Contents

[1 Summary 2](#_Toc321066281)

[2 Introduction 2](#_Toc321066282)

[3 Cryptographic Notes 3](#_Toc321066283)

[4 Connection Card Structure 3](#_Toc321066284)

[5 Alias Computation 4](#_Toc321066285)

[5.1 Alias Check for Uniqueness 5](#_Toc321066286)

[6 Initial Authorization 6](#_Toc321066287)

[7 Reuse of a Connection Card 9](#_Toc321066288)

[8 Distributing Access Point Public Keys 11](#_Toc321066289)

[9 Access Point Authentication 13](#_Toc321066290)

[10 Packet Transfer 14](#_Toc321066291)

[11 Changing Alias 15](#_Toc321066292)

[12 Roaming 17](#_Toc321066293)

[13 Update Packets 19](#_Toc321066294)

[14 Disconnection 21](#_Toc321066295)

[15 Seamless Mobility in Home Operator 22](#_Toc321066296)

[16 Payment to the Operators (Settlement) 24](#_Toc321066297)

[17 Conclusion 26](#_Toc321066298)

[18 References 28](#_Toc321066299)

# Summary

Wireless Mesh Network (WMN) technology ispromising multi-hop, ubiquitous and high speed networking technology for metropolitan broadband wireless access. Being a service providing system, payment is an important component of WMN structures. In our project, namely *SSPayWMN*, we will design and implement a Secure and Seamless Payment scheme for Wireless Mesh Networks. Security of the system developed is not only confidentiality and integrity of the transmitted messages but also anonymity of the users getting broadband access service from the WMN. Moreover, fairness of the payment scheme and enforced honesty of the participants are also important design issues. Seamless handover among the gateways and the operators in the above secure and fair setting will also be provided in SSPayWMN. The implementation and the performance evaluation of SSPayWMN will be performed in a network simulator environment.

In this document, we provide the design of the cryptographicprotocols to be used to achieve our security goals under the light of the system and user requirements previously described in Deliverable 1.

# Introduction

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 studies for broadband access usually fully trust the operators, 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; which are wide coverage, roaming, seamless connection, seamless roaming, anonymity, mutual authentication, two-way honesty, preventing double spending, and unlinkability.

The symbols used in this report are given in Table 1.

|  |  |
| --- | --- |
|  | XOR operation |
|  | Concatenation |
|  | Encryption of X using the key K |
|  | Decryption of X using the key K |
|  | Taking hash of X n times |
|  | Taking HMAC of X using the key K |
| SN | Serial Number |
|  | Nonce created by entity X |
| CR | Connection Request |
| DA | Disconnection Acknowledgement |
| Req | Request |
| RP | Response |
| LHV | Last Hash Value |
| IV | Initialization Vector |
| TS | Timestamp |
| DR | Disconnection Request |

Table 1 The Symbols

# Cryptographic Notes

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.

# Connection Card Structure

Connection Card is the main deed that clientsbuy 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. An operator will decide on a random IV (Initialization vector) and take hashes of it many times. The number of hash operations will be the decision of the client because if the client wants a hundred hash tokens then hash algorithm will run hundred times.

The last hash token will be the first token to use because we want to exploit one-way attribute of hash algorithms. A misuser will not be able to guess the next hash token unless he/she knows the previous one.

Connection Card will hold number of hash tokens and will be able to erase them as hash tokens are used.

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 themarketing 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.

# 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 will use a mechanism that aliases could be computed by the clients and also by the TTP (Trusted third party).

Serial number (SN) of the connection card, which is bought from the TTP, will be used as a base for client’s aliases. An example alias will be computed by performing following operations:

1. Client will pick a random 128-bit unsigned number and call it his nonce .
2. Perform XOR operation with and his nonce,
3. Client will use this alias whenever his identity is required.

One may argue that this kind of alias computation would have a risk of causing same alias for several users. We address this problem by making TTP to check proposed alias to be a unique alias at that point of time.

## Alias Check for Uniqueness

Clients change their aliases periodically. 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 do not want multiple users using same alias at a given time, because it would complicate the processes and may cause some errors.In order to avoid this unwanted situation, we develop asub protocol for uniqueness check in which the TTP makes this control.

