A MOBILE ROLE BASED ACCESS CONTROL SYSTEM USING IDENTITY BASED ENCRYPTION WITH NON-INTERACTIVE ZERO KNOWLEDGE PROOF OF AUTHENTICATION

by

Ambica Pawan Khandavilli

Submitted in partial fulfilment of the requirements
for the degree of Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
March 2012

DALHOUSIE UNIVERSITY

FACULTY OF COMPUTER SCIENCE

The undersigned hereby certify that they have read and recommend to the Faculty of Graduate Studies for acceptance a thesis entitled "A MOBILE ROLE BASED ACCESS CONTROL SYSTEM USING IDENTITY BASED ENCRYPTION WITH NON-INTERACTIVE ZERO KNOWLEDGE PROOF OF AUTHENTICATION" by Ambica Pawan Khandavilli in partial fulfilment of the requirements for the degree of Master of Computer Science.

Dated: March 29th 2012

Supervisor: ________________________________

Readers: ________________________________

____

DALHOUSIE UNIVERSITY

DATE: March 29th 2012

AUTHOR: Ambica Pawan Khandavilli

TITLE: A MOBILE ROLE BASED ACCESS CONTROL SYSTEM USING IDENTITY BASED ENCRYPTION WITH NON-INTERACTIVE ZERO KNOWLEDGE PROOF OF AUTHENTICATION

DEPARTMENT OR SCHOOL: Faculty of Computer Science

DEGREE: MCSC CONVOCATION: May YEAR: 2012

Permission is herewith granted to Dalhousie University to circulate and to have copied for non-commercial purposes, at its discretion, the above title upon the request of individuals or institutions. I understand that my thesis will be electronically available to the public.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

The author attests that permission has been obtained for the use of any copyrighted material appearing in the thesis (other than the brief excerpts requiring only proper acknowledgement in scholarly writing), and that all such use is clearly acknowledged.

____

Signature of Author

TABLE OF CONTENTS

LIST OF FIGURES vi

LIST OF TABLES vii

ABSTRACT viii

LIST OF ABBREVIATIONS USED ix

CHAPTER 1: INTRODUCTION 1

CHAPTER 2: BACKGROUND AND LITERATURE SURVEY 4

2.1 NEAR FIELD COMMUNICATION 4

2.1.1 NFC communication modes *4*

2.1.2 NFC modes of operation *4*

2.1.3 NFC Data Exchange format *6*

2.1.4 Android NFC *8*

2.1.5 Common attacks on NFC phones *9*

2.1.6 Future of NFC *9*

2.2 ROLE BASED ACCESS CONTROL AND SPATIAL EXTENSIONS OF RBAC 10

2.2.1 Enforcing Spatial Constrains on mobile RBAC systems *12*

2.3 ZERO KNOWLEDGE PROOF 17

2.3.1 Zero Knowledge proof as a Digital Signature *19*

2.4 IDENTITY BASED ENCRYPTION 20

2.4.1 Boneh Franklin IBE *23*

2.4.2 Identity Based Signatures and Access Control *26*

CHAPTER 3: PROPOSED FRAMEWORK **28**

3.1 SYSTEM ARCHITECTURE 28

3.1.1 Setup Phase *32*

3.1.1.1 Setting up the Location Device 32

3.1.1.2 Setting up the User 32

3.1.2 Operation phase *33*

3.2 PROTOCOL IN OVERVIEW 33

3.3 DETAILED PROTOCOL STEPS 34

3.3.1 Session Initiation *34*

3.3.2 Session establishment *35*

3.3.3 Session Key Generation *37*

3.3.4 Secure exchange of data *38*

3.3.5 Session re-establishment *38*

3.3.6 Session termination *38*

CHAPTER 4: IMPLEMENTATION **39**

4.1 PAIRING BASED CRYPTOGRAPHY IMPLEMENTATIONS 39

4.2 PEER-TO-PEER COMMUNICATION WITH AN NFC DEVICE 40

4.3 TEST RUNS 41

4.3.1 Normal run of the system *41*

4.3.2 ZKP generation on an android device *44*

CHAPTER 5: SECURITY ANALYSIS **46**

5.1 ROGUE NODE 47

5.2 REPLAY ATTACK 47

5.3 COLLUSION 49
5.4 REFLECTION ATTACK 49
5.5 DENIAL OF SERVICE 49
5.6 THREATS DUE TO NFC COMMUNICATION 50

CHAPTER 6: CONCLUSION AND FUTURE WORK 53

BIBLIOGRAPHY 55

LIST OF FIGURES

Figure 1. NFC modes of operation..5
Figure 2. NFC peer to peer mode..6
Figure 3. NDEF record format..7
Figure 4. NDEF record types..7
Figure 5. Android NFC Intents..8
Figure 6. General architecture of access control..10
Figure 7. RBAC model..11
Figure 8. GEO-RBAC..11
Figure 9. Communication channels within a spatially-aware RBAC..14
Figure 10. Privacy Preserving RBAC model..16
Figure 11. Ali Baba cave problem..17
Figure 12: Generation of keys in traditional public key system..21
Figure 13: Generation of keys in an Identity Based Encryption system..22
Figure 14: System architecture and Communication..28
Figure 12: Normal run of the system..42
Figure 13: ZKP proof packet sent from user to PKG..44
Figure 14: ZKP proof packet sent by a rogue node..48
Figure 15: ZKP and IBE tests on an android device..45

LIST OF TABLES

Table 1: Time taken for executing steps of the protocol.. 52

Abstract

Controlled access to confidential information and resources is a critical element in security systems. Role based access control (RBAC) has gained widespread usage in modern enterprise systems. Extensions have been proposed to RBAC for incorporating spatial constraints into such systems. Several solutions have been proposed for such models and much research has now been directed towards enforcing system policies.

The thesis proposes a security framework for RBAC systems with spatial constraints based on identity based encryption. Integration of identity based encryption and with zero knowledge proof is proposed to provide authentication and information security. We also show how Near Field Communication can be used to establish the integrity of a user's proof of location. We discuss the design choices made in the protocol and explain the protocol implementation. Simulation results in Java validate our model. Furthermore, security analysis has been done to show how our framework protects against well-known attacks.

List of Abbreviations Used

IBE – Identity Based Encryption

RBAC – Role Based Access Control

ZKP – Zero Knowledge Proof

NFC – Near Field Communication

PKG – Private key Generator

R_SM – Resource manager

R_OM – Role Manager

LD – Location Device

NDEF – NFC Data exchange format

Chapter 1: Introduction

The rapid pace of development in wireless technologies and mobile computing has made ubiquitous computing a reality. Enterprise systems are now moving away from stationary (fixed) workstations to mobile workstation such as mobile phones and tablet computers. The motivation for this shift is to increase the availability of information to users. Some of the applications that benefit from the increase in mobility and connectivity may require access to sensitive data. In some highly secure settings such as in military, healthcare and government agencies, confidential data might be restricted to a room or a set of rooms i.e. spatial restrictions are imposed on data [1].

Consider a situation where an employee may have access to confidential information in his/her office but not outside the office. Traditionally, security of such kind is assured by using a fixed workstation and binding the authentication parameters to the workstation. The traditional fixed workstation provides a secure solution, but the deployment and of administration of such a system is very tedious. Mobile devices such as tablets are becoming the norm of industries such as Healthcare; doctors are using these devices to get access to patient records, medical images and other information. With the increase in the use of mobile devices, new challenges arise for users requiring access to secure information from different settings. An example in the health care scenario is the one of a “bored but curious employee”; such an employee may access the records of a high profile client even if she/he does not have a reason to do so [2]. Enterprises also want to keep their innovations and secrets safe from threats both internal and external threats by restricting the access to sensitive information at a certain location. In such a setting, location aware authentication becomes plays an important role.

The requirements for such a system can be manyfold.

- Verification of the user’s location
- Authentication of the user’s identity
- Access control
- Distribution of resources in a secure manner

Verification of the user's location is validating the users claim to a location. Authentication of user's identity is verifying that the user requesting the access to the request is actually the user of the system. Access control is to verify that a user has the permission to access a specific resource

Role based access control (RBAC) with spatial extensions is a good foundation to model such a system. RBAC is a standard for authorization and has gained wide deployment across a variety of organizations. The main advantage of RBAC is a simplified mechanism for authorizing access to confidential/sensitive records on job functions rather than the identities of a user. GEO-RBAC model is one of the first extensions to incorporate spatial extension to RBAC. Along with authorization, authentication of the user's identity plays a very important role in the system. Research is needed to define protocols enforcing these policies in a secure manner.

In this research we propose a framework for mobile role based access control using an Identity Based Cryptosystem with Non-interactive zero knowledge proof of authentication. Our framework is modeled around the system proposed in the paper [2]

To the best of our knowledge, this is the first proposal to integrate to concepts of ZKP and IBE to incorporate authentication into IBE without infrastructure overhead.

We take into account the security requirements of the framework with a special emphasis on the authentication of the user and the secure distribution of data. To achieve this goal we propose an integration of ZKP into an Identity Based Encryption system. Zero-knowledge proof (ZKP) is a protocol which allows a prover to convince a verifier, knowledge of secret information without disclosing any information.

IBE is an emerging technology based on Pairing Based Cryptography. The main advantage of IBE based system over a Public Key Cryptography (PKC) is that in regular PKC there is no correlation between an individual's ID and their public key. Hence a trusted third party is needed to establish this correlation. However, in the case of IBE, the main idea is that known information that uniquely identifies the users (such as email address, IP, etc) can be used to derive its public key. As a result, keys are self-authenticated and certificate by trusted third parties is thus unnecessary.

Efficient implementations of both the IBE and ZKP systems are based on the Elliptic Curves. Common infrastructures for implementations make IBE and ZKP perfect candidates for integration. ZKP provides the authentication and IBE system takes care of the key management and information security.

The rest of the thesis is organized as follows.

The second chapter gives the background details necessary to understand the rest of the thesis. It gives a brief introduction of Near Field Communication; Role based access control, Identity Based Encryption and Zero Knowledge proof of authentication. It also has a discussion about the relevant recent literature, which is close to the work in this thesis.

In the third chapter we propose our framework and explain in detail the protocols involved in the framework.

The next Chapter 4 discusses about the implementation of the framework we provide screenshots of the simulation.

Chapter 5 provides a discussion on the security counter measures and an informal security analysis of the protocol. We end the report with the conclusion and the bibliography.

Chapter 2: Background and Literature survey

2.1 Near Field Communication

Near Field Communication (NFC) is a bidirectional proximity coupling technology based on the ISO14443 and the FeliCa RFID standards. NFC operates in the 13.56 Mhz spectrum and supports data transfer rates of 106, 216, and 424 kbit/s. The communication range of NFC is typically between 2 cm and 4 cm.

NFC has a different set of features other than RFID. The two main new features added in the standards are peer-to-peer communication between two active NFC devices (NFCIP) and the emulation of a passive proximity RFID tag.

NFC has garnered a lot of interest in recent years. Several leading mobile technology companies like Google, Nokia and RIM are trying to leverage NFC technology by launching NFC enabled phones and also launching mobile wallet applications (Google wallet) and several other solutions. NFC technology has developed and matured over the years and is not being tested in several pilot projects around the globe.

2.1.1 NFC communication modes

Active mode: In this mode, the target and the initiator devices have power supplies and can communicate with one another by alternate signal transmission.

Passive mode: In this mode, the initiator device generates radio signals and the target device is powered by this electromagnetic field. The target device responds to the initiator by modulating the existing electromagnetic field.

2.1.2 NFC modes of operation

NFC devices can operate in three different modes based on the ISO/IEC 18092, NFC IP-1 and ISO/IEC 14443 contactless smart card standards. Figure 1 shows a graphical representation of the modes of operation.

Read and write mode

In this mode an NFC enabled device can read or write data to any of the supported tag types in a standard NFC data exchange format.

Peer-to-Peer mode

In this mode a NFC enabled device can communicate with another NFC device using ISO/IEC 18092 standard. The data is exchanged in the NFC data exchange format.

Card emulation mode

In this mode of operation, an NFC enabled device can act as a tag for readers.

For a comprehensive and exhaustive survey of the NFC standards, the introduction to NFC document provided by Nokia is an excellent reading [3].

