secure and seamless payment for wıreless mesh networks

by  
SERHAT CAN LELOĞLU

Submitted to the Graduate School of Engineering and Natural Sciences   
 in partial fulfillment of  
the requirements for the degree of  
Master of Science

Sabancı University  
August 2012

SECURE AND SEAMLESS PAYMENT FOR WIRELESS MESH NETWORKS

APPROVED BY

Assoc. Prof. Dr. Albert Levi ........................................

(Thesis Supervisor)

DATE OF APPROVAL ........................................

© Serhat Can Leloğlu 2012

All Rights Reserved

SECURE AND SEAMLESS PAYMENT FOR WIRELESS MESH NETWORKS

Serhat Can Leloğlu  
Computer Science and Engineering, MS Thesis, 2012  
Thesis Supervisor: Assoc. Prof. Albert Levi  
Keywords: Prepaid Payment Systems, Security, Micropayments, Wireless Mesh Networks

Abstract

Wireless Mesh Networks [1] is a multi-hop wireless networking technology to provide broadband ubiquitous access in metropolitan area. WMN provides flexibility for topology design and has self-organized nodes. These nodes could form routing tables and provide high-speed connection end to end. WMNs are easy to set up and they have manageable overheads.

In the WMN structure that we assume in SSPayWMN project there are mobile clients and operators, who will be charging the service they give. We assume there is more than one operator and users should be able to get service from these operators. In case of a roaming situation, service should not be interrupted and users should continue getting service without noticing operator change has occurred. Related works for broadband access usually trust operators fully, but in real life operators may unintentionally overcharge their users and these cause disputes between the customers and the operators. Even in the cases where the operator is right, it is very difficult to convince the customer since the operators generally do not have justifiable proofs that cannot be denied by the customers.

In SSPayWMN project, our aim is to design a secure payment scheme which is fair to both the operator and users. Using cryptographic tools and techniques, all system parties will make sure that they talk to the correct entity and providing/getting service in an undeniable way. We will design and implement a secure prepaid payment scheme and we will prove our system’s effectiveness by implementing our system on a network simulator. By doing so, we aim to get near real life performance results for critical use cases.

We will design our system considering our main requirements. Our requirements are to be studied under two categories. One of them is network requirements and the other one is cryptographic requirements. They are briefly explained in following sections.

KABLOSUZ ÖRGÜ AĞLARI İÇİN GÜVENLİ VE KESİNTİSİZ ÖN ÖDEMELİ MİKROÖDEME

Serhat Can Leloğlu  
Bilgisayar Bilimi ve Mühendisliği, Yüksek Lisans Tezi, 2012  
Tez Danışmanı: Doç. Dr. Albert Levi  
Anahtar Kelimeler: Ön Ödemeli Sistemler, Güvenlik, Mikro Ödeme, Kablosuz Örgü Ağlar

Özet

Kablosuz Örgü Ağları (KÖA) çok sekmeli, erişimin her yerden sağlanabildiği, yüksek hızlı ve gelişmekte olan bir ağ teknolojisidir. Kablosuz Örgü Ağları ağ yapısında esneklik sağlar ve kendiliğinden kurulabilen dinamik bir yapısı vardır. Bu ağlardaki elemanlar rota tabloları oluşturabilirler ve yüksek hızda iletişim sağlayabilirler. Kablosuz Örgü Ağları kurulması kolay sistemlerdir ve ekstra yükleri azdır.

Bizim KÖA sistemimizde mobil halde kullanıcılar ve verdikleri hizmete göre ücret talep eden operatörler var. Varsayımımıza göre birden fazla operatör olacak ve kullanıcılar istedikleri operatörlerden hizmet almakta özgür olacaklar. Operatör değişikliği olduğu durumlarda servis kesintisiz şekilde devam edecek ve kullanıcı operatör değiştirdiğinin bile farkına varmayacak. Bu alanda yapılan diğer çalışmalar genel olarak operatörlere sonsuz güven ilkesine dayanarak tasarlanmıştır. SSPayWMN projesinde biz operatörlere mutlak güven ilkesini benimsemedik ve sistemimizi bu doğrultuda kurduk. Bunun nedeni operatörlerin istem dışı ücret almalarını engellemek. SSPayWMN sisteminde bir kullanıcı parasını ödemediği hiç bir hizmeti alamaz aynı zamanda kimse kullanıcının almadığı hizmetin parasını da alamaz. Bu proje operatör için de kullanıcı için de isteklerin kriptografik yollarla ispatlanmasına olanak tanıyor.

SSPayWMN projesinde amacımız hem operatörler hem de kullanıcılar için adil bir sistem kurmatı. Kriptografik araçlar ve algoritmalar kullanarak bütün ağ elemanları kullandıkları servisi ve ödedikleri ücerti ispatlayabilme şansına sahipler. Kurduğumuz sistemin güvenli ve hızlı olduğunu bu dökümanda anlatıyoruz ve efektif çalıştığını görebilmek için network simulasyonı sonuçlarını da paylaşıyoruz. Bu sayede gerçek hayattaki uygulamaya çok yakın sonuçlar elde etmiş oluyoruz.

Sistemimizi ana gerekliliklere göre kurduk. Bu gereklilikler iki dalda incelenebilir. Bu alanlar ağ gereklilikleri ve kriptografik gereklilikler. Bu dallar dökümanın ilerleyen sayfalarında açıklanacaklar.

*To my dear Margeret*

Contents

[1. INTRODUCTION 12](#_Toc336438422)

[1.1 Contribution of the Thesis 12](#_Toc336438423)

[1.2 Organization of the Thesis 13](#_Toc336438424)

[2. Background on wireless mesh networks 14](#_Toc336438425)

[2.1 Components of WMN 14](#_Toc336438426)

[2.2 Characteristics 15](#_Toc336438427)

[3. Background on Cryptographic functions 17](#_Toc336438428)

[3.1 Hash Functions 17](#_Toc336438429)

[3.2 Hash Chains 17](#_Toc336438430)

[3.3 HMAC 19](#_Toc336438431)

[3.4 Symmetric Cryptography 20](#_Toc336438432)

[3.5 Public Key Cryptography 21](#_Toc336438433)

[4. Requirements for a secure and seamless mıcropayment scheme ın wıreless mesh networks 23](#_Toc336438434)

[4.1 Requirements of the Network 23](#_Toc336438435)

[4.2 General Overview of the Proposed Scheme 24](#_Toc336438436)

[4.3 Connection Card Structure 27](#_Toc336438437)

[4.4 Alias Computation 28](#_Toc336438438)

[5. Protocols of secure and seamless mıcropayment scheme 30](#_Toc336438439)

[5.1 Initial Authorization 30](#_Toc336438440)

[5.2 Reuse of a Connection Card 32](#_Toc336438441)

[5.3 Access Point Authentication 34](#_Toc336438442)

[5.4 Packet Transfer 35](#_Toc336438443)

[5.5 Changing Alias 36](#_Toc336438444)

[5.6 Update Packets 38](#_Toc336438445)

[5.7 Disconnection 39](#_Toc336438446)

[5.8 Distributing Access Point Public Keys 41](#_Toc336438447)

[5.9 Seamless Roaming (Payment Related) 42](#_Toc336438448)

[5.10 Seamless Mobility in Home Operator (Payment Related) 43](#_Toc336438449)

[6. Payment to the operators (settlement) 45](#_Toc336438450)

[7. Mobility and User Roles 48](#_Toc336438451)

[Network Topology 48](#_Toc336438452)

[User Roles 50](#_Toc336438453)

[User Actions 50](#_Toc336438454)

[Client Types 51](#_Toc336438455)

[Parameters 51](#_Toc336438456)

[User Mobility and Timing 53](#_Toc336438457)

[8. Results 54](#_Toc336438458)

[Access Point Authentication 54](#_Toc336438459)

[End-to-End Base Simulations 55](#_Toc336438460)

[Seamless Mobility and Roaming 56](#_Toc336438461)

[Packet Transfer 56](#_Toc336438462)

[What do these results mean? 57](#_Toc336438463)

[9. Discussion 59](#_Toc336438464)

Acknowledgements

**LIST OF FIGURES**

**LIST OF TABLES**

# INTRODUCTION

Wireless Mesh Networking (WMN) is multi-hop wireless networking technology to provide broadband ubiquitous access in metropolitan area. WMN provides flexible service like ad hoc networks, and supports this flexibility with the centralized structure of Wi-Fi. With the assistance of wireless mesh routers, WMNs are easily established. Moreover, WMNs provide high connection speeds with manageable overhead.

In this project, a secure prepaid payment scheme for broadband Internet access will be designed and developed in a simulation environment. This scheme will be particularly for Wireless Mesh Networks with multiple operators. The users in this system will be able to switch among several operators without needing to re-authenticate themselves.

## Contribution of the Thesis

Authentication, confidentiality, non-repudiation, fraud protection will be provided in our system. The users will not be able to deny using credits for the services actually obtained; the operator will not be able to charge more than the usage amount. Moreover, inter-operator settlement will be performed in a secure way such that each operator will have cryptographic proofs of use for the services that they provide to other operators' customers. In order to provide privacy of individuals, our scheme will provide untraceability such that no unauthorized entity will be able to track down a particular user.

