Efficient Anonymous Key Exchange Protocol for Roaming in Wireless Networks

1Joshua J. Tom, 2Prof. Boniface K. Alese,
3Prof. Olumide S. Adewale, 4Dr. Aderonke F. Thompson
1,2,3,4Department of Computer Science, Federal University of Technology, Akure, Nigeria
e-mail:1tomjoshua.tom@gmail.com, 2bkalese@futa.edu.ng, 3adewale@futal.edu.ng, 4afthompson@futal.edu.ng

Abstract

Group signature schemes existentially provide anonymity, non-repudiation and can make a mobile device untraceable. But using group signature only in designing anonymous key exchange system is time wasting and consumes much of other computing resources hence the use of it must be minimal especially when deployed on resources-constrained mobile devices. In this paper, we propose the combined use of group signature because of its inherent security properties which are very important when a mobile user roams in the insecure wireless network and message authentication code to reduce the huge computational burden occasioned by group signature’s expensive public key operations resulting in unbearable authentication latency. In this paper, we built two authenticators, a signature based authenticator and a message authentication code based authenticator. These models are based on the Canetti-Krawczyk model. We implemented the design using Java 8 on Android Studio 2.2 and tested it on Genymotion based emulator on Oracle VM VirtualBox. The experiment result showed a significant reduction in the authentication time when message authentication code based authenticator was executed compared to authentication time result in the group signature based authenticator. We further compared our model with a similar model and find that ours was superior in efficiency measured from authentication latency when a user roams from his home domain to a visited network and other networks subsequently while maintaining same security characteristics.

Keywords: Wireless network, Roaming, Authentication, security, Anonymity, untraceability, Authentication latency, Group signature, message authentication code, mobile device, Anonymous key exchange, Visited network

1. INTRODUCTION

Nowadays we have witnessed the expansive deployment of wireless networks locally and internationally. This, expectedly, is enabled by availability and affordability of mobile devices including PDAs, smart phones, etc. This proliferation enhances the mobile users’ mobility as these technologies allow people to get connected seamlessly, have access to, and enjoy normal network services as they move about from their local domain to a foreign domain without being limited by geographical coverage of their home network. A wireless network that provides its subscribers with the ability to access services while outside its coverage is said to provide roaming services. Before access is granted to roaming users by foreign networks, these users must be authenticated. Authentication involves ensuring that network services are not obtained fraudulently hence, it is very crucial that the identities of mobile users engaged in roaming must be authenticated to prevent illegal use of resources, etc. However, the authentication credentials of these mobile users must be protected from third parties to ensure privacy of users. Anonymous roaming is a key requirement when people roam among visited networks. Allowing third parties, such as eavesdroppers, access to users’ roaming data can have very serious consequences to the detriment of the user. Firstly, the real identity of the mobile user must not be known by anyone else except the mobile user and the home network. This is referred to as identity anonymity. Secondly, the location of the mobile user should also be kept secret from other network users while abroad. We refer to this as location anonymity. A privacy policy specification can require the home network to gain knowledge of the real identity of a mobile user only when it is extremely necessary such as when the foreign network bills the home network for services enjoyed by the mobile user or when the mobile user exhibits malicious behavior or breaks one the rules a warrant of revocation can be issued by a constituted authority to reveal such user’s real identity.

To realize the scenario above an anonymous authentication protocol design is a good way to achieve it. We must always take cognizance of the limited computational resources of mobile terminal to keep the protocol not too complex. In a roaming scenario, there are typically three entities involved: the home network, the roaming mobile user and the visited foreign network. It is natural for the home network that must have a roaming agreement with the visited network. Where this is applicable, the visited network must authenticate home network’s subscribers before granting them access to its roaming services they roam to visited network. Most existing authentication protocols such as those of [6], [7], [8], and [9] require a mobile user, the visited network, and the home network to participate in the roaming service. These protocols
adopt three-party authentication technique. The disadvantages of this technique include: first, many interactive message flows are needed (not less than 4 flows). Here it takes a long time for the home server to communicate with the visited server because they are far away from each other; second, the home server must always be online and available making it vulnerable to authentication problem; third, according to [10], since the visited server must verify messages by communicating with the home server, there may be a Denial-of-Service (DoS) attack on the protocol because the home server will be verifying a lot of messages sent to it from visited servers. The structure for a three-party authentication is shown in figure 1.

