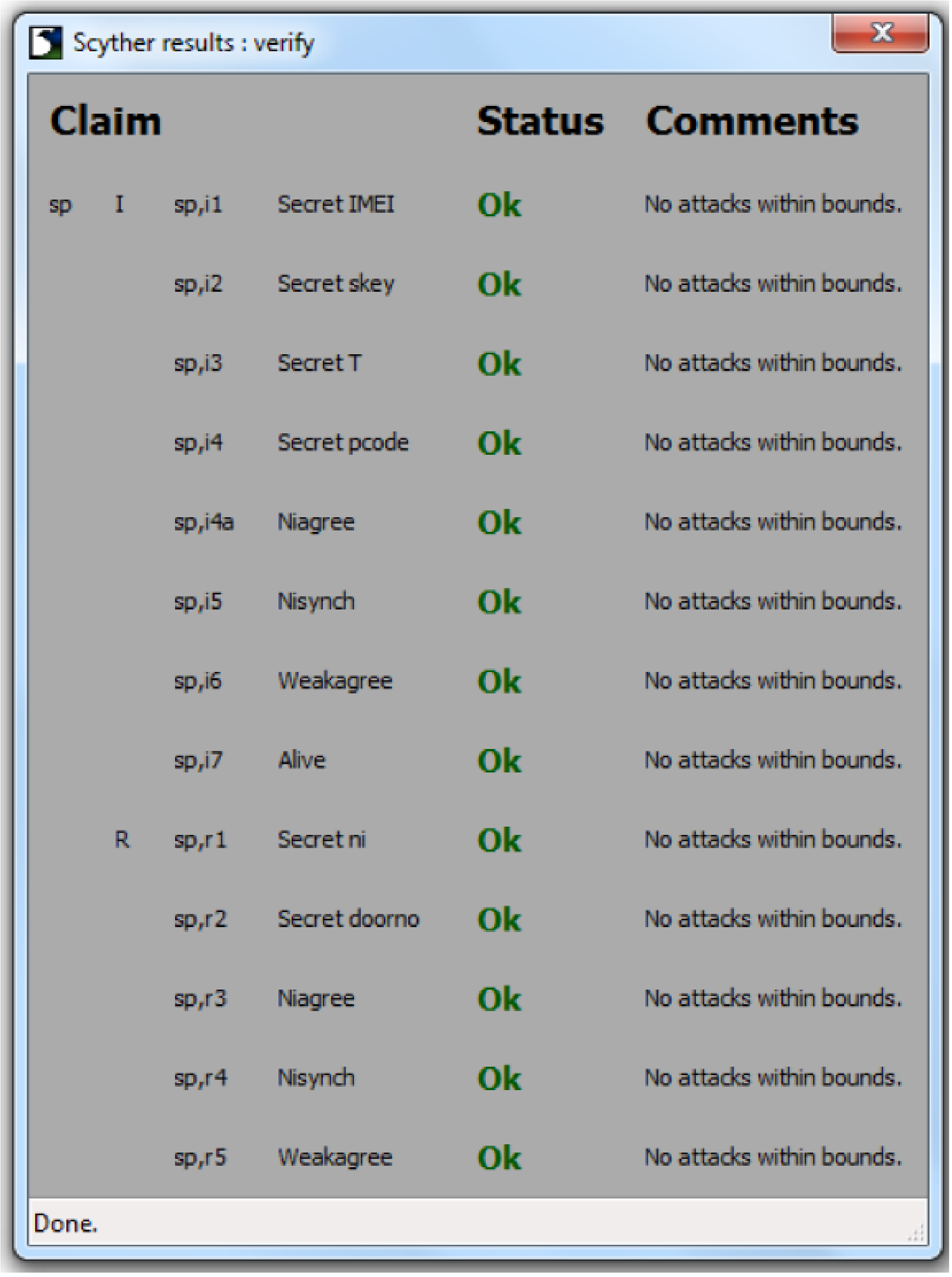
- *SPACS AA→SP: {skey, T', Ts, Sig(SPACS AA)}pk(SPCert)*
- *SP→SPACS AA : {dno,Tc,H(dno,T',Tc)} skey*
- *SPACS AA→SP: {OTP, Ts} skey*

The above mentioned model is challenged using scyther tool against following requirements:

2. *Property 2 & Property 3:* This property is proved and confidentiality of all user data exchanged between SPACS AA server and Smartphone in *Session Key Exchange* and *Door Access Request* phases of authentication protocol is ensured. Public Key Infrastructure (PKI) is used to validate certificates and to handle forgery attacks and *Man-in-the-Middle* attack for session key exchange. Also Cryptographic Hash has been used for message integrity and timestamp is used to ensure the messages are received from authenticated agent and not replayed. No attack tree has been generated by Scyther to show possible attack.
3. *Property 3:* It is proved that an adversary cannot get session key during transmission from SPACS AA server to Smartphone as that message is encrypted with the public key as shown above. Also skey is proved to be unique because of the fact that it is transmitted by agent in one send event and is signed by the private key of server. The adversary will not be able to learn the contents of door access request message and one time passcode as they are encrypted with unique session key transmitted by the server. The new unique session key will be transmitted to smartphone after expiry of the session. Timestamp has been added in the message to ensure synchronization and message integrity is maintained

through cryptographic hash. No attack tree has been generated by Scyther to show possible attack.

4. *Property 5:* This property is proved as the messages are sent and received as defined in the protocol specification and is accordance with the roles defined. Agents get send or receive messages in synchronized order.



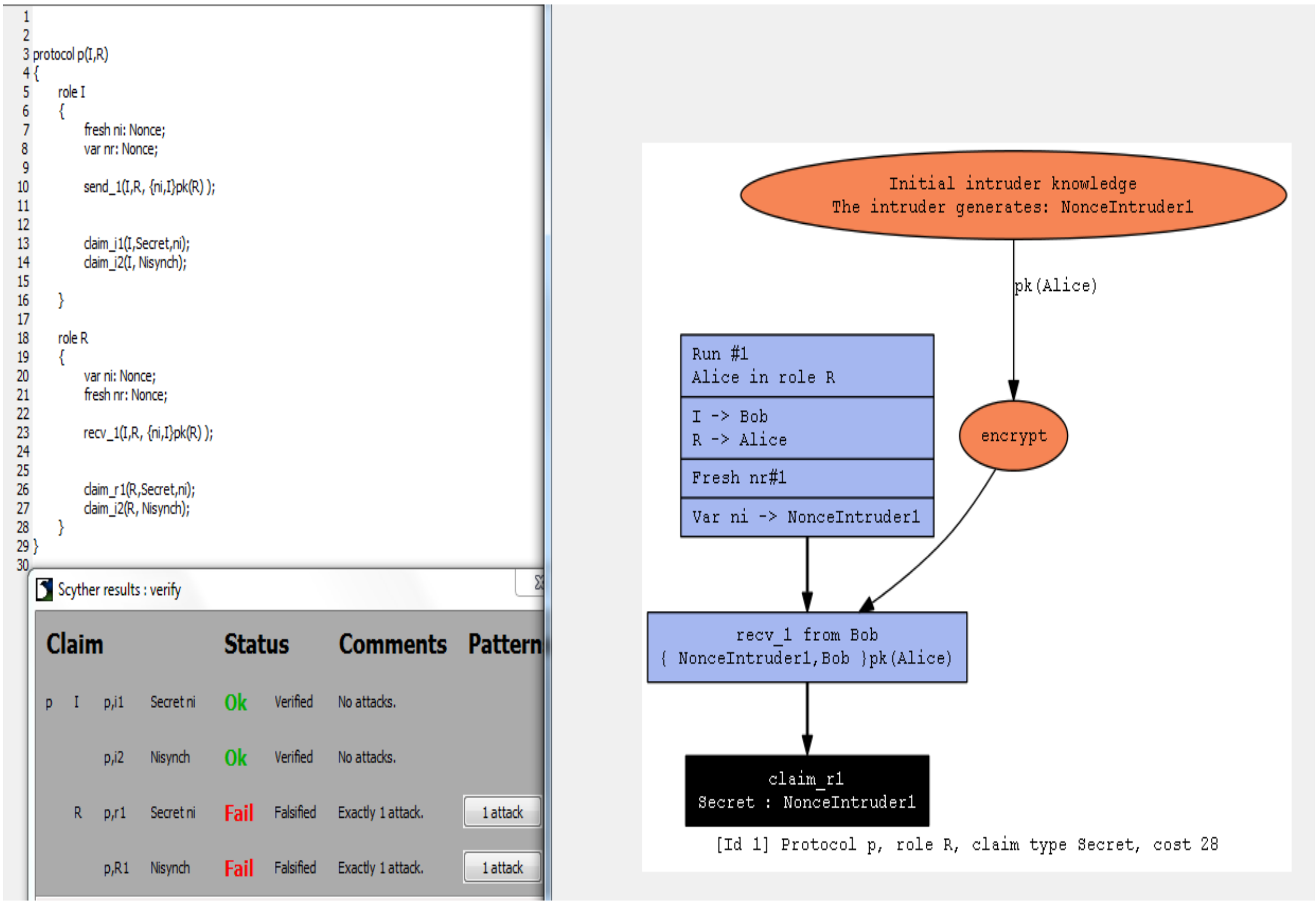
Scyther results : verify

Claim				Status	Comments
sp	I	sp,i1	Secret IMEI	Ok	No attacks within bounds.