Since the clients are mobile, they may hand over among different mesh routers (i.e. access points) of the same operators. They may also roam among different operators, not only due to coverage reasons, but also for having a better quality service. Our system aims to have seamless mobility and seamless roaming for payment purposes such that when the client gets service through a new AP or switch to another operator, authentication and authorization are not performed from scratch.

From security point of view, we aim to have mutual authentication between client and the network in our protocols. Anonymity of the clients and untraceability across different usage periods (a.k.a. unlinkability) are privacy related goals of the protocols.

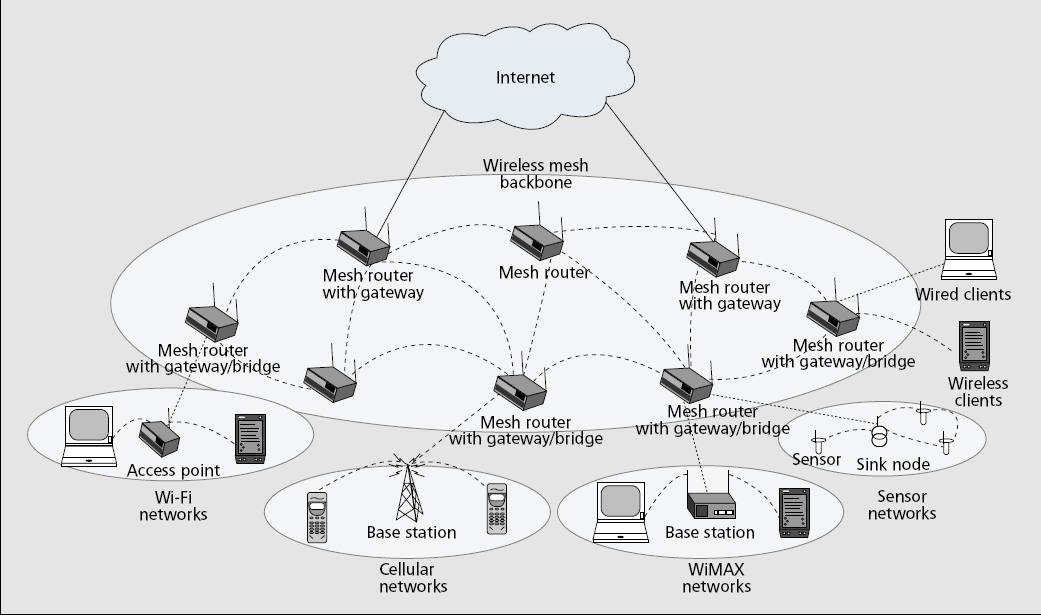
From payment point of view, our main aim is to have a fair system in which all the claimed transactions bear cryptographic proofs. In this way, the clients cannot repudiate using a service and the operators cannot claim for services that they do not provide. The latter is especially important during inter-operator settlement; it is also important to resolve client disputes.

## Organization of the Thesis

The organization of the thesis is as follows…

# Background on wireless mesh networks

Wireless Mesh Network (WMN) is multi-hop wireless networking type, designed as an alternative to traditional centralized wireless networking achieved by access points. WMNs offer flexibility and easy deployment since nodes in the network form neighborhood dynamically. Wireless mesh routers enables WMN to provide broadband access and connectivity within the network. WMNs achieve high connection speeds with reasonable delays. See Figure 1 for a sample WMN infrastructure.



Wireless Mesh Networks are a special kind of ad hoc networks. The main idea is the same, which is to provide connectivity from every node of the network without using a central node. Mesh routers do not need any access point to talk with each other they only need to be close enough. Neighborhood relationships and routing tables are formed dynamically at the setup phase of the network and even if the topology changes during the run mesh routers could adapt to new structure and form neighborhood information once again.

## 2.1 Components of WMN

There are three kinds of WMNs; they differ according to functionality of their nodes. We will briefly explain these three kinds as following.

**Infrastructural Backbone**: Figure shows the infrastructural backbone, where dash and solid lines represent wireless and wired connectivity respectively. This kind of WMN consists of mesh routers, forming up a mesh backbone to enable clients to connect to them. Internet service is provided by special mesh elements called gateways. Gateways are located within mesh backbone but they do not interfere to routing protocols or relay any packets. They are always source or the destination of the packets.

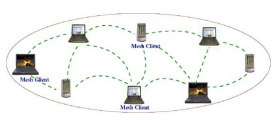
Infrastructural backbone enables clients to connect to mesh network using conventional wireless or wired technology. Mesh routers in the bakbone has bridging capabilities so it is possible to connect to these routers using Wi-Fi, WiMAX or wired networks.

This WMN type is the most commonly used one and we will be using this type in our proposed scheme as well.

Infrastructural mesh networking is useful for community and neighborhood networks. Mesh routers could be placed at the top of the buildings and provide broadband access to any client passes by or stays at these buildings.

**Client WMNs:** This kind of WMN enables peer-to-peer connection between client nodes. It is more likely an adhoc network. In this architecture clients form up the network, handling all the relaying and routing. They also provide end user applications to the customers, so there is no need for mesh routers in this structure.

A client sends a packet through a route and that packet is hopped by any intermediate node to reach to destination. The Figure shows the client WMN architecture.



**Hybrid WMNs:** As its name states this kind of architecture is a mix of the first two architectures. The Figure shows the general structure. In this networking style, a client WMN could connect to a infrastructural mesh backbone via wireless or wired communication. The infrastructural part provides broadband access to any kind of network, whereas client WMN provides adhoc communication to local clients.

## 2.2 Characteristics

The characteristics are explained as follows:

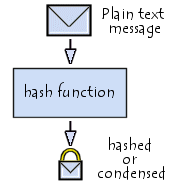
* **Multi-hop Wireless Network:** The main goal to deploy wireless mesh networks is to extend the coverage of any given network. It is available to extend the connectivity without establishing new base stations or sacrificing the channel capacity.
* **Support for Adhoc Networking:** Adhoc networking provides self-forming, self-healing and self-organization. It is fairly easy to deploy such a network.
* **Mobility:** Mesh routers do not have mobility usually, whereas clients could be mobile or stationary depending on the scenario.
* **Bridging:** Mesh networks provide service for other types of networks, it is possible to connect a Wi-Fi network to the backbone or it is even possible to connect with backbone using an Ethernet cable.

# Background on Cryptographic functions

We will give brief information for cryptographic functions. In 3.1 we will explain hash functions. In 3.2 we will describe hash chain which we have used as tokens for prepaid internet service. 3.3 will give details of message authentication code, which is also known as HMAC. 3.4 about symmetric encryption and 3.5 will give detailed information about public key cryptography.

## 3.1 Hash Functions

Hash functions are irreversible mathematical functions that generate an alphanumerical value given an input block.

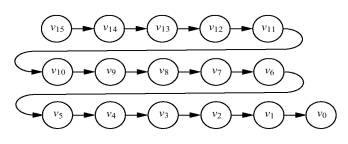


<http://en.kioskea.net/contents/crypto/signature.php3>

* Hash functions may take various sized inputs but their output length is fixed.
* Hash functions are completed in a fairly small amount of time, which makes them very eligible for security systems.
* It is fairly easy to generate a hash output but it is infeasible to find the input by only knowing the output. That feature is called irreversibleness of hash functions.
* Different inputs may generate the same output but it is infeasible to find an alternative input by only knowing a sample input. That feature is called strong collision resistance.

## 3.2 Hash Chains

Hash chains are formed by applying a hash algorithm to an initial value and using the output as an input for following hash functions. Length of the hash chain is determined by the number of times the hash algorithm is run.



<http://www.stanford.edu/class/cs259/WWW04/projects/project01/01%20-%20writeup.html>

Since hash functions are irreversible, it is easy to go forward in the chain but it is not computationally feasible to go backwards. Which means a person could find any value on the chain if he knows the initial value but this situation is not possible for a person who knows the last value on the chain.

Hash chains are formed using a hash algorithm to a result of hash algorithm several times. Each output becomes a member of the hash chain and in systems that use hash chains they chain members are used in reverse order, meaning that the last output is the first one to be used.

Hash chain with 3 elements is denoted as:

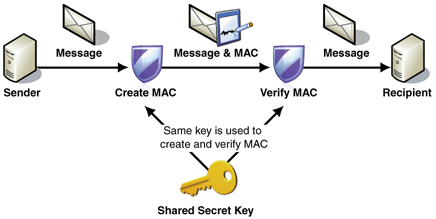
More generally, hash chain with *n* elements is denoted as:

n

One-wayness of the hash chains makes these values to be used as prepaid service tokens since it is not feasible to find the next token by knowing the previous one but it is easy to confirm next token by looking at the previous one.

Hash chains are easy and fast to compute that’s why they are widely used in cryptographic systems. Especially in SSPayWMN, we have a delicacy for delay. We cannot afford big delays for computations so hash chains are the perfect tools for us to use in our system.

## 3.3 HMAC



<http://msdn.microsoft.com/en-us/library/ff648434.aspx>

Message Authentication Codes (MACs) are used for authentication purposes. A shared secret key is used for MAC functions and this secret key authenticates two entities to each other.

