A GROUP AUTHENTICATION MODEL FOR WIRELESS NETWORK SERVICES BASED ON GROUP KEY MANAGEMENT

Huy Hoang Ngo, Xianping Wu and Phu Dung Le
Faculty of Information Technology, Monash University, Australia

Keywords: Group authentication, group key management, forward secrecy.

Abstract: Group authentication provides authentication for members of communication group between services and users in insecure environments such as wireless networks. Most of the group authentication models do not consider the risk of compromised share secrets in the group under various security threats such as cryptanalysis. Although authentication key exchange in groups can benefit from group key management to minimise this risk, the group key management schemes are inefficient for authentication services. In this paper, a group authentication model for wireless networks services using a group key management is presented. The group key management is specially designed with forward secrecy and session keys for efficient and secure key exchange. Based on this secure session keys, a dynamic group authentication scheme provides a secure and efficient group authentication for wireless network users and services.

1 INTRODUCTION

Authentication is the major security component in information systems. Authentication protects the systems from unauthorised accesses from malicious sources. The classical authentication scheme deals with two trusted parties communicating over an insecure environment. In this scheme, these parties share secrets with each other. By proving the ownership of the secrets, a party can create the trust on its identity. The source of the authentication can be either an individual user, an application or a single service in the system. Normally the target of the authentication process is a service. Each service maintains a database of its own user identities and authentication keys. The cost to maintain the user identity databases for different services is expensive especially in wireless networks.

Instead of verifying individual identities as in the classical authentication scheme, group authentication allows members to prove their memberships of the group from a centralised user identity database. There has been much research on group authentication. In (Dijk et al., 1998), (Hanaoka et al., 2002), and (Zwierko and Kotulski, 2005)schemes, the secret sharing is used to provide group authentication. However, these schemes did not consider the compromised secret keys risk. Using the non compromised authentication keys assumption, Homage (Handley, 2000) utilised Diffie-Hellman and group signatures to provide a secure group authentication. However, Jaulmes and Poupard(Jaulmes and Poupard, 2002) mentioned the vulnerability of forgeable valid proof and compromised identity. Martucci et. al. (Martucci et al., 2004) proposed the use of hash function to calculate the challenge and authentication key from the time stamp. To prevent replay attacks, clock synchronisation and limited challenge life time were employed. On one hand, the pre-shared key as input for hash function becomes vulnerability cryptanalysis attacks. Group key management (Rafaeli and Hutchinson, 2003), (Challal and Seba, 2005), (Amir et al., 2001) is another approach of authentication key exchange in group authentication. On the other hand, the previous group key management mechanisms did not focus on efficiency for group authentication.

In this paper, we present a group authentication model based on group key management. In this group authentication model, the users and services are divided into clusters. Each cluster has its own secure policy and requirement. A modified version of group key management is employed to exchange authentication keys in the clusters. This group key management is built for efficient authentication key exchange in wireless networks. Users can anonymously authenticate to services in the same cluster by using the cluster

Hoang Ngo H., Wu X. and Dung Le P. (2008).
A GROUP AUTHENTICATION MODEL FOR WIRELESS NETWORK SERVICES BASED ON GROUP KEY MANAGEMENT.
In *Proceedings of the Tenth International Conference on Enterprise Information Systems - ISAS*, pages 182-188
DOI: 10.5220/0001687201820188

authentication key from the group key management.

The rest of the paper is organised as follows. Section 2 introduces the authentication model. The group key management is illustrated in section 3. Section 4 presents the group authentication module. The model is analysed in section 5. A case study is investigated in section 6 and section 7 concludes the paper.

2 THE GROUP AUTHENTICATION MODEL

The group authentication model contains two main components: a group authentication module and an authentication group key management. The authentication module uses the group key management to exchange authentication keys of the clusters. Based on these keys, the authentication module provides a secure and anonymous authentication service to members in the clusters. The model is shown in Fig 1.

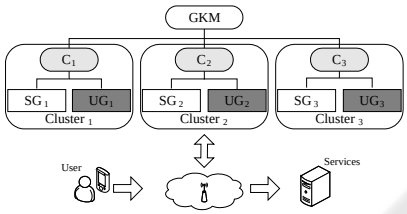

Figure 1: Group Authentication Model.

The group authentication module and the group key management are two different layers in the group authentication framework. These layers are shown in Fig 2.