![Three-Party Authentication Architecture](image)

Figure 1: Three-Party Authentication Architecture

To improve upon the lapses of most three-party authentication protocols, recently anonymous authentication schemes involving only the roaming mobile user and the visited network have received significant research attention and have made some progress. These protocols which do not need the real-time participation of the home network in the authentication process are referred to as two-party authentication protocols shown in figure 2. Examples of two-party protocols include those of [11] and [12].

![Two-Party Authentication Architecture](image)

Figure 2: Two-Party Authentication Architecture

Although authenticating a mobile user is pertinent, the privacy of the user is paramount. Roaming authentication protocols exposes the users’ privacy in terms of their identities and location information at the authentication phase. The disclosure of a user’s identity may allow unauthorized entities to track the user’s movement history and current location and therefore create users’ behavior profiles. Any illegal access to information relating to the user’s location without his attention can be a serious violation of his privacy. Identity and location anonymity are important properties for roaming services and these anonymity characteristics are the focus of this paper.

### 2. RELATED WORKS

In [1] an anonymous authentication protocol based on group signature for mobile roaming networks was carried out. The protocol involves only a mobile user and the user’s visited server, without the real time involvement of the user’s home server until the user’s identity (ID) needs to be revealed. The protocol gives a method that to reduce the computational burden the mobile terminals, minimizes expensive pairing operations. It was shown that the protocol greatly reduces the average authentication latency while security and anonymity are still preserved. The limitation of this protocol is that foreign servers are given information to identify revoked users and this information can enable the foreign servers to link other protocol runs involving the revoked user. Hence the protocol does ensureunlinkability.

In [2] the authors proposed a privacy-preserving universal authentication protocol, called Priauth, which provides strong user anonymity against both eavesdroppers and foreign servers, session key establishment, and achieves efficiency. Most importantly, Priauth provides a way to trace users’ signatures in individual period, an approach that adequately tackles the problem of user revocation and ensures unlinkability while supporting strong user untraceability. But the protocol utilized expensive and resources-hungry bilinear pairing-based group signature both when the user connects to the first foreign network and in subsequent connections to other foreign networks. This has great negative impact on the overall authentication process. A short group signature scheme that supports Verifier-Local Revocation (VLR) is constructed in [3]. In this model, revocation messages are only sent to signature verifiers and to both signers and verifiers. This is appealing for systems providing attestation capabilities. Their signatures are as short as standard RSA signatures with comparable security. The security of their group signature scheme is based on the Strong Diffie-Hellman assumption and the Decision Linear assumption in bilinear groups. They gave a precisemodel for VLR group signatures. The weakness of their protocol lies in the fact that protocol runs involving the same user could be linked. Therefore the scheme does not satisfy the backward unlinkability. Again, since the protocol does not consider differentiation in application of the group signature during different stages of the authentication, the computational load involved is bound to be high.

In [4] the authors identified verifier-local revocation as an approach of membership revocation in group signatures. In the approach, only verifiers are involved in the revocation mechanism, while signers have no

www.ijsret.org
involvement. They proposed VLR group signature schemes with the backward unlinkability from bilinear maps and asserted that their protocol is suitable for mobile environments since signers have no load. Here, we do nor seem to agree totally with this assert since group signature schemes are known to be naturally computationally demanding in terms of computing resources. We believe that anonymity, untraceability, revocation, etc. could still be achieved by using a scheme which is computationally less demanding.

The authors in [5] proposed a novel protocol to achieve privacy-preserving universal authentication protocol for wireless communications. This is to secure the communication as the data are sensitive or to ensure that the user is billed for services rendered in their scheme, revoking multiple users is to associate a key with every nonempty subset of users in the group such that if one or more users are revoked, the VA uses the key associated with the subset of the remaining users to encrypt the new key and transmits the new group key to them. The advantage of this approach is that the communication overhead is only one message for revoking any number of users. However, the number of keys stored by the VA and the users is exponential in the size of the group. Also, this approach suffers from all the users in the subgroup being automatically revoked. In such case, their protocol runs can be linkable. This approach has two costs overheads; Storage is computed in terms of keys that each user (respectively, VA) maintains and revocation cost is computed in terms of the encryptions performed, and the number of messages transmitted, by the VA.