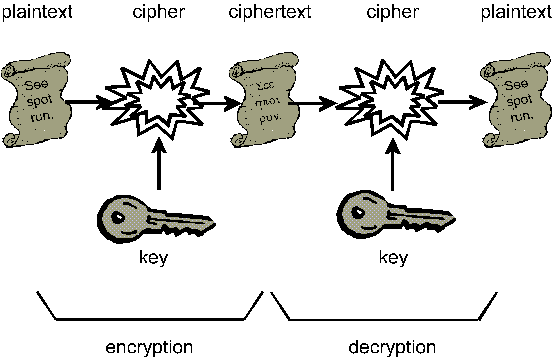
As it is explained in <http://tools.ietf.org/html/rfc2104>.

*HMAC* (*K*,*m*) = *H* ((*K* ⊕ *opad*) ∥ *H* ((*K* ⊕ *ipad*) ∥ *m*))

Where

* *H* is a cryptographic hash function,
* *K* is a secret key padded to the right with extra zeros to the input block size of the hash function, or the hash of the original key if it's longer than that block size,
* *m* is the message to be authenticated,
* ∥ denotes concatenation,
* ⊕ denotes exclusive or (XOR),
* *opad* is the outer padding (0x5c5c5c…5c5c, one-block-long hexadecimal constant),
* and *ipad* is the inner padding (0x363636…3636, one-block-long hexadecimal constant).

## 3.4 Symmetric Cryptography



<http://www.unicom.com/pw/pubnetinfosec/>

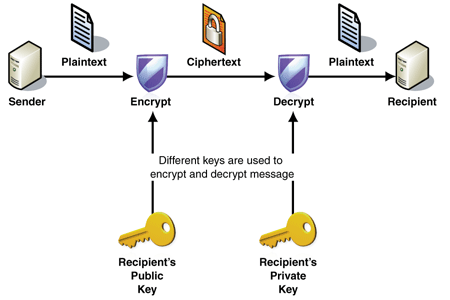
Symmetric cryptography is one of the most important ways to achieve security in science of cryptography. It requires two parties that are agreed on a single secret key. Key size depends on the algorithm but it is preferred that it should be at least 128 bits long.

In this context a party uses its secret key to encrypt a plain text message and sends it to the other party. Considering the environment the link between the two parties is assumed to be insecure. Since the message is encrypted an eavesdropper could not decrypt the message and cannot understand what it is about even if he was successful to acquire the message.

When the second party receives the message he will use the same secret key on the received cipher text. Only then the message will be able to be read.

Symmetric cryptography is the oldest way of cryptography and there are a lot of algorithms to achieve symmetric cryptography. Most recent and secure ones are AES, 3DES, Blowfish …etc.

## 3.5 Public Key Cryptography



<http://msdn.microsoft.com/en-us/library/ff647097.aspx>

Public Key Cryptography (PKC) is the mostly used cryptographic tool in this project. It is vitally important for this project to achieve authentication and confidentiality.

In PKC there are two types of keys. First of them is public key the other one is private key. These two keys are computationally related. Public key is accessible by everybody in the system whereas the private key is only known to its owner.

There are two ways to use PKC. The first one is using public key for encryption and private key for decryption. This scheme provides confidentiality.

* Sender uses recipient’s public key over a plaintext for encryption. The output is a cipher text.
* Sender sends the cipher text to the recipient.
* Recipient receives the cipher text and uses his private key to decrypt the message. The output is plain text.

An eavesdropper, who has managed to acquire the cipher text, could not understand the content of the cipher because he does not know the private key of the recipient.

The other way to use PKC is using private key for digital signature. In this scenario:

* Sender uses his own private key to sign a plain text.
* Sender sends the signed message to recipient.
* Recipient uses sender’s public key to verify the signature on the message and if the operation is successful than it means the message came from the signer indeed. Because no one in the system is able to generate that signature but only the signature owner.

RSA is the most common PKC algorithm. The key length depends on the algorithm and RSA could use different key length as well. The algorithms are called RSA-1024, RSA-2048. In SSPayWMN we used RSA-2048 for PKC.

# Requirements for a secure and seamless mıcropayment scheme ın wıreless mesh networks

For a payment scheme designed for Wireless Mesh Networks requires following attributes:

* **Wide Coverage:** Users should be getting service within a large area.
* **Roaming:** Users should connect and maintain their connection and continue to get service even while they are moving. Designed connection method should apply to different operators.
* **Seamless Connection:** Users should be able to switch between access points as they move without noticing it.
* **Seamless Roaming:** Users should be able to switch between operators as they move without noticing it.
* **Anonymity:** It should not be feasible to track down a user’s network actions from their payments (unless law enforcement requires doing so).
* **Mutual Authentication:** For preventing malicious use of network, both user and network should be mutually authenticated. Moreover, man-in-the-middle and replay attacks must be prevented.
* **Two-way honesty:** clients cannot deny that they did not take service. Operators cannot claim that they provide service more than they actually provided. These are to be guaranteed using strong cryptographic protocols.
* **Preventing Double Spending:** A payment token should not be able to be used to get more services that its value. In particular, the payment token should not be used twice or more.
* **Unlinkability:** It must not be possible to relate connection sessions of the users with other connection sessions. In this way, higher level of privacy could be provided.

## 4.1 Requirements of the Network

In SSPayWMN project, our design of Wireless Mesh Networks will not only consists of mesh backbone but also Wi-Fi clients and wired servers. Mesh backbone will basically relay the packages from clients to server to make the users able to get service.

Servers of the operators are wired and will be communicated via regular 802.3 Ethernet protocol in its local area. Mesh backbone will communicate within itself using 802.11s protocol. Clients will use 802.11a/b/g Wi-Fi protocols to connect to the access points/mesh routers.

Mesh backbone will use Hybrid Wireless Mesh Protocol (HWMP) [16] routing for finding the shortest path to the gateways.

The system will be implemented in a simulation environment using a simulation tool such as OMNeT++ [2], ns-3 [3], etc. We will measure the delay metrics using some application scenarios in this simulation environment.

## 4.2 General Overview of the Proposed Scheme

Since WMNs are service providing system, they should have support for user identification, authentication as well as authorization and accounting. In our SSPayWMN project, our main goal is to design and develop a secure payment infrastructure for WMNs that also considers users' privacy and fairness. The basics of the system model, roles, entities and requirements have been identified in Deliverable 1. As mentioned there, our system model assumes mobile clients and operators, who will be charging the service they give. The operator's mesh backbone is made of several mesh routers which are actually Access Points (APs) with IEEE 802.11s support. This backbone is connected to operator's server via a gateway. There is also a TTP (Trusted Third Party) which may be reachable through operator. These system components are listed, together their icons used in the protocol figures, in Table 1.

|  |  |
| --- | --- |
| D:\My Documents\albert\tt proje\phone.png | Mobile user (client) |
| D:\My Documents\albert\tt proje\accessPoint.png | Access Point (AP) with mesh routing capability. From now on in this document, it is called as AP, but please note that it also has routing capability. |
| D:\My Documents\albert\tt proje\cloudWithoutDots.png | Mesh backbone of the operator |
| D:\My Documents\albert\tt proje\gateway.png | Gateway (GW) that connects the mesh backbone to outer world and also to the operator's server |
| D:\My Documents\albert\tt proje\operator.png | Operator's server (OP). Keeps necessary logs and user info. |
| D:\My Documents\albert\tt proje\trustedThirdParty.png | Trusted Third Party (TTP). Payment related logs are mostly to be generated by the TTP. |

Since the clients are mobile, they may hand over among different mesh routers (i.e. access points) of the same operators. They may also roam among different operators, not only due to coverage reasons, but also for having a better quality service. Our system aims to have seamless mobility and seamless roaming for payment purposes such that when the client gets service through a new AP or switch to another operator, authentication and authorization are not performed from scratch.

From security point of view, we aim to have mutual authentication between client and the network in our protocols. Anonymity of the clients and untraceability across different usage periods (a.k.a. unlinkability) are privacy related goals of the protocols.

From payment point of view, our main aim is to have a fair system in which all the claimed transactions bear cryptographic proofs. In this way, the clients cannot repudiate using a service and the operators cannot claim for services that they do not provide. The latter is especially important during inter-operator settlement; it is also important to resolve client disputes.

The protocols detailed in this deliverable are designed by considering the abovementioned requirements. The symbols used in this document are given in Table 2.

|  |  |
| --- | --- |
|  | XOR operation |
|  | Concatenation |
|  | Encryption of using the key |
|  | Decryption of using the key |
|  | Taking hash of times |
|  | Taking HMAC of using the key |
|  | th element of the hash chain (usage order) |
|  | Public key of TTP |
|  | Private key of TTP |
|  | th Access Point or its identity |
|  | th Operator or its identity |
|  | Public key of |
|  | Private key of |
|  | Serial Number |
|  | Nonce created by entity |
|  | Previous Alias |
|  | New Alias |
|  | Public key certificate of |
|  | Initialization Vector |
|  | Timestamp |
|  | Connection Request |
|  | Disconnection Request |
|  | Roaming Request |
|  | Change Alias Request |
|  | Mobility Request |
|  | Response (used in various protocol as positive acknowledgment) |
|  | Disconnection Acknowledgement |
|  | Roaming Acknowledgement |
|  | Mobility Response |

In our protocols, we use relevant cryptographic primitives. For public key encryption and signature purposes, we use 2048 bit RSA [1]. For symmetric encryption and decryption, we use AES-128 [2]. SHA-256 [3, 4] algorithm is employed as hash function and to form hash chains. For Challenge-Response protocols we use HMAC [3, 4] algorithm.

