Anonymity and Untraceability in Mobile Networks *

Didier Samfat   Refik Molva
EURECOM Institute
2229, Route des Crêtes
06904 Sophia-Antipolis - FRANCE
{*molva,samfat*}*@eurecom.fr*

N. Asokan
Department of Computer Science
University of Waterloo
Waterloo, Ont. N2L 3G1, CANADA
nasokan@uwaterloo.ca

Abstract

User mobility is a feature that raises many new security-related issues and concerns. One of them is the disclosure of a mobile user's identity during the authentication process, or other procedures specific to mobile networks. Such disclosure allows an unauthorized third-party to track the mobile user's movements and current whereabouts. Depending on the context, access to any information related to a mobile user's location or activity without his consent can be a serious violation of his privacy. This new issue might be seen as a conflicting requirement with respect to authentication: anonymity requires hiding the user's identity while authentication requires the user's identity to be revealed in order to be proved. What is needed is a single mechanism reconciling both authentication and privacy of a mobile user's identity. The basic solution to this problem is the use of *aliases*. Aliases insure anonymity by hiding the user's identity as well as his relationship with domain authorities. In this paper, we present a classification scheme to identify the different pieces of information which should be protected from legitimate network entities and unauthorized third parties. We then present an efficient method for the computation of aliases and apply it to a new set of inter-domain authentication protocols. We demonstrate that these protocols can be designed to meet various degrees of privacy requirements. In designing these protocols, we try to avoid the drawbacks of authentication protocols in existing mobile network architectures such as CDPD and GSM.

Keywords: authentication, anonymity, mobility, security, GSM, CDPD, alias, location privacy

*The work described herein was funded by the IBM Zürich Research Laboratory

1 Introduction

Distributed computer systems are becoming increasingly common in the daily lives of many people. Often, people have to interact with distributed systems in order to obtain goods and services. Payment with credit cards, access to the Internet are some examples of such interactions. A common feature of these transactions is authentication: users are required to provide a claimed identity and prove it. In open distributed systems an eavesdropper can easily watch the messages exchanged during this process. If sufficient care is not taken, he may be able to see or infer information such as who is involved in a specific transaction, where and when it was performed etc. The protection of identity information or *anonymity* is therefore an important issue in the design of authentication protocols.

In systems where mobility is possible (both wireless networks and wireline networks with support for mobility), information about the locations and migration of users is vulnerable in the same manner. Typically, whenever a user, registered in his *home domain*[1], shows up in a foreign domain, he may need to prove his identity and solvency to the domain being visited (*visited domain*) in order to obtain services. Usually, the home domain needs to be contacted in this process. The messages exchanged may reveal private information about traveling patterns and locations of users to eavesdroppers located along the message paths. Therefore, in addition to anonymity, protection of movement information or *untraceability* is another similar issue.

In general, there are many aspects to the privacy of a user in distributed systems. Privacy of message contents is often the primary, if not the only, security issue addressed in the design of distributed systems. Encryption is a common technique used to achieve privacy of message contents. The preceding paragraphs illustrate privacy issues of a different kind: the protection of "secondary" information. This includes information about identities of entities involved in a transaction or meta-information arising out of the activities of and the interactions between entities in a distributed system (e.g. the number of times a mobile user has visited a certain store in a given period of time).

For the purposes of this paper, we use the word *privacy* to mean the protection of secondary information; we do not concern ourselves with the privacy of message contents within the context of this paper.

An intuitive solution to the problem of providing privacy to a roaming user is to assign *aliases* for him while he is traveling. Aliases must be chosen in such a way that it is possible to provide the required kind of privacy. For example, using a long-term alias makes it difficult to provide untraceability. The degree of privacy required depends on various factors such as the security policy being enforced, the cost vs. benefit trade-off etc. For complete privacy, only the user should know any information regarding his identity, location or movements. Depending on the circumstances, a lesser degree of privacy may be satisfactory or even mandated.

In this paper, first we present existing mechanisms for providing privacy, notably in banking systems, followed by a *new* classification of the different levels of privacy requirements. We then discuss other issues in mobile networks, such as user interface and resource limitations, that have an impact on the provision of privacy. Finally, we present a summary of the drawbacks of solutions in existing mobile networks and then describe and evaluate a set of strong authentication protocols suitable for mobile users, with provision for privacy.

2 Privacy in Other Contexts

Provision of anonymity is not a new feature. David Chaum [12, 13] and others have done extensive work in developing techniques for secure, untraceable electronic transactions in banking environments or in fixed networks. In this section, we describe some of Chaum's approaches in particular and discuss their

[1] A *domain* consists of regions and entities that fall within a common administrative control.

applicability to mobile computing environments. The following is a bare-bones description of some of the detailed techniques outlined in [13].

The first mechanism provides untraceability of payment transactions. A bank announces public keys corresponding to various denominations of money. A customer can convert his money into a "digital coin" by generating a serial number (which is derived from a random number), "blinding" it with another random number and getting the bank to sign the blinded quantity with the public key of the desired denomination. The bank will deduct the amount of money corresponding to the chosen denomination from the customer's account. The customer then "un-blinds" the signed digital coin by removing the blinding random number. Now, the digital coin can be given to a shop along with the corresponding serial number. The shop can verify that it is indeed a valid coin since it knows the bank's public key. When the shop presents the digital coin to the bank, the bank will credit the shop's account and record the serial number of the coin in order to disallow double spending. The bank cannot link the coin and the customer because it cannot know the serial number when it signed the coin. This approach is essentially a capability-based approach which obviates the need for authentication. Notice that the the shop cannot detect double spending all by itself. Only the bank has the serial numbers of coins already used. Therefore, at least when large sums are involved, the shop will have to perform on-line validation of the coins presented to it before its transaction with the user is completed.

The second approach allows a user to obtain credentials from an organization and present them to a *different* organization without either organization being able to link these activities to the same user. The user identifies himself by a different alias with each different organization. He can obtain a credential Cr_A from Organization A, to which he is known by the alias a. The credential Cr_A is essentially a signature using A's secret key. However, it also contains information about the various aliases by which the user is known to different organizations. Organization A, however, cannot extract these other aliases during the signing process. This is achieved by using the same blind signature technique described earlier. The user can present Cr_A to a different organization B to which it is known by the alias b. B will be able to verify that Cr_A was in fact signed by A and that it belongs to b. But it cannot infer any information about its other aliases.

