Untraceable Mobility or How to Travel *Incognito* *

Giuseppe Ateniese[†] Amir Herzberg[‡] Hugo Krawczyk[§] Gene Tsudik[¶]

January 14, 1999

Abstract

User mobility is rapidly becoming an important and popular network feature. This is especially evident in wireless/cellular networks where user mobility raises a number of important security issues and concerns. Foremost among them is the ability to track mobile users' movements and whereabouts. Ideally, no entity other than the user himself and a responsible authority in the user's home domain (if any) should know both the real identity and the current location of the mobile user. At present, most environments supporting user mobility either do not address the problem at all or base their solutions on assumptions that are specific to today's cellular phone networks.

This paper discusses a number of issues related to anonymity and location privacy in mobile networks. It reviews current state-of-the-art approaches, identifies their exposures of anonymity and proposes several low-cost solutions which vary in complexity, degree of protection and assumptions about the underlying environment.

Keywords: Anonymity, Mobility, Privacy, and Wireless Networks.

1 Introduction

A typical situation arising in a mobile environment occurs when a user (or a device owned by a user) registered in a certain *home domain* requests services while visiting another *foreign* domain. Typically, a foreign domain needs to verify that the user is entitled to services, and/or charge the user for these services. Furthermore, in order to obtain some basic service, such as forwarding user's incoming data to the foreign domain, the user's home domain also needs to identify the user. This issue has received considerable attention in the recent years [1, 3, 4, 8, 6, 7].

This mobile user scenario raises some concern for *privacy*. In particular, a mobile user would usually prefer to remain anonymous with respect to the foreign domain, as well as to potential eavesdroppers. Ideally, only the home domain authority should be informed as to the mobile user's real identity, itinerary and current whereabouts. (In some cases, it is even necessary to hide user's movements and whereabouts from the home domain.) We refer to these issues collectively as the *mobile anonymity problem*.

One obvious anonymity approach is for a home domain to identify each traveling user by a unique *alias* (e.g., a random quantity) instead of a meaningful user name. However, this provides only very basic protection. A fixed alias still makes it possible for a malicious observer to *link* multiple communication sessions emanating from the same user. Consequently, a mobile user can be *tracked*. This serves as motivation for stronger, *unlinkable* anonymity, using *dynamic aliases*, i.e., a distinct and unique alias for each connection.

In this paper we discuss a range of methods for generating dynamic aliases in order to prevent tracking and address the anonymity problem for mobile users. The methods vary in complexity, assumptions about the mobile devices and in degree of protection.

*An earlier version of this paper was presented at IEEE Workshop on Mobile Systems and Applications [20].

[†]DISI, Università di Genova, Via Dodecaneso 35, 16146 Genova - Italy. *ateniese@disi.unige.it*

[‡]IBM Haifa Research Lab - Tel Aviv Annex, Israel. *amir@haifa.vnet.ibm.com*

[§]Department of Electrical Engineering, Technion, Haifa 32000, Israel, and IBM T.J. Watson Research Center, New York, USA. *hugo@ee.technion.ac.il*

[¶]USC Information Sciences Institute, Marina del Rey, CA. *gts@isi.edu*

In this paper we focus on techniques for anonymity where the user's home domain is always aware of the user's current location. There are two reasons for this:

- Many environments involve communication through the home domain (at least initially) for every inter-domain hop made by a mobile user.
- Most fee-based mobile communication services aggregate roaming charges in a single location – the home domain – and subsequently present a user with a collective bill. Therefore, the user's movements become known to the home domain.

We note that this level of anonymity (where the home location is aware of the user's location) is not as strong as the anonymity provided by most electronic cash (e-cash) payment mechanisms, such as those described in [9, 10]. In e-cash mechanisms, no entity – neither seller nor bank, nor even the combination of the two – is aware of where a particular user spends her money.

