# PRIVATE AND ANONYMOUS AUTHENTICATION IN MOBILE NETWORKS

Konstantinos Saninas, George C. Polyzos

*Mobile Multimedia Laboratory*
*Athens University of Economics and Busines*
*Patision 76, Athens, Greece*

Keywords: Authentication, Privacy, Anonymity, Mobile Networks Security

Abstract: Anonymity and privacy is a major area of research in the mobile networks environment. This is due to the typical features of such a network, which provides access to services regardless of the user's location and movements. In a typical scenario of a user roaming among different network providers, it is essential to provide anonymity, untraceability and overall security to the user. The use of a proper authentication protocol is a way of providing security to the network and, if designed properly, protection of user's private information. In order to understand the design of such protocols we define the security and privacy requirements that they should satisfy. We present and examine several authentication protocols, which have been designed to provide user authentication in mobile networks. Some of them claim to provide anonymity and untraceability. We examine if they satisfy these basic requirements. The basic protocols are compared to each other in regard to three factors: security, privacy and computational load to the user. It is shown that most protocols can't protect anonymity and untraceability of the user without sacrificing basic security requirements.

## 1 INTRODUCTION

Wireless and mobile networking constitute an emerging area of research and development that is expected to get even more attention and investment as these technologies become more friendly to the common user. There are several reason for this rapid development; mainly the mobility of the network's elements and the distributed access of data and equipment. With the use of wireless and mobile networks we are able to provide new decentralized services that make daily tasks easier and more fun. Due to these characteristics, wireless connectivity suggests higher security risks and more threats especially for the privacy of the user. Private information of the user is being compromised, while travelling through the wireless medium.

These risks are even greater in the typical case of user roaming between different network providers. In such a scenario like this, there are several entities involved:

- The mobile user (*M*),
- The "Home Domain" of the user (*H*),
- The "Remote Domain" (*R*) that is visited in a typical roaming situation,
- Third legitimate entities or domains, and finally
- Third malicious entities like eavesdroppers.

The user usually belongs to his "home domain", where he can have access to the services that it provides, but it is desirable to have access to other networks as well. Third legitimate entities include networks that have established a certain trust with the "Home Domain" of the user. They may or may not have knowledge of user's movements and assorted information. Third malicious entities are the usual adversaries that are trying to attack the networks, have access to services with unauthorized methods or impersonate a legitimate user. They could also be eavesdroppers that can monitor the communication between two other entities. There are usually two kinds of an attacker: the passive one that can only monitor and record messages in a communication line, and the active one that can also modify, after, or inject messages in the communication line.

In a typical scenario of a user visiting his "Home Domain", the user can be authenticated using a common authentication protocol like Kerberos (Neuman and Ts'o, 1994). After he provides his credentials and the network verifies them, the user has access to the services. In a case of roaming to a "Remote Domain" user authentication includes more steps. The user must provide proper credentials to prove that he belongs to a different, though collaborating network. This suggests that at least

two messages have to be exchanged by the two networks. The remote network wants to know if the user belongs to the network that he claims to, and the home network replies accordingly. After the home network verifies the solvency of the user, the authentication of the user at the remote domain is completed and he can access any or some of the services provided. It is evident that the second scenario poses more threats for the user's privacy. We will see that there are several levels of protection. For the purpose of this discussion, we assume that the protection of privacy revolves around three important security goals.

**Data confidentiality:** keeping the data exchanged by two communication parties secret to all other unauthorized parties.

**Anonymity:** the confidentiality of the user's pseudonym or any other information that could reveal the real identity of the person.

**Untraceability:** the confidentiality of any information regarding the person/user's movements.

To provide confidentiality of the data travelling through the wireless medium, we must encode the communication with a shared symmetric secret-key. Usually the same applies to protecting the pseudonym of the user; it's best to avoid sending the pseudonym unencrypted, thus minimizing an attacker's chance to associate the pseudonym with a certain session, or impersonate the user.

# 2 DESIGN REQUIREMENTS

A proper authentication protocol designed for mobile or wireless networks should protect the privacy of the user along with providing proper authentication. There are several requirements that these protocols should satisfy, divided into three categories: general requirements (Bird et al., 1993, Bird et al., 1991), security requirements (Boyd and Park, 1998, Horn and Preneel, 1998) and privacy requirements (Samfat et al., 1995).