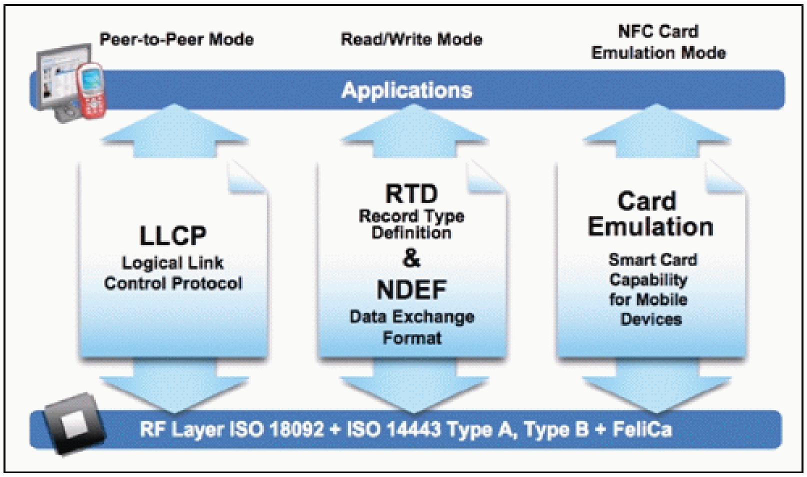


Figure 1. NFC modes of operation

Several use cases can be derived from the different operational modes. The operational mode that has become quite popular recently is the peer-to-peer communication mode. The peer-to-peer mode which is sown in Figure 2 abstracts the different standards and technologies to give a very clear implementation strategy for the NFC developer. Protocols such as the logical link Control Protocol (LLCP), Simple NDEF Push Protocol

(SNEP) and NDEF Push Protocol have been established to facilitate the use of this mode of operation.

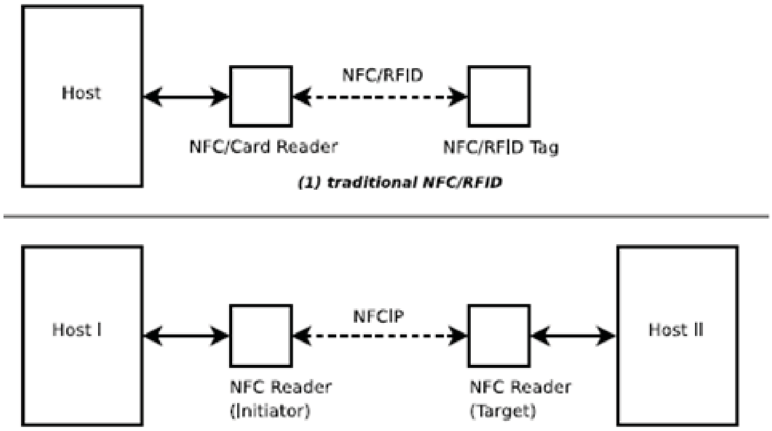


Figure 2. NFC peer to peer mode

2.1.3 NFC Data Exchange format

Figure 4 shows the format of an NFC Data Exchange Format message (NDEF). NDEF format standardizes how to store data on a smartcard that is compatible with one of the NFC Forum tags. NDEF is a lightweight, binary message format that can be used to encapsulate one or more application-defined payloads of arbitrary type and size into a single message. An NDEF message is composed of one or more NDEF records. There can be multiple records in a NDEF message. Basically NDEF message is array of NDEF records. How many records we can encapsulate in a NDEF message that depends on our application and the tag type.

NDEF can be classified into different types; the different types are shown in Figure 4

These tags can be used to store for example bookmarks, business cards, alarm clock settings, Smart Posters information, Call or SMS Requests and several other objects.

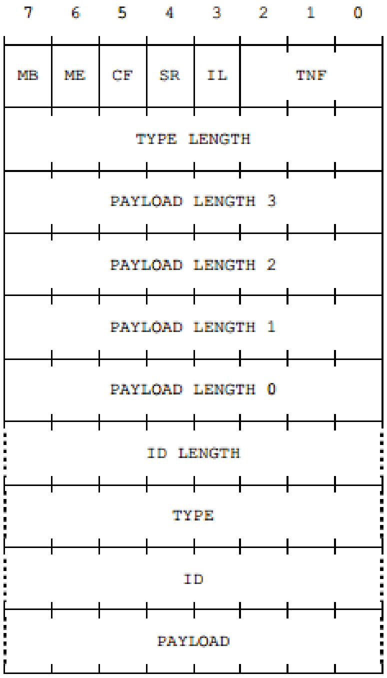


Figure 3. NDEF record format

Type Name Format	Value
Empty	0x00
NFC Forum well-known type [NFC RTD]	0x01
Media-type as defined in RFC 2046 [RFC 2046]	0x02
Absolute URI as defined in RFC 3986 [RFC 3986]	0x03
NFC Forum external type [NFC RTD]	0x04
Unknown	0x05
Unchanged (see section 2.3.3)	0x06
Reserved	0x07

Figure 4. NDEF record types

2.1.4 Android NFC

The Android mobile operating system has one of the most robust NFC technology stack. Google who is the chief contributor to the Android Open Source project collaborated with NXP, the pioneer and major contributor of NFC to develop API's that abstract the different NFC protocols. NFC was first introduced in the Gingerbread[1] version of the operating system and supported only tag reading and writing in the beginning. Android continued to roll in updates to its NFC stack and it now supports reading and writing tags and peer-to-peer communication to other NFC FORUM enabled devices. Android Beam, which abstracts the peer-to-peer communication between android-based NFC smart phones, is a new feature that has been added to the Ice-Cream sandwich iteration of the Operating System and has become quite popular.

In Figure 5, we can see how the Android Operating system handles when it reads a tag or receives NDEF formatted data via a peer-to-peer communication. The OS packages the data that it received neatly into an NDEF format and hands it over to the intent that can handle the type of data (intent is an abstract description of an action to be performed). For example, a tag that holds URL formatted NDEF will launch the browser application.

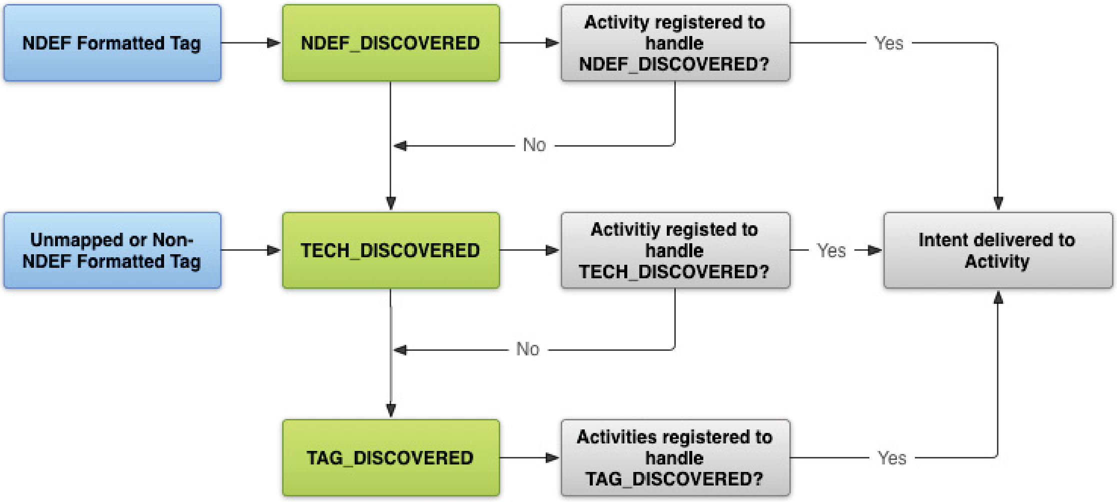

NDEF Formatted Tag
NDEF_DISCOVERED
Activity registered to handle NDEF_DISCOVERED?
Yes
No
Unmapped or Non-NDEF Formatted Tag
TECH_DISCOVERED
Activitiy registed to handle TECH_DISCOVERED?
Yes
Intent delivered to Activity
No
TAG_DISCOVERED
Activities registered to handle TAG_DISCOVERED?
Yes

Figure 5. Android NFC Intents

[1] Android Gingerbread is the 5th iteration of the Operation System rolled out by Google.

2.1.5 Common attacks on NFC phones

Researchers investigated a series of attacks on NFC enabled phones [4]. The authors implement a series of attacks by using malicious NDEF messages that can initiate a Bluetooth connection or trick the user into installing malicious software onto the phone. The attacks listed in the paper are not threats directly related to NFC, but threats that can be initiated by using malicious NDEF records and by exploiting the vulnerabilities of the underlying operating system. We discuss about how we implement the application to minimize these threats in the security analysis discussed in Chapter 5.

NFC technology has been deployed in many use cases. Payments, ticketing, access control form the major application scenarios along with several other innovative applications. One of the major advantages of NFC that we make use of in this framework is the ability to bind a user to a certain location and time securely with the proper system and implementation in place.

2.1.6 Future of NFC

NFC is facing the same roadblocks as RFID did for real life deployment i.e. security. NFC Forum has released a set of standards for secure NFC communication, protocols such as NFC-SEC provide cryptographic standards for NFC using Elliptic curve Diffie Hellman and AES. Active research is also being done to prevent attacks on NFC communication using malformed NDEF messages. The NFC research lab at the Upper Austria university of Applied sciences, Hagenberg, Austria has been actively investigating on NFC security and published papers such as [5] as a recommendation to the standards. Very recently, the author of this thesis has co-founded a company called Alfred NFC [6]. The company is working on a product that enables the users of modern Smartphone's to tie their identity to a specific location in a secure manner.

The future of NFC in consumer products is looking with the increase of researchers in NFC protocol and communication security, several pilot projects being conducted over the globe and the push given by mobile phone manufactures and operating systems by including NFC in their products.

2.2 Role based access control and spatial extensions of RBAC

One of the key components of any extensive security solution is access control. RBAC is one of the most widely adopted access control approaches.

Access control determines whether a user of the system has access to a protected resource or service under a given circumstance. When a user of the system tries to access a resource, the access control service checks the rights of the subject against a set of authorizations. Authorizations encode the access control policies of the organization. Figure 6 shows the general architecture of access control.

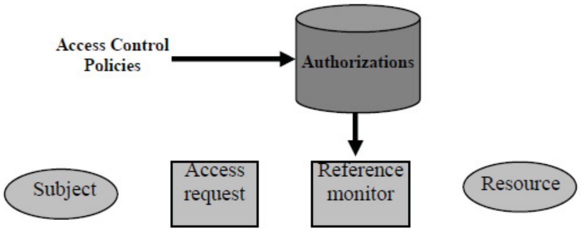


Figure 6. General architecture of access control

The subject constructs an access request. The access request generally encapsulates the identity of the user and the resource that the subject wants to access. The reference monitor also generally needs to know which action the subject wants to perform on the resource (common actions include read, write and edit). The reference monitor intercepts the access request and runs it along the authorizations. One of hurdles in deploying a simple access control is the high cost of administration and maintenance of access control lists or similar access control data structures. To reduce such costs notion of a simple access control has modified and extended, one notable extension is Role Based Access control where the access to resources is not assigned to users directly but to entities which are referred to as roles. Instead of saying that Doctor John has access to records of patients, John is assigned to the role of doctor and the doctor role has access to records of patients.

Roles are assigned to job descriptions, because there are fewer roles than users, considerable savings in administration can be achieved.

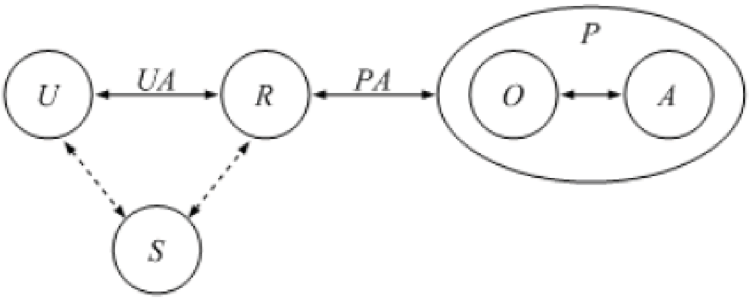


Figure 7. RBAC model

RBAC is modeled on a set of users U, Permissions P and a set of roles R. Users are associated with roles using a user role assignment relation *UA*, where the relation is a set of pairs of the form *(u,r)*, meaning that user *u* is assigned to role *R*. Permissions are similarly associated with roles using a Permission-role assignment relation *PA*. The Users interact with the RBAC system by authenticating themselves and activating a session from a set *S*. RBAC also has the notion of role hierarchy where the role high up the hierarchy will inherit the permission assigned to lower roles.