The remainder of the paper is organized as follows. In the next sub-section, we overview the problem scope. We then outline our model of anonymity in Section 2. Current anonymity approaches are discussed in Section 3. Next, several new anonymity approaches are described and contrasted in Section 4.

1.1 Application Scope

One application domain in obvious need of anonymity is cellular telephony. Indeed, the anonymity problem has received the most attention in the wireless/cellular context. The two leading cellular telephone standards, GSM in Europe and CDPD in North America, early on recognized the need to protect the identities of mobile users. (We review their respective approaches in the next section.)

Somewhat surprisingly, anonymity has not been addressed in the more traditional, wired network and internetwork environments. A possible explanation is that user mobility is not yet widespread in today's wired networks. One exception is the *anonymous remailers* which have been steadily gaining popularity on the Internet, e.g., Mixmaster, similar Cypherpunks remailers [20] and others [16]. (An anonymous remailer is, essentially, a clearinghouse for anonymous newsgroup postings and anonymous electronic mail.) Another popular application domain for anonymity services is anonymous web access, such as that provided by the *Anonymizer* service [21] and the *Crowds* system [25].

In summary, even though it is most pressing in mobile network environments, anonymity is an issue wherever user mobility is supported. We now describe two such environments; our results are applicable to both, with obvious adjustments.

Anonymity can be an important factor in the all-too-familiar electronic banking. Most banks offer some form of electronic services and participate in inter-bank ATM networks such as CIRRUS or STAR. Typically, a customer is issued an ATM card by his "home" bank. Armed with an ATM card, a customer can (among other things) withdraw cash from a multitude of ATM-s located throughout the world. The ability to obtain instant cash is very useful, however, the customer's identity is revealed every time an ATM is used. This makes it possible for foreign banks or ATM providers to track the movements and whereabouts of the customer.

Another familiar environment is the ubiquitous credit card payment system. A typical consumer who is shopping and paying for goods or services with a credit card (e.g., Master Card, VISA, American Express, etc.) discloses his identity to the payee (i.e., the retailer or service provider.) Moreover, the identity of the payee is revealed to the central clearinghouse and, subsequently, to the organization that issued the credit card. All this is, strictly speaking, not necessary. Ideally, a consumer should not have to reveal anything to the payee other than the confirmation of his good standing with respect to the credit card. Conversely, the identity of the payee should not necessarily be made known to the consumer's credit card issuer. There have been several proposals for (limited) anonymous credit cards, e.g., Low et al. [14].

2 Model and Measures of Anonymity Mechanisms

The anonymity problem occurs in many different settings, as described above. However, most can be mapped into the following simple model, which we use in this paper.

In our model we assume that the "world" is partitioned into administrative domains. Every user has a permanent home in one domain and each domain has at least one Authentication Server (AS) – an entity that performs authentication, key distribution, alias resolution and other security-related tasks. Our task is to allow the AS in the home domain to perform these functions, while the user is in another domain, without exposing the identity of the user.

Our model contains three types of entities:

- U_x - Mobile device or user (we do not distinguish between the user and the device). Scenarios and solutions differ in the capabilities of the mobile device. In particular, devices differ along the following properties:
  - State: some devices have only fixed (static) state, while others have dynamic state.
  - Clock: some devices include a reliable real-time clock.
  - Computation power: devices may be able to compute computationally-intensive public key operations, or be limited to simpler, shared-key cryptography operations. In the extreme, the user device may not be able to compute at all.

  Powerful devices (with dynamic state, clock and computation power) are expensive. Some of our solutions are applicable even when the user does not have a real device, but only fixed state (which may be printed on paper). Note that this scenario is important, since sometimes even a simple device is not affordable; cf., the widely used S/key authentication protocol, which does not require a device [12]. Finally, we stress that the *size* of the state is also important.
- AS_x - Authentication Server of the home domain; also referred to as the *home agent*. Solutions differ on the storage and processing requirements for the home domain.
- AS_y - Authentication Server of the foreign domain. This is sometimes referred to as the *base station* (since these are fixed locations to which the mobile unit communicates via an antenna). Other names in the literature include *foreign agent*, *visited location* and *remote domain*.