Figure 1 shows thissub protocol, namely Alias Changing Round (ACR). ACR will be incorporated into authentication protocols to secure alias changing and also alias authentication, as will be discussed later.

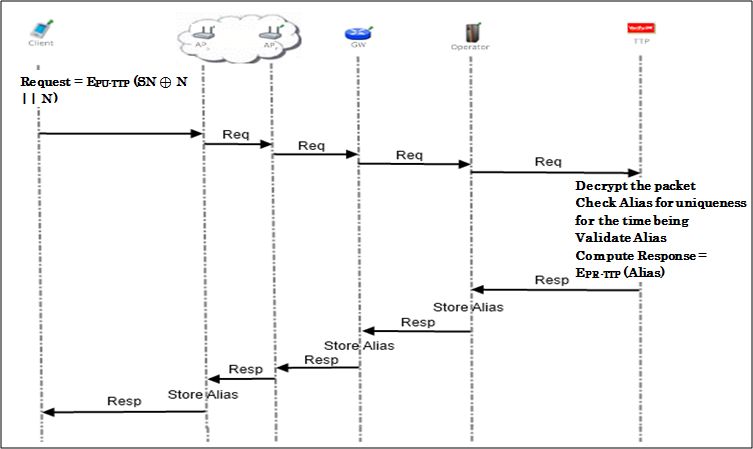


Figure 1 One Alias Changing Round

Figure 1 shows one round of ACR. In the case of there is another user using the proposed alias, TTP will detect it and will not validate the alias. In this case, TTP declines the alias and asks the user to form a new alias. This would cause a second round of ACR.

Because of the fact that alias is a 128-bit value, it is almost impossible to choose the same alias with another user at a given time, that’s why most of the time one round of ACR will be able to change or select an alias.

ACR is described below.

1. Client forms a request:

Client sends that request to the serving access point.

1. Serving access point forwards this request to the mesh backbone to make the request reach the gateway.
2. GW receives the request and relay the packet to the operator.
3. The Operator receives the request and relay the packet to the TTP.
4. TTP receives the request.

TTP decrypts it with its private key. Check alias’ uniqueness at that point of time.

If alias is not unique it will send a decline notice to the client and client will repeat the steps 1 to 4, but most likely alias will be unique and TTP will compute.

TTP sends it to the gateway, which forwarded the request in the first place.

1. GW receives the response and decrypts it using TTP’s public key.

It stores the alias.

GW sends response to the mesh backbone.

1. Serving Access Point receives the response and decrypts it using TTP’s public key.

It stores the alias.

APSsends response to the client.

# 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 will achieve a mutual authentication and authorization of the user.

In Figure 2, 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.



Figure 2 Initial Authorization

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.Oncethe 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. Access Points (APs) deliver packets in a multi-hop manner. Higher hop count means higher delay.

Connection medium between mesh backbone and gateway (GW) 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 hisalias periodically as he continues to get service.

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.

Client sends this to .

1. generates an entry for the client in the database:

refers to the IP address of the client. For the initialization phase, Status is set as not authenticated. is null at this point, and we will receive it from home operator in the reply packet. and will be explained in Section 4.6 (Packet Transfer). field is for preventing double use, and Denial-f-Service attack. It is set as true until the end of period or a disconnection request is received. routes to its gateway, through relaying access points ()in the mesh network.

1. Gateway creates client info in its database:

All the information is same with the , except the *Last AP IP* field. This is for routing reply packet from operator to the serving AP. Moreover, it is used for a seamless connection while a client is moving to another coverage area. After storing the necessary information, gateway relays packet to the operator.

1. Operator does the same operation with gateway, i.e. it creates a client entry:

All of the information is same with gateway. Operator could not tell if the authorization will fail or succeed, it could not decrypt the connection request (CR) since it is encrypted with TTP’s public key.

Hence operator relays to TTP to ask if the client is valid or not.

1. TTP receives connection request () and decrypt it using its private key.

TTP checksalias' uniqueness, start over ACR if necessary

It compute

TTP checks and relevancy. Store and

TTP computes

TTP sends to operator.

1. Operator receives and decrypts it using public key of TTP.

It will get and and store these values. The client will use the values resulting from XOR operation as an alias until he will change his alias.

Operator will send the to the gateway.

receives and decrypt it using public key of TTP.