## 4.3 Connection Card Structure

*Connection Card* is the main deed that clients buy from operators and use to get Internet service. We use a prepaid system, in which connection cards include credits as tokens. Hash tokens are generated using hash chains as discussed below. Connection cards also have unique Serial Numbers (), which are to be used for alias computation.

Tokens for getting Internet service are basically links in a hash chain. For each set of tokens, the operator picks on a random IV (Initialization Vector) and take hashes of it many times. The number of hash operations is actually the number of token in a set. For example, if the client wants a hundred hash tokens, then the hash of IV is taken hundred times. More formally a hash chain with, say, 100 tokens are constructed in the following way.

…

is the first token to use. Then we use the token in the increasing order of token index. In this way, we exploit one-way property of hash algorithms such that an attacker cannot learn the next token even if he knows the previous ones.

The operators inform the TTP (Trusted Third Party) about the associations between and corresponding so that TTP validates them as needed in the protocols.

Connection Cards are refillable with hash tokens, which are to be sold by the operators. We assume a free market strategy in the marketing of the hash tokens. The prices or campaigns related for the marketing of hash tokens are to be decided by the operators. In other words, operators would compete with each other to sell hash tokens. They also compete with each other to provide high-quality service for broadband access in the WMN since the users are assumed to have free roaming.

Serial Number, , is a 16 digit alphanumeric and case sensitive value. In this setting, we are able to support up to users.Hash tokens are to be generated using SHA-256 hash algorithm; so they are 32 bytes long.

We foresee that smart cards can be used as connection cards. A simple Connection Card with 4 KB memory could store a and more than 100 hash tokens.

## 4.4 Alias Computation

In SSPayWMN project, we use aliases as temporary identifiers for clients. Aliases change frequently using a secure protocol that we design. By changing the aliases frequently, we provide anonymity in our system to some extent. We use a mechanism in which the aliases could be computed by the clients and also by the TTP (Trusted third party).

Serial number () of the connection card is used as the base for client’s aliases. The computation process of an alias is given below:

1. Client picks a random 128-bit unsigned number and calls it his nonce, .
2. Perform XOR operation with and this nonce, the resulting value is the alias:
3. Client uses this alias whenever his identity is required.

Aliases are to be updated periodically. The related protocols will be explained later in this document, but in this section we explain how a new alias is computed using the old previous one. Basically, the client picks a new nonce and XORs this nonce with its current alias to compute the next one. More formally, in order to change previous alias, , to a new alias, , the client performs the following steps.

1. Client picks a random 128-bit unsigned number as a nonce value, .
2. Perform XOR operation with and this nonce, the resulting value is the new alias:

The nonce values used in changing the aliases are to be sent in encrypted messages to the TTP in the related protocols. Thus only the client and the TTP can relate the aliases originated from a particular .

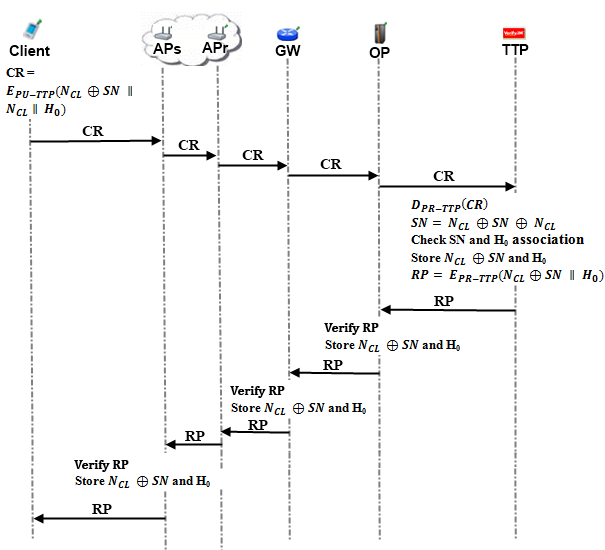
One may argue that this kind of alias computation would have a risk of causing same alias for several users. Aliases are 128-bit values; even if it is a very small possibility to have the same alias with another client at a given point of time, there is still a nonzero probability. We address this problem by making TTP to check proposed alias to be a unique alias at that point of time. This check is embedded in related protocols as will be described later.

# Protocols of secure and seamless mıcropayment scheme

## 5.1 Initial Authorization

Initial Authorization is the beginning of everything. Whenever a client buys some new hash tokens from an operator, he will need to authorize himself to TTP. Initial Authorization Protocol, shown in Figure 1, achieves mutual authentication and authorization of the user.

In Figure 1, connection between client and serving access point () is either Wi-Fi (802.11b/g) or cellular. The access point is a member of a mesh backbone and a particular access point is to be selected according to its transmission power. Since we assume all access points have same attributes, the serving access point will most likely be the closest access point to the mobile client.



Connection between serving access point and relaying access point is wireless, and uses 802.11s protocol [5].This mesh backbone is like a cloud from the mobile user’s perspective. It is a black box; which receives packets from mobile user and delivers them to the gateway in a multi-hop manner. Mesh backbone uses HWMP (Hybrid Wireless Mesh Protocol) protocol [6], which is a hybrid routing protocol. Once the mesh nodes deliver a packet through a route, they remember this route when they try to send a packet again, which will decrease packet delivery time.

Connection medium between mesh backbone and gateway () is wireless. Gateways and operators communicate through wireless medium also. The connection between an operator and TTP is either wireless using WiMAX 802.16 protocol or wired using 802.3 protocol.

Our mobile client introduces himself to the operator using *Initial Authorization* protocol. Trusted Third Party (TTP) already knows mobile user’s serial number () and first element of his hash chain (). Mobile user does not want to reveal his to any adversary because that will be used all the time; it is as valuable as mobile client’s identity. To achieve anonymity, mobile client computes an alias and use this value instead of . Mobile client will change his alias periodically as he continues to get service (related protocols will be explained later).

Initial Authorization steps are described below.

1. Client computes an alias using a nonce that he generated.

Client forms a connection request and encrypts this connection request using TTP’s public key, with RSA-2048.

(assume that connection card has 100 credits)

Client sends this to .

1. receives the connection request and relays the request through mesh backbone.
2. Gateway receives the and relays it to the operator.
3. Operator relays to TTP.
4. TTP receives connection request () and decrypts it using its private key.

TTP checks alias' uniqueness within its database of users, it would make the client start over the protocol if alias is not unique.

It computes

TTP checks and association. Store and

TTP computes

TTP sends to operator.

1. Operator receives and verifies the signature using public key of TTP.

It gets and and stores these values. The value of is the client's alias until he changes it.

Operator sends to the gateway.

1. receives and verifies the signature using public key of TTP.

It stores and .

relays to through mesh backbone.

1. receives and verifies the signature using public key of TTP.

It gets and and stores these values.

sends to the client.

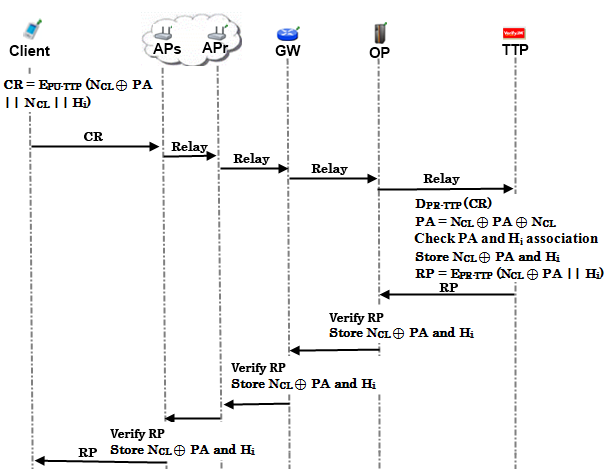
1. Client gets and understands that he is authorized to get service.

In this protocol, we assume that the packet has been sent in encrypted manner in hop-by-hop basis during its route. For the sake of simplicity, we have not shown this encryption and corresponding decryptions in the figure.

## 5.2 Reuse of a Connection Card

The users may disconnect before using up all the credits in a connection card. Reuse of a connection card protocol allows a user to connect using the remaining credits in a card. This protocol does not differ much from initial authorization protocol. The main difference here is instead of sending first hash token; client sends whichever token is the next one. Alias will change before the protocol starts. In our example case that we use below while explaining the protocol, client will send *i*th hash token and try to authenticate himself again. Another difference is that the client changes its alias during the connection. The crucial point here is that TTP should be able to update last hash value entry of the client in the database and associate it with the new alias.

Figure 2 demonstrates how the protocol actually works end-to-end.



Reuse of connection card protocol is described below.

1. Client computes a new using a nonce that he generated.

*,* where stands for *Previous Alias* of the client

Client forms a connection request and encrypts this connection request using TTP’s public key, with RSA-2048.

Client sends this to .

1. receives this connection request and relays it to the through the mesh backbone.
2. Gateway relays to the operator.
3. Operator relays to TTP.
4. TTP receives connection request () and decrypt it using its private key.

TTP checks alias' uniqueness, start over the protocol if necessary.

It computes

It checks and association and stores (i.e. the new alias) and .