One way to implement a blind signature scheme is described in [13]. Suppose Alice wants to have a number (corresponding to a coin or some other data) signed by Bob without Bob's knowing what the number is.

First, Bob computes a public modulus n, a public exponent d and a matching secret exponent d^{-1} using the RSA algorithm [15]. All subsequent arithmetic is carried out modulo n.

In order to get a blind signature from Bob, Alice first picks a random number r and derives a data string s from it. The idea is to require a certain structure in the data string; for example, in the case of digital coins, it can be deemed that a valid data string (sequence number of the coin) should be a substring repeated; in this case concatenating r to itself will yield a suitable sequence number. Alice also picks a random number b and computes $b^d.s$. This quantity is then handed to Bob. Bob cannot determine s since it has been "blinded" by multiplying it with b^d.

He now "signs" this entire quantity by raising it to the power d^{-1}. Bob may demand that Alice submit several candidates and ask her to un-blind a subset chosen by Bob in order to make sure that the data conforms to the required structure. The result, $b.s^{-d}$ is handed back to Alice. Alice can now un-blind this by dividing it with b. In the end, Alice possesses a signed number and Bob does not know what the number is. This description is necessarily brief. See [13], [12] for further details.

One could envision ways in which these techniques can be adapted to a mobile network. The primary problem with these techniques is the amount of resources they require. Storage and processing capacity is typically at a premium on mobile devices. In the case of wireless networks, network bandwidth is also a limited resource.

Secondly, these techniques strive to provide *complete* untraceability. This may not be acceptable under all security policies. For example, complete untraceability also sets the stage for "perfect crimes."

This calls for a general approach that can provide different levels of untraceability without imposing extensive resource requirements that might render the solution impractical in mobile networks.

3 Classification of Privacy Requirements

As we alluded to in the preceding sections, the required level of privacy depends on various factors like the cost incurred by providing this service, the perceived benefits from such a service, practical constraints and so on. To help choose the level of privacy necessary for a given environment, it is necessary to develop a classification scheme to represent the various possible levels of privacy.

Specific privacy requirements can be represented by means of a two dimensional matrix. Therows of the matrix correspond to the objects of privacy, that is, the information which can violate the privacy of users when disclosed, and the columns of the matrix represent the subjects that consist of the entities whose access to the private information should be prevented or allowed according to the privacy requirement that is expressed by the matrix. If a subject represented by a column of the matrix should be prevented access to a particular privacy information represented by a line of the matrix, the corresponding matrix entry is marked 0 (no information). If this subject may access all or some of the privacy information then the entry is marked 1 or s respectively.

In the case of mobile networks, the subjects are: the user (U), the home domain (H), the remote domain (R), legitimate network entities (L) (such as other authorized third parties involved in a transaction) and eavesdroppers (X) (unauthorised third parties). Since U is a trivial subject class containing just one entity that *always* has access to all the traceability information about itself, we omit it in subsequent discussion.

The objects of privacy requirement are: the full identity of the user u, the identity of the home domain (affiliation) h, and the identity of the remote domain r.

This scheme gives rise to a whole spectrum of anonymity and untraceability levels. We identity five particular cases of interest as to the knowledge of relationship between the entities and the identification information.

- C_1: **Hiding User Identity from Eavesdroppers.** Most of the existing solutions address this basic privacy requirement. The current location of the user cannot be directly associated to the user's identity. The resulting policy can be formulated as follows:

	H	**R**	**L**	**X**
u	*1*	*1*	*1*	**0**
h	*1*	*1*	*1*	*1*
r	*1*	*1*	*1*	*1*

  In the Global System for Mobile communications (GSM) [1], the use of Temporary Mobile System Identifiers (TMSIs) is intended to meet this requirement. When the user appears for the very first time in a new foreign domain, he needs to establish temporary residency with the foreign administrative authority. In this step, a long-term alias can be assigned to the user for the duration of his stay. The main problem with this approach is that all activity performed in the remote domain can be linked to this single alias. After a while, the relationship between this alias and the user's home domain may be discovered by traffic analysis.

  However, a basic requirement is that derivation of successive aliases should not lead to the disclosure of the identity. We need to envision different ways of assigning aliases to mobile users during their migration so that this additional requirement is met as well. A more secure alternative is to assign a different alias each time the user accesses a service in the visited domain. This avoids the disclosure of the user's relationship with the foreign authority (subject R).

- C_2: **Hiding User Identity from Foreign Authorities.** In an environment where there is only a privacy of class C_1, the visited domain can still track the user's movements. In some situations, there is no need for the foreign authority to know the identity of the user – it may only need proof of his solvency and enough information to bill his home authority. In this case, the policy is expressed by:

	H	R	L	X
u	*1*	**0**	**0**	**0**
h	*1*	*1*	*1*	*1*
r	*1*	*1*	*1*	*1*

- C_3: **Hiding Home Domain Identity from Third Parties.**

  A stronger privacy requirement is to hide the relationship between the mobile user and his home domain from intruders in order to prevent the disclosure of the user's identity by inference.

  For instance, if an aliased user x visiting a remote domain in France wants to authenticate to his home domain WhiteHouse.Gov and an intruder happens to know that the only users from WhiteHouse.Gov currently in France are President@WhiteHouse.Gov and Vice.President@WhiteHouse.Gov, the intruder can conclude that x in fact corresponds to one of these two identities.

  This requirement is expressed by the following table:

	H	R	L	X
u	*1*	**0**	**0**	**0**
h	*1*	*1*	**0**	**0**
r	*1*	*1*	*s*	*s*

- C_4: **Hiding Home Domain Identity from Foreign Authorities.** When the mobile user needs to be authenticated in a foreign domain, the foreign authority needs to contact the user's home authority in order to confirm the good standing of the user. If the number of potential traveling users from the home domain is small in the visited domain, the foreign authority can guess the identity of the user as described in the previous section. However, in environments where there are other means of establishing solvency (or where solvency is not an issue), a higher level untraceability is possible by preventing even the foreign authority from knowing the identity of the home authority. This policy corresponds to:

	H	R	L	X
u	*1*	**0**	**0**	**0**
h	*1*	**0**	**0**	**0**
r	*1*	*1*	*s*	*s*

- C_5: **Hiding User Behavior from Home Authority.** In some cases it might be important for a mobile user to hide his migration from his home authority. This requirement is especially important if perfect secrecy of user behavior should be guaranteed by the system, that is, if no one other than the user should know about the user's location. The resulting policy can be formulated as follows:

	H	R	L	X
u	**0**	**0**	**0**	**0**
h	*1*	**0**	**0**	**0**
r	*s*	*1*	*s*	*s*

In other words, no entity has any information about the user's whereabouts. This principle of course would contrast the intent of a "big brother" towards global observation of users' behavior.