It will get and and store these values. The client will use the values resulting from XOR operation as an alias until he will change his alias.

will send the to the client.

1. Client will get the and understand that he is authorized to get service.

# Reuse of a Connection Card

Reuse of a connection card 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. In our example case that we use below while explaining the protocol, client will send 28th hash token and try to authenticate himself again.

Figure 3 demonstrates how the protocol actually works end-to-end.

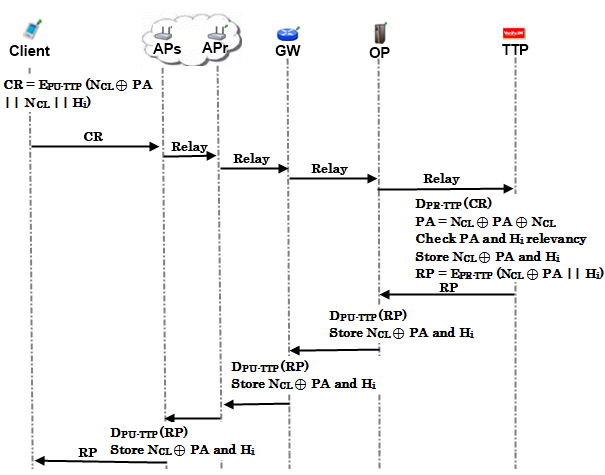


Figure 3 Reuse of a Connection Card

The crucial point here is that TTP should be able to update last hash value entry of the client in the database.

Reuse of connection card protocol is 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.

Client sends this to .

1. updates the entry for the client in the database:

On this entry, AP will update , attributes of the entry.

1. Gateway updates client entry in its database:

On this entry, gateway will update , attributes.

1. Operator relays to TTP to ask if the client is valid or not.
2. TTP receives connection request () and decrypt it using its private key.

TTP checks alias' uniqueness, start over ACR if necessary.

It compute

It checkandrelevancy and storeand

It compute

It send to operator.

1. Operator receives and decrypts it using public key of TTP.

It getsand and store these values. The client will use the values resulting from XOR operation as an alias until he will change his alias.

Operator sends the to the gateway.

1. receives and decrypt it using public key of TTP.

It getsand and store these values. The client will use the values resulting from XOR operation as an alias until he will change his alias.

sends the to the client.

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

# Distributing Access Point Public Keys

Achieving seamless mobility in home operator and also to support roaming, we embed an on-demand public key distribution mechanism in *Initial Authorization* and *Reuse of a Connection Card* protocols.By doing so, we will be able to provide seamless mobility in every possible roaming scenario, and control the extra load onthe traffic by doing it on-demand.

In Figure 4, a generic model for public key distribution is shown. As can be seen in this figure, public distribution is an add-on to the *Initial Authorization* and *Reuse of a Connection Card* protocols. If the public key is distributed within Initial Authorizationprotocol, then first token in the hash chain is used . If the protocol in which the public key distribution is embedded isReuse of a Connection Cardprotocol, then the hash token of could be one of hash tokens in the chain.

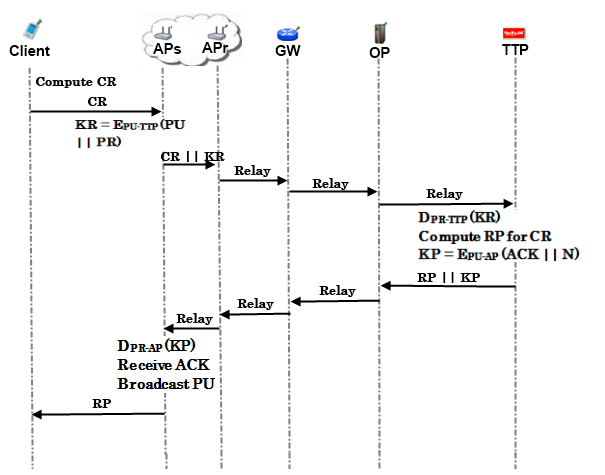


Figure 4 Distributing Access Point Public Keys

Distributing Public Keys algorithm is described below.

1. Client forms a connection request as described earlier and encrypts it using TTP’s public key.

Client sends it to Serving Access Point APs

1. APs receives that connection request and if it does not have a public key adds a Public Key Request (PKR) to the CR and sends it to TTP.