## 2.1 General requirements

General requirements mostly concern important principles for the design or implementation of a secure authentication protocol. Application of these principles does not ensure the success of the design process nor provides any proof of security. In order for the authentication protocol to be reliable and have a minimum security, it should be:

**Nonce – Based:** the protocol should be nonce-based instead of using timestamps. Nonces or timestamps are often used to provide assurance of freshness in the messages. Timestamps have a lot of disadvantages, especially when they are used in a distributed environment. They require the use of synchronized watches in order for every entity to be able to check the timestamp. This need usually suggests the existence of a trusted third authority, which duty is to send messages of clock synchronization to all other parties. The extra overhead in network traffic by these messages is an additional cost that makes the use of nonces more attractive.

**Secure against all usual attacks:** the resulting protocol should be tested against all the usual attacks that an adversary or a cryptanalyst could employ. This does not prove that the protocol is secure, but is a minimum requirement to ensure that an attacker could not break the protocol easily.

**Usable at any network layer:** the protocol should use small packets so that it can be used on lower layers that have a fixed size of packets.

**Usable at any computational base:** wireless and mobile networks are often comprised of small devices from the user side. With the emergence of new, more powerful mobile devices this phenomenon will become more apparent. Nevertheless, the computational capability of mobile devices is, and will continue to be, limited. Unfortunately, encryption operations have an important computational cost. That is even more evident in public key cryptosystems, when one communicating party encrypts data with its private key. Thus, the user should not have to make many encryption operations.

Also, the mobile device's need for power during the authentication should be minimized. Towards low battery consumption and simplicity of implementation, the number of exchanged messages and communication flows must be small. Usually, in the case of two communicating parties three messages are exchanged between the authentication server and the user. When a second authentication server is engaged (ETSI, 1993), there are five or six communication flows.

**Make use of any cryptographic algorithm:** the protocol should be designed in a way that can be used with most of the cryptographic algorithms, symmetric or public key.

**Minimum need for storing shared secrets:** we should not assume that the storage of secret data is a simple thing for the mobile user. Mobile devices like phones and PDAs are not capable of storing a large amount of data. Also, the user must have the secret key always with him, even if storing the key on a

smart card runs the risk of loosing it. The protocol design must not require the user to store a large amount of data. Also, it should define safety procedures for the case that a user looses his key.

## 2.2 Security requirements

A secure authentication protocol must meet the following security requirements, defined by Horn & Preneel for the ASPeCT protocol, which was a candidate for the UMTS authentication protocol. A more detailed review of the following can be found in (Menezes et al., 1996).

**Mutual authentication of user and network:** the main objective of an authentication protocol is to authenticate the user. Nevertheless, it is optimum both the user and the network to be authenticated to each other. Network authentication is an important requirement especially in the case of roaming, where a third malicious entity could impersonate a network provider.

**Agreement on a secret session key with mutual implicit key authentication:** Mutual key authentication means that both communicating parties are assured that no other party aside the second identified party may gain access to the secret key.

**Mutual key confirmation:** By key confirmation both parties are assured that the other entity has possession of the secret key. It doesn't provide that at the time of confirmation, the other party is identified. If both requirements of key confirmation and key authentication arc satisfied, then we can say that we have achieved explicit key authentication.

**Mutual assurance of key freshness:** by assurance of key freshness, the resulting key is different from any other key that has been used on previous protocol runs. This is usually achieved by adding random numbers in the process of key generation. This requirement is related with the property of mutual key control, where no party has unilateral control of the resulting key. This is apparent in protocols like the GSM authentication protocol, where the home network alone chooses the key.

**Non-repudiation of origin for relevant user data:** non-repudiation is the prevention of denial of previous actions or data by an entity. This property is achieved with the use of digital signatures, although it is useful mostly for billing and accounting.

## 2.3 Privacy requirements

In a typical roaming environment there are several entities involved. The achieved user privacy depends on the amount of private information that the authentication protocol manages to withhold from other entities, legitimate or not. Hence, we can classify the privacy level, depending on the awareness of the involved entities about the user's identity and/or location. Samfat et al. have defined five levels of privacy from C1 to C5 (followed by our own remarks):

**C1: Hiding User Identity from Eavesdroppers.** Protecting the user from third malicious entities is the most common privacy requirement that authentication protocols should satisfy. This is usually accomplished by either encrypting the user identity with the public key of the visited network before sending it, or generating user aliases when the user visits a foreign domain. In the first approach, there must be a way of distributing securely public keys (e.g. a Certification Authority). In the second approach, the `alias' generation procedure should be made in a sufficiently random manner and provide “forward secrecy” (Diffie et al., 1992). This means that if an alias is compromised, the eavesdropper can deduce no information about previous aliases.