In view of related literatures, it is obvious that the focus of most of the literatures is either choosing primitives to ensure privacy while trading off computational expense on the communicating parties or being resource efficient while trading off privacy concerns. Hence, this paper proposes a new model that pays attention to both privacy of the mobile user and computational efficiency through reduced latency.

3. MODELLING AND SIMULATION

3.1 Methodology

We divide the authentication process into two; (1) when the mobile user first roams to foreign network (2) when the mobile user subsequently roams to a different foreign network. We propose to use these different approaches because group signature is quite an expensive cryptographic scheme therefore minimal usage is advised. Our protocol deploys group signature scheme message authentication code in the two authentication processes respectively. The reason for this combination is that group signature existentially provides anonymity and untraceability and message authentication code is less expensive in terms of computational complexity. In the group signature scheme, when U roams to a visited network V, U can sign messages on behalf of the group without showing its ID. By verifying the group signature, V is sure that U is one of the valid users of H. The home server does not need to be always online and few message flows are needed for authentication [12].

In consideration of the resources restrictiveness of mobile devices, the foreign server is made to sign messages destined for a mobile user using the Elliptic Curve Digital Signature Algorithm (ECDSA) scheme since ECDSA signature verification takes less time compared to its group signature counterpart. Details of our protocol model are as follows.

Our protocol is based on the system architecture is shown in figure 3 and is consists of two phases: an initialization and authentication phases. The authentication phase is divided into two parts as earlier stated: authentication process when U connects to V₁ and authentication process when U connects to V₂.

![Figure 3: Protocol System Architecture Model](image)

To achieve revocation and unlinkability, the verifier local revocation with backward unlinkability (VLR-GS BU) group signature variant is used when U connects to the first visited server, V₁ and mutual authentication code (MAC) when U roams to the second visited server, V₂.

3.2 Protocol Initialization Phase

H (also known as the group manager) computes initialization parameters. The parameters include

(i) \( H \) selects a random number \( \gamma \in \mathbb{Z} \) and computes a group manager private key (gmpk) computed as:

\[
gmsk = \gamma
\]

(ii) The group manager public key (gmpk) is given as,

\[
gmpk = (g_1, g_2, h_1, h_2, \ldots h_r, w)
\]

where given \( F_q \) a finite field with an elliptic curve \( E \), \( G_1 \) a multiplicative cyclic group of prime order \( p \) and \( G_2 \) a multiplicative group of exponent \( p \), with some power of \( p \) as its order, \( g_1 \) is a generator of \( G_1 \) and \( g_2 \) is an order-\( p \) element of \( G_2 \). The elements \( g_1 \) and \( g_2 \) will be selected at random as part of system setup. \( h_1, h_2, \ldots h_r, w \).
... let $h_T$ represent randomly selected $h_j \in G$ for each interval $j$ and $w$ is given as $w = g_2^y$.

(iii) $H$ uses the above computed parameters to generate a vector of $N$ user’s secret keys $usk$ and a vector of $N \times T$ revocation tokens $(urt)$ for each registered user with current time intervals to ensure unlinkability as follows:

$$usk = (usk[1], usk[2], \ldots, usk[N])$$  \hspace{1cm} (3)

$$urt = (urt[1][1], \ldots, urt[1][T], \ldots, urt[N][T])$$  \hspace{1cm} (4)

(iv) The user secret key for $N$ users is given as:

$$usk[i] = (A_i, x_i)$$  \hspace{1cm} (5)

Where $A_i = g_1^{1/(y + x_i)}$ for all $i \in [1, N]$ and $x_i \in \mathbb{Z}$ is selected randomly.

(v) Next, randomly selected $h_j \in [1, T]$ is used to compute the revocation token at time interval $j$ of user $U$ with secret key $(A_i, x_i)$ as:

$$B_{ij} = h_j^{x_i} \quad \text{for all } i \in [1, N] \text{ and } j \in \mathbb{N}$$  \hspace{1cm} (6)

(vi) $H$ computes an alias for each registered user intending to roam using secret splitting mechanism from:

$$\text{alias}_U = (w || ID_H) \oplus ID_U \oplus ID_H$$  \hspace{1cm} (7)

Where $ID_H$ is the Home Server identity, $ID_U$ is the identity of the user, $U$ and $\oplus$ is an Exclusive-Or operator.