Later on, we will use the term *domain* (home or foreign) to actually mean the authentication server (AS) in that domain. Consider a user U_x (whose home domain is D_X) who travels to a foreign domain D_Y. First, U_x needs to be authenticated and a temporary record must be created in D_Y so as to facilitate subsequent accesses in D_Y. In other words, if the user plans to linger within D_Y for some time, it makes sense to establish some temporary "home" for him instead of having to contact D_X for every access.

The authentication and temporary record establishment procedure can be abstracted as shown in Figure 1. The exact format of the authentication flows is **not important** in this context; our results apply to most mobile authentication procedure (in particular, to those described in [4, 3, 13]). Regardless of the authentication specifics, the identity of U_x must be somehow communicated to AS_x. Since U_x cannot communicate with AS_x directly, all communication has to flow through the local authority, AS_y. The first authentication flow, as shown, must include a user identification field denoted $SUid$. Similarly, the second flow (from AS_y to AS_x) must also include some form of user identification; it is denoted $\overline{SUid}$.[1]

The most important aspect of this protocol, with respect to user identity confidentiality, is the computation of $SUid$.

Communication is carried out only through the foreign domain, i.e. either mobile unit to foreign domain or foreign domain to home domain. The communication channel between the foreign domain and the mobile unit is referred to as the *airlink*, since in many mobile environments this channel is wireless (e.g., cellular). The foreign domain to home domain communication channel is referred to as the *backbone*, since it traverses a fixed, wired network. Note

[1] The reason for different notation is that $SUid$ and $\overline{SUid}$ may differ, e.g., one may be an encrypted version of the other.

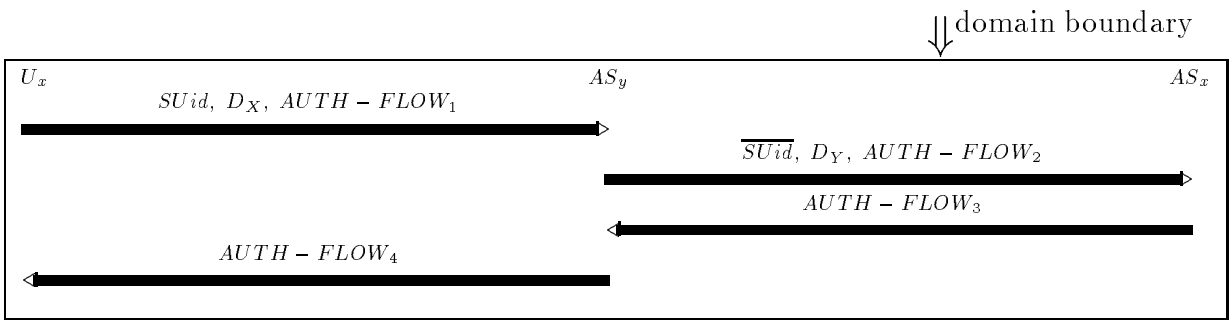

Figure 1: Model of Mobile User Authentication Protocol

that there are environments where mobile anonymity is required, but where the communication media used for these channels do not match these names (e.g., the mobile uses a wireline connection, or the foreign domain itself uses wireless communication to the home domain).

In some protocols, there is also communication between different foreign domains or with other entities. Such communication and entities are also assumed to belong to the backbone network.

2.1 Identity Confidentiality

The central issue in maintaining a secret identity is to prevent anyone from discovering a correspondence between a mobile user, using an alias, and a user name registered in a particular home domain. The intuitive first-step solution is to assign a traveling alias to every mobile user or device when away from the home domain. This alias can be fixed or constantly-changing.

A fixed alias allows attackers to *link* different sessions of the same user. Tracking a user's movements may ultimately expose the identity of that user. Also, an exposure of the real user identity in just one session would lead to exposure in other (both future and past) sessions.