		sp,i2	Secret skey	Ok	No attacks within bounds.
		sp,i3	Secret T	Ok	No attacks within bounds.
		sp,i4	Secret pcode	Ok	No attacks within bounds.
		sp,i4a	Niagree	Ok	No attacks within bounds.
		sp,i5	Nisynch	Ok	No attacks within bounds.
		sp,i6	Weakagree	Ok	No attacks within bounds.
		sp,i7	Alive	Ok	No attacks within bounds.
	R	sp,r1	Secret ni	Ok	No attacks within bounds.
		sp,r2	Secret doorno	Ok	No attacks within bounds.
		sp,r3	Niagree	Ok	No attacks within bounds.
		sp,r4	Nisynch	Ok	No attacks within bounds.
		sp,r5	Weakagree	Ok	No attacks within bounds.

Done.

Figure 5.1 : Scyther Generated Verification Results

In case of any attack tree generated during protocol runs which indicates that protocol is prone to attacks. An example of such protocol is shown in the figure below that shows that protocol compromises confidentiality and hence adversary succeeded in breaking security as claimed by protocol designer.



```


protocol p(I,R)
{
	role I
	{
		fresh ni: Nonce;
		var nr: Nonce;

		send_1(I,R, {ni,I}pk(R) );


		claim_i1(I,Secret,ni);
		claim_i2(I, Nisynch);

	}

	role R
	{
		var ni: Nonce;
		fresh nr: Nonce;

		recv_1(I,R, {ni,I}pk(R) );


		claim_r1(R,Secret,ni);
		claim_i2(R, Nisynch);
	}
}

```

Claim				Status		Comments	Pattern
p	I	p,i1	Secret ni	Ok	Verified	No attacks.	
		p,i2	Nisynch	Ok	Verified	No attacks.	
	R	p,r1	Secret ni	Fail	Falsified	Exactly 1 attack.	1 attack
		p,R1	Nisynch	Fail	Falsified	Exactly 1 attack.	1 attack

Initial intruder knowledge
The intruder generates: NonceIntruder1
pk(Alice)
Run #1
Alice in role R
I -> Bob
R -> Alice
Fresh nr#1
Var ni -> NonceIntruder1
encrypt
recv_1 from Bob
{ NonceIntruder1,Bob }pk(Alice)
claim_r1
Secret : NonceIntruder1
[Id 1] Protocol p, role R, claim type Secret, cost 28