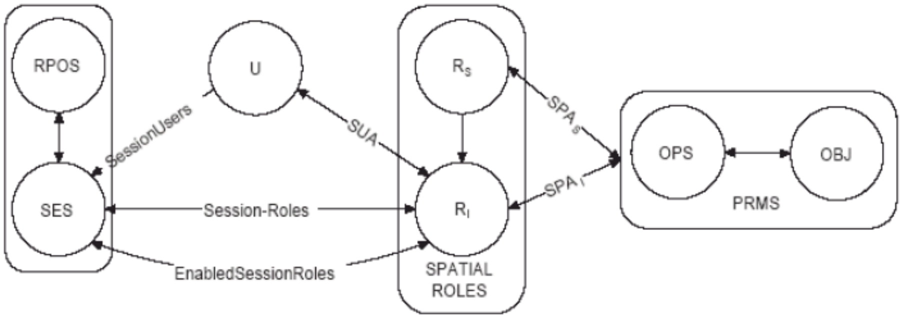


Figure 8. GEO-RBAC

GEO-RBAC: Another important extension of the RBAC model is GEO-RABC. This model incorporates spatial awareness to RBAC. In GEO-RBAC roles are associated with spatial extents; such extents represent spatial regions, in the reference space, that are of interest to the application domain. Locations in GEO-RBAC model are logical concepts, such as “in the operation theater” or “at the laboratory” rather than the actual physical coordinates, like GPS. The locations are mapped to physical locations [7].

In the figure 8, Ri and Rs represent the sets of role instances and role schemas respectively; RP OS is the set of real positions; and U, SES, OPS and OBJ are the sets representing users, sessions, operations and objects respectively

Returning to the example of the health care system, access to a patient's record could be restricted to the spatial role *Doctor-in-Ward*. If the same user logs in from his home office, he would be assigned the spatial role *Doctor-at-Home*. In either case, the credentials and authentication process are identical. The only difference is the factor of the location.

The GEO-RBAC also proposes having a distinction between a role schema and a role instance. A role schema specifies a role name, for example a Doctor in ward while a feature type is for example a Hospital. A role instance is obtained from the role schema by instantiating the feature type to a specific feature. An example of role instance of the role schema <Doctor-in-ward, Hospital> is <Doctor-in-ward, Queen Elizabeth>. Another important feature in the GEO-RBAC scheme is the difference between role enabling and role activation. A role is enabled when the user is in the spatial constraints but is activated only when the user chooses to activate the role during the current session.

2.2.1 Enforcing Spatial Constrains on mobile RBAC systems

Enforcing spatial constraints to mobile RBAC is a very interesting problem. In [2] the authors use NFC to solve this problem. They proposed a set of protocols to enforce the location policies with a GEO-RBAC backbone. GEO-RBAC associates spatial extents to traditional roles.

XACML

Authors use XACML for defining the access policies. XACML stands for eXtensible Access Control Markup Language. The standard defines a declarative access control policy language implemented in XML and a processing model describing how to evaluate authorization requests according to the rules defined in policies.

Common terminology

Policy Administration Point (PAP) – Point that manages policy

Policy Decision Point (PDP) – Point that evaluates and issues authorization decisions.

PEP – Policy Enforcement Point – Point which intercepts user's access request to a resource and enforces PDP's decision

PIP – Policy Information Point – Point which can provide external information to a PDP, such as LDAP attributes information

Architecture and Protocols

Architecture proposed in the system is a ticket granting architecture where the user submits an access request and the access granting authority. Four principals form the core of the architecture.

Users: principal making the request. Generally refers to the device used for the request

Location Device (LD): Physical device storing the location information. Assumed to be pre-installed in strategic locations and are stationary.

Resource Manager (R_SM): Responsible for the requested resource.

Role Manager (R_OM): maps a user to a set of roles. Responsible for evaluating the location claim and credentials presented. It returns a list of active roles to the R_SM, which evaluates the request in relation to defined policy.

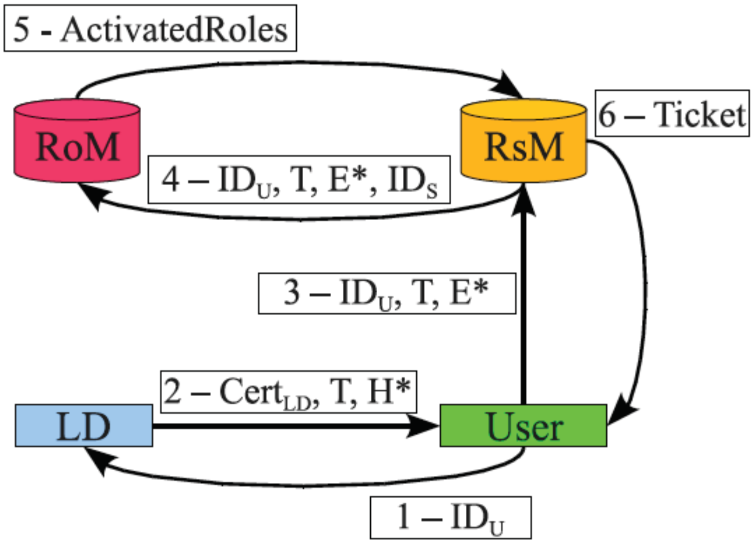


Figure 9. Communication channels within a spatially-aware RBAC [2]

Initial request protocol

In the initial request phase, the user requests access to a resource.

User device sends its identifier to the Location device, which binds the proof of location to the requesting device. Using the identifier received, the LD computes a hash of the identifier with the current timestamp and sends the Hash, timestamp and a certificate to the user.

The user then calculates an encrypted package which contains the requested role, the proof of location hash sent by the Location Device, the user's password and the two certificates (certificate of the Location Device and the certificate of the user) which were signed by the Role Manager. The package is encrypted with a symmetric key.

The resource manager forwards the encrypted packet and the Identifier ID_U along with the session identifier ID_S to the role manager. The role manager after receiving the forwarded request populates a list of activated roles for the request and sends the information back to the resource manager. The resource manager receives the list of activated roles, applies the access control policies and grants a ticket to the user for the access of the resources.

The main objectives of the authors was to propose a model for enforcing spatial constraints to mobile RBAC, they use NFC technology in the peer to peer mode to bind a user to a certain location by sending an identifier from the user device to the Location Device.

The paper does a good job in addressing the problem of enforcing spatial restrictions on mobile networks, and defines a set of protocols for enforcing the whole scheme. The authors assume that the system takes care of all the cryptographic operations necessary. With the use of certificates to authenticate the user and the location device, there is necessity for the role manager to refresh the certificates and the user password in regular intervals.

In this research we extend the work done by the authors to propose a cryptographic framework for mobile RBAC with spatial constraints. We use some of the architectural features discussed in the paper and tailor a cryptosystem that can provide security, privacy, convenience and ease of deployment and management.

Along with proposing a solution for enforcing spatial restriction to mobile RBAC, there has been further research done in the area. In [8] the authors have suggested using hardware identifiers for principals such as Location Devices (refer the previous section) which bind the users to a location. They propose generating the hardware identifiers using the concept of physically unclonable functions (PUF). PUF’s have been used in computer security mainly in cryptographic solutions.

The fundamental idea of PUF’s is to create a random pairing between a challenge c and a response r, the random behavior is based on that fact that no two instances of a hardware design can be identical. The authors use PUF’s derived from ring oscillators for their system. The idea of using PUF’s to bind a user access request or the identity of a certain user to a location is a very innovative and can avoid a lot of computations on the software level.

Another important work that has been done in the area is the privacy preserving enforcement of spatially aware RBAC.

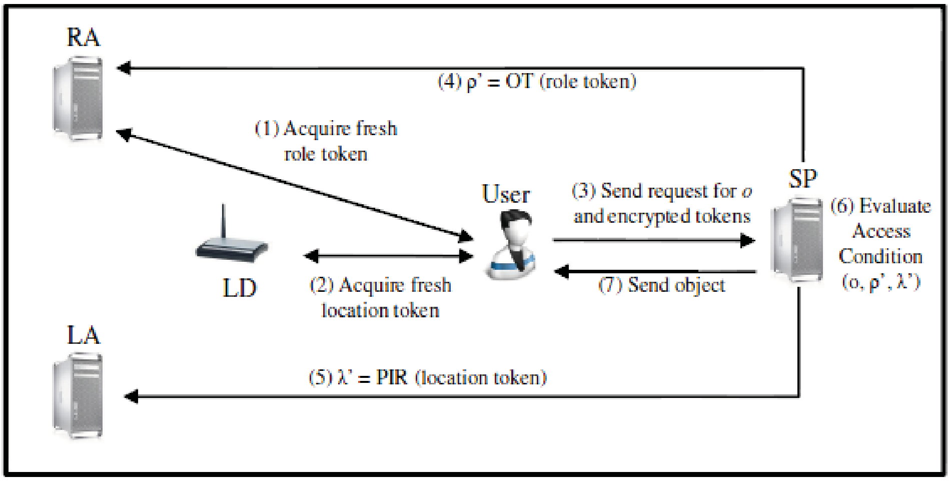

RA
(4) ρ' = OT (role token)
(1) Acquire fresh role token
User
(3) Send request for o and encrypted tokens
SP
(6) Evaluate Access Condition (o, ρ', λ')
LD
(2) Acquire fresh location token
(7) Send object
LA
(5) λ' = PIR (location token)

Figure 10. Privacy Preserving RBAC model [1]

The model is very similar to the one discussed in [2]. The main objective of the architecture is to allow the policy enforcer to enforce its policies correctly, while preventing the disclosure of the user's identity, role or location. They use a combination of cryptographic techniques with a separation of duties between several components. There is only one trusted component in the architecture which setups all the cryptographic parameters, it does not participate in the regular functioning of the protocol. The backend principles involved in the protocol are the role authority (RA), Service provider (SP), Location authority (LA) and an identity authority (IA). IA is the trusted third party that establishes all the cryptographic secrets of the system.

A client wanting to make a request contacts the relevant SP that controls the resource, providing a pair of tokens with the request. SP cannot verify the tokens, but gets the RA and LA to authenticate the tokens. SP then initiates two protocols for oblivious transfer and Private Information Retrieval with the LA to get additional data used for policy evaluation. SP has no knowledge about the mapping from encoded policy to the original policy. SP can determine whether the policies were satisfied but can never know the original unencoded policy. Thus the SP does not know the users role or location.

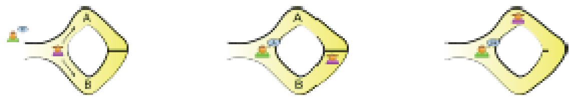
The system in this paper was proposed taking into account that an administrator can be malicious. The primary goal of the framework is to protect the users privacy; they achieve this by distributing the functionalities to different principles.

2.3 Zero knowledge proof

In simple words Zero knowledge proofs (ZKP) are proofs that show a statement to be valid without revealing anything except the veracity of the statement to be proven. With the rise in ubiquitous computing, we are using mobile phones for daily tasks. There is a need to preserve the privacy and not reveal information that can be abused by hackers. Zero knowledge proof can be used when someone needs to prove the possession of critical data without revealing the actual data. We will give a small background about the Zero Knowledge Proof protocols in this section.

Zero knowledge proofs on a high level can be of two instances. Interactive zero knowledge proofs and non-interactive proof.

In interactive proof systems, multiple messages are exchanged between the prover and the verifier in the form of challenge and responses. In the non-interactive proof system, only one message is exchanged.



(a) 1st Step. (b) 2nd Step. (c) 3rd Step.

Figure 11. Ali Baba cave problem

In both the systems the objective of the prover is to convince the verifier about the truth of an assertion. The verifier can the either reject or accept the proof.

The classic example for explaining ZKP is the Ali Baba cave problem.

1. Victor waits outside the cave as Peggy goes in

2. Peggy randomly takes either path A or B inside the cave

3. Victor enters the cave and shouts the name of the path he wants her to use to return either A or B, chosen at random

4. Peggy does that using the secret word if needed to open the magic door

5. The above steps are repeated n times until Victor is convinced that Peggy knows the secret word

If Peggy does not know the secret word, since Victor chooses path A or B at random. Peggy has a 0.5 chance of cheating at one round. If the steps are repeated for many rounds, Peggy's chance of successfully guessing all of Victor's requests is very low.

A zero-knowledge proof is said to obey the properties of completeness and soundness.

A proof is complete, if given an honest prover and an honest verifier, the protocol succeeds with overwhelming probability and sound if the probability of a dishonest prover to complete the proof successfully is negligible [4]. Additionally, a protocol which consists of a proof of knowledge must have the zero-knowledge property: there exists an expected polynomial-time algorithm which can produce, upon input of the assertions to be proven – but without interacting with the real prover, transcripts indistinguishable from those resulting from interaction with the real prover.

Zero knowledge proofs have seen a wide variety of implementation based on the Discrete Logarithmic problem. In this section we will discuss a few Zero knowledge proof protocols based on the Elliptic curve Discrete Logarithmic Problem (ECDLP)

The ECDLP can be defined as follows. Given an elliptic curve E over a field F of order n, a generator G € E/Fn and a point B € E/Fn, it is computationally hard to find x such that B = x.G

Schnorr's protocol is a simple and frequently used proof of knowledge. The protocol is defined for a cyclic group Gq of order q with generator G.

In order to prove knowledge of $x = log\, g_y$, the prover interacts with the verifier as follows:

In the first round the prover commits herself to randomness r; therefore the first message $t = g^r$ is also called *commitment*.