Therefore, it is desirable to use *unlinkable* aliases. Informally, we consider the aliases unlinkable if the attacker is unable to identify an alias used in a given session, even if the attacker is aware of all the aliases used in other sessions. We say that the aliases are *conditionally-linkable* if the attacker is able to identify the alias only in certain (unusual) conditions.

We note that in many mobile systems, complete unlinkability against the foreign domains may be impossible. For example, in the cellular milieu, a user can migrate from one domain to the next (adjacent domain) while actively using the phone. A common technique called *hand-over* is used to pass the call state between adjacent domains. In doing so, it is inevitable that the two domains – both of which can be foreign – discover at least a segment of the user's path.

Another dimension is determined by the type of attackers that the anonymity mechanism must protect against. Different levels of anonymity may be assured against attackers on the airlink (which is usually considered vulnerable) and against attackers on the fixed backbone network, and attackers controlling the foreign domain itself. Furthermore, especially for the airlink, we distinguish between *passive attackers*, i.e., those who just eavesdrop, and *active attackers*, i.e., those who also inject, modify and corrupt messages. Clearly, it is best to have a solution which protects against attackers on the backbone and foreign domain as well as the backbone, where the attackers may be active.

2.2 Location Privacy (from home)

Finally, a relatively non-obvious anonymity issue is keeping the identities of the (visited) foreign domains secret from one's own home domain. This has little or no appeal to cellular telephony since incoming calls for a mobile user are

typically routed (at least initially) through the home domain. Nonetheless, customers wishing only outbound service, using an anonymous payment mechanism, may benefit from this feature for the reason that it keeps the individual's traveling and spending habits private. As an example, we can envisage that, upon making a purchase, a credit card customer authorizes a certain amount and the merchant *anonymously* verifies that the same amount is authorized by the credit card company. However, this type of anonymity is not a prime goal of this paper.

2.3 Hiding the home domain

The identity of a mobile user may be properly viewed as a pair: the home domain and the user identity within that domain. In many cases, anonymity requires hiding the identity of the home domain as well as the identity of the user within that domain, e.g., if the home domain is small in terms of its user population.

At the same time, a foreign domain often needs to know the identity of the home domain, in order to route messages to it and possibly for charging and/or authorization. (As a side note, one can envisage an environment where communication between domain authorities is *anonymized* by a central *clearinghouse* or a mix network. In this case, it is necessary to assign aliases to domains so that a traveling user can reference his home domain by an alias; it is then left up to the central clearinghouse to resolve domain aliases. We do not elaborate on such solutions as they require infrastructure alterations.)

2.4 Robustness and fault-tolerance

Anonymity solutions often rely on synchronized changes to the state of the mobile device and its home domain such as selecting a new alias each time. However, communication networks and devices, in particular in mobile settings, are prone to failures. Robustness to failures is therefore an important property of anonymity solutions.

We will consider *absolute robustness* (resiliency to arbitrary number of failures) or *limited robustness* (resiliency up to a limited number of failures). We will consider robustness to two kinds of failures: 1) loss of messages, and 2) loss of state. Clearly, the latter is a more severe problem.

3 Existing Approaches

State-of-the-art in mobile telephony is exemplified by *Global System for Mobile communications* (GSM), mostly used in Europe, and *Cellular Digital Packet Data* (CDPD) in North America. Outside of telephony, the recently proposed extensions to the Internet Protocol (IP) to support host mobility are likely to have major impact on mobility in data communications. In this section we provide a brief overview of anonymity services in GSM, CDPD and mobile IP.

3.1 Anonymity in GSM

An active mobile unit (cellular phone) in GSM is always under control of the local base station (whether at home or at a foreign domain). Whenever a mobile unit crosses *cell* boundary, a different base station takes over the handling of the unit. The transfer of state is sometimes referred to as the hand-off process. If the two adjacent cells belong to different domains, a somewhat more involved process takes place. Both are discussed below.