**C2: Hiding User Identity from Foreign authorities.** This is a higher privacy requirement in which the user identity is kept secret from the visited domain. The solvency of the user must be proven by the home network, so that the user can access any service in the foreign domain. Although, it seems that it is necessary to have at least two more messages exchanged by the two networks, this is not mandatory when the protocol uses certificates. This requirement is important when it is desired that foreign networks should not be able to track the user's movements.

**C3: Hiding Home Domain from Third Parties.** We can further enhance the privacy of the user, by hiding the identity of the home domain from third legitimate (cooperating networks) or malicious entities (i.e. the visited `foreign' network is not included). Thus, third parties cannot suggest the identity of the user by deducting a smaller group of origin. The real identity should be assumed from all possible identities, instead of assuming one from just the users of network *X org'*. Furthermore, in a peer to peer confederation of networks, an operator or provider could not keep statistics of movements of users that belong to another network.

**C4: Hiding Home Domain Identity from Foreign Authorities.** This class adds the foreign network to

the group of entities unaware of the home domain identity. This is useful for protecting the origin of the user and to prevent the collusion of neighbouring foreign networks. If a user is moving then he should authenticate himself subsequently on neighbouring foreign networks. In a collusion scenario, all the foreign entities have to do to deduce the user's identity, is to search for users from the same home domain, who accessed their networks at different but close time intervals.

**C5: Hiding user Behaviour from Home Authority.** There are cases that a user wants maximum privacy in order to hide his movements from his home network. The result is that no network can have any knowledge about the user's movements or his location at a specific moment.

# 3 AUTHENTICATION PROTOCOLS

## 3.1 Authentication in GSM

The Global System for Mobile Communications (ETSI) standard was a European effort for standardization of mobile communications. GSM tries to provide user authentication, data confidentiality and key management, but the authentication protocol has several known limitations. At the GSM Authentication Protocol every mobile entity has a unique identifier, known as International Mobile Station Identity (IMSI) that links him with his home network. The Authentication Server of the home network shares a key $K_{MH}$ with every entity that belongs to it. When a mobile user *M* appears in a remote domain *R*, he presents his *IMSI*. *R* understands the identity of the home network *H* and sends a request. *H* responds with a set of triads, that *R* can use to verify *M*'s identity. Every triads is comprised by a random challenge *RAND*, a response to the challenge *SRES*, which is a function of *RAND* and $K_{MH}$, and a key $K_{MR}$, which is a different function of the same arguments. With the triad, *R* can challenge *M* sending him *RAND*. Only an entity that knows $K_{MH}$ can compute the corresponding *SRES*. The identity of *M* can know be verified. As soon as *M* is authenticated, he and *R* can use $K_{MR}$, which *M* can compute in a similar way, for the encryption of the transmitted data. Here is the protocol:

$1. M \rightarrow R : M$

$2. R \rightarrow H : M, R$

$3. H \rightarrow R :< RAND, SRES, K_{MR} >$

$4. R \rightarrow M : RAND$

$5. M \rightarrow R : SRES$

where *RAND* is a random number, $SRES = \{RAND\}_{K_{MH}}$, $K_{MR} = \{RAND\}_{K_{MH}}$ using though a different symmetric algorithm.

In the previous protocol *GSM* assumes that the network between *H* and *R* is secure. But this can't be a certainty, especially in open networks. The privacy of the user is protected by providing anonymity with the use of Temporary Mobile Station Identities (*TMSIs*). Nevertheless, *IMSI* must be revealed for the computation of *TMSI*. Obviously, in *GSM* the protection of privacy and user anonymity relies on the trust between the providers.

## 3.2 The Varadharajan/Mu Protocol

Varadharayan and Mu( )presented a series of protocols for user authentication in wireless and mobile networks. These protocols covered a series of scenarios and situations of communication between mobile devices. Some of these protocols allowed two users to communicate with end-to-end security, without the intervention of third servers or networks. We will focus on one protocol presented at these papers that concerned a typical scenario of user roaming (called by the authors as inter-domain protocol).