PKR is computed as:

1. GW receives the and relays it to operator.
2. Operator receives CR and it sees that there is a PKR came along with this CR.

It checks the relevancy of SSID and OpID, which is its own id, and checks if requesting access point does not have a public key.

If it finds out APs is already given a public key

# Access Point Authentication

After validation of Reply Packet received from operator, starts a challenge-response protocol for mutual authentication between client and . This protocol is depicted in Figure 5.



Figure 5 Access Point Authentication

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 the
   1. Client drops the packet if it is not the *AP* 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 ) which is stored as *Last Hash Value* () as the key of HMAC:

sends response to the client.

1. Client also hashes the challenge and uses the stored *Hash Value* () as the key of HMAC. Then compares. If it is authenticated, client sets the as *authenticated*.
2. Instead of adding a new step to the challenge-response protocol, as soon as gets the next token from the client, it sets the *Status* of the client as *associated*.

# Packet Transfer

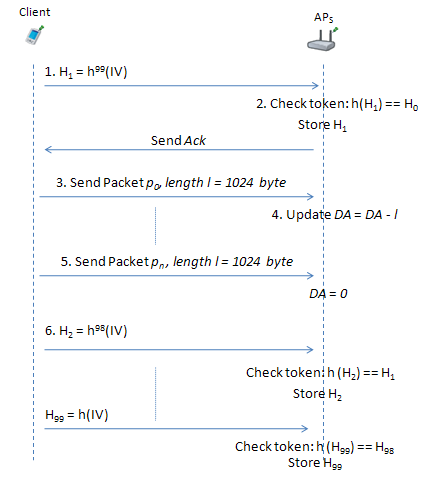


Figure 6 Packet Transfer

After Mutual authentication of client and , client starts to send packets as shown in Figure 6.

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 *Client Entry* in the database. Then:
   1. Checks if
   2. If true sends acknowledgement (*Ack*)to *Client*, and sets *Client* as *associated*
3. Client sends first 1024 byte data packet *p0*.
4. decreases*Decrement Amount* () with the length of the packet (here 1024 byte). When client is first authenticated was set as the *Service Cost* of the .
5. value is also stored in client’s database. Therefore, client proceeds with sending data packets until value drops to 0.
6. Client sends next hash value (*)*.
   1. If goes below 0, with every received packet increments *Packet Exceed* field by packet length. Hence, client could proceed with sending packets for a while.
   2. When is received and validated, is set to *Service Cost* of the and *Packet Exceed* value is decremented from . Then *Packet Exceed*is set to 0.
   3. If *Packet Exceed* valueexceeds a predefined limit. Client is set as *Not Authenticated,* and all the packets received from this client will be ignored.
7. Every time goes below 0, client sends next token.

# 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 often because that would slow down the entire operation due to extra delays caused. That’s why we define a threshold time value to make sure that access points would ask clients for new aliases after a certain period of time. Attackers or access points themselves will know that aliases are changed but will not know the mapping between old aliases and the new ones. This protocol is used also 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 “Change Alias” task 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.

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 .

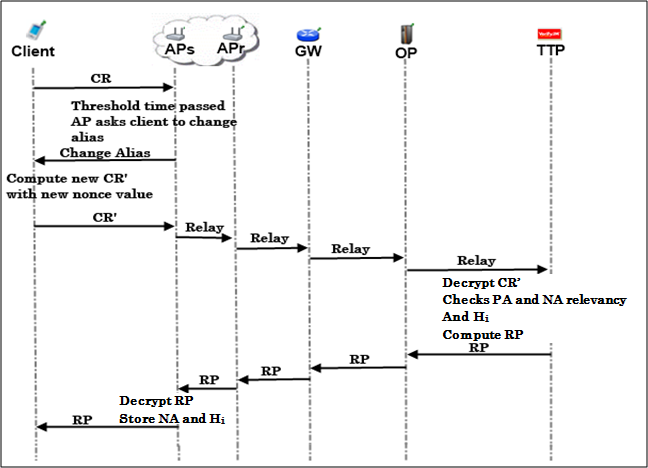


Figure 7 Changing Alias

Changing Alias Protocol is shown in Figure 7 and described below.