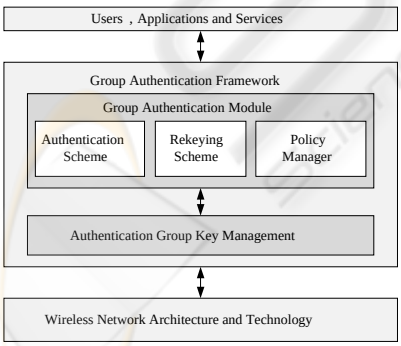

Figure 2: Group Authentication Architecture.

3 THE GROUP KEY MANAGEMENT

Group key management is the most important framework for secure group communication. In (Challal and Seba, 2005), Challal and Seba implied that the major problems of group key management are confidentiality, authentication, watermarking and access control. Based on the above problems, there are five security requirements for group key management: forward secrecy, backward secrecy, collision freedom, key independence and minimal trust. Among these security requirements, forward secrecy and backward secrecy are very important for group communication confidentiality. Forward secrecy implies that users who have left the group cannot obtain any future key. Backward secrecy means that users who have joined the group cannot obtain any previous keys.

Although authentication is among the issues for group key management, most of proposed group key management protocols focus on data confidentiality which requires both forward secrecy and backward secrecy. On the other hand, authentication does not need backward secrecy. In other words, users that join a group can obtain the previous keys. These keys are neither re-used for authentication nor used for encryption sensitive data in group communication. Therefore, when a new user or a service joins a group, the re-keying process in group key management is unnecessary. Based on this characteristic, an efficient group key management for group authentication is proposed.

3.1 Notation

- ug_1, ug_2 : groups of users.
- sg_1, sg_2: groups of services.
- c_1, c_2: clusters.
- s_{11}, s_{12}: services in service group sg_1.
- u_{21}, u_{22}: users in user group ug_2.
- $K_{S11}, K_{S12}, K_{S22}$: individual authentication key of services.
- $K_{U12}, K_{U21}, K_{U22}$: individual authentication key of users.
- K_{S1}, K_{S2}: group key of service group sg_1 and sg_2.
- K_{U1}, K_{U2}: group key of user group ug_1 and ug_2.
- K_{C1}, K_{C2}: the current cluster authentication keys of cluster c_1 and c_2.
- GKM: group key management server.
- N_1, N_2: nonces (random number).
- $\{\mathrm{x}\}\mathrm{k}$: message x is encrypted by the key k.
- $h(x,y)$: hash function of message x and key y.
- DK: distributed key

- $SK_1, SK_2, ...SK_N$: session keys
- $A \rightarrow B\{message\}$: A sends *message* to B using unicast.
- $A \Rightarrow B\{message\}$: A sends *message* to B using multicast or broadcast.

3.2 The Group Key Management Architecture

In this group key management framework, the services and users are divided into n clusters. The services in a cluster are grouped into one service group. Users in this cluster are also grouped into one user group. Users in the cluster C_1 share the cluster authentication key K_{C1} to authenticate to the service group sg_1 in this cluster. In this group key management, one user may present in several groups. This situation illustrates that one user can have authorisation access to multiple services in different service groups in different clusters. The group key management can have a tree structure as shown in Fig 3. The second level of the tree encloses the clusters.The third level of the tree contains the users and service groups. Each service group ug_1 is depicted by a key K_{S1}. Each user group ug_2 is depicted by a key K_{U2}. Service group sg_1 shares the cluster key K_{C1} with user group ug_1 in the second level.

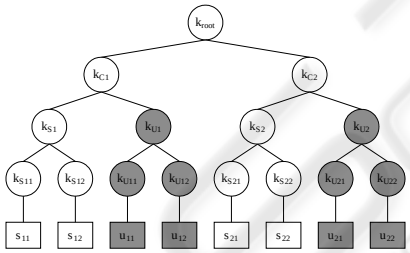

Figure 3: Group Key Management Tree.

Because services in the system have higher privilege than the users, the services are separated from the users in the cluster. If an application wants to join the service group in a cluster to provide the service, it has to authenticate itself to the group key management first. Therefore, not anyone in the system has authorisation to provide services. These services are divided into different clusters because of their different requirements of security. Each service group has different security policy and requirements. The services in the same group share the same security policy and requirements. Therefore, users in group ug_1 can use the same key K_{C1} to authenticate to different services in the same service group gs_1. These services share not only the same authentication key K_{C1} but also the same security requirements.