Figure 5.2 : Scyther Generated Verification Attack Tree

5.2 Scyther SPDL Script for Protocol Verification

Following is the Scyther SPDL script used for verification of SPACS Authentication & Authorization protocol:

```
//================SPDL Script ====================

usertype InternationalMobileEquipmentIdentity, TimeStamp, SessionKey, ResourceId,
PINCode;
const Fresh:Function;

// I for Smartphone, R for SPACS Server, X for Intruder

protocol intruder(X)
{
role X
{
fresh nx: Nonce;
 fresh Tx: TimeStamp;
var ni: Nonce;
var Ti: TimeStamp;
var  IMEI: InternationalMobileEquipmentIdentity;
fresh Tkx:Ticket;
var nr: Nonce;
 var Tr: TimeStamp;
var skey:SessionKey;
var T:Ticket;
fresh skeyx:SessionKey;

recv_1!X(X,X,I,R, (ni,Ti,{H(ni,IMEI,Ti)}sk(I)));
send_1!X(X,X,I,R, (nx,Tx,{H(ni,IMEI,Ti)}sk(I)));

recv_2!X(X,X,R,I, (nr,Tr,{H(ni,IMEI,nr,Tr)}sk(R)) );
send_2!X(X,X,R,I, (nx,Tx,{H(ni,IMEI,nr,Tr)}sk(R)) );

recv_3!X(X,X,R,I,{skey,Tr,T,{H(T,skey,Tr)}sk(R)}pk(I));
send_3!X(X,X,R,I,{skeyx,Tx,Tkx,{H(T,skey,Tr)}sk(R)}pk(I));

}
}

protocol sp(I,R)
 {
 role I
  {
        fresh ni: Nonce;
        var nr: Nonce;
        fresh Ti: TimeStamp;
        var Tr: TimeStamp;
        hashfunction H;
        fresh  IMEI: InternationalMobileEquipmentIdentity;
        fresh doorno:ResourceId;
        var skey:SessionKey;
        var pcode:PINCode;
        var T:Ticket;

//Mutual Authentication and Registration Phase Message
        send_1(I,R, (ni,Ti,{H(ni,IMEI,Ti)}sk(I)));
        recv_2(R,I, (nr,Tr,{H(ni,IMEI,nr,Tr)}sk(R)) );

//Session Key Phase Message
        recv_3(R,I,{skey,Tr,T,{H(T,skey,Tr)}sk(R)}pk(I));

//Door Access Request Phase
        send_4(I,R,{T,doorno,H(Ti,T)}skey);
        recv_5(R,I,{pcode,Tr}skey);

// Security properties Claimed by SmartPhone
        claim_i1(I,Secret,IMEI);
        claim_i2(I,Secret,skey);
        claim_i3(I,Secret,T);
        claim_i4(I,Secret,pcode);
        claim_i4a(I,Niagree);
        claim_i5(I, Nisych);
   }

    role R
   {
        var ni: Nonce;
        fresh nr: Nonce;
         var Ti: TimeStamp;
        fresh Tr:TimeStamp;
        hashfunction H;
        var IMEI:InternationalMobileEquipmentIdentity;
        var doorno:ResourceId;
        fresh skey:SessionKey;
        fresh pcode:PINCode;
        fresh T:Ticket;

   //Mutual Authentication and Registration Phase Message
        recv_1(I,R, (ni,Ti,{H(ni,IMEI,Ti)}sk(I)));
        send_2(R,I, (nr,Tr,{H(ni,IMEI,nr,Tr)}sk(R)) );

   //Session key Phase Message
        send_3(R,I,{skey,Tr,T,{H(T,skey,Tr)}sk(R)}pk(I));

   //Door Access Request Phase
        recv_4(I,R,{T,doorno,H(Ti,T)}skey);
        send_5(R,I,{pcode,Tr}skey);

   // Security properties Claimed by SPACS Server
         claim_r1(R,Secret,IMEI);
        claim_r5(R,Secret,skey);
        claim_r2(R,Secret,doorno);
        claim_r4(R,Nisynch);
        claim_r3(R,Niagree);
   } }
```

Chapter 6

Conclusion

"A conclusion is the place where you got tired thinking"
Martin H. Fischer

In this thesis, we have designed and implemented architecture and communication protocol for smartphone based authentication and authorization for PACS. The designed communication protocol is an extension of FIPS-196 challenge/response protocol and provides two-factor authentication (Smartphone & PIN). The communication protocol is divided into three phases: *Authentication, Session key Exchange* and *Door Access Request* phase. The protocol provides mutual authentication between smartphone and SPACS Authentication and Authorization server. Furthermore, the usage of symmetric key cryptography provides an efficient solution to achieve confidentiality of messages exchanged between components of SPACS. In order to ensure the presence of the legitimate user in the premises, the system uses a pass-code feature that is only valid for one time usage. Hence using our protocol, user's smartphone can act as authenticator in the system. Since, the solution may be used by the non-technical persons so it is designed that it should be user friendly and require minimum efforts for configuration of security parameters. Considering the smartphone user's skills, the protocol and mobile application transparently handles the security credentials. In order to validate the security of designed protocol, automated security protocol verification tool Scyther is used. We found that no attack tree was generated by Scyther and the results obtained against our security claims, it is verified that our security protocol resists against Man-in-the-Middle, replay and attacks on confidentiality of user's credentials.

6.1 Future Work

We have designed architecture for SPACS and evaluated the secrecy and reliability of our designed Authentication and Authorization Protocol using automated security protocol analyzer Scyther [2]. To reduce average entry time and avoid human typing mistakes occurred during keying pass code at electronic door key panels, the pass code received by SPACS application for smartphone can be replaced by encrypted OTP (One Time Passcode) QRCode and electronic door key panels with cameras to capture them. This approach can be beneficial in congested or crowded places like bus stands or subways and will be more users friendly and efficient. Furthermore, more formal verification of our designed Authentication and Authorization Protocol for SPACS can be done using AVISPA [22] and ProVerif [23] for comparative study.

Bibliography

[1] Lawrence O'Gorman, "Comparing Passwords, Tokens, and Biometrics for User Authentication", IEEE journal and Magazines, Vol. 91, No. 12, Dec. 2003, pp. 2019-2040.