1. Client forms a connection request (CR).

Client sends CR to access point.

1. Access Point receives this CR. Access Point often checks a threshold time, and when it’s time exceeds the predefined threshold time value (e.g. an hour or a day; it is up to network designer).

Access Point broadcasts Change Alias order to all of its clients.

1. Client receives change alias command.

Client computes a new alias by picking up a new random nonce N’CL.

Client forms a new

It sends this CR’ to AP.

1. APs updates the entry for the client in the database:

On this entry, AP will update HashValue, Token in Use colons of the entry.

1. Gateway updates client entry in its database:

On this entry, gateway will update HashValue, Token in Use colons of the entry.

1. Operator relays CR’ to TTP to ask if the client is valid or not.
2. TTP receives connection request (CR’) and decrypt it using its private key.

TTP

It checks

It computes (

TTP sendsRP to operator.

1. Operator receives RP and decrypts it using public key of TTP.

It will get and store these values. The values resulting from XOR operation will be used as an alias by the client until he will change his alias.

Operator sends the to the gateway.

1. APs receives RP and decrypt it using public key of TTP.

It will get and store these values. The values resulting from XOR operation will be used as an alias by the client until he will change his alias.

APs sends the RP to the client.

1. Client gets the RP.

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

# Roaming

When the clients need to get service from an access point of a foreign operator, they roam between their home operator and foreign operator. In this kind of situation, foreign operator asks TTP for authentication of the client. In this way, we won’t be communicating with home operator, but only with the TTP, which reduces the load on the network.

In our system TTP is the only entity that could provide initial authentication and permissions. Client could receive service only authenticating to the access point only if they are authenticated by TTP.

This rule is valid for roaming situations as well. Clients need to authenticate themselves to the TTP and then start to get service from access points.

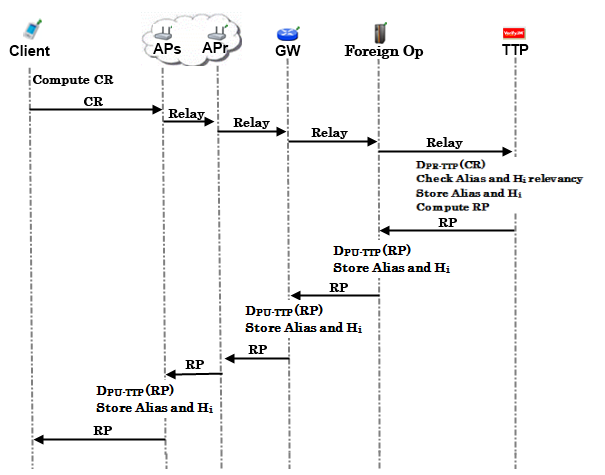


Figure 8 Roaming

Roaming protocol is shown in Figure 8 and described below.

1. Client forms a connection request (CR) for the foreign operator.

Client sends it to serving access point (APs).

1. Serving access point forwards this connection request to the next relaying access point (APr).
2. Gateway relays this connection request to the foreign operator.
3. Foreign operator understands that incoming packets belong to a connection request but it does not know whether it is from operator’s own clients or not.

As for every connection request, foreign operator sends that connection request to the TTP.

1. TTP receives connection request and decrypts it using its private key.

TTP checks for alias’ uniqueness, starts over ACR if necessary.

Since TTP knows the correlation between the alias and hash value, it is able to decide client is authenticated or not.

TTP forms a response

Sends that RP to the gateway, which had forwarded the connection request in the first place.

1. Gateway (GW) receives the response and decrypts it using TTP’s public key.

GW stores and H0.

GW forwards RP to the relaying access point that sent the CR.

1. Relaying access point forwards the RP to the serving AP, without touching the packets.
2. APs receives the response and decrypts it using TTP’s public key.

APs stores and H0.

APs forwards RP to the relaying access point that sent the CR.

1. Client receives the RP and considers himself authenticated.

# 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 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 these threshold time values, access points send update packets to gateways, and gateways send update packets to operator. This mechanism is depicted in Figure 9 and explained below.