In the following sections, we explain the procedures of the group key management for authentication in wireless networks. Similar to the group key management in (Damodaran et al., 2006), the session keys are used for three operations in this group key management: join, leave and manual rekeying.

3.2.1 Service Join

When the service s_{22} wants to join a service group, it sends the join request to the key server. Attached to the join request is a distributed key DK. The distributed key DK is a key generated by the service and exchanged with the key server. This key is used to compute session key SK. When the key management server receives the join request, it generates and sends back to the user a random number N_1 as a challenge. Both service s_{22} and key management server compute the set of session keys $\{SK_1, ... SK_N\}$ from the distributed key DK as following procedures:

$SK_1 = h(DK, K_{S22})$
$SK_2 = h(DK, SK_1)$
...
$SK_N = h(DK, SK_{N-1})$.

The session keys in the key set $\{SK_1, ...SK_N\}$ are used to encrypt communication messages between the service and key management server. These session keys can be used once. Service s_{22} encrypts the challenge N_1 by the first session key SK_1. Key management server sends back to the service a response with the current cluster keys K_{C2} and the current group key K_{S2}. Because the backward secrecy is not required to prevent service from accessing previous group communication, rekeying after joining in other group key management architecture is redundant. The messages between service and key management server are illustrated as follows:

1. $s_{22} \rightarrow GKM : \{join, s_{22}, sg_2, \{DK\}K_{S22}\}$
2. $GKM \rightarrow s_{22} : \{N_1\}$
3. $s_{22} \rightarrow GKM : \{N_1, N_2\}SK_1$
4. $GKM \rightarrow s_{22} : \{N_2 + 1, K_{C2}, K_{S2}\}SK_2$

3.2.2 User Join

The join operation of users is similar to that of services. If u_{12} wants to join user group ug_1, four following messages are used.

1. $u_{12} \rightarrow GKM : \{join, u_{12}, ug_1, \{DK\}K_{U12}\}$
2. $GKM \rightarrow u_{12} : \{N_1\}$
3. $u_{12} \rightarrow GKM : \{N_1, N_2\}SK_1$
4. $GKM \rightarrow u_{12} : \{N_2 + 1, K_{C1}, K_{U1}\}SK_2$

3.2.3 Service Leave

Opposite to join operation, forward secrecy is very essential to leave operation. When a member in service group leaves the group, it is warranted that the service cannot receive future group communication details. In other words, the service can no longer get future authentication keys to provide services to that group. The leave operation can either be invoked by services or initiated by the key server. The leave operation for services has two steps. The first step is the leave request. The second step is the rekeying procedure. The rekeying procedure only affects one service group and its correlative user group in the same cluster. If s_{11} wants to leave group, it sends a leave request encrypted by the next session key SK_{S11} in the session key set of service s_{11}. The next step is to rekey the group key K_{S1} and cluster key K_{C1}. The key management server sends to all other members in the service group using uni-cast the new keys K_{S1} and K_{C1} encrypted by the next session keys in their session key sets. When all the keys in the key set are used, the service generates a new distributed key DK itself and sends it to the key management server to invoke the operation that re-generates the new session key set. In the rekeying process, the key server also multi-casts the new cluster key to user group ug_1. The messages in the system is described as follows:

1. $s_{11} \rightarrow GKM : \{leave, sg_1, s_{11}, h(K_{C1}, SK_{S11})\}$
2. $GKM \rightarrow s_{12} : \{newK_{S1}, newK_{C1}\}SK_{S12}$
3. $GKM \Rightarrow u_{11}, u_{12} : \{newK_{C1}\}K_{U1}$

3.2.4 User Leave

The leave operation for users is also similar to the leave operation for services. It can also be either invoked by the users or initiated by the key server. The first step is the leave request. The second step is the rekeying procedure. The rekeying procedure warrantees that the users leaving the group can no longer authenticate to the services in that group. If u_{21} wants to leave the group ug_2, the following messages are sent for rekeying:

1. $u_{21} \rightarrow GKM : \{leave, ug_2, u_{21}, h(K_{C2}, SK_{U21})\}$
2. $GKM \rightarrow u_{22} : \{newK_{U2}, K_{C2}\}SK_{U22}$
3. $GKM \Rightarrow s_{21}, s_{22} : \{newK_{C2}\}K_{S2}$