The verifier replies with a *challenge c* chosen at random.

After receiving c, the prover sends the third and last message (the *response*) $s = r + cx$.

The verifier accepts, if $g^s = ty^c$.

Elliptic curve variant of Schnorrs protocol

Prover computes random r and computes the point $A=r.G$

Prover sends the point A to Verifier

Verifier computes the random $c = HASH\ (G, B, A)$ and sends c to Prover

Prover computes $m=r+c.x$ and sends m to verifier

Verifier checks that $P = m.G\text{-}c.B = (r+c.x).G - c.B = r.G + c.x.G - c.x.G = r.G = A$

2.3.1 Zero Knowledge proof as a Digital Signature

The elliptic curve variant of the Schorrs protocol is executed for one round. Verifiers coin flip (repeating the proof of knowledge till the verifier is convinced with high probability) that the prover is honest. Adi Shamir et al. [9] in their paper propose the use of a hash function and the agreement on an initial message m can remove the interactivity from protocols such as the schorrs protocol. In addition to the parameters used in the previous protocol, a new parameter is used. The point P € E/F_n represents the pre-shared message that the prover wants to send to the verifier.

Prover

Generates random r and computes the point $A = r.G$

Prover computes $c = HASH\ (x.P,r.P,r.G)$

Prover computes $s = r+c.x(mod\ n)$

Prover sends to the verifier the message: "$s||x.P||r.P||r.G$"

Verifier

Computes $c = HASH\ (x.P,r.P,r.G)$

Verifier checks that $s.G = (r+c.x).G = r.G + c.x.G = r.G + c.B = A + c.B$

Verifier checks that $s.P = (r+c.x).P = r.P + c.xP$

In this protocol, the prover simulates both the prover and the verifier with the use of a hash function. The prover sends only one message and the verifier either accepts or rejects. The prover generates a random number as in the previous protocols but the verifiers random choices are simulated by hashing the input along with a value calculated from the Provers choice of r. The Verifiers random choice depends on the provers random choices and it is hard to fake the outcome. The value of c is a challenge for the prover as it is computed from a hash function. In order to cheat, the prover who does not know x would try to find s satisfying $s.G = r.G + c.x.G$ which is an instance of the discrete logarithmic problem. Enumerating random r values would be hard as the hacker would have to find a matching value of c.

2.4 Identity Based Encryption

IBE is a public-key encryption technology that allows users to calculate a public key from an arbitrary string. The ability to calculate keys as needed gives IBE system different properties than public key encryption. Although there are probably few situations in which it is impossible to solve any problem with traditional public-key technologies that can be solved with IBE, the solutions that use IBE may be much simpler to implement and much less expensive to support than alternatives.

In [10] the author gives some interesting advantages of IBE over the traditional public key systems. In implementations of traditional public-key system that uses digital certificates to manage public keys, either a user, or an agent working on behalf of the user generates a public-private key pair randomly, in which the public key contains all the parameters needed for using it in cryptographic calculations. Random generation of keys is not strictly required by the public-key algorithms that are used in such systems, but it is used by existing standards that define the use of such algorithms. After the public key is created, the key along with the identity of the owner of the key is digitally signed by a certificate authority to create a digital certificate that is then used to transport and manage the key. In some applications, it may be necessary to recover private keys that are lost or unavailable. In such cases there is need of a key recovery agent.

In a traditional Public key system, the identity of the user is carefully verified before a CA is issued to him, which is a usually an expensive process. The process of generating public-private key pairs can also be computationally expensive.

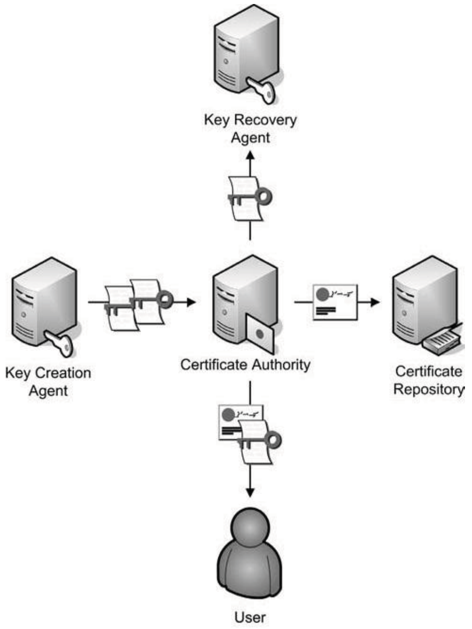


Figure 12: Generation of keys in traditional public key system [10]

Because generating keys and verifying user identities can be expensive, the digital certificates are issued with long validity periods.

The concept of identity based encryption originated from Adi Shamirs paper in 1984. The paper described a rough outline of the properties that such a system should have, but he was not able to find a secure and feasible technology for its practical implementation.

An IBE based system has similarities with traditional public key systems, but is also quite different in other ways. Traditional public keys contain all the parameters needed to use the key, but to use an IBE; a user typically needs to get a set of public parameters from a trusted third party. With these parameters, a user can then calculate the IBE public key of any user and use it to encrypt information to that user.

The recipient of IBE-encrypted information has to authenticate to the PKG, he then calculates the private key that corresponds to the public key. The key is distributed to the authorized user.

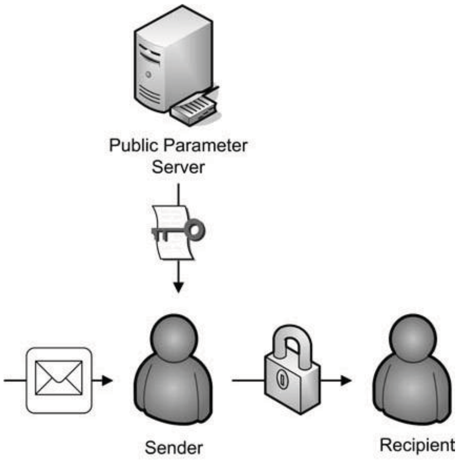


Figure 13: Generation of keys in an Identity Based Encryption system [10]

The core properties of an IBE based cryptography schema can be enumerated as follows

Any kind of string can be used as an IBE encryption key (public key). The string can be a sequence of characters or bytes such as a role, a text, email address, a picture or a set of instructions. Information is encrypted by using the key derived from this string. The trusted third party or the PKG is the only entity that can generate the corresponding decryption key.

The second property is the delay of generation of decryption key. The decryption key associated with an encryption key can be generated later than the encryption key.

The third and a very important property is the reliance on a trusted third party for the generation of Private Keys.

An Identity based system consists of four algorithms.

- Setup: Initialize all the system parameters
- Extraction: Calculate IBE private key from PKG and an identity using system parameters
- Encrypt: Encrypt information using an IBE public key calculated from system parameters and an identity
- Decrypt: Decrypt information using an IBE private key calculated from PKG master key and an identity

There have been several cryptographic schemes that are based on Identity based encryption. Cocks IBE scheme, Boneh-Franklin IBE scheme, Boneh-Boyen IBE scheme and the Sakai-Kasahara IBE scheme. Boneh-Franklin IBE scheme is the most popular of all the schemes. In the next section we will talk in detail about this scheme.

2.4.1 Boneh Franklin IBE

The Boneh Franklin IBE system was the first practical and secure IBE system invented. Before we discuss about the Boneh Franklin IBE system, we should have a little background about Pairing Based Cryptography which is the backbone of Identity Based Encryption systems.

Pairing Based Cryptography

The idea of pairing is to construct a map between two cryptographic groups which allows for new cryptographic schemes based on the reduction or transform of one problem in one group to a different problem in the other group. The problem in the group which is being mapped to might be an easier problem. Tate and Weil pairing are the well-known implementations of these pairings. The pairings were initially used as cryptoanalysis tools to reduce the complexity of the discrete logarithmic problem on weak elliptic curves.

Bi-linear pairings

$G1$, $G2$ and GT are cyclic groups of prime order r.

Let g1 be the generator of $G1$ and $g2$ is the generator of $G2$.

Bilinear pairing or bilinear map e is an efficiently computable function e: G1 X G2 -> GT

1. Bi linearity $\forall a,b \in Zr$ it holds that $e(g1a,g2b) = e(g1,g2)ab$
2. Non-degeneracy $e\,(\mathrm{g1},\mathrm{g2})\;!=\;1$

In cryptography G_1 and G_2 are usually taken from certain elliptic curves over finite field F_q while G_t is taken from the extension fields $F_q k$.

Let P be the generator of G1 and G2, the following properties hold for a pairing

It is bilinear that is:

$e\,(aP,bQ)=e(P,Q)^{ab}$
$e(P1 + P2\,,Q)=e(P1\,,Q)e(P2\,,Q)$
$e(P,Q1 + Q2\,)=e(P,Q1\,)e(P,Q2\,)$

It is non-degenerate:

$e\,(P,Q)!=1$

Algorithms

As discussed earlier, every IBE scheme has four main algorithms, the setup, extract, and encryption and decryption algorithms. Here we list the algorithms in brief.

Setup algorithm

INPUT: a security parameter k, an elliptic curve E, a plaintext bit length n
OUTPUT: $BFParams = (G1, GT, e, n, P, sP, H1, H2, H3, H4)$ and master secret s

1. Select a prime p and prime power q with $p\ /\#E\ (Fq\ )$ and
$P2\ /\#E(Fq\ )$ and such that the bit security level provided by p and q meets the required security parameter k . For best performance, p should be a Solinas prime.
2. Select a random $P \in E\ (Fq\ )\ [\ p]$ and let $G1 = <P>.$
3. Let k be the embedding degree of $E\ /\ Fq$; select a pairing
$e\hat{}: G1\ X\ G1 \rightarrow F^*qk$.
4. Let $G_T = <e(P, P)>.$
5. Select a random $s \in Z^*p$ and calculate $sP.$
6. Select appropriate cryptographic hash functions
$H1: \{0, 1\}^* \rightarrow G1,$
$H2: GT \rightarrow \{0, 1\}\ n,$
$H3: \{0, 1\}\ n\ X\ \{0, 1\}\ n \rightarrow Z^*p$ *and*
$H4: \{0, 1\}\ n\ X\ \{0, 1\}\ n \rightarrow Z^*p.$
7. The master secret is the value $s.$
8. The public parameters are BFParams = (G1, GT,e, n, P, sP, H1, H2, H3, H4).

Key Extraction Algorithm

INPUT: A string ID representing an identity and a set of public parameters
$BFParams = (G_1, G_T, e, n, P, sP, H_1, H_2, H_3, H_4).$
OUTPUT: The private key $sQID$
1. Calculate $sQID = sH1\ (ID).$

Encryption algorithm

INPUT: A plaintext message M of length n bits, a string ID representing the identity of the recipient of the ciphertext, a set of public parameters
$BFParams = (G_1, G_T, e, n, P, sP, H_1, H_2, H_3, H_4).$
OUTPUT: A ciphertext $C = (C1, C2, C3)$
1. Calculate $QID = H1\ (ID)$.
2. Select a random σ ∈ {0, 1}.
3. Calculate $r = H3\ (\sigma, M).$

4. Calculate $C1 = rP$.
5. Calculate $C2 = \sigma \oplus H2\,(e(rQID\,,\,sP\,))$.
6. Calculate $C3 = M \oplus H4\,(\sigma)$.
7. *Sets the ciphertext to* $C = (C1, C2, C3)$

Decryption algorithm

INPUT: A ciphertext $C = (C_1\,, C_2\,, C_3\,)$*, a set of public parameters BFParams*
$= (G_1\,, G_T, e, n, P, sP, H_1\,, H_2\,, H_3\,, H_4\,)$*, a private key sQID .*
OUTPUT: A plaintext message M or an error condition
1. Calculate _ = $C_2 \oplus H_2\,(e\,(sQ_{ID}\,, C_1\,))$.
2. Calculate $M = C_3 \oplus H_4\,(\sigma)$
3. Calculate $r = H_3\,(\sigma, M)$ and then calculate rP. If $C_1 \uparrow rP$ then raise
an error condition that indicates an invalid ciphertext. Otherwise,
return the plaintext M.

IBE is a great way to provide confidentiality of data, but it fails in the other requirements of an information security solution. A hybrid solution that uses IBE for encryption and a technology that provides digital signatures might

2.4.2 Identity Based Signatures and Access Control

Identity Based Encryption and access control Combining Authentication with Role-Based Access control Based on IBS was proposed [11] to provide a solution for cryptographically providing authentication and role based access control for large organizations. The scheme is based on an Identity Based Signature scheme, the scheme is used for both user authentication and role based authorization at the same time. They achieve this by extending the elements user and role in RBAC to include Identity-based cryptography. Each user uses his/her identity as a public key and has a set of private keys corresponding to the roles assigned to them. The manager checks the validity of the users identity and activated roles by verifying the signature. The cons of this system are the storage of private keys on the users end. A key is generated for every unique role and it gets difficult to manage when the number of roles increases.