It can be noticed that the requirements defined by each class C_i form a subset of class C_{i+1} because class C_{i+1} is obtained by increasing the constraints of class C_i.

4 Limitations and Conflicts of Privacy with Other System Requirements

In any complex system, conflicting requirements need to be balanced with appropriate trade-off during system design. Various factors, both technical and non-technical, influence the level of privacy that is appropriate in a given environment.

In this section, we describe how hardware limitations and organizational policy regarding service provisions influence the application of the privacy solution.

4.1 End-User Interface

In the case of a network that supports mobility of users (but not of computing devices), a user may have only a password or PIN for the purpose of authentication [11]. Users who need to provide their credentials in order to access services are forced to rely on the available public access equipment (i.e workstation or public terminal). We can reduce this exposure by using traveling aliases in order to avoid revealing the identity of the user to entities under the control of the visited domain. Therefore, even if the public access equipment knows the password of the user, it does not know *whose* credential it is.

The alias used should be a character string as easy to remember as the usual user identity. The only condition on the generation of this alias is that it should not be related to the user-name at the home domain. The choice of such an alias is not subject at all to the same considerations as the generation of a secret password. In the case of passwords, the security requirement is to make guessing the value hard. The goal is to choose an unusual string as a password so that a straightforward search through the list of known words would not yield its value. Unlike passwords, aliases are not secret. Therefore an alias can be sent in clear-text because there is no need to use unusual strings as aliases; common words of the dictionary are sufficient.

Nevertheless, a similar attack can be made in this kind of situation. Having the password of the user, a malicious workstation can scan a list of pre-established aliases in order to know whom the password belongs to. Little can be done in this situation unless the user changes his alias regularly. In that case, the user will need a list of traveling aliases that are easy to remember.

In networks that support portable computing devices, a mobile user in possession of a trusted device (e.g., smartcard, portable phone) can benefit from reduced exposure by having better random aliases which can be changed more frequently and in a transparent way. In fact, the end-user terminal can share additional specific information with his home authority in order to generate strong random aliases assuming that the mobile unit has non volatile memory.

4.2 Solvency of Mobile Users

Classes C_4 and C_5 may be contradictory with other (unrelated to security) system requirements. A typical situation arises when the home domain is the entity that is expected to vouch for the solvency of a user while he is traveling. In this case, the home domain must always have the possibility to revoke the mobile user's account at any time. However, it is not always the case that solvency needs to be underwritten by the home domain. For example, if the original solvency guarantee from the home domain contains limits (in terms of both amount of resources and time) and all domain level servers trust each other to some extent, a foreign domain can rely on the solvency guarantee from the previous domain from which the user wandered in. Solvency may also be defined in non-monetary terms using an anonymous capability scheme in order to perform access control. For example, a domain may be willing to provide free services to members of a club. In this case, a foreign domain may again trust any other domain level server to vouch for a user's membership in the club. In such situations, C_4 and C_5 requirements become meaningful.

4.3 Accounting and Billing

Provision of higher levels of privacy requires that the foreign domain authority be kept "in the dark" about the identities of visiting users. This level of privacy can be undesirable when accounting and billing are involved. In the case of classes C_1, C_2 and C_3, the foreign authority can still keep track of the user by recording proof of use and later asking the home authority for a "refund."

However, requirements for classes C_4 and C_5 conflict with the need for billing. If the foreign authority does not know the identity of the home domain, it will not be able to later bill the mobile user. Note that class C_5 also conflicts with the need for the home to forbid a user to consume resources in a foreign domain, after having revoked the account privileges of the user.

Alias solutions are not in contradiction with accountability/billing provided that they allow the home authority to recover the user's identity in order to invoice him or by having recourse to sophisticated techniques based on digital cash as described in [12] and elsewhere.

4.4 Incoming and Outgoing Calls

The privacy that can be achieved on a communication highly depends on the direction of the call with respect to the mobile user. In the case of outgoing calls privacy of type C_5 can be envisioned provided that the solvency of the mobile user can be guaranteed by a trusted third party.

On the other hand achieving privacy at a degree higher than C_3 is very difficult on incoming calls. In all the existing mobile network architectures, incoming calls are routed to the mobile user by an agent located in the home domain. For instance, in GSM, the routing agent is the HLR (Home Location Register) which knows the address of the MSC (Mobile Switching Center) currently serving an active mobile unit.

In order for this agent to be able to reach the mobile user, the agent continuously keeps track of the whereabouts of the mobile user. Thus this inherent networking feature violates the requirements of both C_4 and C_5 since in order to inform the routing agent on the actual location of the mobile user the authorities in the remote domain have to communicate with the ones in the home domain after each move of the mobile user. This conflict calls for an *anonymous routing* scheme that would eliminate the need to keep track of the user's whereabouts at the home domain. Since there is no practical solution for anonymous routing other than the wasteful full broadcast method, providing privacy of type C_4 and C_5 for incoming calls remains an open issue.

4.5 Intrusion Detection

Intrusion detection is an important service requirement in open networks [21]. In the case of mobile networks, this requirement may conflict with privacy requirements.

The basic principle of intrusion detection is to monitor and record the behavior of the users. Intrusion detection systems therefore require access to the user's identification information per operation at various granularities ranging from individual user names to a group or domain identification. The disclosure and recording of this information thus contrasts the privacy requirements at different levels.

5 Review of Existing Approaches

In this section, we review existing approaches for privacy in mobile networks.

The Global System for Mobile (GSM) [1] is the first digital cellular network to attempt to provide privacy to its subscribers. In GSM, privacy is provided by using aliases known as Temporary Mobile Subscriber Identifiers (TMSI). The main concern with GSM is when a user first switches on his portable phone: his identity, known as the International Mobile Subscriber Identifier (IMSI), is transmitted in the clear through the radio path (the TMSI is only allocated after this step). In this case, if the user is continuously tracked, his identity is revealed; it is therefore possible for an eavesdropper to correlate this IMSI with the TMSIs assigned subsequently. A similar situation arises when synchronization of TMSI between the user and the home entity is lost – the user is again forced to send his IMSI in the clear.