[2] Scyther: http://people.inf.ethz.ch/cremersc/scyther/

[3] Y.W. Kao, G.H. Luo, H.T Lin, Y.K. Huang and S.M. Yuan, "Physical access control based on QR code", Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), IEEE 2011, pp. 285-288.

[4] R. Marx, H.S. Fhom, D. Scheuermann, K.M. Bayarou and A. Perez, "Increasing security and privacy in user-centric Identity Management: The IdM card approach", P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), IEEE 2010, pp. 459-464.

[5] Lopez, G., Canovas, O., Gomez-Skarmeta, A.F. and Girao, J., "A swift take on Identity Management", IEEE Computer, 2009, pp. 58-65.

[6] Dmitrienko, A., Sadeghi, A.R., Tamrakar, S. and Wachsmann, C., "SmartTokens: delegable access control with NFC-Enabled smartphones", Springer, 2012. Trust and Trustworthy Computing, pp. 219-238.

[7] K. Lu, A. Ali, K. Sachdeva and K. Krishna, "A Pragmatic Online Authentication Framework using Smart Card", SERVICE COMPUTATION 2011, The Third International Conferences on Advanced Service Computing. pp. 84-91.

[8] Daradimos, I. and Papadopoulos, K. and Stavrakas, I. and Kaitsa, M. and Kontogiannis, T. and Triantis, D., "A Physical Access Control System that utilizes existing networking and computer infrastructure", EUROCON 2007, The International Conference on Computer as a Tool, pp. 501-504.

[9] "Access control with mobile phones: the future with Near Field Communications". [Online] http://www.sourcesecurity.com/news/articles/co-3108-ga.5735.html accessed on 12, Sep, 2012.

[10] "A Recommendation for the Use of PIV Credentials in Physical Access Control Systems (PACS)". [Online] http://csrc.nist.gov/publications/nistpubs/800-116/SP800-116.pdf accessed on 15, Sep, 2012.

[11] Di pietro, R.; Gianluigi Me; Strangio, M.A., “A two-factor mobile authentication scheme for secure transactions”, Mobile Business, ICMB 2005 International Conference, pp.28-34, July 2005.

[12] Isobe, Y.; Yoichi Seto; Kataoka M., “Development of Personal Authentication System using fingerprint with Digital Signature technologies”, System Sciences, Proceedings of the 34th Annual Hawaii International Conference, pp.9, Jan 2001.

[13] Hassinen, M.; Hypponen, K., “Strong Mobile Authentication”, Wireless Communication Systems, 2nd International Symposium, pp. 96-100, Sept 2005.

[14] Leong, A. and Fong, S. and Yan, Z., “A Logical Model for Detecting Irregular Actions in Physical Access Environment”, Database and Expert Systems Applications, 18th International Workshop DEXA IEEE 2007, pp. 560-564.

[15] Kriplean, T. and Welbourne, E. and Khoussainova, N. and Rastogi, V. and Balazinska, M. and Borriello, C. and Kohno, T. and Suciu, D., “Physical access control for captured RFID data”, IEEE, 2007.Pervasive Computing, pp. 48-55.

[16] Chague, S.; Droit, B.; Yanushkevich, S.N.; Shmerko, V.P.; Stoica, A., “Biometric-Based Decision Support Assistance in Physical Access Control System”, Bio-inspired Learning and Intelligent Systems for Security, BLISS ’08, pp.11-16, Aug 2008.

[17] D. Song, “Athena: A new efficient automatic checker for security protocol analysis”, In PCSFW: Proceedings of the Computer Security Foundations Workshop, IEEE Computer Society Press, pp.192, 1999.

[18] C. Cremers and S. Mauw, “Operational semantics of security protocols”, In S. Leue and T. Systa, editors, Scenarios: Models, Transformations and Tools Workshop 2003, revised Selected Papers, vol. 3466 of LNCS, Springer, pp. 66-89 2005.

[19] D. Dolev and A. C. Yao, “On the security of public-key protocols”, IEEE Transactions on Information Theory, 2(29):198–208, 1983.

[20] Jorstad, I., and T. Jonvik, "Strong authentication with mobile phone as security token", In Mobile Adhoc and Sensor Systems, 2009, IEEE 6th International Conference on, pp. 777-782, IEEE, 2009.

[21] Suoranta, Sanna, André Andrade, and Tuomas Aura, "Strong Authentication with Mobile Phone", Information Security (2012): pp. 70-85.

[22] AVISPA: http://www.avispa-project.org

[23] ProVerif: http://prosecco.gforge.inria.fr/personal/bblanche/proverif/