3.2.5 Manual Rekeying

The manual rekeying is a group and cluster keys renewing process in the system. It is not related to either join or leave operation. After a number of authentication attempts, group and cluster keys used for authentication become vulnerable under key compromised and cryptanalysis attack risks. This manual rekeying operation helps the system to reduce the above risks. The more regularly rekeying process is invoked, the more secure the group authentication model is. However, regular rekeying also uses a great deal of resource in the authentication model. Different groups of services require different security policies. The policy manager in the group authentication framework determines when and how the manual rekeying to be invoked to obtain the security and efficiency requirement for services. The rekeying process of cluster C_1 is specified as the following messages.

1. $GKM \Rightarrow sg_1 : \{newK_{C1}\}K_{S1}$
2. $GKM \Rightarrow ug_1 : \{newK_{C1}\}K_{U1}$

4 THE GROUP AUTHENTICATION MODULE

The group authentication module has two parts: an authentication protocol and a policy manager. The authentication module uses the cluster keys from the group key management to perform group authentications to the services. The policy manager decides when the manual rekeying operation is invoked to renew cluster and group keys to obtain the most suitable security and efficiency requirement.

4.1 The Group Authentication Protocol

4.1.1 Notation

- U: user.
- S: service.
- K_C: the current authentication key.
- $h(X,Y)$: hash function of message X with the key Y
- K_US: the session key.

4.1.2 The Group Authentication Protocol

In the group authentication protocol, user U uses the secrets K_C to authenticate to the services in the same clusters. These keys are the current and previous cluster keys from the group key management. Therefore, all the members in user group and service group share this secrets as authentication keys. Based on this group authentication keys, U can authenticate to the services without revealing its identity. The authentication protocol is shown below:

1. $U \rightarrow S : N_1$
2. $S \rightarrow U : \{N_1 + 1, N_2, K_{US}\}K_C$

3. $U \rightarrow S : \{N_2 + 1\}K_{US}$

U generates a nonce N_1 and sends it to S as the authentication request. S generates a session key K_{US} and sends to U with the nonce N_1 and another nonce N_2 as a challenge encrypted by the cluster key K_C. U decrypts the challenge to extract the session key K_{US} and N_2. He trusts the session key K_{US}. The last message is the exchange of nonces N_2, encrypted by the session key K_{US}. The nonces exchange process is used to prove the ownership of the session key K_{US} and cluster K_C.

4.2 Policy Manager

The services in different clusters have different security and efficiency requirements. Security requirements often conflict with efficiency requirements. Higher security requirement often demands many system resources. The policy manager is the component that sets the policy when the cluster and group keys are manually rekeyed to match with the security and efficiency requirements of each cluster. For each cluster in the system, the policies can be set for rekeying as follows:

- after each authentication request to a service in a group.
- after a constant period of time (depend on the number of users and services in the group and the average amount of authentication).
- both two above conditions.

5 DISCUSSION AND ANALYSIS

We highlight the important issues that are raised in the group authentication model: security and efficiency.

5.1 Security

The security of this authentication model is analysed through its two layers: the authentication module and group key management. The security of the group authentication module heavily depends on the group authentication keys. This group authentication keys are managed by the group key management.

5.1.1 Security of Group Key Management

The proposed group key management inherits the security features of the previous group key management schemes. Although this group key management does not enforce backward secrecy, it satisfies the following security goals:

1. Non Group Key Confidentiality: Members who are not members of the group cannot obtain the shared secrets group keys within the group. Therefore a principal that does not belong to a cluster cannot obtain the current cluster key. When another principal leaves a group, he is also no longer able to obtain the current cluster key.

2. Key Authenticity: The key management server only accepts requests from authenticated group members.

The final issue of group key management for group authentication is the risk of compromised cluster keys from replayed and cryptanalysis attacks. In order to obtain the current cluster key to authenticate, attackers capture messages from group key operations to extract the cluster key. However, the cluster key is manually rekeyed after a period. This operation can reduce the risk of compromised cluster keys from cryptanalysis attacks. Although the cryptography using in group key management is symmetric encryption, cryptographic keys encrypting the cluster keys are short term session keys. If a session key is compromised, it has a new value in the next message. Even in the worst scenario, when an attacker has guessed all the correct values of session keys, the current set of session keys is expired after a period of time. A new set of session keys is generated for the authorised parties. This feature can minimise the risk of compromised session keys and cluster keys.