It computes

It sends to operator.

1. Operator receives and verifies the signature using public key of TTP.

It gets and , stores these values.

Operator sends to the gateway.

1. receives and verifies the signature using public key of TTP.

It gets and and store these values.

sends to the .

1. receives and verifies the signature using public key of TTP.

It gets and and store these values.

sends to the client.

1. Client gets and understands that he is authenticated to get service.

In this protocol, we assume that the packet has been sent in encrypted manner in hop-by-hop basis during its route. For the sake of simplicity, we have not shown this encryption and corresponding decryptions in the figure.

## 5.3 Access Point Authentication

After authentication processes of the client with the TTP, a second authentication step begins. Client and access point will mutually authenticate each other for safe communication.

Figure 3 describes the protocol briefly.



APS authentication is described below.

1. sends a challenge request to the client which started connection.
2. When client receives this challenge request, it sends a 128 bit challenge to
   1. Client drops the packet if it is not the that he sent connection request.
   2. Client drops the packet if there was not any authentication request.
3. hashes this challenge, and uses relevant hash value (here , but it could be any if the authentication protocol runs after the reuse of a connection card protocol) as the key of :

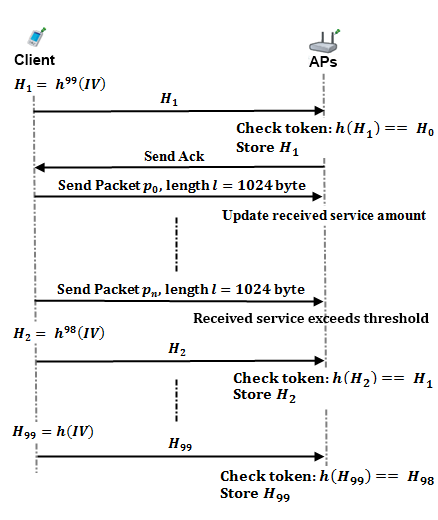
sends response to the client.

1. Client also HMACs the challenge and uses the stored hashvalue () as the key. Then it compares the result with the one that access point sent.

If it is authenticated, client starts to use access point to get Internet service.

## 5.4 Packet Transfer

After mutual authentication of client and client starts to send packets as shown in Figure 4.



1. Before sending data packet, client sends next token first. (It causes client to spend one token for each connection session.)
2. gets from the database. Remember stored client’s alias and hash token in *Initial Authorization* and *Reuse of a Connection Card* protocols.

Then:

* 1. Checks if
  2. If true sends acknowledgement () toClient and updates currently used hash value as .

1. Client sends first 1024 byte data packet .
2. serves the client without wanting any other hash token for a predefined value. This value depends on the operator.
   1. Every time client sends data packets access point will update the amount of service the user got.
   2. Whenever service amount value passes the predefined threshold client will send the next hash token.

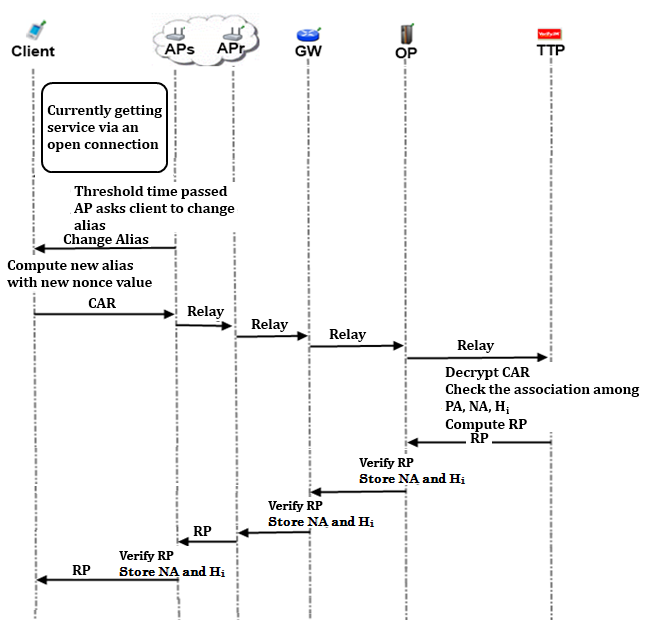
## 5.5 Changing Alias

We achieve anonymity property by using aliases, but tricky part here is achieving unlinkability. We have to change aliases on a basis that an adversary, who knows a certain client’s alias, could not be able to trace client’s activity on his home network, and also could not trace his movements among the operators or access points.

To be able to change alias in a safe way, we have to communicate with TTP but we do not want to bother TTP very often because that would slow down the entire operation due to extra delays caused. That’s why we make use of periodic timer value to make sure that access points would ask clients for new aliases after a certain period of time. Attackers or access points themselves would know that aliases are changed but would not know the mapping between old aliases and the new ones. Such a protocol is also used in Mix Networks [7].

In this way, we prevent any type of attack that would aim to analyze network traffic of access points and examine connection requests. By passing the task of enforcing alias changing to the access points, we gain a more generalized control over the clients. No attacker would understand which client wanted to change his alias, because all the clients getting service from a particular access point have requested to change their aliases at that particular time.

We need to keep actual change alias operation on the client, because client and the TTP should be the only ones who know association between an alias and a client’s .



Changing Alias Protocol is shown in Figure 5 and described below.

1. Client continues to get service, in other words uses the *Packet Transfer* protocol.

When the Alias Change Timer value expires, Access Point broadcasts "Change Alias" command to all of its clients. This times value is a system parameter; typical value is a couple of hours.

1. Client receives "Change Alias" command.

Client computes a new alias by picking up a new random nonce and computing , where is the new alias and is the previous alias .

Client forms a Change Alias Request ()

It sends this to .

1. receives and relays it to the via mesh backbone.
2. Gateway forwards to operator.
3. Operator relays to TTP.
4. TTP receives Change Alias Request () and decrypts it using its private key.

TTP checks for new alias' () uniqueness and starts over the protocol if not unique.

TTP computes

It checks and association and stores and .

It computes (

TTP sends to operator.

1. Operator receives and verifies the signature using public key of TTP.

It gets and , and stores these values.

Operator sends to the gateway.

1. receives and verifies the signature using public key of TTP.

It gets and , and stores these values.

sends to the .

1. receives and verifies the signature using public key of TTP.

It gets and , and stores these values.

sends to the client.

1. Client gets the .

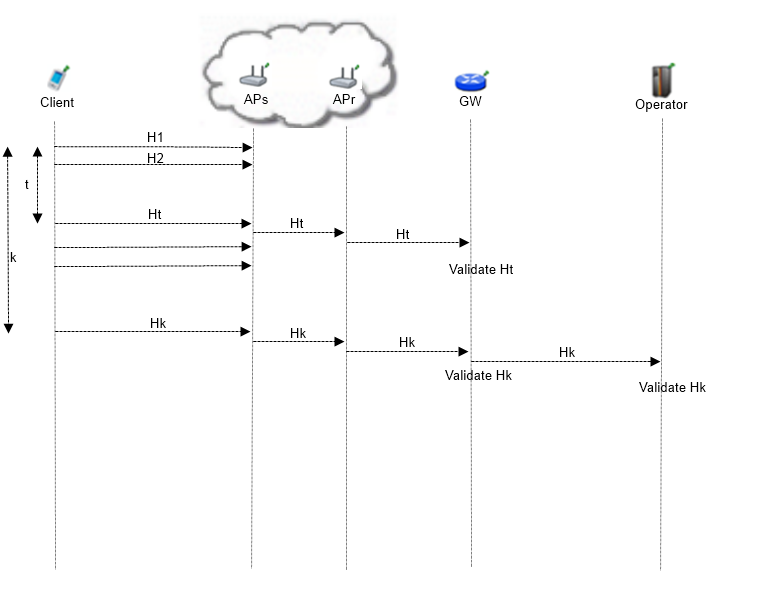
Client decrypts it using TTP’s public key and update his last used hash value and new alias.

In this protocol, we assume that the packet has been sent in encrypted manner in hop-by-hop basis during its route. For the sake of simplicity, we have not shown this encryption and corresponding decryptions in the figure.

## 5.6 Update Packets

In our usual flow, after authentication, access points do the accounting. Because of the fact that they keep the last alias and token of the client they are able to validate next token by performing hash operation to the token they kept and compare it with new coming hash token. But it is essential to send periodic updates to the operator. This is essential because we want to provide a seamless mobile communication, even when user steps out from one access point’s region to another’s. In this kind of situation, clients should authenticate themselves by showing themselves to gateway only. By doing that, we bypass operator and we can decrease authentication time significantly.

In order to use abovementioned protocol, gateways should be aware of client’s lastly used token and connection status. From security point of view, it would be ideal to update gateway entry at every time when a new token is used by the client. However, this would be very inefficient and would increase network traffic. That’s why we define threshold time values for access points and gateways. After passing this threshold time values, access points send update packets to gateways, and gateways send update packets to operator. This mechanism is depicted in Figure 6 and explained below.