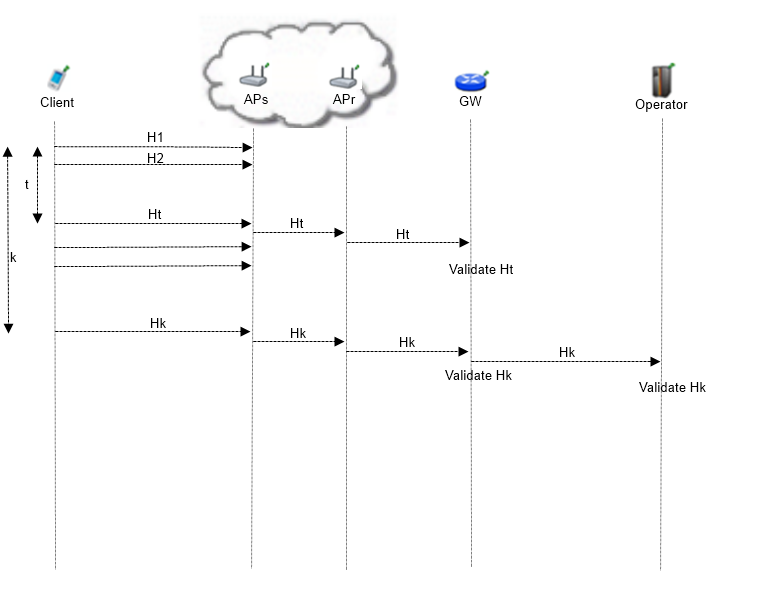


Figure 9 Update Packets ( is Access Point threshold, is gateway threshold)

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 newlyreceived token.

# 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 10..

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.

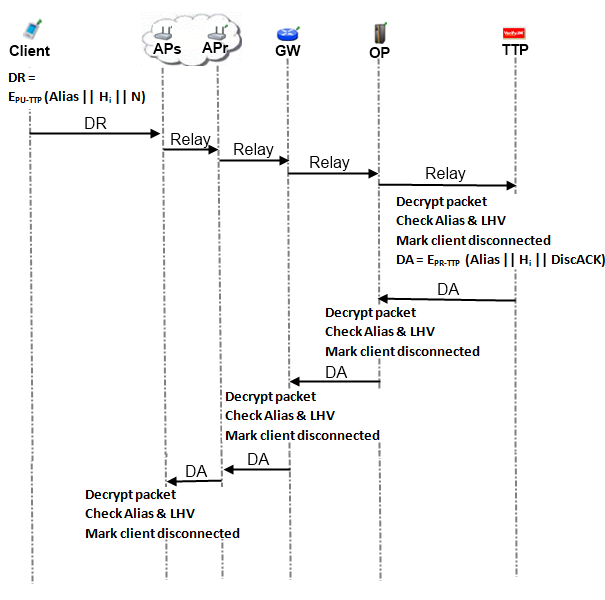


Figure 10 Disconnection from Home Operator (: Disconnection Request, : Disconnection Acknowledgement)

In our disconnection protocols we do not bother TTP for the disconnection process, because it is enough the operator to know about the active users. In order the operators to collect their money in return of their provided service, they will only need the starting token and the ending token of the session. This process will be explained later in the document in more detail.

Disconnection protocol is shown in Figure 11 and described below.

1. Client forms a disconnection request

Client sends it to the access point.

1. Access Point receives the DR and prepares itself to disconnect that particular IP Address of the Client.

Relay the DR to the mesh backbone, to make it reach to the GW.

1. GW receives and forwards the DR to the Operator.
2. Operator receives and forwards the DR to the TTP.
3. TTP receives the Alias and LHV. Checks the relevancy of Alias and the hash token, if the relevancy holds computes a DA.

TTP sends that DA to Operator, which had sent the DR in the first place.

1. Operator receives DA and deletes Client entry from its client table.

Operator relays DA to GW.

1. GW receives DA and deletes Client entry from its client table.

Relay DA to the mesh backbone.

1. Serving Access Point eventually gets the DA and disconnects the particular IP Address which corresponds to the Alias it received.

Before completing disconnection process, it sends final disconnection update to the client to make him sure that he is successfully disconnected.

# Seamless Mobility in Home Operator

When a client moves out of the coverage area of associated AP, a new connection initialization process should be started. If the client is still in the same operator’s zone, it is possible not to start the authentication phase from scratch.