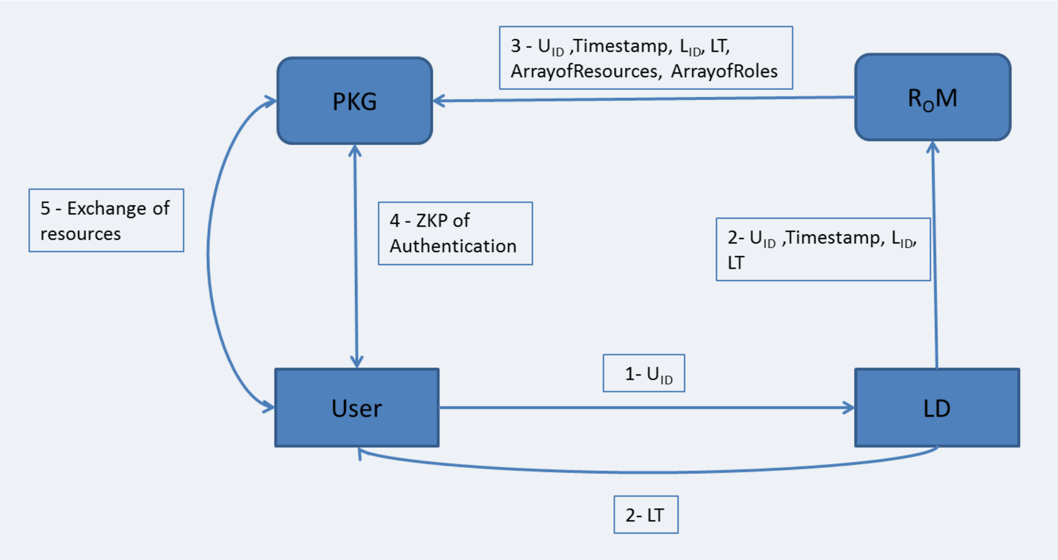
From the literature discussed in this section, the models and protocols proposed in [2] and [8] are the most similar to the work proposed in this thesis. The proposed model borrows the base architecture from the model in [2] but significant changes have been proposed in the design principles and the cryptographic extensions to the model. Identity Based Encryption schemes have been used in combination with Role Based Access Control. Our work is also different from the work proposed in [11] as we use IBE for securing information access rather than the actual enforcement of the RBAC protocols. We also discuss in brief about NFC and how threats modelled around the communication can be mitigated.

Chapter 3: Proposed Framework

3.1 System Architecture

In this chapter we propose our framework for mobile RBAC using IBE with ZKP. The framework is modeled along the lines of the work by [2], and acts as a secure cryptographic framework for enforcing the policies of the model. As discussed in Chapter 2, systems such as this have security requirements that fall into the following categories.

- Verification of the user's location. Validating the users claim to a location.
- Authentication of user's identity is verifying that the user requesting the access to the request is actually the user of the system.
- Access control is to verify that a user has the permission to access a specific resource
- Safe distribution of resources



PKG
R_OM
User
LD
3 - U_{ID} ,Timestamp, L_{ID}, LT, ArrayofResources, ArrayofRoles
5 - Exchange of resources
4 - ZKP of Authentication
2- U_{ID} ,Timestamp, L_{ID}, LT
1- U_{ID}
2- LT

Figure 14: System architecture and Communication

Let us discuss how the framework deals with each of these requirements

Principles involved

User – The User principle is a user of the system (The description of the user is strongly bound to the physical mobile device that he carries). He can be assigned to one or more roles in the system. The user is assigned an id and credentials when he/she is enrolled into the system. The id and credentials are used to authenticate the user to the PKG. The mobile device that the user carries should be capable of performing cryptographic operation, capable of networking operations and some sort of near field communication to communicate to the location device. NFC, a derivative of Radio frequency identification is used in this system.

Location Device – The location device is installed at strategic locations in the system. The LD holds information that can authenticate a user to a particular location in the system. It is assumed that the role manager and LD can communicate securely. The LD ideally is an embedded device that is capable of talking to the users mobile device and capable of cryptographic operations.

Private Key Generator [PKG]: The PKG is the Trusted Third Party in the system. It plays a crucial role during the setup of the system and also during the normal operation phase of the system. It has the responsibility to authenticate the user against the credentials supplied and also regulates the access to resources based on the information given by the role manager.

Resource Manager [R_sM]: The resource manager maps the resources to the resource id's, it works closely with PKG during the distribution of reources.

Role Manager [R_oM] – The role managers maps the user to a set of roles. The R_oM accepts user's requests through the Location Device, it is responsible for verifying the location claims of the user, list a set of active roles and resources that are accessible to the user for that session and delegate the information to the information to the Resource manager.

Assumptions

Time restrictions are to be applied wherever necessary. The R_OM can assign a time restriction on the user's request based on the access control policies. In our framework we assume that the R_OM has a set of policies that can initiate a session for the user. For every session initiated the R_OM sends a list of resources that the user has access to at that particular time and location for the time period assigned by the role manager. The PKG accepts requests for resources for the duration of the session. When the session has expired, the user is denied any access to resources and has to initiate the process all over again to gain access to resources.

We also assume that the Private key Generator who assigns the cryptographic parameters to all the principals is not malicious and trusted by all the parties.

Verification of the Users Location claim

Verification of the users location claim is a very important criterion for the framework. In traditional systems, the access to sensitive resources was restricted to the workstations which were fixed and the verification credentials were embedded into the workstations. In order to accommodate mobile workstations, we need to modify the method in which a location claim is made.

Near field communication is a technology that gives us the advantage of proximity. With the recent advancements in NFC such as the establishment of peer to peer communication protocols, sensitive data can be stored in a device without the threat of the tag being "sniffed" by a powerful reader. The proximity needed for establishing the communication between two devices that are capable of NFC makes NFC an automatic choice for the user to "check in" at a particular location.

The Location device and the user's mobile device are assumed to be capable of near field communications. Once the user claims his location to the Location device, security must be in place to make sure that the Location Device place can communicate to the Role Managing authority securely and authenticate itself to the authority. The Location Device is preloaded with an initial seed. The seed is used to produce a secure random number that will be used to compute a Location Token with every request that goes to the Role Managing authority. The random number produced for the current transaction will be used as the seed for the next transaction to make the entire process more secure. The detailed steps on how the location token is computed and other steps for this process will be explained in detail later in the section.

Authentication of Users Identity

The second and a very important security consideration for an access control system is the authentication of the users. Once the location of the user has been established by the Location Device, it is now the duty of the Private Key Generator to verify whether the user is an authentic user and if the user is really who he says he is before any exchange of resources takes place. This process of authentication is twofold. The user has to verify that he is indeed a user of the system and that he has established his location to a location device. Strong authentication schemes can be difficult to deploy and maintain, an example of such an authentication scheme is the use of one-time passwords for all the users in the system. The proposed framework handles authentication of the users using Zero Knowledge proof of authentication and using Location Tokens which are valid for a session/ time period. ZKP proofs strong authentication and are easier to deploy than one time passwords to all the users of the system. Complete details would be discussed later in the chapter.

Access control

We will not discuss in detail how the access control policies are enforced in the system. We plan to use a variant of the GEO-RBAC model discussed in chapter 2

Secure distribution of resources

After the user has been authenticated and his location established at a certain place and time, the PKG now is ready to serve the user the resources that he requested for. The distribution of resources has to be done in a secure manner. The obvious choice would be the use of a one-time session key for the encryption and safe distribution of resources between the PKG and the user. Identity based Encryption has several advantages over traditional public key encryption techniques as discussed in Chapter 2. The ability to calculate keys make short-lived keys (session keys in this case) makes IBE an ideal choice of encryption for the framework.

3.1.1 Setup Phase

In order for our system to be functional, a number of setup steps have to be followed. The PKG plays a very important role during this process. The PKG is responsible for setting up the parameters and credentials for the user and the Location device to communicate with the R_OM and other entities of the system.

3.1.1.1 Setting up the Location Device

The location device is an embedded device, which is to be deployed at strategic locations in the system. After the PKG has been deployed, the PKG has to set up a few parameters in the LD.

Seed for the pseudorandom number generator – The LD has a secure pseudorandom number generator. The PKG assigns a seed for each Location Device that is deployed. The seed is used to generate a random number which in turn is used to generate a location. The Location Token is generated for each request made to the LD.

3.1.1.2 Setting up the User

Every user that wants to have access to resources has to go through a setup process. On a higher level, he is assigned an id and also the list of roles that he can activate in the system.

Along with the higher level parameters, the PKG assigns some parameters which are used by the system to authenticate the user and provide secure access. These are the parameters required for the ZKP packet generation and the parameters required for the IBE system.

- Generator G of the elliptic curve
- Pre-shared point P on the elliptic curve
- Point B on the elliptic curve where B=x.G
- Hash function which is used to calculate the ZKP packet

3.1.2 Operation phase

After the parameters have been successfully setup, the system can now start functioning in the normal operation phase. The normal operational phase consists of the following four phase

- Session Initiation
- Session establishment
- Distribution of resources
- Session re-establishment
- Session termination

The Let us discuss the protocol on a high level and then detail the steps involved in each step.

3.2 Protocol in overview

1. [User -> LD: U_{ID}]: The user's device sends its unique identifier to the location device. This step is for the user to check-in at a particular location.

2. [LD -> R_OM: U_{ID},Timestamp, L_{ID}, Location Token]: The location device on receiving a request from the user sends the user's identification, the time and the Location Token to the R_OM. The Location Token is used by the R_OM to decide whether the request came from an authentic Location Device. This binds the user to the location at that given time. It is assumed that the Location Device is in a fixed position and can communicate in a secure manner to the role manager.

3. [R_OM -> PKG: User,ArrayofRoles,ArrayofResource ID's, Time] : The role manager listens for requests from Location Device. After receiving a request, the

R_OM authenticates the Location Device and checks whether the request has arrived from a valid user id. If the two conditions are valid, the R_OM sends information regarding the user and the resources he has access to during the time period, the location token to the PKG.

4. The PKG after receiving the request access forwarded by the R_OM, places the request in the pending request buffer and waits for the user to send a ZKP proof of verification. After the user sends his authentication credentials, the PKG checks the credentials and verifies the timestamp to check if the session has timed out. The PKG initiates a Modified Diffie-Hellman with non-interactive Zero-Knowledge Proof to exchange the session keys.

5. The protocol is now in the normal operation phase and Data exchange can take place between the PKG and the user by using the session key generated.

3.3 Detailed Protocol Steps

3.3.1 Session Initiation

The user who wants to access resources at a particular location has to establish his location to the role manager using the Location Device. Ideally, the location device is a NFC device that is capable of communication with a NFC enabled mobile device. The LD sends the request packet, which consists of the Location Token, User id (U_{ID}), Location ID (L_{ID}) and Timestamp. After receiving the Access Request, the R_OM verifies that the request came from a legitimate LD, the U_{ID} exists in the system and if the user has any roles to play at that location and time. After verifying the request, the R_OM computes the array of roles that the user is eligible for and forwards the data to the PKG.

Step 1: Start
Step 2: User sends his U_{ID} to the location device
Step 3: The location device receives the U_{ID} and calculates the Location Token

```
ComputeLocationToken($U_{ID}$,TimeStamp)
      PRN = PRNG (LAST_PRN)
```

```
        Location_Token = HASH ($U_{ID}$,TimeStamp,PRN)
        LAST_PRN = PRN
        return Location_Token
```

Step 4: Location device sends (Location_Token || TimeStamp || U_{ID} || L_{ID}) to R$_O$M

Step 5: Location device sends the (Location_Token) to the user

Step 6: R$_O$M receives the request packet and checks if the request came from a valid Location Device.

```
    VerifyRequestPacket(Location_Token,$U_{ID}$,$L_{ID}$,TimeStamp)
        START
        Location_Seed = SeedStore($L_{ID}$)
        PRN = PRNG (Location_Seed)
        Computed_Token = Hash($U_{ID}$,TimeStamp,PRN)
        IF (Computed_Token = Location_Token && isValidUID($U_{ID}$) &&
        userHasRoles)
        THEN
              Store Request_Packet
              Update SeedStore($L_{ID}$) = PRN
              Goto Step 6
              Construct ArrayOfRoles
              Construct ArrayOfResources
        ELSE
              Refuse Request
        STOP
```

Step 7: ROM forwards the (ArrayofRoles||ArrayOfResources||U_{ID}||Location_Token||validityOfSession) to the PKG

Step 8: R$_S$M stores the request and waits for the user to send a resource request

Step 9: Stop

3.3.2 Session establishment

User who has initiated the session needs to establish his identity before the actual data exchange takes place. A session is initiated when LD established that a user has checked in at a location and the R$_O$M passes the request to the PKG after verifying the location token. The PKG is now ready to handle request from the user. There are two important steps that take place during session establishment. The user sends a request to the PKG with a request to start the session. The request packet contains the Zero knowledge proof of authentication, and the location token.