Over the airlink, GSM protects the identity of the mobile unit and its home location while transmitting them to the local foreign domain from the previous visited foreign domain. Each mobile unit, when registering with a foreign domain, gets a temporary identity (TMSI).[2] The mobile uses the TMSI instead of its actual identity, whenever it "talks" with this foreign domain. When moving to a new foreign domain, the mobile unit sends the previous TMSI and LAI (Location Area Identifier) of the previous foreign domain.

If the previous foreign domain is unreachable, current foreign domain can give up and ask the mobile to reveal its actual home and identity. This fall-back process can be exploited by an active attacker who, masquerading as a

[2] Temporary Mobile System Identifier.

foreign domain, can ask the mobile to reveal its identity claiming to have no contact to the previous base. Most implementations, in this case, would reveal the true identity of the mobile unit (called *IMSI*, International Mobile System Identifier).

Even greater opportunities for hostile tracking exist on the inter-domain level. When a mobile unit crosses the domain boundary or is simply activated in a new domain, a registration process takes place. The purpose of the latter is to establish (at the domain level) the necessary state for the mobile unit. In the course of registration, the mobile unit is authenticated with the direct aid of its home location (see [1, 4]) and a TMSI is assigned. However, authentication of the mobile involves its *real* identity (IMSI) communicated in the **clear** over the airlink.

In summary, the GSM design focuses on mobile unit's anonymity over the air link. The design offers no anonymity against foreign domains. In fact, every base station discovers not only the mobile's identity but also its previous and next base stations.

Furthermore, all information between foreign domains flows in the clear. Hence, an attacker can easily discover identities and locations by passive eavesdropping on the inter-foreign domain communication on the backbone. An active attacker may, in fact, use a foreign domain as an oracle that reveals the identities. Therefore, it does not even need to intercept messages but can issue them at its convenience. For this, the attacker would claim to be another foreign domain to which the user has connected in the past.

A final note on GSM is that its procedure relies heavily on synchronized state in the mobile unit. If this state is lost in either the mobile or previous foreign domain, anonymity is compromised. This requires base stations to keep state of mobiles even after they left. The scheme is geared primarily to support a mobile moving between adjacent foreign domains.

3.2 Anonymity in CDPD

In many respects, CDPD is very different from GSM. Differences start with the terminology. In CDPD, the equivalent of a GSM base station is *Mobile Data Base Station* (MDBS). Unlike a base station in GSM, MDBS is a low-level entity that is not involved in any security-related activity. In fact, an MDBS does not even take part in inter-cell hand-over of mobile unit's state. All of the mobility management as well security, functions are concentrated in the *Message Data Intermediate System – MD-IS.* Each MD-IS controls an *area* (i.e., domain) covered by a number of MDBSs. Therefore, it only makes sense to discuss anonymity with respect to inter-area mobility.[3]

Upon arrival to a new area, the mobile unit first engages in a Diffie-Hellman key exchange protocol[2] with the local MD-IS. As a result, both parties obtain a shared secret key. Subsequently, the mobile unit encrypts its real identity (Network Equipment Identifier – NEI) and transmits it to the local MD-IS.

While seemingly more secure than GSM, the CDPD approach has two major drawbacks. First, it allows the local MD-IS to discover the *real* identity of the mobile unit. As we argued above, the identity of the mobile unit should not be revealed to the local authority. It is sufficient for the mobile's identity and current standing to be corroborated by the home domain authority. The second problem is due to the nature of the Diffie-Hellman key exchange protocol. Its purpose is to establish a secret key on-the-fly. This means that an active attacker masquerading as the local domain authority can engage in the key exchange protocol with the mobile unit and obtain a shared key. The mobile unit then transmits its real identity enciphered with the new key and the intruder can simply decipher the entire transmission. (One obvious fix for this problem is to introduce domain-level certificates.)

See [11] for suggestions on how to improve CDPD security, including its anonymity aspects.