In Figure 11 protocol is demonstrated briefly. The user will send 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 will find the public key, and use this public key to encrypt the credentials of the user.

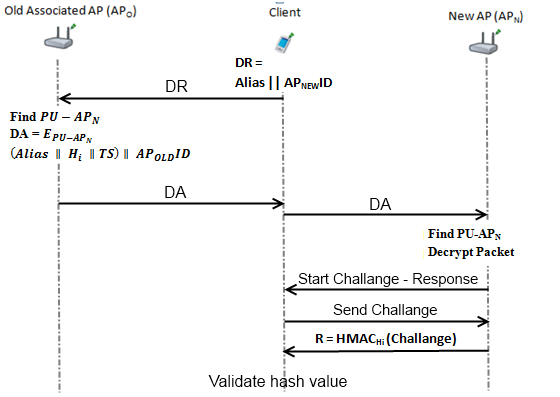


Figure 11 Seamless Mobility in Home Operator

In Figure 11, client moves to the new AP’s () coverage area from old AP (). Seamless mobility is described as follows:

1. Client sends his alias and identity information of the new AP () to .
2. concatenates Alias, , timestamp , and encrypts it with the public key of . It relays this packet to via client.
3. decrypts packet with the private key of itself and checks whether packet is expired or not.
4. If it is not expired, stores and Alias*.* Then sends a challenge request to the client.
5. When client receives this challenge request, it sends a challenge to the .
6. hashes this challenge, and uses relevant *Hash Value* (here ) as the key of HMAC:

sends response to the client.

1. Client also hashes the challenge and uses as the key of HMAC. Then client compares it with the received packet. If it is authenticated, client sets the as the new authenticated AP.

# 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 CRsin 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 theXOR 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.

We need to consider misuse of logs by the operators. Consider the situation of a client:

* Gets service from his home operator between H0 and H10
* Gets service from a foreign operator between H11 and H20
* Gets service from his home operator between H21 and H30

In this type of situation home operator has two CRs and DRs.Whereas foreign operator has a CR and DR. Let’s look at the look at the logs of home operator:

We see that home operator served between H0 and H10 also served between H21 and H30. Home operator would want to take money for serving between H11 and H20. It could pretend that it has served the client between H11 and H20 by not sending Log2 and Log3. Because Log2 indicates that client is disconnected from the operator at H10 and Log3 indicates that Client started to get service from the operator at H21. By sending only Log1 and Log4 home operator tells TTP that it served the client between H0 and H30. Operator would want money for serving 30 hash tokens.

In that kind of situations there should be a foreign operator which has served between H11 and H20. Foreign operator would have two logs as follows.

Foreign operator proves that it has served between H11 and H20 by showing the signed RP and DA.

TTP would see that it has already paid home operator for service to that particular client between H11 and H20. 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 money. 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.

# Conclusion

In this section, we discuss which of the 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. Hence, clients in the same operator can move from one *AP* to another without any interruptions in their connections.

Seamless roaming: Mobility of the clients from one operator’s zone to another is not provided without connection interruptions. This requirement is not met yet, but our infrastructure supports it.

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 by the server. 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 protocol provides unlinkability by changing aliases periodically. There occurs linkability between the times clients change their aliases. The period of time to change the aliases is a choice of the designer.

# References

1. [1] Rivest, R., Shamir, A., and Adleman, L. (1978) A method for obtaining digital signatures and public-key cryptosystems, *Communications of the ACM*,21(2): 120–126.
2. [2] FIPS PUB 197 (2001) Announcing the Advanced Encryption Standard (AES), http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
3. [3] Trappe, W., and Washington, L. (2006) Introduction to cryptography with coding theory, *Person Education, Inc.*
4. Stallings, W. (2006) Cryptography and network security, *Person Education, Inc.*
5. Joseph D. Camp and Edward W. KnightlyThe IEEE 802.11s Extended Service Set Mesh Networking Standard
6. Kai Yang, Jian-feng Ma, Zi-hui Miao (2009) Hybrid routing protocol for wireless mesh network, Computational Intelligence and Security – CIS ’09
7. David Chaum. Untraceable electronic mail, return addresses, and digital pseudo- nyms. Communications of the ACM, 4(2), February 1982.