The Zero knowledge proof of authentication is constructed according to the ZKP as Digital signature which was discussed earlier in Chapter 2.

Step 1: If the user has a Location_Token that he received after sending his User id to the LD, he can make a request if the role is valid. User calculates the ZKP authentication packet to send along with the request to the PKG

```
#Preshared Parameters
         Generator G
         x such that B = x.G
         HASH function
         Point P
         #Assigned Password
         Pass
      CalculateZKPPacket()
         START
             PRN = PRNG (Pass)
             Pass_temp = PRN
             A = randGenerated.G
             c = HASH (x.P,r.P,r.G,Pass)
             s = r+c.x
             ZPK_Packet = s||x.P||r.P||r.G
            RETURN ZPK_Packet
           STOP
```

Step 2: User sends the request packet (ZPK_Packet||U$_{ID}$||Location_Token)
Step 3: PKG receives the user request and checks the request buffer if he has a valid request from the user. If the user has a valid role the PKG continues to verify the

Location_Token. If the Location_Token is verified the PKG then continues to verify the ZKP packet sent by the user.

```
#Preshared Parameters
      Generator G
      x such that B = x.G
      HASH function
      Point P
      Can retrieve the password of the user
VerifyZKPPacket (ZKPPacket)
      START
      Received_x.P = ZKPPacket[x.P]
      Received_r.P = ZKPPacket[r.P]
      Received_r.G = ZKPPacket[r.G]
      c = HASH (Pass,Received_x.P, Received_r.P,Received_r.P)
      IF ( s.G = (r+c.x).G = r.G + c.x.G = r.G + c.B = A + c.B ) && (s.P =
             (r+c.x).P = r.P + c.xP)
      THEN
      User verified
      generatesessionKeys(UID)
      initiateKeyExchange(UID)
      ELSE
      Refuse Request
      STOP
```

3.3.3 Session Key Generation

After the user has been authenticated, the session is established and resources can be shared with the user. Distributed messages are encrypted using Identity Based Encryption. We use the basic Identity based Encryption scheme proposed by Boneh and Franklin [12]

A session is created for a fixed period of time; the resource manager can make the decision. After the session has been timed out, the user has to re-establish his location by passing his credentials to the Location device/authenticator to establish a new session.

3.3.4 Secure exchange of data

After the distribution of the session key, the user is free to request for resources. The resources requested by the users are distributed encrypted using the session key generated. There are several ways for the users to request for resources. One way would be to have a set of tagged objects in the location and the user can scan a tag with his NFC enabled phone to be able to request for the resource. Another option would be the PKG to send the list of resources that the user has access to for the session and the user can choose which resource he wants to access.

3.3.5 Session re-establishment

After the timeout, if the user wants to have continued access to the resource, he is required to go through the process of authentication all over again. This includes the checking in at the Location Device and sending an access request to the PKG

3.3.6 Session termination

The session terminates either when the time period expires or he can send a session termination request to the PKG.

Chapter 4: Implementation

The prototype of the framework has been implemented in Java. Java is a very good language for prototyping and it has several inbuilt modules that help in building cryptographic systems. The Role Based Access control component has not been implemented, the main focus of the implementation is to provide the Zero Knowledge proof authentication and Identity based encryption system. The user, Location device and the PKG have been simulated in a test environment. Since the end system user of the framework uses a mobile device, we also implemented a small test app on an android based cell phone and profiled the time taken for the generation of a zero knowledge proof of authentication and decryption using an identity based encryption system. Experiments have also been done on the amount of time it takes to send the user id over NFC to a Desktop NFC reader running an Ubuntu machine. In this section we will explain the details of the simulation and some sample screen shots of the test run.

The principals in the simulation use standard java ports to communicate with each other. The three principles are represented by the following files. Node.java is the representation of the user, LocationDevice.java represents the location device and provider.java represents the PKG (the functionalities of the role manager reside inside this as well) and a controller program that controls the workflow.

4.1 Pairing Based Cryptography implementations

Pairing based Cryptography is the backbone of Identity Based Cryptography has garnered a lot of interest in recent years. After the breakthrough paper [12] several implementations of Pairing Based Cryptography have come up. The PBC library developed by Ben Lynn is the most popular of the implementations. To bring the PBC library to Java, Angelo De Caro [13] developed a java wrapper for the PBC library. In [14] the authors describe an object-oriented approach to an IBE system. The paper details about the implementation of an IBE system and the algorithms involved in such a system. For this implementation we use a java based pairing library called jPair, we selected the library because of the ease of use and fast prototyping.

We use two elliptic curves in the implementation. The first curve is for the Zero Knowledge proof of authentication and the other curve is used for the pairing. JPair API defines a set of predefined pairings that provide a 1024 bit security.

4.2 Peer-to-Peer communication with an NFC device

We also implemented an Android-based app that can communicate with an NFC desktop reader to test the peer-to-peer communication. The android application was developed using the android NFC API which gives provides us the functionality of pushing an NDEF message to another NFC reader. The android nfc stack is one of the most advanced near field communication stack on a mobile operating system. The desktop-based application, which was the simulation of a location device, was developed using the open source libnfc api and libnfc-llcp. The application receives data from the phones, calculates the Location Packet and sends the data back to the phone in a single tap on the Location Device. The application was developed on the Ubuntu variant of Linux.

We used a Nexus S Android device, which runs the gingerbread version of the android .We tested the communication using two NFC communication protocols. The NDEF Push Protocol developed (NPP) which was developed by Google as a part of its Gingerbread Operating System release and Simple NDEF push protocol (SNEP) which was proposed by the NFC-FORUM as a standard for communication over Peer-to-Peer using NFC. Both the protocols performed well for simple data transfer, but the SNEP standard proposed by NFC-FORUM is a much more reliable protocol for the data transfer and can handle larger amounts of data.

Ideally our vision of a Location Device is a standalone system which can be deployed in different strategic locations in the system. The embedded system should be capable of near field communication and networked communication over LAN. It also should be able to perform some hash functions to calculate a unique location token for each session that it initiates.

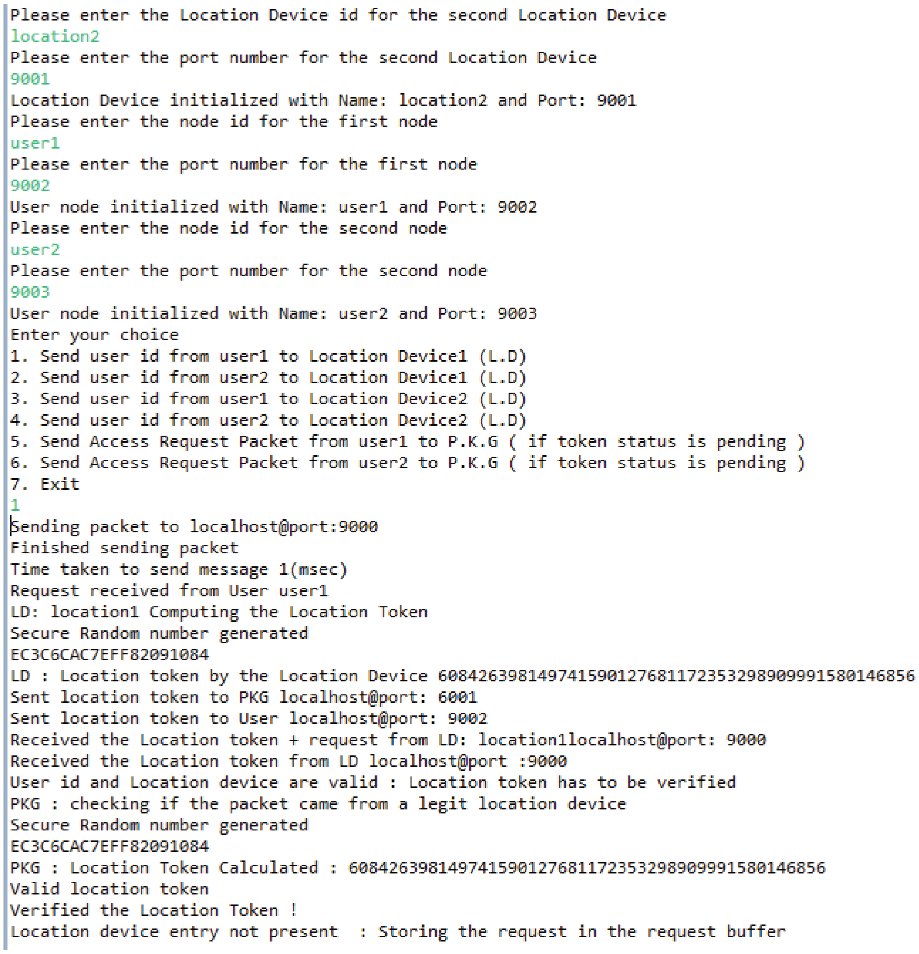
4.3 Test Runs

The first screenshot represents a sample run of the framework. The controller program initially allows the user to initialize two Location Devices and two users of the system providing the port number for communication. After getting the details from the user, the PKG assigns the parameters required for the normal operation phase to all the principals.

The location devices are initialized with the seed value for calculating the secure pseudo random numbers and the users are assigned the parameters required for Zero Knowledge proof of authentication and the IBE exchange. After the initialization, the system is ready for normal operation. There are basically two options that a user can choose from. The first one is sending the user id to a Location Device (the communication is done using ports in this test setup, but we will post results on how the actual RF communication takes place later in this section) and sending the access request to the PKG authenticating using the Zero Knowledge proof of authentication. The first screenshot shows the state of the system after it was initialized with two location devices, two users and when the user1 sends a "check in" request to the Location device 1.

4.3.1 Normal run of the system

Let us walk through the steps that take place when the user chooses the option sending the user id to Location Device 1 to check in at that location.

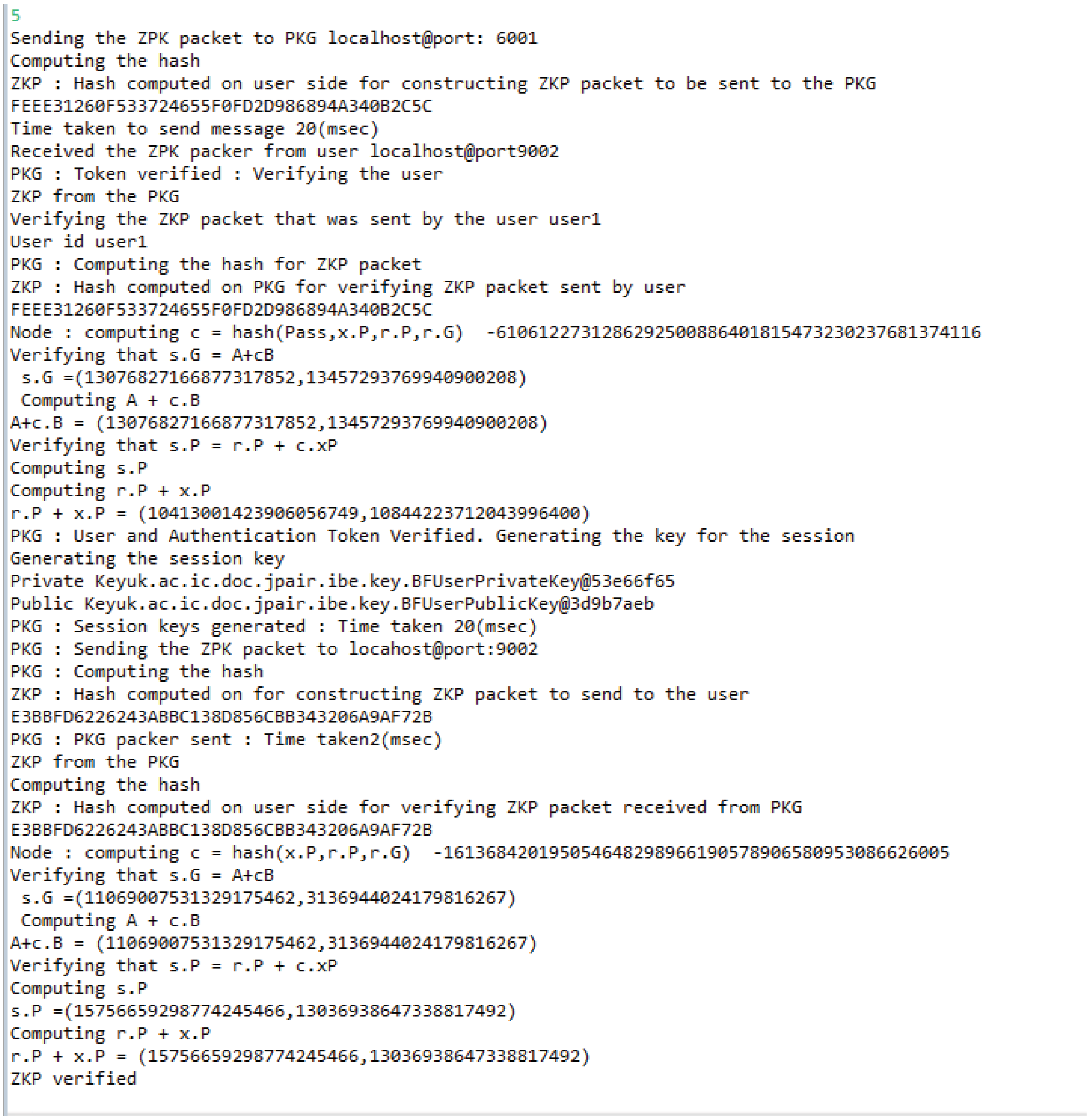


```
Please enter the Location Device id for the second Location Device
location2
Please enter the port number for the second Location Device
9001
Location Device initialized with Name: location2 and Port: 9001
Please enter the node id for the first node
user1
Please enter the port number for the first node
9002
User node initialized with Name: user1 and Port: 9002
Please enter the node id for the second node
user2
Please enter the port number for the second node
9003
User node initialized with Name: user2 and Port: 9003
Enter your choice
1. Send user id from user1 to Location Device1 (L.D)
2. Send user id from user2 to Location Device1 (L.D)
3. Send user id from user1 to Location Device2 (L.D)
4. Send user id from user2 to Location Device2 (L.D)
5. Send Access Request Packet from user1 to P.K.G ( if token status is pending )
6. Send Access Request Packet from user2 to P.K.G ( if token status is pending )
7. Exit
1
Sending packet to localhost@port:9000
Finished sending packet
Time taken to send message 1(msec)
Request received from User user1
LD: location1 Computing the Location Token
Secure Random number generated
EC3C6CAC7EFF82091084
LD : Location token by the Location Device 608426398149741590127681172353298909991580146856
Sent location token to PKG localhost@port: 6001
Sent location token to User localhost@port: 9002
Received the Location token + request from LD: location1localhost@port: 9000
Received the Location token from LD localhost@port :9000
User id and Location device are valid : Location token has to be verified
PKG : checking if the packet came from a legit location device
Secure Random number generated
EC3C6CAC7EFF82091084
PKG : Location Token Calculated : 608426398149741590127681172353298909991580146856
Valid location token
Verified the Location Token !
Location device entry not present  : Storing the request in the request buffer
```