This scenario has a lot in common with the Samfat et al. protocol, although it is simpler in implementation. Mobile user *M* has a `subliminal' identity $M_S$, known only to himself and his home network *H*, which contains a serial number and a timestamp. The home domain is the only one that can match this identity with the real name of the user. Also, there is a symmetric key $K_{MH}$ shared by the user and the home network. In a similar way, a key $K_{RH}$ exists for the communication between the home domain and the remote domain (*R*). The protocol is comprised by the following messages:

$1. M \rightarrow R : M_S, H, N_M, Token_{MRH}$,

$\{h(M_S, H, N_M)\}_{K_{MR}}$

$2. R \rightarrow H : R, H, N_R, M_S, Token_{MRH}$,

$\{h(R, H, N_R, M_S, Token_{MRH})\}_{K_{RH}}$

$3. H \rightarrow R : H, R, N_R, \{M'_S\}_{K_{MH}}$,

$\{h(H, R, K_{MR}, M_S, N_R)\}_{K_{RH}}$,

$\{K_{MR}, M_S\}_{K_{RH}}, \{h(H, M'_S, N_M)\}_{K_{MH}}$

$4. R \rightarrow M : R, M_S, \{K_S\}_{K_{MR}}, \{h(R, M_S, K_S)\}_{K_{MR}}$,

$\{M'_S\}_{K_{MH}}, \{h(H, M'_S, N_M)\}_{K_{MH}}$

where $K_{MR} = f(K_{MH}, M_S, R)$ and $Token_{MRH} = \{M, H, R, N_M\}_{K_{MH}}$.

The protocol consists of four message flows, because the foreign network asks the home network to verify the identity of the user. The authentication of the user M is based on the token $Token_{MRH}$, which is send from *M* to *R* and finally to *H*. This token is not readable by R since it is encrypted with $K_{MH}$ which R doesn't have. $2.R \rightarrow H : M, R$

The hashed first message is signed with the key $K_{MR}$. Notice that the remote network can't generate this key, thus cannot check the signature. Nevertheless, the remote network passes the token and waits for confirmation. After the home network verifies the solvency of the mobile user, sends its verification in message 3. Included in the message are the new `subliminal' identity of the user, and the key $K_{MR}$ along with the old identity of the user. The home networks signs the message twice, for the remote network and then for the user with the keys $K_{RH}$ and $K_{MH}$ respectively. After receiving message 3, the remote network learns the key $K_{MR}$ and verifies the signature received at message 1. At message 4, the user is authenticated and the remote network sends to him the new subliminal identity $M'_S$ and the session key $K_S$.

The protocol of Varadharajan and Mu protects the anonymity of the user with the use of subliminal identities, similar to *GSM*'s *TMSI*s. The renewal of the identity is more secure than GSM, because the foreign network is not aware of the new identity. Protecting the aliases of the user in a confidential way, we prevent the association of a user with certain behaviour by malicious entities. Overall, the protocol is very simple, using only symmetric cryptography and avoiding costly operations associated with public cryptography.

One weakness of the protocol is the existence of static keys between the users and their home network and also between all the network providers. This suggests that if the keys remain the same for a long time the security of the system decreases. In order to make the system more secure we should add a key management mechanism that will renew all the shared keys after a period of time. In that case the users of a network would renew their keys, whenever they would login to their network. Another problem is the renewal of the subliminal identity of the user. The protocol doesn't provide a verification of acceptance of the new identity by the user. The new identity could never reach the user and the home network wouldn't know it. That could lead to inconsistencies to the database of the home network, and the user would not be able to be authenticated in the future.

## 3.3 The ASPeCT Protocol

The ASPeCT protocol (Horn and Preneel, 1998) was a candidate for the authentication and key exchange in the Universal Mobile Telecommunications System (*UMTS*) and it is based on public key cryptography. It is designed for authentication of a user from a Value Added Service Provider (*VASP*) as well as from a UMTS network.

Besides user *M* and network *R* there is a Certification Authority (*CA*), which provides verification of the entities' public keys. Public keys of *M* and *R* are $y_M = g^{x_M}$ and $y_R = g^{x_R}$ respectively, where $g$ is a generator known by both sides and $x_M = y_M^{-1}$, $x_R = y_R^{-1}$ the private keys as in Diffie-Hellman key exchange protocol (Diffie et al., 1992). Also, we use the following parameters:

- hashing functions *h1, h2 and h3*,
- message *X* signed by an entity *U* is $\{X\}_{y_U^{-1}}$,
- ***chd*** are data concerning charging
- ***pay*** are the data concerning payment,
- $T_M$ is the timestamp issued by user *M*,
- $T_R$ is the timestamp issued by *R*.