5.1.2 Security of the Authentication Module

The following notations description are the BAN notations (Rubin and Honeyman, 1993) using in analysing the security of the protocol.

Notation

- $U \overset{K_{US}}{\longleftrightarrow} S$: U and S may use the shared key K_{US} to communicate.
- $S \models X$: S believes X.
- $\sharp(X)$: X is fresh. X has not been sent before in any messages.
- $S \Rightarrow X$: S has jurisdiction over X: S's beliefs about X should be trusted.
- $U \triangleleft Y$: U has received message Y. U can read and repeat Y.
- $U \mid\sim X$: U has sent a message including the statement X. U believes X when he sends it.

The group authentication protocol is analysed using BAN Logic(Burrows et al., 1990) . First, it is transformed into the idealised form.

1. $U \rightarrow S : N_1$

2. $S \rightarrow U : \{N_1, N_2, U \overset{K_{US}}{\longleftrightarrow} S\}K_C$
3. $U \rightarrow S : \{N_2, U \overset{K_{US}}{\longleftrightarrow} S\}K_{US}$

To analyse the security of the group authentication protocol, we make the following assumptions:

$U \models U \overset{K_C}{\longleftrightarrow} S \qquad S \models U \overset{K_C}{\longleftrightarrow} S$

$U \models S \Rightarrow U \overset{K}{\longleftrightarrow} S \quad S \models U \overset{K_{US}}{\longleftrightarrow} S$

$U \models \sharp(N_1) \qquad S \models \sharp(N_2)$

Once we have the assumption and the idealised form, we can start to verify the authentication protocol. Sending message 2 leads to:

$$U \triangleleft \{S \mid\sim (N_1, N_2, U \overset{K_{US}}{\longleftrightarrow} S)\}K_C$$

in more details

$$U \models S \mid\sim (N_1, N_2, U \overset{K_{US}}{\longleftrightarrow} S)$$

This message contains nonce N_1 that U believes to be fresh. So we can deduce:

$$U \models \sharp((N_1, N_2, U \overset{K_{US}}{\longleftrightarrow} S))$$

Hence, U knows S exists. Using the nonce verification rule, we have:

$$U \models S \models (N_1, N_2, U \overset{K_{US}}{\longleftrightarrow} S)$$

or

$$U \models S \models U \overset{K_{US}}{\longleftrightarrow} S$$

From the jurisdiction rules, we can deduce

$$U \models U \overset{K_{US}}{\longleftrightarrow} S \ (1)$$

In the third message, again, since S believes that nonce N_2 is fresh, we can deduce

$$S \models U \models (N_2, U \overset{K_{US}}{\longleftrightarrow} S)$$

and make the conclusion

$$S \models U \models U \overset{K_{US}}{\longleftrightarrow} S$$

or

$$S \models U \overset{K_{US}}{\longleftrightarrow} S \ (2)$$

From (1) and (2), we can obtain the final belief to conclude that the protocol achieves its goals.

5.2 Efficiency

The efficiency of the authentication model is also derived from two main components: the group key management and the authentication module.

5.2.1 The Efficiency of Group Key Management

The following problems are discussed:

1. Rekeying is a process to re-new authentication and group keys after being used. This process reduces the compromised key risks and other security risks. Not many group authentications have this process. In comparison with the approaches using centralised and synchronised distributed key database, the rekeying process in group key management is clearly more secure and efficient with the hierarchical tree structure.
2. In comparison with other group key managements, the proposed group key management has more efficient join operation without rekeying overhead.
3. Handoff is a regular problem in wireless networks. Handoff can cause major perfomance header in rejoining to a group key management. The rejoin procedure after an handoff often causes the major overhead from rekeying to warrantee backward secrecy in the other group key managements. However, this group key management does not have backward secrecy. Therefore handoff does not affect the efficiency of this group key management and the group authentication model.

5.3 Anonymity

Group authentication provides users and other services partially anonymous authentication to the services. Although group members have to authenticate individually with the group key management, their identities are not revealed to the services in the same cluster which they authenticate to. The only information that services can extract from the authentication messages includes the group keys and group, cluster identities. Therefore the group authentication can provide anonymous authentication to the services.