3.3 Common Aspects

Both GSM and CDPD view their network environment as divided into two parts: air links and fixed network. The former is the "ether" over which subscribers communicate with base stations and the latter is the rest of the network, i.e., the medium over which base stations, message switching centers and other "wired" elements communicate.

[3] In the current context, we can assume that there is a one-to-one correspondence between domains and MD-ISs.

The air links are considered wide-open and vulnerable while the fixed network is considered secure. Thus, anonymity and other security services are either not implemented or greatly relaxed over the fixed network. This is a reasonable approach if dedicated, private links make up the entire fixed backbone network. However, not all mobile environments have the luxury of a secure backbone.

Another underlying assumption is the benevolence of home and foreign domains. This means that current whereabouts and movements (path) of the mobile user are known to the home domain authority. Also, the user's identity and the identity of its home location are exposed to the foreign domain.

3.4 Anonymity in Mobile IP

Unlike GSM and CDPD, IP started out as a protocol for *fixed* networks. However, in the recent years efforts have been under way to make IP mobility-aware. The cast of characters in mobile IP includes:

- MN - mobile node (host)
- HA - home agent
- FA - foreign agent

An MN that wanders into a foreign domain establishes a relationship with the local FA. A three-way handshake among MN, HA and FA allows data traffic destined for MH to be re-routed (via HA and FA) to MN's current location. Security in mobile IP is left entirely up to IP's own security mechanisms, i.e., no additional, specifically mobile, mechanisms are defined. As pointed out in [13] protection from traffic analysis may be obtained by using encrypted bi-directional tunneling between MN and HA. However, there is no provision to hide the MN's identity from the FA or other network elements in the FA's domain.

In a recent work, Fasbender et al. [6] identified some traffic analysis issues (including anonymity and location secrecy) in mobile IP and suggested the Non-Disclosure Method (NDM) based roughly on the Chaum's MIX approach [15]. While theoretically sound, NDM requires non-trivial infrastructure alterations due to the introduction of MIX-s. It also raises some concerns regarding the performance overhead of switching IP packet traffic through a chain of MIX-s, each performing encryption.

4 Anonymity Mechanisms

In this section we present and discuss several mechanisms supporting user mobility. We start with the most primitive, *low-tech* case: a wired network (e.g., the Internet) where mobile users gain network access through a multitude of entry points (workstations, hosts, dial-up terminals, PDAs, etc.) without possessing any trusted personal equipment. In spite of limitations inherent to this "unsophisticated" environment, a certain degree of anonymity can be provided.

We then proceed to a more sophisticated scenario where users are assumed to possess a personal trusted device (e.g, a smartcard or a token-card.) First we consider anonymity protection scheme based on the conventional, shared-key cryptography model. Finally, we discuss the case with the a device based on public key cryptography.

Note that all mechanisms discussed below may be complemented with the two-flow Diffie-Hellman[2] key exchange protocol over the airlink, *a la* CDPD[3]. In this case, the entire procedure becomes resistant to passive intruders over the airlink since all messages can be enciphered under the new key. We consider this enhancement orthogonal to our techniques.

4.1 Device-less (Time-based) Anonymity

Initially, we address the anonymity problem by developing solutions suitable for relatively low-tech mobile environments where users are equipped with weak authentication devices or no devices at all. Instead, authentication is based on state remembered by the mobile user and on simple selection operations within this state.

We begin with a solution that takes advantage of the the user's clock (i.e., a watch or a wall-clock). Every domain D_X selects a domain-wide time interval δ_X which is expected to be relatively coarse, e.g., one hour or one day.

A user's alias $SUid$ is computed as:

$$SUid = F(U, T_u, PW_u)$$

where F is a published strong one-way function, such as the FIPS 186 one-way function [18] based on the hash function SHA-1[17]. T_u is the current time *rounded* to the nearest δ_X value. If the user is **not** equipped with a smartcard-like device, PW_u is the user's password which he enters into the public workstation or some such terminal. For a smartcard-bound user, PW_u can be either: 1) a strong key stored within the smartcard (for those smartcards that lack a keypad or other means of input), or 2) a combination of the smartcard's key and the user's password (for smartcards with input capabilities).