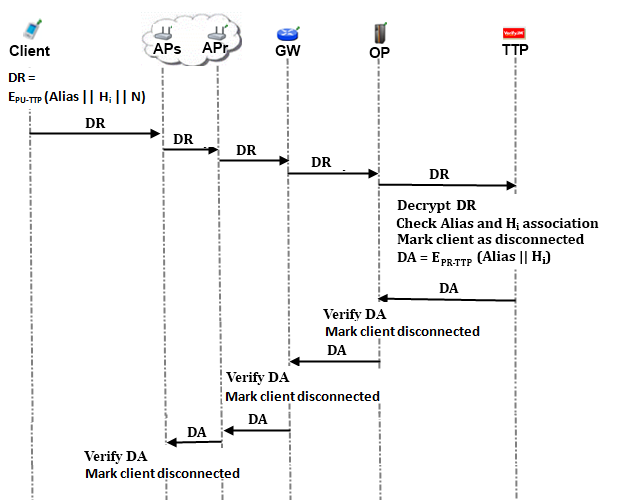


1. After client sends the first token he uses, at the current session, access point starts to count the time passed. After units of time (value of is a system parameter), access point sends lastly used hash token to the relaying access points.
2. Relaying access points forward the token to the gateway.
3. Gateway receives the token and updates the client entry. Gateway updates the last used value for the token.
4. Gateway starts to count the time passed from the lastly arrived token. After units of time (value of is a system parameter), gateway sends lastly used token to the operator.
5. Operator receives the token from gateway and updates the client entry by changing the last used value attribute with the newly received token.

## 5.7 Disconnection

To be able to run Reuse Connection Card Protocol, we have to establish proper disconnection. Our Update Packets protocol brings stability to the system in case of a connection interruption, but we assume that most of the users will be disconnecting from the operator using the disconnection protocol that we explain in this section and in Figure 7.

These protocols are designed for the sake of operators, to make them aware of how many users they are serving at a point of time. That information will bring them the opportunity to organize their servers accordingly, deciding on their marketing strategies using traffic density, etc.



Disconnection protocol is shown in Figure 7 and described below.

1. Client forms a disconnection request

Client sends it to the access point.

1. Access Point receives and prepares itself to disconnect that particular client.

AP relays to the mesh backbone, to make it reach to the .

1. receives and forwards the to the Operator.
2. Operator receives and forwards the to the TTP.
3. TTP receives the and . It checks the association between the and this hash token; if the association holds, then it computes a disconnection acknowledgement ().

TTP sends to Operator.

1. Operator receives , verifies the signature on it and marks client as disconnected.

Operator relays to .

1. receives , verifies the signature on it and marks client as disconnected.

It relays to the mesh backbone.

1. Serving Access Point eventually gets the , verifies the signature on it and disconnects the particular client which corresponds to the it received.

In this protocol, we assume that the packet has been sent in encrypted manner in hop-by-hop basis during its route. For the sake of simplicity, we have not shown this encryption and corresponding decryptions in the figure.

## 5.8 Distributing Access Point Public Keys

Achieving seamless mobility in home operator and also to support seamless roaming, we embed a public key distribution mechanism in SSPayWMN system.

In Figure 8, a generic model for public key distribution is shown. This protocol has two parts; one is certificate generation for access point public keys, the other one is distribution of the public keys. The part between operator and the TTP is offline. This part of the protocol runs during set-up, before the deployment of the access points in the field.



Distributing Public Keys algorithm is described below.

1. Operator generates public/private key pairs for the access points in its mesh backbone and embeds these keys to them before the deployment.

Operator forms an access point list (); which consists of access points and their corresponding public keys.

Operator sends this list to the TTP through a secure channel or in offline manner.

1. TTP receives the and starts to generate certificates for every access point and public key pair.

Certificates are formed as:

TTP stores these certificates for distribution.

We make use of other protocols (such as *Initial Authorization* or *Reuse of a Connection Card* protocols) of SSPayWMN for certificate distribution. Suppose an AP does not possess its certificate. In such a case whenever this access point gets a connection request it will concatenate a certificate request to the packet. When the TTP receives such a request, it concatenates corresponding certificate to the connection response. Then, TTP sends the connection response and together to operator.

1. Operator receives the connection response and the certificate and relays these packets to the access point through gateway and mesh backbone.
2. Access point gets its certificate and broadcasts it to the nearby access points. It also stores this certificate.

## 5.9 Seamless Roaming (Payment Related)

When the clients need to get service from an access point of a new operator, they roam between old operator and new one. In this kind of situations, we do not bother TTP and save time and computational power.

Because of the fact that every access point has its public/private key pairs and ability to broadcast public keys, we can handle roaming in a seamless way without running the authorization process from scratch. As it is shown in Figure 9, client gets a signed roaming ticket from its old access point and uses this signed ticket to maintain to get Internet service from a new operator.



Roaming protocol is shown in Figure 9 and described below. In this protocol, the client would like to switch from its old operator () to a new one (). In this setting, is the last access point that the client got services from . , is the access point that the client would like to continue to get services in network.

1. Client sends a Roaming Request () to .
2. receives and forms a Roaming Acknowledgement ().

sends to the client.

includes the roaming ticket that the client uses to get services from . It is signed by and encrypted for , thanks to public key distribution mechanism that we employ.

1. starts the disconnection protocol for the client after sending .

This disconnection protocol runs in parallel with the roaming protocol. Thus it does not put an extra delay in roaming. Old operator () stores disconnection acknowledgement () to support its claim to get funds for the services that it provided until roaming occurs. TTP stores the information that this disconnection is due to a roaming to in order not to get confused when disconnects without a connection request reached to it.

In this scheme, ’s signed ticket serves as a formal document, which represents the beginning of the session with .

1. Client receives and forwards it to the new operator ().
2. decrypts using its private key.

It gets the signed ticket of the . stores this signed data to use it for collecting funds from TTP.

verifies the signature over this signed ticket using ’s public key. Then, it checks in order to decide whether the ticket has expired or not.

Then, starts a challenge-response protocol with the client.

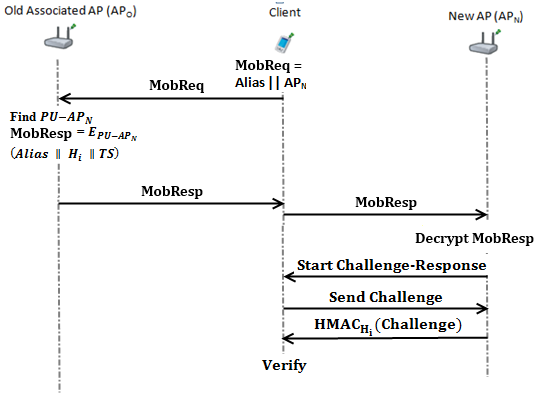
1. Client sends a 128-bit challenge to .
2. computes

sends this HMAC value to the client.

1. Client receives this HMAC. Moreover, it computes the same HMAC using . If the computed one and the received one are the same, then it authenticates .

## 5.10 Seamless Mobility in Home Operator (Payment Related)

When a client moves out of the coverage area of its associated AP or if another AP provides a better service, the client may want to hand off to another AP. For the payment purposes, we develop a seamless mobility protocol in order to avoid a full authorization process during such a handoff. This protocol is shown in Figure 10. The user sends only the ID of the new access point. This protocol uses the service of “Distributing Access Point Public Keys” protocol. With the ID of the new access point, the old access point pops up the public key, and use this public key to encrypt the credentials of the user. In this way, the necessary user and payment information to continue the service at the new AP is sent to it in an encrypted way.



In Figure 10, client moves to the new AP’s () coverage area from old AP (). Seamless mobility is described as follows:

1. Client forms a mobility request () and sends it to .
2. concatenates Alias, , timestamp , and encrypts it with the public key of . It relays this mobility response () to via client.
3. decrypts using the private key of itself and checks whether it is expired or not.
4. If it is not expired, stores and *.* Then sends a challenge request to the client.
5. When client receives this challenge request, it sends a 125-bit challenge to .
6. computes

sends this HMAC value to the client.

1. Client receives this HMAC. Moreover, it computes the same HMAC using . If the computed one and the received one are the same, then it authenticates .

# Payment to the operators (settlement)

In our system, operators claim their money from the TTP, as they show their service logs. A log proves a service that has been provided between a connection request and a disconnection request.

Operators store CRs of the clients; we described CRs in Initial Authorization and Reuse of a Connection Card protocols. When a client makes a disconnection request, operator stores the disconnection request as well. After receiving the DR, operator forms its log as follows.

Where TS stands for timestamp. We make operators to add timestamps to make TTP’s job easier.

Let’s look at what is going to happen when TTP receives these two logs from an operator.

1. TTP will sort the logs according to their TS value; this sorting process would ease the operation.
2. TTP first processes CR. CR is a request; which is encrypted with the public key of TTP. CR consists of Alias, nonce and the first hash token to be used to get service.

Consider

TTP decrypts it using its private key, and gets SN by the XOR operation:

Note that SN’s first token used is Hf.

1. TTP decrypts the Signed Connection Response using its public key, and gets the alias and the hash token. TTP compares the values with the ones in connection request. If they match, then it is a valid log.
2. The abovementioned log is only a service starter; operator needs to show service ending log to claim its money from the TTP.

Service ending log naturally has a larger TS value; that is why that log comes later in the sorted list of logs.

TTP takes the ending log and decrypts DR using its private key.

TTP gets Alias, nonce and hash token from the decrypted DR. TTP makes the XOR operation: and gets the SN. Note that SN used is the hash token came with the DR to end the service.