Another point of contention with GSM is that the fixed network is assumed to be secure, and all visited domains know the identity of the mobile unit. Location data are transmitted in the clear through the fixed network, thereby depriving the user of his privacy.

In contrast to GSM, Cellular Digital Packet Data (CDPD) [10] has a more secure approach. Before the authentication procedure takes place, the mobile unit engages a Diffie-Hellman key exchange protocol in order to share a secret session key with the foreign authority. Next, the mobile unit enciphers its identity with this new key and transmits it to the foreign authority, which deciphers the encrypted identity with the same shared secret key.

The first drawback of this approach is that it allows the foreign authority to know the identity of the mobile unit. The second drawback stems from the nature of the Diffie-Hellman protocol, which allows an intruder to masquerade as the foreign authority (using what is called the "man-in-the-middle" attack) and to discover, among other things, the mobile unit's identity.

With respect to the requirement classification of Section 3, both GSM and CDPD only partially covers the requirements of case C_1. Even if these approaches are reasonable in their limited contexts, they are not sufficient if higher levels of privacy is required. Providing complete privacy to mobile users requires hiding the user's identity and activities from both unauthorized parties (eavesdroppers) and authorized parties (remote administrative authorities) as described in Section 3.

In [19], Cooper and Birman consider the problem of providing privacy in a mobile network. They identify three kinds of privacy: content, participant, and location. Content privacy is provided via traditional mechanisms using shared key or public key cryptography. Location privacy can be provided by the use of *mixes* as described by Chaum [14]. To provide participant privacy, the authors use the notion of a *message service*. The message service consists of a number of replicated servers that act as repositories of messages. Participants in a message exchange can send or retrieve messages from this exchange. Using a technique that is similar to secret sharing schemes, a participant can effectively retrieve a message without the exchange or an observer knowing *which* message is being retrieved. The design of the message exchange assumes that each replicated server receives messages in exactly the same relative order.

However, the effectiveness of such technique relies on the assumption that the replicated servers can be included in the architecture of the mobile network. It is not reasonable to expect this requirement to be met on all systems. In our work, we neither require nor preclude replication of servers. Consequently, the proposed techniques are independent of the implementation details of replication in environments where it is available.

6 Reconciling Authentication and Anonymity

Anonymity might be seen as a conflicting requirement with respect to authentication, as anonymity requires hiding the user's identity in contrast with authentication that requires the user's identity to be revealed in order to be proved. In this section we present a solution for the computation of aliases which can be used to protect the identities of the different parties involved in the authentication process.

6.1 Initial Assumptions

We begin by stating that a user has one *home* which is the administrative domain where he is registered on a long-term basis. When accessing the network in each visited domain, the mobile user is authenticated with a traditional server-based authentication mechanism such as Kerberos [2] or KryptoKnight [3]. Users of a given domain are registered with the Authentication Server (AS) of that domain. The AS of a domain can be replicated or partitioned within the domain but the set of all partitioned and duplicated ASs represents a single domain-level authority.

We assume that the user has a personal device which can store information in a non-volatile memory, because little can be done for protecting the privacy of mobile users having only their user-name and password (or PIN) for authentication as described in Section 4.

The user needs a universal identification (for example home user identification) to which only the home domain can link the different aliases. This identification can be a number or a string allocated to the user at subscription time. This is particularly important for a central authority, especially when accounting and billing are involved.

6.2 Design Criteria

In order to insure good privacy to the mobile user during his migration, the alias generation must take into account the following design criteria:

- *One-time-aliases.* In order to hide the relationship between the actual user name and the alias each alias should be used at most once per security process. If the same alias is used in several security processes, a correlation might be established between the behavior of the user in his home domain (identified by the user name) and the behavior in remote domains of an anonymous user (identified by the alias) thus leading to the disclosure of the identification of the anonymous user. Furthermore one-time aliases limit the effect of a privacy breach to a single security process, if an alias can be "broken" at all.

- *No correlation between aliases.* Similarly, in order to prevent the identification of the user based on the correlation between the user name and the alias, not only the re-use of aliases should be avoided but also the relationship between a user name and the aliases that represent the same user should be difficult to establish.

- *Domain separation.* Even when assuming conspiracy of all visited domains (except the home domain) the identity of the user should not be discovered.

The first two criteria on aliases are similar to the properties required for nonces in authentication protocols in that a nonce is a number used only once and the value of a nonce is unpredictable even when all the previous values are known.

The solution should also allow the protection of the identity of some or all of the authorities involved in the authentication, in order to fulfill some of the untraceability requirements as described in Section 3.

6.3 Protocol Building Blocks

We base our design on the one-way authentication protocol (see Figure 1) borrowed from *KryptoKnight*, an authentication and key distribution service developed at IBM Research [4]. The reason for this choice is that KryptoKnight benefits from having strong authentication protocols, and provides some insurance

$$A \xrightarrow{\overbrace{E_{K_{ab}}(A \bigoplus E_{K_{ab}}(T_a \bigoplus E_{K_{ab}}(N_a)))}^{Token_{K_{ab}}(A,T_a,N_a)}, A, N_a, T_a} B$$

Figure 1: One-Way Authentication Protocol

$$A \xrightarrow{N_a, \overbrace{Token_{Kab}(N_a \bigoplus C, N_b, N_a \bigoplus A) \bigoplus K_s}^{TICK_{K_{ab}}(A,B,C,K_s)}} B$$

Figure 2: Ticket containing a shared secret key K_s

of security with respect to a number of attacks. The cryptographic messages used in KryptoKnight present some qualities that make them attractive for use in building strong untraceable authentication protocols.

Due to space constraints, we do not go into the details of and the rationales behind the KryptoKnight approach. The reader is referred to the various KryptoKnight papers (e.g. [5]) for more information.