3.3 Authentication Phase

(1) Authentication Process when $U$ connects to $V_1$:

A pairing-based verifier-local revocation group signature with backward unlinkability and the conventional digital signature scheme, ECDSA are used in this process. The process is shown in figure 4 and illustrated as follows:

(i) $U$ selects a random number, $N_U$ and computes $N_U G$ (we assume $G_1 = G_2 = G$, from bilinear map) and sends $(ID_H, N_U G)$ to $V_1$.

(ii) $V_1$ selects a random number $N_{V_1}$, computes an ECDSA signature, $\sigma_{V_1}$ with its secret key, $sk_{V_1}$ as follows:

$$\sigma_{V_1} = \text{ECDSA}.sig(sk_{V_1}, N_{V_1} G, NUG)$$  \hspace{1cm} (8)

and sends $(ID_{V_1}, N_{V_1} G, \sigma_{V_1})$ to $U$. $ID_{V_1}$ is the identity of the first network visited by $U$. After that, $V_1$ computes a session key used between $V_1$ and $U$ as follows

$$k_{UV_1} = N_{V_1} (N_U G)$$  \hspace{1cm} (9)

(iii) $U$ verifies $V_1$ ‘s signature, $\sigma_{V_1}$ with $V_1$ ‘s public key, $pk_{V_1}$ by running ECDSA verification algorithm, ECDSA.ver($pk_{V_1}, N_{V_1} G, NUG, \sigma_{V_1}$) and accept or reject connection based on the verification result as follows:

$$\sigma_{V_1} = \begin{cases} 
1, \text{ valid accept connection} \\
0, \text{ valid reject connection}
\end{cases}$$

If the signature $\sigma_{V_1}$ equals to 1, $U$ computes a session key to between $U$ and $V_1$ as follows:

$$k_{UV_1} = N_{V_1} (N_U G)$$  \hspace{1cm} (10)

It then computes a temporary alias by encrypting $\text{alias}_U$ given to it by its home network using its session key $k_{UV_1}$. Then it computes a group signature, $\sigma_u$ with $usk[i]$ as:

$$\sigma_u = G.\text{Sig}(gmpk, usk[i], j, \text{alias}, N_U G)$$  \hspace{1cm} (11)

and sends (alias, $\sigma_u$) to $V_1$. Otherwise, if the signature $\sigma_{V_1} = 0$, connection rejected.

(iv) $V_1$ verifies the signature from $U$ with home server’s public key, $gmpk$ by running verification algorithm, $G.\text{Ver}(gmpk, usk[i], j, \text{alias}, N_U G, \sigma_u)$.

$$\sigma_u = \begin{cases} 
1, \text{ valid, allow connection} \\
0, \text{ otherwise disallow}
\end{cases}$$

$V_1$ allows connection if $\sigma_u = 1$ and disallow otherwise.

(2) Authentication Process when $U$ connects to $V_2$:

Message Authentication Code (MAC) reduces roaming authentication time when $U$ connects to $V_2$. This is shown in figure 5 and illustrated as follows:

(i) $V_1$ passes (alias, $ID_H, k_{UV_1}$) to $V_2$.

(ii) $U$ selects a new random number $N_U'$, encrypts the message (alias, $ID_{V_1}$, $N_U' G$) using its session key, $k_{UV_1}$, and send $E_{k_{UV_1}}(\text{alias}, ID_{V_1}, N_U' G)$ to $V_2$.

(iii) $V_2$ takes out $k_{V_1 U}$ from message in (i) above and use it to decrypt alias to get $\text{alias}_U$. Then it selects a random number $N_{V_2}$, computes a session key $k_{V_2 U}$ and a new alias by decrypting the alias received from $V_1$ from:

$$k_{V_2 U} = N_{V_2} (N_U' G)$$ and alias’ $= E_{k_{V_2 U}}(\text{alias}_U)$

then, it computes a message authentication code (MAC) value $\sigma_{V_2}$ using $\sigma_{V_2} = \text{MAC}_{k_{V_2 U}}(N_U' G, N_{V_2} G)$ and sends $(ID_{V_2}, N_{V_2} G, \sigma_{V_2})$ to $U$.