For some services that require clear identity for authentication, the authentication can combine the group identity with the individual identity for authenticate individually to the services. This combination can form two-factor authentication for the services.

6 CASE STUDY

In this section, an authentication system for a portal of wireless multimedia content services is examined. In this system, users are allowed to access services through paid memberships from different wireless service providers. Services are provided by different companies. These services are classified into four groups of services based on the package that the portal provides. The service group one contains online news, music, mobile games and video on demand. These services are provided by a wireless media company. This company also provides the services in service group two as an advanced package. Service group two has wireless news, music, video on demand, children education games, and wireless TV. Service group three includes stock exchange information, currency exchange and real estate services. The last group encloses adult services from an adult entertainment company for restricted memberships.

The group authentication model is applied to provide anonymous authentication for the portal. The user is only required to authenticate to the group key management at portal once. After that he is allowed to access the different services from different groups depend on his subscription.

The following table compares the advantages and disadvantages between

1. this group authentication,
2. Zwierko's group authentication scheme (Zwierko and Kotulski, 2005),
3. Damodaran's group key management for group authentication (Damodaran et al., 2006), and
4. the traditional individual authentication Kerberos model in this case study.

Table 1: Comparison between different group models.

	1	2	3	4
Anonymity	Yes	Yes	Yes	No
Number of Messages	3	6+	3	6+
GKM	Yes	No	Yes	No
Rekey	Yes	Yes	Yes	No

The comparison table shows that this group authentication model is more efficient in rekeying and authentication operations in groups. It is also more secure by using session keys in every authentication session. Besides, it also can provides the anonymity which is very important in wireless networks communities. In this model, not all processes can provide services freely to the system. They have to authenticate to the group key management to obtain the right to provide the services to a certain group. So that an adult entertainment service cannot provide access to the users in the normal children entertainment groups.

7 CONCLUSIONS

The paper proposes a group authentication using forward secrecy group key management. The forward secrecy group key management is used for secure authentication key exchange. Based on this efficient group key management, the group authentication protocol uses three messages to provide anonymous group authentication for the users and the services in the same clusters. Depending on the security and efficiency policy of each cluster, the rekeying operation is invoked to renew the previous cluster and group keys. The rekeying operation makes the group authentication keys become one time session group authentication keys. The rekeying process in group key management can minimise the compromised authentication key risks from security threats. The authentication model can be applied securely and efficiently for group of services and users in wireless networks.

REFERENCES

Amir, Y., Nita-Rotaru, C., and Stanton, J. R. (2001). Framework for authentication and access control of client-server group communication systems. *Lecture Notes in Computer Science*, 2233:128.

Burrows, M., Abadi, M., and Needham, R. (1990). A logic of authentication. *ACM Transactions on Computer Systems*, 8(1):18–36.

Challal, Y. and Seba, H. (2005). Group key management protocols: A novel taxonomy. *International Journal of Information Technology*, 2(1):105–118.

Damodaran, D., Singh, R., and Le, P. D. (2006). Group key management in wireless networks using session keys. *Proceedings of the Third International Conference on Information Technology: New Generations*, pages 402–407.

Dijk, M. V., Gehrmann, C., and Smeets, B. (1998). Unconditionally secure group authentication. *Designs, Codes and Cryptography*, 14(3):281–296.

Hanaoka, G., Shikata, J., Hanaoka, Y., and Imai, H. (2002). Unconditionally secure anonymous encryption and group authentication. *Lecture Notes In Computer Science*, 2501:81–99.

Handley, B. (2000). Resource-efficient anonymous group identification. *Financial Cryptography*, 1962:295–312.

Jaulmes, E. and Poupard, G. (2002). On the security of homage group authentication protocol. *Lecture Notes In Computer Science*, 2339:106–116.

Martucci, L., Carvalho, T., and Ruggiero, W. (2004). A lightweight distributed group authentication mechanism. *INC2004 - Fourth International Network Conference*, pages 393–400.

Rafaeli, S. and Hutchinson, D. (2003). A survey of key management for secure group communication. *ACM Computing Surveys*, 35(3):309–329.

Rubin, A. D. and Honeyman, P. (1993). Formal methods for the analysis of authentication protocols. (CITI Technical Report 93-7).

Zwierko, A. and Kotulski, Z. (2005). A new protocol for group authentication providing partial anonymity. *Next Generation Internet Networks*, pages 356–363.