Figure 12: User Request sent to location device

The first step involves the user checking in to a location device by sending his/her identity to the Location to the device. After receiving the identity, the LD calculates the location token. The token is calculated by calculating the hash of the secure random number generated the user id and the timestamp when the request was received. The LD sends over the user request to the R$_O$M manager (we do not have a role manager implemented that maps the users id to the role, the PKG handles the request sent by the LD in the test system). The LD simultaneously sends a location Token back to user who initiated the request. After receiving the request, the PKG unpacks the request to check if the request came from a valid Location Device and the User Id is valid. The PKG checks the Location token by generating the pseudorandom number from the seed associated with the Location token. We used the Java implementation of SHA1PRNG with a 10 bit secure random number as the seed. The seed for the next iteration is the result of the previous secure random number generation. To make sure that the seed values never get desynchronized and to be able to come back to steady state, the PKG maintains a buffer of all the seed values generated. After the verification of the location device has been done, the request is stored in the request Buffer and the PKG is now ready to accept a request for resources from the User.

The user receives a notification of a check-in at a particular location and he can now initiate an access request to resources from the PKG. When the User wants to request for a resource, he has to authenticate himself to the PKG. The authentication process involves the user generating a Zero Knowledge proof of authentication packet and sending it to the PKG along with the Location Token that it received from the Location Device checks in. The process can be seen in the screen shown below, the systems outputs a series of self-explanatory messages that follow the ZKP protocol discussed in the Chapter 2 with an enhancement to the protocol, each user is given is assigned a password when they are setup to use the system. The PKG initially checks if the user has a pending request in the request buffer, if the Location Token that is associated with the user in the request buffer is equal to the Location Token sent by the user and, the PKG continues to verify the ZKP authentication sent by the user. The output of the hash function is used to produce a 10-bit BigInt type object, which is recommended by Java for cryptographic protocols. The BigInt is used to perform arithmetic operations such as addition with the elliptic curve points.

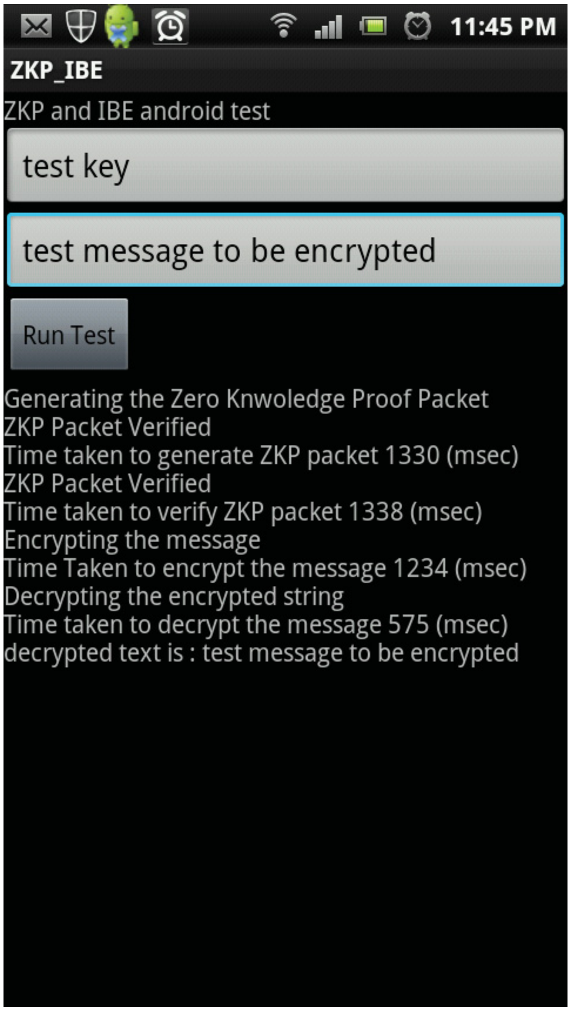


```
5
Sending the ZPK packet to PKG localhost@port: 6001
Computing the hash
ZKP : Hash computed on user side for constructing ZKP packet to be sent to the PKG
FEEE31260F533724655F0FD2D986894A340B2C5C
Time taken to send message 20(msec)
Received the ZPK packer from user localhost@port9002
PKG : Token verified : Verifying the user
ZKP from the PKG
Verifying the ZKP packet that was sent by the user user1
User id user1
PKG : Computing the hash for ZKP packet
ZKP : Hash computed on PKG for verifying ZKP packet sent by user
FEEE31260F533724655F0FD2D986894A340B2C5C
Node : computing c = hash(Pass,x.P,r.P,r.G)  -6106122731286292500886401815473230237681374116
Verifying that s.G = A+cB
 s.G =(13076827166877317852,13457293769940900208)
 Computing A + c.B
A+c.B = (13076827166877317852,13457293769940900208)
Verifying that s.P = r.P + c.xP
Computing s.P
Computing r.P + x.P
r.P + x.P = (10413001423906056749,10844223712043996400)
PKG : User and Authentication Token Verified. Generating the key for the session
Generating the session key
Private Keyuk.ac.ic.doc.jpair.ibe.key.BFUserPrivateKey@53e66f65
Public Keyuk.ac.ic.doc.jpair.ibe.key.BFUserPublicKey@3d9b7aeb
PKG : Session keys generated : Time taken 20(msec)
PKG : Sending the ZPK packet to locahost@port:9002
PKG : Computing the hash
ZKP : Hash computed on for constructing ZKP packet to send to the user
E3BBFD6226243ABBC138D856CBB343206A9AF72B
PKG : PKG packer sent : Time taken2(msec)
ZKP from the PKG
Computing the hash
ZKP : Hash computed on user side for verifying ZKP packet received from PKG
E3BBFD6226243ABBC138D856CBB343206A9AF72B
Node : computing c = hash(x.P,r.P,r.G)  -161368420195054648298966190578906580953086626005
Verifying that s.G = A+cB
 s.G =(11069007531329175462,3136944024179816267)
 Computing A + c.B
A+c.B = (11069007531329175462,3136944024179816267)
Verifying that s.P = r.P + c.xP
Computing s.P
s.P =(15756659298774245466,13036938647338817492)
Computing r.P + x.P
r.P + x.P = (15756659298774245466,13036938647338817492)
ZKP verified
```

Figure 13: ZKP proof packet sent from user to PKG

4.3.2 ZKP generation on an android device

We ported the code for the generation of Zero Knowledge proof packet on an Android machine to test the time required to generate the packet. The application was installed on an Xperia X10 running the gingerbread version of the android operation system. Figure 15 shows the screenshot of the test.



```
Generating the Zero Knwoledge Proof Packet
ZKP Packet Verified
Time taken to generate ZKP packet 1330 (msec)
ZKP Packet Verified
Time taken to verify ZKP packet 1338 (msec)
Encrypting the message
Time Taken to encrypt the message 1234 (msec)
Decrypting the encrypted string
Time taken to decrypt the message 575 (msec)
decrypted text is : test message to be encrypted
```

Figure 14: ZKP and IBE tests on an android device

Chapter 5: Security Analysis

We present an informal security analysis of the system in this section. The principal focus of the framework is to secure the access to sensitive resources from malicious users, so the authentication of the user to the system will be primarily discussed in this section. The authors in [8] have enumerated the threats that are common to authentication protocols. The common attacks are replay, collusion, reflection, denial of service and typing, eavesdropping and modification.

Before we discuss the various threats and scenarios; let us recall the security that the ZKP proof of authentication provides us. Please refer to the algorithm in Chapter 2 Section 2.3.1 for the algorithm in the discussion. The security of the ZKP protocol is from the fact that calculating the value of c is a challenge for the prover as it is calculated from a hash function and out of the provers control. If the prover does not know the newly calculated value of Pass and the preshared parameter x it is very difficult to find satisfying s.G = r.G + c.x.G which is an instance of the Discrete Logarithm problem.

The communication between the user and the PKG/R_SM and the key generation is done using the Boneh Franklin IBE scheme. The security of the scheme is based on the hard Bilinear Diffie Hellman Problem, which is believed to be difficult to reverse in real time.

Assumptions

We put forward a few assumptions while stating the threat models.

Integrity of the Private Key Generator (PKG) and Role Manager (R_OM): The PKG is responsible for distributing the cryptographic parameters to all the principles. It also has access to all the resources of the system. Hence we assume that the PKG is a secure server, which is protected by suitable software and network security as well. The R_OM is the principle that can assign roles, resources to the user and also determine the time period for which a session is valid. The unique “seed password” gives to the users is stored safely on the user’s phone in a secure manner. We also assume that the primitives communicate in a predefined packet structure and any request which is not in this structure will not be processed.

The LD is placed at strategic locations according to granularity of spatial constraints required by the enterprise. Ideally this system is suitable for indoor location verification; it can be extended for outdoor use as well. The communication between the User and the LD is done use Near Field communication. The communication between the Location Device and the role manager is done using a wired infrastructure and is assumed to be secure.

5.1 Rogue Node

The rogue node is a malicious user of the system. We have two implementations of the rogue node. The first implementation has legitimate public parameters of the system, but the intentions are malicious. The second implementation is a malicious hacker who does not have the public parameters but can capture communication and replay the communications to the PKG.

5.2 Replay Attack

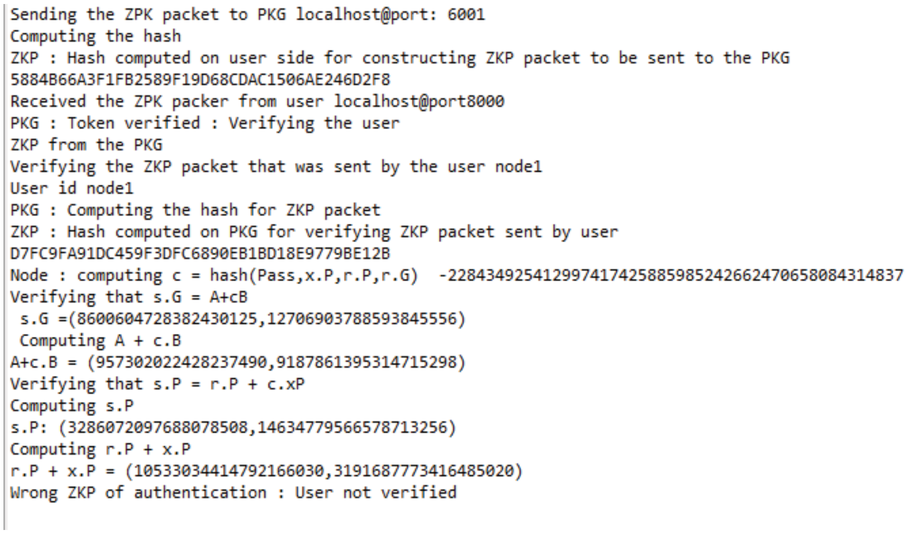
The goal of the replay attack is for the user or an eavesdropper to reuse a piece of data as a part of the false request. There are a few scenarios on how the attacker could try to gain illegal access to the resources. Let us consider the scenario where the user is a legitimate user of the system and tries to convince the PKG that he is another user.