As specified, $SUid$ is unintelligible to AS_y. The only information AS_y is able to obtain is that the mobile user is registered in D_X. In the second flow, AS_y transmits $SUid$ (along with other authentication information) to the user's claimed home domain authority AS_x.

The crucial issue is how AS_x determines that $SUid$ corresponds to the locally-registered user U_x. It does so by maintaining an up-to-date table which, for every native user, lists the corresponding $SUid$ value. This translation table is computed for every δ_X interval. Since AS_x already stores the values of U and PW_u for every user, it has all the necessary information to compute up-to-date translation tables.

We note that, since $SUid$-s are not dependent on the users' current location, the translation tables can be pre-computed **off-line** and well in **advance**. This is particularly the case when a relatively coarse δ_X value is used, e.g., one hour or one day. Pre-computation is, of course, commensurate with increased space requirements (to store the alias tables) but it makes the protocol more efficient. In fact, the user can just carry the list of aliases on paper, or as a computer file, and use standard authentication mechanisms (that does not support anonymity). These requirements are similar to those of the popular S/Key authentication mechanism [12].

Finally, establishing the *real* identity of the mobile user is only half the work; AS_x must then verify the authentication information supplied in the second flow. However, this is unrelated to the problem at hand. (See, for example, [4] for a treatment of this subject.)

To summarize, this approach provides a reasonable level of conditionally-linkable anonymity, with minimal requirements on the user's device. Robustness is mainly a function of the clock synchronization, and clever techniques can deal with drifts. In the extreme, the user can just use a list of aliases on paper (but that has clear disadvantages with respect to both usability and security).

4.1.1 Reducing Computational Overhead

In an environment where only a few users travel outside their home domain, it can be quite inefficient and even wasteful to pre-compute, maintain and search time-based alias tables for all users. In this case, one simple way to reduce overhead is to require users to inform their domain authority (AS_x) in advance of traveling. This enhancement allows the domain authority to keep track of only those users that are currently traveling.

Note that this does *not* imply that users need to disclose their complete itinerary in advance; they simply need to register the beginning of each trip *abroad.* Upon notification, AS_x adds the user to a special list which is used for time-based alias computation. However, it is not necessary for the user to inform AS_x upon completion of each trip; AS_x can deduce that a certain user has returned *home* when this user tries to log in with his real userid.

4.1.2 Maintaining Loosely-Synchronized Clocks

Our assumption about the user maintaining a coarse clock, loosely-synchronized with the home domain authority is quite realistic for most environments. Clearly, a user equipped with a smartcard can rely on the smartcard's clock to keep track of T_u. For an unequipped user, a public workstation's internal clock will suffice. It is, in fact, possible for the user to enter the time manually from a wall-clock or a wristwatch (depending on the granularity of δ_X, of

course.) Despite the ease of maintaining T_u, it is still conceivable that in some cases, keeping track of T_u is not possible for some reason. To handle this situation, the protocol can be modified in a way that either: 1) the local domain AS_y provides T_u, or 2) the home domain AS_x provides T_u. In either case, T_u has to be supplied to the user (or his device) a priori, i.e., in an extra flow preceding flow 1 in Figure 1. This can be done in cleartext since T_u is not a secret value.

4.1.3 Limitations

As demonstrated above, the most important factor in traveling *incognito* is to have frequently changing and seemingly unrelated aliases. (If constant or long-term aliases are used, tracking and identity-correlation becomes possible.) Ideally, an alias is fully *disposable*, i.e., used only once. The present method is not up to that standard – it allows aliases to be re-used within the configurable δ_X time interval. Consequently, if a user migrates through multiple domains within a δ_X interval, he becomes vulnerable to limited identity tracking.