1. TTP takes the Signed Disconnection Response and decrypts it using its public key. TTP gets the alias and the hash token from it, and compares the values with the ones came with the DR. If the values match, TTP considers the log as a valid service ending log.
2. After validating the logs, TTP performs the hash operation over service ending hash token until it reaches the service starter hash token. TTP counts these hash operations. This count is mapped to funds for the provided service.

Now, let us analyze a sample misbehaving case where a client:

* gets service from his home operator between and
* gets service from a foreign operator between and
* gets service from his home operator between and

In this type of situation home operator has two CRs and DRs. Whereas foreign operator has one CR and DR. Below are the logs of home operator:

We see that home operator served between and also served between and . Home operator would want to take money for serving between and . It could pretend that it has served the client between and by not sending and . Because indicates that client is disconnected from the operator at and indicates that Client started to get service from the operator at . By sending only and home operator tells TTP that it served the client between and . Operator would want money for serving 30 hash tokens.

In this scenario, the foreign operator, which has served between and , has the following two logs.

Foreign operator proves that it has served between and by showing the signed RP and DA.

TTP would see that it has already paid home operator for service to that particular client between and . This means that home operator tricked TTP to pay more.

In our system, TTP is the one who has the authority. It pays operators their earnings. If it finds an operator misbehaving, it could give a penalty to the operator and do not pay for future services, or there could be several other kinds of penalties, since TTP has the proof it could bring the subject to the court as well.

# Mobility and User Roles

We aimed to be close to reality in SSPayWMN simulations. That understanding would ease the process, implementing our project in real life. Ns-3 is a very successful network simulator and it is proven to be close to reality but there is still work to do considering network topology and properties system entities.

In our simulations we will implement before mentioned system entities. Mobile clients, mesh routers, gateways, operators and trusted third party. Network topology will be suitable for real time situations. Users will not be considered as same, they will be part of a group in which group members shows similarities but again they will not be exactly same.

In results section there are real time scenario results and also unit test. Unit test will show protocols’ performance under no stress. Real time scenario will show how our protocols would respond to any real time situation and how their performance would be.

In unit tests there is only one client whereas in real time simulation there are 300 mobile clients.

## Network Topology

Our simulations are done for a metropolitan area, which is 1 km2. Access points which happen to be mesh routers at the same time are located on a grid. That grid has 100 meters of step length. Figure 1 shows the mesh router topology briefly.

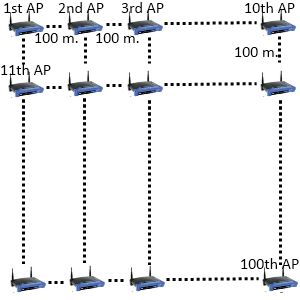


Figure 1 Mesh Router Topology

As seen on Figure 1 mesh routers are put at the corners of a grid with 100 meters between each neighbor. That topology will lead us to implement a Manhattan scenario [1] in which every user could move on a grid.

Mesh backbone will use 802.11s protocol and provide high speed communication. Every mesh router will be neighbor with maximum 4 other mesh routers. Some mesh routers will be close to a gateway which would lead us to have mesh routers with maximum 5 neighbors.

Gateways are located inside the mesh backbone. They will be accessible via mesh routers. We aimed to make a mobile client to be able to communicate with a gateway through maximum 3 hops. That is necessary for speed and throughput, because mesh backbone is the most important and critical part of the network. Performance of wired and Wi-Fi technologies are proven to be efficient and fast. Communication between gateways and operators is wired. Mobile clients communicate with access points/mesh routers via Wi-Fi, so these parts of the network do not add a considerable overhead. Our experience from the simulations showed us, after 5 hops mesh network could drop the packet or bring an unacceptable delay to packets. That is why we kept gateway numbers high as they are close to half of the mesh router count.

In our real life scenario simulation we have 300 mobile clients, 100 access points, 50 gateways, 2 operators and 1 trusted third party.



Figure 2 Network Topology

Dotted lines indicate wireless communication and straight lines indicate wired communications. Wireless communication is either 802.11s or 802.11b/g. Wired communication uses 802.3 protocol.

## User Roles

In our secure and seamless prepaid system for wireless mesh networks, we intend to serve a large variety of users. Network clients differ in behavior considering their network usage frequency, time and the fact that if they are roaming often or not.

For the sake of simplicity we define user roles and corresponding behaviors for these user types. To achieve randomness in client behavior we will define probabilities for certain kind of actions and these actions will have discrete uniform distribution. Which means, an A kind of user will use packet transfer protocol only for %n, and perform roaming for %(100-n). That is the case if we have only packet transfer and roaming in our action list. We will define what the actions in following chapters are.

A critical part of our simulations is defining the network client types. We thought of differing the clients considering their jobs. Jobs of clients will affect their behavior in the network, and of course they will always be likely to perform their more common actions but it is always possible for a network client to perform unexpectedly because of the randomness of the system.

We will simply perform uniform distribution over the actions, and random variables will be used to decide which action will be performed next by the client. Client will have a more common action, comparing to other actions. Common action will have a bigger probability to happen and random variable will most likely to come out in the area of common action probability area. But again it is possible to perform an unexpected behavior even if its probability is %1.

## User Actions

Actions are basically the protocols described before, we have a protocol for every kind of action.

Some protocols serve to the same action together like “Initial Authorization” and “Reuse of a Connection Card”. A user initially authorizes self once and then every time he starts to get service again he will use the “Reuse of a Connection Card” protocol.

The actions are described as follows

* **Become Active:** A user could be inactive for hours or days no matter, but he/she should become active and start to get service as he/she pleases.
* **Disconnect:** Also could be called becoming inactive. A user should be able to disconnect and stop getting service when he/she closes the device running or run the disconnection protocol.
* **Packet Transfer:** It is basically getting Internet service from the access points. It covers only getting service from the same access points for some time. This protocol only uses Wi-Fi 802.11 b/g to communicate.
* **Roaming:** Roaming is starting to get service from a foreign operator’s access point. Every access point broadcasts its operator, which the access point belongs to. A user does not use a foreign access point until he/she is far from a home access point. Roaming is performed a lot by the clients, who likes or obliged to travel more.
* **Seamless Mobility:** The clients, who travel from a place to another close place, use seamless mobility. Access points usually have 100 meters between one to another. So if a client is walking through a long road or traveling with a car he would be using “Seamless Mobility” protocol. That protocol would be highly used during the rush hours when people are waiting the traffic to dissolve.

## Client Types

We have thought to divide users using their jobs. We will define parameters for the probabilities that clients will perform which action in which percent.

User types are described as follows:

* **Students:** This kind of clients will get service in the morning and in the afternoon; because of the fact that they are going to school they do not have so much time to spend in Internet. We thought to make this kind to be active in the morning with a high probability, and be inactive during the school hours. Students will be active again when they are out of school and they will begin to use “Seamless Mobility” because they will travel to their homes. Until the midnight students will continue to get service from their homes in an immobile way. Of course the above mentioned scenario is most likely to happen scenario, it is possible for a student to go out with his/her friends and use “Seamless Mobility” and “Packet Transfer” protocols all the time.
* **Daily Workers:** This kind of clients is the easiest to handle clients. As day starts they begin to get service in an immobile way and then they go to work. Until the working hours are done they continuously use “Packet Transfer” protocol. When they return home they authenticates themselves and start to use “Packet Transfer” again until they sleep.
* **Non-Workers:** This type of users is meant to get Internet service in an immobile way, from a laptop or a PC. Because of the fact that they are immobile and do not leave their houses a lot they do not use “Roaming” and “Seamless Mobility” a lot. They would use “Packet Transfer” usually.

## Parameters

Every action A in our action set AS:

Equation 1 holds for every kind of user but distribution of the probabilities differ for user types. Ai would have more probability value from Ai+1 for students; on the other hand Ai+1 may be more likely to happen then Ai for business people.

Let’s take students as an example for early hours for the day.

Students would have the probability distribution such as:

If a student becomes active he/she would check the Internet for a little while and disconnect afterwards. For this time period students would most likely to use “Packet Transfer” only. In case of student become active:

Note that these probabilities are guessed only and could be played with and optimized.

Considering these probabilities a student will be active with probability of %70, and then use “Packet Transfer” for probability of %60. In every half of a second client node will decide what to do next. We will have a threshold time value to prevent the occurrences of becoming active and immediately disconnecting. After disconnection the same threshold value will hold also.

As one can see “Seamless Mobility” has a very small chance to occur for a student in the morning. That is so because we assume the student to be at his/her home and immobile relative to an access point.

These probabilistic values would change for a student by the time he/she gets away from the school. Please note again we will give preliminary and guessed values for probabilities.

By the time a student gets home he/she would behave as follows:

If a client were to become active he/she would behave as follows:

Considering these values a student is more likely to get service from the access point by using “Packet Transfer”. Student has a small possibility to move and change access point he/she is getting service from.

All users’ probabilistic values in a nutshell could be shown as:

becomeActiveProbNonWorker = {40, 60, 60};

becomeActiveProbStudent = {20, 20, 80};

becomeActiveProbWorker = {20, 99, 20};

stayActiveProbNonWorker = {90, 98, 80};

stayActiveProbStudent = {30, 20, 98};

stayActiveProbWorker = {30, 99, 20};

The notation prob = {x, y, z} indicates three part of a day. x stands for first part y is the second and z is the third part of the day. The names for the probabilistic values are self explanatory. For example stayActiveProbNonWorker probabilistic array consists of probabilistic values for the non-worker group. First value is 90; which means an active non-worker at the first part of the day has a %90 chance that he/she will stay active at the next minute.