(iv) $U$ computes a session key between $U$ and $V_2$ as

$$k_{UV_2} = N_U' (N_{V_2} G)$$

It then verifies the MAC value $\sigma_{V_2}$ by computing

$$\sigma'_{U} = \text{MAC}_{k_{UV_2}}(N_U' G, N_{V_2} G)$$

and compare it with $\sigma_{V_2}$ received from $V_2$. If $\sigma_{V_2} = \text{MAC}$ value $\sigma'_{U}$, then $\sigma_{V_2}$ is valid and $U$ updates its ID and computes a new MAC value as:

$$\text{alias''} = E_{k_{UV_2}}(\text{alias}_u)$$

$$\sigma''_{U} = \text{MAC}_{k_{UV_2}}(\text{alias''}, N_U G, N_{V_2} G)$$

and sends (alias’’, $\sigma''_{U}$) to $V_2$. Otherwise $U$ rejects the connection.

www.ijsret.org
(v) $V_2$ verifies the MAC value $\sigma''_U$ by computing
$$\sigma''_2 = MAC_{K_2U}(\text{alias}''', \text{N}'\text{UG}, N_2G)$$
then it compares it with MAC value $\sigma''_U$ received from U. If $\sigma''_2 = \sigma''_U$, $V_2$ takes alias''' as U’s temporary identity and $k_{UV_2}$ as new session key. Otherwise $V_2$ rejects the connection. If $k_{UV_2} = K_{V_2U}$ and alias''' = alias'' secure re-authentication.

4. IMPLEMENTATION AND RESULTS
The Protocol is implemented using Java 8 on Android Studio 2.2, and tested in a Genymotion supported VirtualBox to provide the needed virtualization support.

Figure 4: The Key Generation Initial Interface

Figure 5: roaming_table Design Structure

Figure 6: The Key Generation Result Interface

Figure 4 shows the key generation interface, where the home server generates the group public key $gpk$, a vector of N user secret keys $usk$, a vector of Nxt user revocation tokens $urt$, and a group master secret key $gmsk$. These keys are stored in MySQL database named roaming_table (see figure 5 below).

Figure 5 shows the output of the implemented key generation algorithm including a secret key for a roaming user, revocation tokens for the interval, in this case one week (7 days).

Figure 7: Group signature verified by $V_1$.

This shows group signature received from mobile subscriber for verification. The result of implementation of the verification algorithm is shown in figure 7 as the received group signature is verified by the FS-Authenticator. The total signature-based authentication is also shown to be 117095 milliseconds.

Figure 8: ECDSA signature verification
The implemented ECDSA signature is shown in figure 8 as the mobile user received the signature from FS-Authenticator for verification 8. When mobile user roams to V₂, V₁ sends mobile user’s authentication information to V₂. Figure 9 and 10 show key exchanges between V₁ and V₂.

5. ANALYSIS

5.1 Security Analysis
The design of our protocol is based on the CK-Model [13]. Based on this, we design our key-exchange protocol in an idealized model, and then translate it using general tools to obtain security in the realistic model. It includes three parts: Authenticated-links Adversarial model (AM); Unauthenticated-links adversarial model (UM); Authenticators. AM is considered as an idealized model where the communication links are perfectly authenticated; UM is the realistic setting of adversary-controlled links; Authenticators are the so-called general algorithm tools that translate a protocol which is Session-key secure (SK-secure) in AM into a protocol which is SK-secure in UM. Since the protocol design in this paper follows from the CK model, our protocol guarantees security.

5.2 Anonymity and Unlinkability
User anonymity is achieved due to the existential anonymity of a special group signature algorithm which employs verifier local revocation. With this algorithm, V is not able to obtain the identity of the real signer since it does not have U’s revocation token u[i][j]. Only U’s home server H has. User untraceability is also achieved by the anonymity of the deployed scheme. When U₁ exists in the revocation list, RL of H during a particular interval j, V can obtain u[i][j] and uses it to make sure that U₁ is revoked for interval j. V cannot link U₁’s protocol run during any interval J₁ to u[i][j]. because the revocation token of each user changes for every interval j where j₁ ≠ j. Based on this, our authentication and key exchange system can achieve anonymity and unlinkability of U₁’s protocol runs during past and future periods. Besides, nobody including V₁ knows U’s new temporary ID, so none of the entities taking part in the communication protocol can identify and trace U. Even if V₁ gets N₁₂G and N₁G, it still cannot compute Kᵥ₂ᵤ = N₁₂G(N₁G) due to DDH assumption, and thus cannot get alias which is encrypted using KV₂U.

From the above analysis, our protocol can ensure secure and anonymous key exchange between the communicating parties.