The protocol is as follows:

$1.M \rightarrow R : g^{r_M}, CA$

$2.R \rightarrow M : r_R, h_2(K_{MR}, r_R, R),\ chd, T_R, Cert_R$

$3.M \rightarrow R :$
$\{\{h_3(g^{r_M}, g^b, r_R, R, chd, T_M, pay)\}_{y_M^{-1}}, Cert_M, pay\}_{K_{MR}}$

The session key is computed by *M* as $K_{MR} = h_1(r_R, (y_R)^{r_M})$ and from *R* as $K_{MR} = h_1(r_R, (g^{r_M})^{x_R})$.

For the protection of the user's privacy, the user certificate is not sent until message 3, where it can be encrypted with the session key. Although this seems to be beneficial to the protection of the user's identity, it is not the proper method to do so when charging is involved. The actual problem is that the user signs the charging data and states his identity on the same message. This allows an intruder to intercept the message and withhold from the provider the fact that the user has paid.

# 4 COMPARISON

We compare the presented protocols, based on the requirements for security and privacy of the user, defined earlier. Since the protocols are designed for use in mobile and wireless networks, we provide the computational load on the mobile user for each protocol. This is a measure of efficiency and adaptability to a network comprised of small devices. The computational load is measured in how many times the mobile device has to perform a cryptographic operation, symmetric or public-key. It is common that public key operations have a lot more computational load than symmetric ones. Furthermore, we decided not to measure the computational load in modular multiplications, since other types of asymmetric cryptosystems can be used, that are more efficient (e.g. elliptic curve cryptography).

# 5 CONCLUSION

Comparing the protocols regarding security, privacy and efficiency the conclusions are educational. It is obvious that the more secure a protocol is, the more computational needs has (i.e. the security/computational cost relationship is proportional). But it was apparent enough, that protocols that were designed for optimal privacy are not so secure and vice versa.

Security requirements defined here are very important because they mainly concern mutuality. When these requirements are fulfilled, the user has control over the progress of the protocol. That is, the server doesn't decide unilaterally on the keys issued by the protocol. The protocols that manage to provide a proper level of security use mainly public key cryptography. The reason is that we usually need a Diffie-Hellman exchange in order to give the user the ability to contribute to the computation of the session key or exchange nonces with the server. “The ASPeCT protocol” make use of this technique successfully. The drawback of the extensive use of public key cryptography is the computational load that made the use of these protocols forbidding in the past.

Of course, the most difficult goal to pursue is privacy. The difficulties in pursuing this goal are the generation of aliases and the encryption of user information from entities that participate in the protocol.

# 6 REFERENCES


Bird, R., Gopal, I., Herzberg, A., Janson, P., Kutten, S., Molva, R. & Yung, M. (1993) Systematic Design of a Family of Attack-Resistant Authentication Protocols. *IEEE JSAC*.

Bird, R., Gopal, I., Herzberg, A., Molva, R., Janson, P., Kutten, S. & Yung, M. (1991) Systematic Design of two-party authentication protocols. IN J.FEIGENBAUM (Ed.) Santa Barbara, CA, Springer Verlag.

Boyd, C. & Park, D.-G. (1998) Public Key Protocols for Wireless Communications. *The 1st International Conference on Information Security and Cryptology (ICISC'98).*

Diffie, W., Oorschot, P. C. V. & Wiener, M. J. (1992) Authentication and authenticated key exchanges. *Designs, Codes, and Cryptography*, 2**,** 107-125.

Etsi (1993) GSM - Security Related Network Functions. European Telecommunications Standards Institute.

Horn, G. & Preneel, B. (1998) Authentication and Payment in Future Mobile Systems. *European Symposium on Research in Computer Security (ESORICS '98).*

Menezes, A., Oorschot, P. V. & Vanstone, S. (1996) *Handbook of Applied Cryptography*, CRC Press.

Neuman, C. & Ts'o, T. (1994) Kerberos: An Authentication Service for Computer Networks. *IEEE Communications Magazine*, 32.

Samfat, D., Molva, R. & Asokan, N. (1995) Anonymity and Untraceability in Mobile Networks. *ACM International Conference on Mobile Computing and Networking*.