## User Mobility and Timing

In our simulations clients are able to move from one place to another according to their speed. The time and direction of their movement is random but chances are affected by user roles. For example when school is over, a student is most likely to move towards his/her target destination at a metropolitan city as it is shown on Manhattan scenario.

Considering user roles and their probabilities clients are assigned random movement times. Clients are also assigned a random target access point. The client moves from its current access point to the target access point on the grid. As it goes on its way, if there is a continuing internet service going on client forms up a new connection with the access point which is closest to client’s current location.

Depending on the new access point’s affiliated operator this movement results to *Seamless Mobility* or *Roaming* protocols. If new access point’s affiliated operator is same with the one client uses than it means client will perform a *Seamless Mobility* protocol. Client will run *Roaming* protocol otherwise. Differences between *Seamless Mobility* and *Roaming* are mentioned before.

Clients are assigned random speeds and these speeds vary from 1 km/h to 6 km/h. The clients are thought to move without a motor vehicle.

# Results

We have completed unit simulations of the protocols. Unit simulations consist of one user trying to get internet service using our secure prepaid system.

## Access Point Authentication

Figure 3 shows the performance of *Access Point Authentication* protocol.

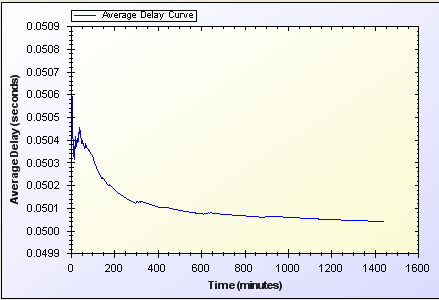


Figure 3 Access Point Authentication Unit Simulation

Access Point Authentication simulation is performed for a node, authenticating itself to an access point every minute.

In our simulation we see that around a 0.05 seconds delay we can support access point authentication.

Another important result is that we do not have unreasonable and unexpected rises or falls for delay. The average delay shows variation at the beginning of the simulation but it gets a balance in time and shows a stabile behavior.

## End-to-End Base Simulations

What we mean from base simulation is that some protocols are the core of our system like *Initial Authorization* or *Reuse of a Connection Card*. These protocols are very similar to each other considering the length of packages sent client to server and vice versa.

Figure 4 shows the performance of *Initial Authorization*, *Reuse of a Connection Card*, *Disconnection* and *Change Alias* protocols.

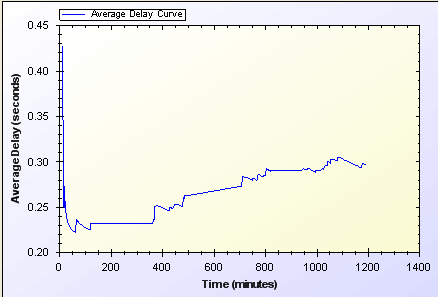


Figure 4 Initial Authorization, Reuse of a Connection Card, Disconnection, Change Alias

We have performed a unit simulation for a client sending encrypted requests to operator every minute.

The chart starts with a delay that shows variation between 0.43 and 0.25. This unstable behavior is caused by different initial packet delays. System needs some packets to set up paths between mesh nodes. That’s why we have this kind of variation between initial results but then it is clearly seen that we get to a stabilized behavior over time.

In our simulation we saw that first packets have close to 0.5 second network delay but eventually it reaches a reasonable average delay. We have a 0.3 seconds delay for end-to-end communication for above mentioned protocols.

## Seamless Mobility and Roaming

*Seamless Mobility* and *Roaming* protocols are very crucial for our system. We need a reasonable performance for these protocols because it needs to be seamless to the user. Mobile devices should decide themselves where to change access point.

Figure 5 shows the performance of these simulations.

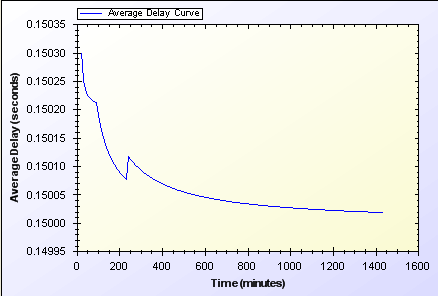


Figure 5 Seamless Mobility, Roaming

*Seamless Mobility* and *Seamless Roaming* protocols have the same behavior since client sends same length of packets and receives the same as well.

We have performed a unit simulation for a client changing access points every second. The difference between seamless mobility and roaming is the serving operator of the new access point is the same with the old access point or not.

In our simulation we saw that we have 0.15 seconds of network delay for access point change.

## Packet Transfer

Packet transfer will be the mostly used protocol in our system. That’s why we need a very small amount of network delay for this protocol.

Figure 6 shows the performance of this protocol.

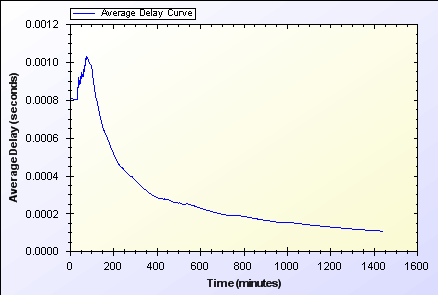


Figure 6 Packet Transfer

At the beginning of the simulation we have a higher network delay because of system set up. Access points talk to each other to see who is around and receive neighbor access point beacons.

Packet transfer unit simulation shows a user sending a packet to an access point every minute.

Our simulation shows that we could handle this request in a very short amount of time close to 0.0002 second.

## What do these results mean?

As one can see probabilistic values would change for the same type of client time to time. All of the probabilistic values are changeable and will be optimized soon.

Out uniform probability distribution model enables us to simulate real time scenarios in simulation environment, and get results closer to real time situations.

In our simulations we will have user types and their total numbers as parameters. We will also add randomness to the system, to provide different outcomes from the same simulations. The average of those simulations would cover even the most unexpected situations and this attribute of the simulations will help us to handle every possible state of the system.

Our results show that the SSPayWMN system works in very reasonable amount of time. It is secure and fast. Unit simulations set an example for how the system will behave in empty hours. We are very happy about our simulation results they show that our hard work paid up and gave good results to us.

# Discussion

In this deliverable, we designed the protocols to be used in the SSPayWMN protocol. With these protocols, the following security requirements are met.

*Roaming/mobility*: Reuse of a connection card is possible after attempting first connection. Roaming is supported, when our protocol is implemented in participating *AP*s, and tokens are valid.

*Seamless connection*: Mobility of the users in home operator is supported. The clients in the same operator can move from one *AP* to another by carrying necessary credentials in a secure way and without a new authorization and service interruptions.

*Seamless roaming*: Mobility of the clients from one operator’s zone to another is also supported without service interruptions and without needing to get authorized from scratch.

*Anonymity*: For legal purposes users must give their identities to connection card issuer (*TTP*) for getting connection cards. Therefore, as far as *TTP* keeps clients’ identities secret, users can stay anonymous.

*Mutual authentication*: We have seen how the client is authenticated. Valid token information is received by the *AP*, and with the challenge-response protocol both AP and the Client is mutually authenticated.

If there is an adversary between AP and the Client that intercepts the packet transfer between these two entities, in initialization phase, he can behave like the client. After the authentication phase, the adversary gets service from the Operator. Without getting service, client does not send the next token. Hence, client only loses two tokens in this situation; first is for establishing connection, second is for packet transfer.

If the client is already authenticated, and while sending next token if the packet is captured by the adversary, because of the lack of the Serial Number knowledge, it is not usable by him.

No ultimate trust to operators: In our scheme, users control their balance in the connection cards. Operators cannot generate tokens and it is not possible for the operators to retain unused tokens. Hence, they cannot cheat the users by saying “the token is already used”.

*Three-way honesty*: Since the tokens are issued by *TTP*, only the *TTP* and connection card holder knows all the tokens that are related with a specific connection card. Hence whenever a Client sends a new token, it is not possible for him to say “I did not use it”. Since *TTP* is a trusted third party, in the roaming phase, operators cannot say that they provided service for non-used tokens.

*Preventing double spending*: All the connection card information is stored in the database with *In Use* field. Therefore it is not possible for two users to use the same connection card at the same time. Since the last token information is stored in the database, it is not possible to double-spend a token.

*Unlinkability*: Our protocols provide unlinkability by changing aliases periodically. During the time of validity of a particular alias, the connection of a client can be traced, but once the alias has changed using the designed protocols, the client cannot be traced. The period of time to change the aliases is a system parameter.

With this document, we have finished the planned tasks of the first year of the SSPayWMN project. Next stage is the implementation of the protocols in the selected simulator (ns-3). We will also define the user profiles and parameters for the simulations. Finally, we will run the simulations and make the performance analyses. We expect to finish all these tasks and the project until the end of March 2013, as planned.

# References

1. Chin, Daniel Smith, Richard A Traffic Simulation for Mid-Manhattan with Model Free Adaptive Signal Control, Applied Physics Laboratory Laurel