6. EVALUATION AND COMPARISON
The difference between our protocol and that of [12] is that when U roams subsequently to other foreign networks, UPK operations are constant (still 29.515 ECSR + 12.95Pairing) but only 11.5ECSR are needed in our protocol. The authors in [1] did a similar work and obtained a better result than the result in this work but in their protocol when a user is revoked, the users previous cannot protocol runs are linkable. This means that their protocol does not provide backward unlinkability which our protocol provides.
The protocol developed in this paper is compared with the protocol in [12] in terms of user public key (UPK) operations and UPK computation latency. When U first connects to a network, 8.75 Elliptic curve scalar multiplication (ECSM) plus 12.95 Pairing operations are needed in both protocols. When U roams to another network, UPK operations are still 8.75 ECSM plus 3 Pairing in [12], but only 2 ECSM operations are needed in our protocol.

Let \( X_{UV_1} \) be UPK operations when U connects to V1, and \( X_{UV_2}, X_{UV_3}, \ldots, X_{UV_n} \) be UPK operations when U roams to networks \( (V_2, V_3, \ldots, V_n) \), then we calculate the average UPK operations as follows:

\[
X_{\text{Average}} = \frac{X_{UV_1} + X_{UV_2} + \cdots + X_{UV_n}}{n}
\]

where \( n \) is the number of networks the user has roamed.

Going by equation (12), \( X_{\text{Average}} \) in [12] is 29.515 ECSM + 12.95 Pairing. In the case of the protocol developed in this paper,

\[
X_{\text{Average}} = \frac{29.515 \text{EC} + 12.95 \text{Pair} + 2 \text{EC}(n-1)}{n}
\]

\[
= 11.5 \text{EC} + \frac{29.515 \text{EC} + 12.95 \text{Pair}}{n}
\]

In equation (13) above 29.515 ECSM + 12.95 Pairing represents the user public key operations involved in authentication process when the user roams to the first visited network using group signature scheme in [14]. The time of authentication obtained in our case is 117,095 milliseconds (0.12 seconds approx.).

Let \( L_{UV_1} \) be computation latency for UPK operations when U connects to V1, and \( L_{UV_2}, L_{UV_3}, \ldots, L_{UV_n} \) be that when U roams to networks \( (V_2, V_3, \ldots, V_n) \), then the average latency is computed as follows:

\[
L_{\text{Average}} = \frac{L_{UV_1} + L_{UV_2} + \cdots + L_{UV_n}}{n}
\]

Based on the authentication delay of 117,095 milliseconds obtained in this work (see figure 7), 1 ECSM and 1 Pairing need 2300 milliseconds and 3800 milliseconds respectively in a terminal with a 2.20GHz processor, so \( L_{\text{Average}} \) in [12] would be permanently 117,095 milliseconds, but in our protocol it is:

\[
L_{\text{Average}} = \frac{117095 + (n-1) \times 26450}{n}
\]

\[
= \frac{117095 + 26450n - 26450}{n}
\]

\[
= 90645 + \frac{26450}{n}
\]

\[
= 26450 + \frac{90645}{n}
\]

In this evaluation it can be seen from equation (15) that the bigger \( n \) is the smaller the computation latency. This is not the case for a scheme employing only the expensive group signature example [16].

Figure 11 is the average latency comparison between the protocol developed in [12] and our protocol according to the user’s roaming frequency. From the figure we can see that, the more frequently the user roams among different networks, the less time it will take in authentication.

7. CONCLUSION

According to the results obtained in this work, generating a group signature imposes a significant burden on the computing terminal and needs 117,095 milliseconds for authentication. On first connection to a foreign network, it may be bearable for U to wait this long. When U roams to another network such as V2, re-authentication process has to restart, and U has to wait another 117,095 milliseconds. It will become unbearable for U to be interrupted regularly when U roams among foreign networks frequently. Because group signature weighs so much on computing resources minimal usage of it is needed. It can be adequate to make use of a group signature when U connects to V1, but it is not necessary to do that again when U roams to V2. This is the reason why this work employs message authentication code which greatly reduces roaming authentication time to 339 milliseconds.

User anonymity is easy to be revoked by its home server; untraceability is achieved because when a user is revoked in a particular time interval, say j, he cannot be linked to any of his previous protocol runs; the foreign servers do not have to communicate with the home server to authenticate the user, so it needs few message flows. The analysis shows that the protocol is practical for roaming in wireless networks in which users are roaming frequently.
REFERENCES