The cryptographic token in Figure 1 is computed by applying a strong encryption function E, e.g., DES [9], with K_{ab} as the encryption key, over three inputs: a nonce (N_a), a timestamp (T_a) and the name of the message originator (A). The $\bigoplus$ symbol indicates a bitwise exclusive-OR operation. In the rest of the paper $AUTH_{ab}$ will denote the one-way authentication message of an initiator A to a responder B:

$$AUTH_{ab} = [N_a, T_a, Token_{K_{ab}}(A, T_a, N_a)]$$

In Figure 2, the initiator A is sending a ticket to a responder B containing a session key K_s to be shared by B with a third party, C. In the following sections, $TICK_{K_{ab}}(K_s)$ will denote the expression of this certificate:

$$TICK_{K_{ab}}(A, B, C, K_s) = Token_{Kab}(N_a \bigoplus C, N_b, N_a \bigoplus A) \bigoplus K_s$$

6.4 Alias Computation

The KryptoKnight protocols [7] do not provide privacy as the initiator A sends his identity in the clear to the responder B. Therefore, in order to hide the identity A from unauthorized parties, the only identification information that should be sent over the network is an alias which only the responder B can recognize as the secret representation of A. The alias computation technique is based on public-key cryptography.

Using shared secrets for alias computation necessarily implies that some clear-text information should be sent along with the encrypted value that is the alias in order for the recipient to retrieve the secret to be used to verify the alias. The intruders would then be able to trace this clear-text information to the user's name. In public-key cryptography the recipient does not need to know the identity of the origin in order to decrypt a message that is encrypted under his public-key since the secret key of the recipient is sufficient to decrypt the message. Sending the identification of the source in clear-text is thus not required in any encrypted communication using public-key cryptography.

There are many ways to compute a one-time alias for A to be read by B, using public-key techniques. The distinguishing factor is the assumptions made on the cryptosystem. In theory, the alias can be:

$$ALIAS(A) = P_b(N_a^{'}, A)$$

$P_b()$ denotes the result of the encryption with the responder's public key over two inputs: a nonce $(N_a^{'})$ and the identity of the initiator (A). The comma between the alias parameters indicates the concatenation of different blocks to which the public-key encryption is applied.

However, using such computation implies that the size of the concatenated message is not larger than the block size of the cryptosystem. Otherwise, A must be divided into at least two blocks (A_1 and A_2). The resulting alias will be:

$$ALIAS(A) = P_b(N_a^{'}, A_1), P_b(N_a^{''}, A_2)$$

In order to avoid generating another nonce $(N_a^{''})$ or explicitly dividing A into several blocks, we can use some form of chaining. A common chaining mechanism, often used with shared key cryptosystems, is *Cipher Block Chaining* (CBC) mode. Using a derive version of CBC with public keys, the alias will be computed as follows:

$$ALIAS(A) = P_b(N_a^{'}), P_b(A \bigoplus P_b(N_a^{'}))$$

However, this construction is susceptible to a dictionary attack on the second block. A simpler expression which is resistant to the dictionary attack is as follows:

$$ALIAS(A) = P_b(N_a^{'}, N_a^{'} \bigoplus A)$$

Upon receiving the alias, B obtains $(N_a^{'}, N_a^{'} \bigoplus A)$ by deciphering the alias with his secret key S_b, then the identity A is retrieved by computing $N'_a \bigoplus N'_a \bigoplus A$.

This method allows us to have a standard computation for the alias without making any assumptions about the cryptosystem used (e.g., the block size). The use of a single secret nonce, randomizes the identity A thereby providing additional resistance against guessing attacks on A. It is reasonable to assume that the identity strings are of a standard maximum size.

7 Authentication Protocols with Privacy

Many cryptographic solutions are possible for the problem of providing privacy to mobile users. The main distinguishing factor is the level of privacy needed. We develop three authentication protocols taking into account the five classes of privacy as described in Section 3. In doing so, we avoid the drawbacks of GSM and CDPD. In a mobile computing environment the proposed solutions are more suitable than the "digital cash" approach for reasons related to performance and system requirements as mentioned in Section 2.

7.1 Basic Authentication Protocol with Privacy

The basic untraceable authentication protocol is depicted in Figure 3. The main idea is to allow the user to change his alias on successive security transactions by generating a random alias each time. The following notation is used in this protocol:

- Uid – Universal identification of the end-user U in his home domain
- Uid_x – Identification of the user in domain X
- AS_h, AS_r – Authentication Servers of the home domain and of the remote domain respectively
- K_u – Strong key shared by U and AS_h

- K_{rh} – Long-term key shared between AS_r and AS_h
- K_{ur} – Location-dependent key computed as $F(Uid, AS_r, K_u))$
- P_x, S_x – Public key, Secret key pair of AS_x
- N_x – Nonce issued by entity X
- $P_x(M)$ – Encryption of message M with the public key P_x of AS_x
- $AUTH_{XY}$ – Authentication message computed by X and to be verified by Y (see Figure 1 for the complete format)
- $TICK_{K_x}(K_s)$ – A ticket computed with the key K_x and containing a session key K_s (see Figure 2
- $F(M)$ – A hash function such as MD5 [6] applied on message M
- $\bigoplus$ – exclusive-or operation (xor).

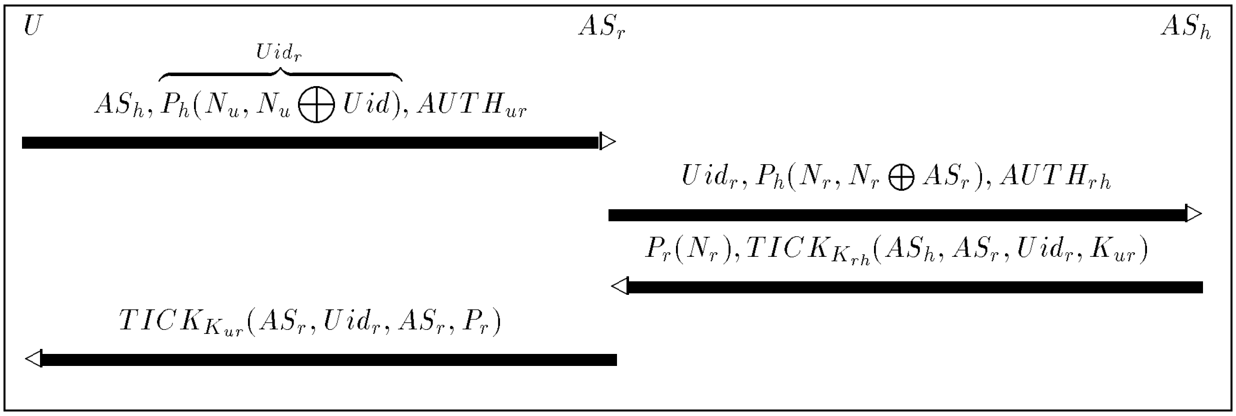

Figure 3: Basic Authentication Protocol with Privacy

This protocol provides class C_1 and C_2 of privacy as the user's identity is not revealed to onlookers, *including* all legitimate authorities except AS_h. Note also that the identity of AS_r is not disclosed to an onlooker located between AS_h and AS_r. The basic requirement is that the user's device must store his home domain public key P_h on a long-term basis. We now turn to the details of the protocol:

1. The user begins by generating a nonce N_u and his location-dependent key K_{ur} and storing them in his device. Next, he computes both his alias $P_h(N_u, N_u \bigoplus Uid))$, and his one-way authentication message using his computed key K_{ur} (the nonce used to compute the alias and $AUTH_{ur}$ are different). Then, he sends these messages to the local AS_r along with the identity of AS_h. Note that at this step, the relationship between the user and AS_h is revealed, but we can add another level of privacy as described in Section 7.2.

2. Upon receipt of the initial message, AS_r issues a nonce N_r, and saves $P_r(N_r)$ as well as the future identification of the user in the remote domain, e.g $Uid_r = F(P_h(N_u, N_u \bigoplus Uid))$ in its database (the reasons for these computations will be explained below). Next, it generates its own alias $P_h(N_r, N_r \bigoplus AS_r)$, then computes its authentication message by replacing the timestamp of the

token in $AUTH_{rh}$ by $AUTH_{ur}$ in order to prevent a guessing attack on $AUTH_{ur}$. Further details on this token chaining technique are described in [7].

3. When AS_h receives the message from AS_r (flow 2), it proceeds as follows:
   (a) It deciphers the user's alias with S_h to obtain $N_u, N_u \bigoplus Uid$. Then Uid is obtained by applying the xor operation once again.
   (b) AS_h recovers the identity of AS_r in the same way.
   (c) Having Uid and AS_r, AS_h is able to look for the corresponding shared secret keys in its database. Next, AS_h generates K_{ur} then recomputes $AUTH_{ur}$ and the chaining token $AUTH_{rh}$.
   (d) A match in this last step authenticates both the user and AS_r without revealing either Uid or AS_r to a third-party.
   (e) As AS_h needs to send a ticket containing the location dependent key K_{ur} to AS_r, it simply returns N_r enciphered with with AS_r's public key along with the ticket.

4. Upon receiving the message from AS_h (flow 3), AS_r looks for $P_r(N_r)$ in its database and retrieves the necessary information in order to read the incoming ticket. Having K_{ur}, AS_r is able to check the integrity of the key by recomputing $AUTH_{ur}$ received. In fact, sending $P_r(N_r)$ avoids the need for AS_h to compute and send its alias in flow 3. This value can be seen as a secret transaction number which identifies the authentication process involving both the user and AS_h. In other words, it allows AS_r to know who is sending the ticket and to whom K_{ur} belongs while insuring anonymity to the home AS as well as to the user.

   The reason for AS_r to record the encrypted form of N_r is to avoid having to decipher it upon receiving the response from AS_h. This has the advantage of reducing the computation operation with the private key S_r, as an immediate comparison can be done.

5. The fourth flow is purely optional as it allows AS_r to give the user his public key P_r (P_r can be given to the user by another means). This key will be used by the user during future authentications to AS_r for the computation of new aliases.

Once the user has established residence in the remote domain with the identification Uid_r , e.g. $F(P_h(N_u, N_u \bigoplus Uid))$,
he may protect this identity on the next single-sign-on using the same alias computation technique as in Section 6.3. The user issues a new nonce N_1 and provides the following message along with his authentication message:

$$P_r(N_1, N_1 \bigoplus Uid_r)$$

Upon receiving this message, AS_r is able to recover Uid_r. Therefore, this alias changing technique allows the user to vary his identity in the remote domain at every authentication process.

7.2 Enhancing the Level of Privacy

The basic protocol does not provide secrecy of the relationship between the user and his home because the identity of AS_h is revealed in flow 1. In order to have an additional degree of privacy as described in Section 3, the user needs to compute an alias for AS_h using AS_r's public key (P_r). This is illustrated in Figure 4. As the mobile user does not necessarily have P_r, before the authentication protocol starts the user may obtain AS_r's public key certificate containing P_r either from AS_r or from a public repository. This enhancement raises the level of privacy to C_3. In the next section, we describe techniques for meeting even higher levels of untraceability requirements.

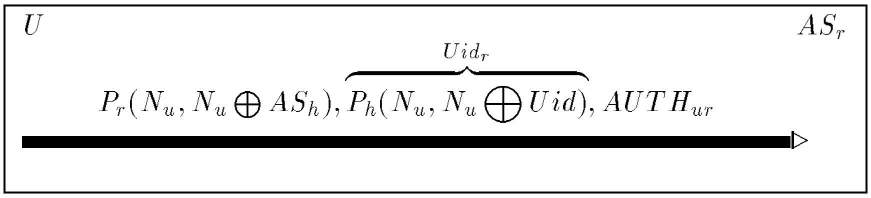

Figure 4: Protocol Hiding Identity of Home Domain Authority from Unauthorized Third Parties

7.3 The "Homeless" Authentication Protocol

So far, we have presented two protocols forcing mobile users to contact their home domain for the purpose of authentication. Figure 5 depicts an authentication protocol which avoids having to "call home." The following additional notation is used:

- $\overline{X}$ – Alias computed for identity X using the public key technique as defined in section 6.3.
- Uid_d – Identification/alias of the user in domain D
- K_{ua} – Key shared by the user with AS_a
- $\hat{K}_{ua}$ – A time-dependent key used only once, $\hat{K}_{ua} = F(K_{ua}, T_u, Uid_a)$