A legitimate user of the system tries to access resources that he is not assigned to. Let us assume that he by-passes the application level security on his mobile device and sends a request to the Location Device impersonating another user's identification. The Location Device sends the request to the R_OM. The R_OM verifies the Location Device and forwards the access request to the PKG. The user now tries to send an access request to the PKG. The user collected the Location Token from the Location Device and tries to use a legitimate location token along with a Zero Knowledge Proof packet of authentication to the PKG. The PKG verifies the Location Token and starts verifying the

ZKP packet. Please note that because of time restrictions which are in place, the access to resources may be denied if the request is made after the expiration of the session.

When calculating c = HASH (x.P,r.P,r.G,Pass) in the verification function of the ZKP packet, the hash does not match because of the “Pass” parameter. The attacker used a packet that he managed to eavesdrop from an earlier transaction that involved an access request to the PKG from the user. The Pass is set another securely random generated number after the completion of the transaction and it is difficult to reverse engineer the Pass from the value of the Hash that it was used to generate.

Because of the fail in the verification of the Zero knowledge packet, the PKG denies the request of the user.



```
Sending the ZPK packet to PKG localhost@port: 6001
Computing the hash
ZKP : Hash computed on user side for constructing ZKP packet to be sent to the PKG
5884B66A3F1FB2589F19D68CDAC1506AE246D2F8
Received the ZPK packer from user localhost@port8000
PKG : Token verified : Verifying the user
ZKP from the PKG
Verifying the ZKP packet that was sent by the user node1
User id node1
PKG : Computing the hash for ZKP packet
ZKP : Hash computed on PKG for verifying ZKP packet sent by user
D7FC9FA91DC459F3DFC6890EB1BD18E9779BE12B
Node : computing c = hash(Pass,x.P,r.P,r.G)  -2284349254129974174258859852426624706580843148 37
Verifying that s.G = A+cB
 s.G =(8600604728382430125,12706903788593845556)
 Computing A + c.B
A+c.B = (957302022428237490,9187861395314715298)
Verifying that s.P = r.P + c.xP
Computing s.P
s.P: (3286072097688078508,14634779566578713256)
Computing r.P + x.P
r.P + x.P = (10533034414792166030,3191687773416485020)
Wrong ZKP of authentication : User not verified
```

Figure 15: ZKP proof packet sent by a rogue node

The malicious user then tries to send an access request to the Private Key Generator. One of the ways he can try to do it is by capturing the request sent by the user. The framework will be able to defend against these kinds of attacks because of the different hash output generated by the Zero Knowledge Proof of Knowledge packet produced for every request sent. Figure 14 shows the step by step verification done on the PKG side.

5.3 Collusion

A collusion attack is an action carried out by malicious users in possession of copy of protected content of the system. A user can check in at one location and pass Location Token obtained to another user of the system. This kind of attack is unlikely but in case of such an access request is received, because the Location Token is constructed by using the user id sent to the LD as an input to the a hash function, the token is bound to one user for a session. Hence the PKG access request fails.

A user can also check-in at one location and send the access request from another location. We believe that by physically placing the resource id's at the location e.g. Placing NFC tags that contain the resource ids at location so that the user needs to scan the resource id to obtain access to the resource. We also believe the time restrictions in place for the access control policies can subdue these kinds of attacks to an extent.

5.4 Reflection attack

Reflection attacks are popular in authentication protocols. In a reflection attack, the attacker can engage in a protocol to get data that can be reused a part of the request. In our framework, Zero Knowledge proof as a digital signature is used as the authentication between the user and the PKG generator that can give access to resources. The use of an initial seed that generates a secure pseudo random which is used as an input for the generation of the ZKP packet makes this attack obsolete.

5.5 Denial of Service

A Denial of Service is the most frequent application level threat detected in any networked system. In our system we try to mitigate DOS attacks by distributing the functionalities over the principles of the system. Moreover the processing capacity of the principles is also taken into account; the LD which is an embedded device and does not have much processing power is not expected to do any lookups about the user ids that it processes.

5.6 Threats due to NFC communication

In [4] the authors detailed attacks that can be launched on an NFC enabled phone. Most of the attacks are launched using malformed NDEF records which use the underlying smartphone operating system vulnerabilities. Our system does not use NFC for communicating between two smartphones, but between an embedded device that is capable of NFC communication and an NFC smart phone (we will use the example of an android based smartphone). The attack vectors that are stated in the literature are not a direct threat to the system but may indirectly threaten the system by taking control of the mobile device that hosts the application.

When an android device receives an NDEF message either via reading a tag or a peer-to-peer communication, it opens an application that can handle the type of NDEF records. For example an NDEF message that contains a URL will be handled by the browser application. Android uses intent filters to have a mapping between the application and the type of NDEF that it can handle. As our system is proposed to be a closed system, one way of mitigating these types of attacks is to use an unknown type NDEF message and let the application handle only the records it is supposed to handle. The LD is under the control of the system so it can be configured to construct an unknown NDEF record to communicate with the user.

Time Complexity

We ran the simulation of the system on a Dell XPS machine with an 8 GB RAM and an i7 Intel processor. We also tested the generation and verification of a Zero Knowledge proof of authentication on a Nexus S device that runs on a 1 GHz Qualcomm Snapdragon processor with a 1 GB RAM. The table below contains the time taken for different steps of the protocol to execute

Step in Protocol	Desktop Machine(Simulation)	Phone
Sending user ID to LD	1 msec	NA
Bi-directional communication between phone and LD	NA	3 s
Generation of Location Token	7 msec	NA
Sending message from LD to R_OM	1 msec	NA
Verification of Location Token and placement in request buffer	1 msec	NA
Generation of ZKP request on Users side	28 msec	1384 msec

Verification of ZKP packet by the PKG and generation of session keys	43 msec	NA
Verification of ZKP packet on Users side	17 msec	1341 msec

Table 1: Time taken for executing steps of the protocol

Chapter 6: Conclusion and Future Work

Role based access control has been an industry standard for securing access to sensitive resources in an organization. For the past several years there has been a gradual change from the use of fixed workstations (computer terminals) to of mobile devices at an enterprise level, accessing information on a mobile device is a convenient option for accessing timely information for employees but some of the resources that need to be accessed are sensitive in nature and need strong authentication. Location of the principle/user who wants to access the resources is also considered essential in some applications and scenarios. Several models have been proposed to incorporate spatial constraints into Role based Access Control [15], but now researchers are devoting much research into finding solutions for a secure enforcement of such models. In this work, we propose a framework for a mobile role based access control system based on the model proposed by [2]. We identify the security requirements of such a system and propose cryptographic extensions using Identity Based Encryption and non-interactive Zero Knowledge proof of authentication. We propose a set of protocols from session initiation to session termination that use the cryptographic extensions to enforce the model proposed.

To establish the location securely we use near field communication. NFC allows the user of a mobile device to tie their identity to a specific location in a secure way when we have the right infrastructure in place. The proximity-constrained feature of NFC can be used while enforcing spatial constraints to the mobile RBAC systems. We present an informal security analysis describing how the protocol is able to mitigate well known attacks against authentication protocols. We also discuss some of the application level attacks on NFC based mobile phones and list out precautions that can be taken to prevent such kind of attacks.

We developed a proof of concept of our system using Java. We used some available implementations of pairing based cryptography and developed our system using that as the base. We also developed a prototype for a peer to peer communication between an SCM 3711 NFC desktop reader and an android Nexus S phone which is capable on NFC.

From the investigation done we feel that the design choices used in the framework provide robust authentication and information exchange. Identity Based Encryption provides a perfect platform for secure transfer of Information, but it lacks authentication. Authentication is a very important design consideration in a Role Based Access control system; to achieve this we propose the integration of Zero Knowledge proof of authentication into an Identity Based Encryption system. The most efficient implementations of IBE and ZKP are based on Elliptic curves; hence the infrastructure in place for IBE can be reused for ZKP and vice versa.

Future Work

Enforcing spatial constraints to RBAC access constraints is a very interesting challenge, the framework tries to address the security requirements of such a system but there is potential for research in several areas. One issue is keeping track of when the user leaves a location. We try to address the problem by enabling time restrictions for a session, but this is not a foolproof solution to the problem. A possible solution is to use the information provided by the Wi-Fi access point and other location based services that are provided by a mobile device.

We also plan to test the framework and the security protocols. The product developed at Alfred NFC [6] has the necessary communication and computational and design as the Location Device discussed in the protocol which provides a test bed for the practical implementation of the protocols.

Bibliography

[1] M. Kirkpatrick, G. Ghinita and E. Bertino, "Privacy-Preserving Enforcement of Spatially Aware RBAC," in *IEEE Transaction Dependable and Secure Computing*, 2012.

[2] M. Kirkpatrick and E. Bertino, "Enforcing spatial constraints for mobile RBAC systems," in *SACMAT '10 Proceedings of the 15th ACM symposium on Access control models and technologies*, 2010
.

[3] Nokia, "Introduction to NFC," 19 April 2011. [Online]. Available: http://www.adafruit.com/datasheets/Introduction_to_NFC_v1_0_en.pdf. [Accessed 12 march 2012].

[4] R. Verdult and F. Kooman , "Practical attacks on NFC enabled cell phones," in *Third International Workshop on Near Field Communication*, 2011.

[5] M. Roland and J. Langer, "Digital Signature Records for the NFC Data Exchange Format," in *Second International Workshop on Near Field Communication*, 2010.

[6] "Alfred NFC," Universal NFC Cloud Connect inc , [Online]. Available: http://alfrednfc.com/. [Accessed 12 March 2012].

[7] E. Bertino and M. Kirkpatrick, "Location-Aware Authentication and Access Control - Concepts and Issues," in *2009 International Conference on Advanced Information Networking and Applications*, 2009.

[8] M. Kirkpatrick and S. Kerr, "Enforcing Physically Restricted Access Control for Remote," in *CODASPY '11 Proceedings of the first ACM conference on Data and application security and privacy*, 2011.

[9] A. Fiat and A. Shamir, "How To Prove Yourself: Practical Solutions to Identification and Signature Problems," in *Proceedings on Advances in cryptology*.

[10] L. Martin, Introduction to Identity-Based Encryption, ARTECH HOUSE, 2008.

[11] J. Wang, J. Yu, D. Li and Z. Jia, "Combining Authentication with Role-Based Access Control Based on IBS," in *International Conference onComputational Intelligence and Security*, 2006.

[12] D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairing," in *CRYPTO '01 Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*, 2001.

[13] A. D. Caro and V. Iovino, "jPBC: Java pairing based cryptography," in *2011 IEEE Symposium on Computers and Communications (ISCC)*, 2011.

[14] A. Duffy, T. Dowling and L. Owens, "An Identity Based Encryption System," in *PPPJ '04 Proceedings of the 3rd international symposium on Principles and practice of programming in Java*, 2004.

[15] E. Bertino, B. Catania and M. L. Damiani, "GEO-RBAC: a spatially aware RBAC," in *SACMATSymposium on Access Control Models and Technologies*, 2005.

[16] M. C. Mont, P. Bramhall and K. Harrison, "A flexible role-based secure messaging service: exploiting IBE technology for privacy in health care," in *2003. Proceedings. 14th International WorkshopDatabase and Expert Systems Applications*, 2003.

[17] E. Bertino and M. Kirkpatrick, "Location-based access control systems for mobile users: concepts and research directions," in *SPRINGL '11 Proceedings of the 4th ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS*, 2011.

[18] I. Chatzigiannakis, A. Pyrgelis, P. Spirakis and Y. Stamatiou, "Elliptic Curve Based Zero Knowledge Proofs and their Applicability on Resource Constrained Devices," in *Eighth IEEE International Conference on Mobile Ad-Hoc and Sensor Systems*, 2011.

[19] "Introduction to libnfc," [Online]. Available: http://www.libnfc.org/documentation/introduction. [Accessed 2012 March 2012].