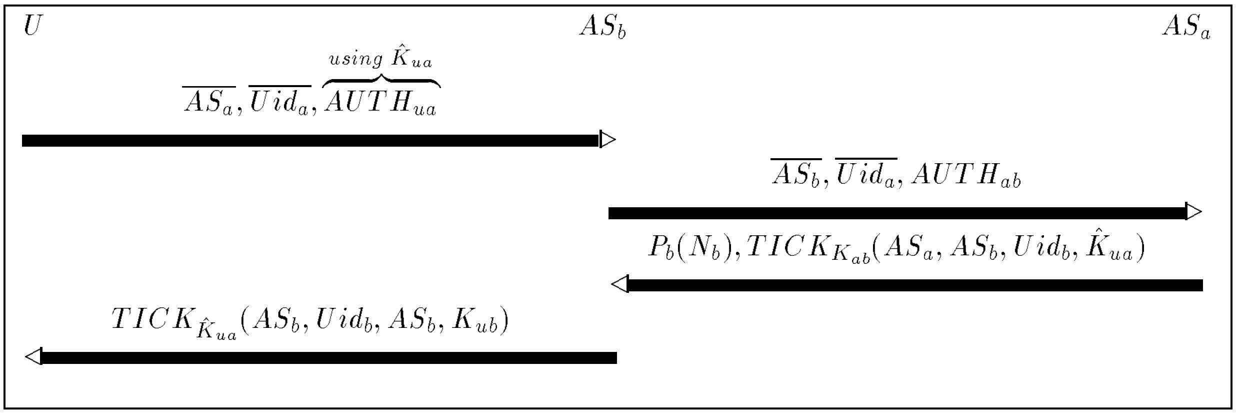

Figure 5: "Homeless" Authentication Protocol

The basic idea of this protocol is that the user needs only request a recently visited domain A to guarantee his solvency to the domain currently being visited, B. We assume that the user has already been authenticated in domain A and shares a secret key K_{ua} with AS_a[2]. As the home is no longer involved in the protocol, each domain must generate its own key to be shared with the user while he is visiting its domain.

Therefore, the user computes an alias ($\overline{Uid_a}$) for the identification Uid_a he has in domain A and another alias for the AS of this domain ($\overline{AS_a}$). Then, he generates the time dependent key $\hat{K}_{ua}$ which he uses to compute $AUTH_{ua}$.

[2]This step may have involved the home domain using the basic protocol.

Upon receipt this message, AS_b computes his authentication message (like AS_r in the basic protocol) and sends it to AS_a. If both U and AS_b are successfully authenticated by AS_a, then AS_b receives $\hat{K}_{ua}$. This one-time key allows AS_b to give his key K_{ub} to the user in flow 3. K_{ub} will be used for every authentication of U in domain B.

This protocol achieves class C_4 privacy as the home domain's identity is hidden from the foreign authority. One may think that the first foreign authority to execute the protocol will learn the user's home location, as the home authority acts as the trusted third party between between them. However, assuming that the home domain is not in collaboration with the other authorities[3], there is no means for the foreign authority to know that the trusted third party is in fact the home domain. What the foreign authority knows is that the user has provided the identification of a trusted server which can guarantee his solvency.

Class C_5 untraceability is provided partly by this protocol. The home domain will know about the first migration of the user in a remote domain. This is the only information that the home can obtain in a limited time assuming no collusion among domain level authorities. Also, each domain server may know *some* information about the user's movements, namely the domain which provided the authentication to it and the domains to which it was subsequently asked to provide authentication about the user. This protocol is better characterized as between classes C_4 and C_5.

8 Protocol Evaluation

In this section we contrast our proposal with other possible designs. We also evaluate it with respect to our design goals.

8.1 Other Possible Designs

Other solutions to the privacy problem that make use of short-lived aliases are also possible.

One intuitive approach is to have a pre-computed list of aliases kept on the user equipment and his home AS. However, this requires common state to be shared between the mobile equipment and the home domain. The aliases can only hide the identity of the user in the foreign domain, but not his relationship with the home authority.

Another problem arises when all aliases of a list have been used: in this case, the home domain must generate a new alias list and communicate it to the mobile unit. This requires either a secure channel between the user and his home, or an additional secure protocol to transfer the new aliases. These features are not always available in a mobile environment.

A final remark on such solutions is that the mobile equipment and the home AS must be continuously synchronized in order to choose the same alias in the same time. Otherwise, an additional mechanism is needed for the recovering the common state when synchronization is lost. Our alias computation method avoids all these constraints.

In particular, our solution avoids the need for synchronization altogether by using public key cryptosystems. Another approach is to make use of something that is typically already synchronized in distributed system. Herzberg et al [17] propose a method where a user's short-lived alias is computed as a function of the current time (measured with a fairly rough granularity, say to the hour) and the user's secret key (known only to the user and his home AS). The home AS continually recomputes (in the case of our example, every hour) the mapping between each user and his current alias.

Another technique proposed in the same paper is to compute short-lived aliases by encrypting the user's identity by a secret key known only to the home AS. During every authentication, the home AS provides the next alias (appropriately concealed from prying eyes) to the user. This approach too avoids the need for synchronization altogether. In particular, it allows the home AS to change its secret key

[3] AS_h does not notify AS_r of its affiliation with the user.

fairly easily unlike in our approach. However, their solutions provide only up to class C_2 privacy and cannot be easily upgraded to provide higher degrees of privacy.

Table 1 shows the differences between our protocols and existing solutions according to the classification scheme of section 3.

	GSM	**CDPD**	**Herzberg et al. Protocols**	**Basic Protocol**	**Homeless Protocol**
Privacy Policy	*Partly* C_1	*Partly* C_1	C_1 *and* C_2	C_1 *and* C_2	C_4 *and partly* C_5

Table 1: Comparison of Authentication Protocols

As described in Section 5, GSM and CDPD protocols present some weakness that allow all entities in the network to know the full identity of the user.

8.2 Computational Complexity of the Protocols

The expensive computations of a public key cryptosystem remain the private key operations. Public key algorithms such as RSA choose the keys in order to minimize both the signature verification process and the public key encryption process [15]. In the case of the low exponent RSA algorithm, the encryption takes only two modular multiplications in the basic protocol (another two modular multiplications are needed for the C_3 protocol) and thus minimizes the computation on the mobile unit.

For the normal 512-bit RSA encryption/decryption operation , it takes about 768 modular multiplication and the time for the encryption of the low exponent RSA is only 1/384 of that required for normal RSA operation. With the fastest RSA chip [22], a normal encryption/decryption operation takes 10^{-3} second.

In addition to this, new public key algorithms with lower complexity have been developed for wireless networks and are efficient in real time [16]. Therefore, in evaluating the efficiency of the protocol, we count only the total number of private key operations.

In the basic solution, AS_h needs to decipher both the user's alias and that of AS_r. In the enhanced version of the protocol, one additional operation is needed, as AS_r needs to perform another decryption operation on the alias of AS_h.

Thus, there are two private key operations in the basic solution and three private key operations in the second solution. However, these operations are performed by authentication servers which are not limited by computational resources like mobile units.

8.3 Evaluation of the Alias Solution

The alias computation solution meets the design goals as described below:

- **One-time-use alias.**

  The first time the user accesses the network, he computes an alias. On a subsequent access he is endowed with a new alias by using the chaining alias technique.

- **No direct relationship between aliases.**

  To the extent that the random number N_u in $P_h(N_u, N_u \bigoplus Uid)$ is generated by a good pseudo-random number generator, it is impossible to establish any correlation between the aliases of the same user and to predict the future values of aliases based on its past values.

- **Domain separation.**

  Even in the presence of conspiracy involving every visited AS, it is impossible to determine any relationship between aliases (random numbers) of different domains and to derive the user's identity from them, provided that the home is not part of the conspiracy[4]. Only the home domain is able to recover the user's identity from the first alias transmitted by her.

In order to minimize the computation at the mobile unit during call setup, the device can precompute a set of aliases when idle. This measure will increase the efficiency of the protocol, but will require additional non-volatile memory in the user equipment.

8.4 Analysis of the Basic Authentication Protocol

An analysis of the basic authentication protocol using the BAN logic [18] appears in [23]. It uses BAN logic to show that the protocol steps and the initial assumptions about the various beliefs held by the participants can be used to derive the goals of the authentication process. It must be noted that a BAN analysis does not provide any absolute proof. Moreover, the analysis does not address the question of privacy. We need an extended logic to reason about the preservation of anonymity and untraceability. The classification of various levels of privacy described earlier, and the notation used to describe the various levels, can be a starting point for such extensions.

9 Conclusion

This paper discusses issues related to the privacy of mobile users and their implications. It presents a general classification of privacy requirements and specifically identifies five common classes.Existing solutions for preventing the unauthorized tracking of a user's migration in systems such as GSM and CDPD exhibit some weaknesses in providing users good privacy. We have presented an efficient method for the computation of aliases that hides the identity of every entity involved in the authentication process. The method allows the user to change his aliaswithout revealing the his identity. Therefore, we have provided a new set of authentication protocols with privacy. These protocols maintain a strict separation of domains, avoiding the sharing of domain-specific security information between domains. These authentication protocols are discussed in framework of five classes of privacy requirements.

The need for higher levels of privacy for mobile users may conflict with other system requirements such as intrusion detection. Future work will include solutions reconciling such conflicts.

10 Acknowledgements

The authors are grateful to Gene Tsudik and Jay Black for their support and useful comments.

[4]As noted in Section 7.3, successive domains along a user's path can collude to infer *some* information about the user's movements.

References

[1] M. Rahnema, *Overview of the GSM System and Protocol Architecture*, IEEE Communications Magazine, April 1993.

[2] C. Neuman, T. Ts'O *Kerberos: An Authentication Service for Computer Networks*, IEEE Communications Magazine, Vol. 32 No. 9, September 1994.

[3] R. Molva, G. Tsudik, E. Van Herreweghen, S. Zatti, KryptoKnight *Authentication and Key Distribution System*, Proceedings of ESORICS'92, November 1992.

[4] R. Bird et al. *Systematic Design of a Family of Attack Resistant Authentication Protocols*, IEEE JSAC, June 1993.

[5] R. Bird et al. *The KryptoKnight Family of Light-Weight Protocols for Authentication and Key Distribution*, IEEE/ACM Trans. on Networking, February 1995.

[6] R. Rivest, *The MD5 Message Digest Algorithm*, Internet DRAFT, July 1991.

[7] R. Molva, D. Samfat, G. Tsudik, *Authentication of Mobile Users*, IEEE Network Magazine, Special Issue on Mobile Communications, March/April 1994.

[8] W. Diffie and M. Hellman, *New Directions in Cryptography*, IEEE Transactions on Information Theory, November 1976.

[9] National Bureau of Standards, *Federal Information Processing Standards*, National Bureau of Standards, Publication 46, 1977.

[10] *Cellular Digital Packet Data (CDPD) System Specification*, Release 1.0, July 19, 1993.

[11] European Telecommunications Standards Institute, *Universal Personal Telecommunications*, ETSI NA7 WP1, November 1992.

[12] D. Chaum, A. Fiat and M. Naor, *Untraceable Electronic Cash*, Proceedings of Crypto'88, August 1988.

[13] D. Chaum, *Security Without Identification: Transactions Systems to Make Big Brother Obsolete*, CACM Vol. 28, No. 10, October 1985.

[14] D. Chaum, *Untraceable Electronic Mail, Return Addresses and Digital Pseudonyms*, CACM Vol. 24, No. 2, October 1985.

[15] R. Rivest, A. Shamir and L. Adleman, *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*, CACM Vol. 21, No. 2, February 1978.

[16] M J.Beller, L F. Chang, Y. Yacobi *Security for Personal Communications Services: Public-Key vs. Private Key Approaches* Proceedings of 2nd International Symposium on Personal, Indoor and Mobile Radio Communications, October 1992.

[17] A. Herzberg, H. Krawczyk, G. Tsudik *On Traveling* Incognito Proceedings of First IEEE Workshop on Mobile Computing and its Applications, December 1994.

[18] Michael Burrows et al., *A Logic of Authentication*, Digital Systems Research Center, Technical Report 39, February, 1990.

[19] David A. Cooper and Kenneth P. Birman *Preserving Privacy in a Network of Mobile Computers*, IEEE Symposium on Security and Privacy, May 1995.

[20] D. Samfat, V. Devernay, C. Bonnet *A GSM Platform for Intrusion Detection*, Proceedings of IEEE International Conference on Communications, Vol. 2, pp 766-771, Seatle, June 1995.

[21] B. Mukherjee, L.T. Herbelein, K.N. Levitt *Network Intrusion Detection*, IEEE Network Magazine, Vol.8 No. 3, May/June 1994.

[22] M. Shand and J. Vuillemin *Fast Implementations of RSA Cryptography*, Proceedings of the 11th IEEE Symposium on Computer Arithmetic, Los Alamitos, CA, 1993.

[23] D. Samfat, R. Molva, N. Asokan *Untraceability in Mobile Networks*, to appear in the Proceedings of MobiComm'95, Berkeley, November 1995.