There are two alternative approaches:

1. Make aliases dependent on the visited domain
2. Maintain tight synchronization between the user and his domain authority

If the name of a foreign domain is included into the computation of an alias, correlation of identity becomes impossible since a user migrating from one foreign domain to the next (even within a very short time) will do so under two unrelated aliases. The main drawback of this approach is that the home domain authority is not able to predict its users' movements. Therefore, when AS_x is presented with $SUid$ and the name of D_Y, it is not able to resolve $SUid$ immediately, since there is no pre-computed translation table. Instead, for every registered user, AS_x has to compute the appropriate $SUid$ value using the name of D_Y as one of the inputs. This puts a substantial load on AS_x.

Another possibility is to maintain tight synchronization between the user (or, rather, personal device of the user) and the domain authority. This synchronization can be on the basis of time, secret sequence numbers or identically-seeded random number generators. This approach provides the highest level of security since it guarantees that an alias is never reused. However, it suffers from the same drawback as the domain-dependent aliases. Furthermore, it requires every user to have a reliable, tamper-proof *personal* device.

4.2 Device-Based Anonymity: Home-Centric Approach

Although suitable for personal devices (such as smartcards), the time-based anonymity approach described in the previous section is truly appealing only to an *unarmed* user, i.e., a user without any personal gadgets. We now turn to environments where users are assumed to have a personal device (gadget) which will allow us to improve the anonymity.

A user equipped with a personal device, e.g., a laptop computer, smartcard, or an intelligent cellular phone, can rely on the device to perform complex computations as well as provide secure and non-volatile storage of strong keys and other state information. Naturally, this type of environment is much more amenable to a wide range of anonymity solutions.

The main idea is that, in each registration or network access, the mobile identifies itself by an alias that was generated and communicated to it by the home domain in the previous registration. These aliases appear unintelligible to a hostile (and, perhaps, active) observer of the communication. The home authority generates each one-time alias by encrypting the name of the mobile using a strong secret key **known only** to the home authority *itself*. Since alias computation is performed entirely by the home domain authority and the mobile plays an essentially *passive* role, we refer to this approach as *home-centric*.

In computing an alias, encryption is done probabilistically, i.e., by adding some random *salt* to the name before encryption. This yields several benefits:

- Aliases for the same mobile are always different
- An alias gives away no information about the true identity of the mobile
- Compromise of one mobile unit does not compromise aliases of others and does not reveal *previous* aliases for the same mobile unit

Furthermore, the home authority only needs to perform one decryption operation in order to recover the true identity of a mobile user. Since this can be done using a strong symmetric encryption function (e.g., triple-DES), the resultant solution is very efficient.[4] (We note that a solution with totally random aliases assigned by the home authority would require keeping additional state and performing a table search in order to identify a user.)

Another important property of the present approach is that the strength of the underlying encryption function as well as the size of the encryption key is entirely up to to the individual home domain authority. In other words, a home domain authority can choose any encryption function with any size key. Moreover, it is free to change keys and *even* encryption functions periodically without any impact on the mobile users and with no significant performance degradation. This is because any change in the alias computation procedure is transparent to the mobile users as long as the home authority remains *backward-compatible*, i.e., it continues to "recognize" old aliases computed under older keys or encryption functions. In fact, there is no requirement to use conventional encryption; a home domain is free to use public key encryption with no impact on its users.

One important issue in the present solution is the traceability of a mobile user by an intruder. If we assume (naively) that one-time aliases are communicated in the clear then an intruder (even without knowing the true identity of the user) can track the user's movements by correlating the alias supplied by the home authority in one session with the one used by the mobile user in the next session. Therefore, it is necessary to encrypt the new alias when it is being communicated by the home domain to the mobile user. However, this has no impact on other communication between the mobile user and the home authority, i.e., only the new alias must be encrypted.

This method is illustrated in Figure 2. We assume that the network access is initiated by the mobile user U_x. It provides the current alias $ALIAS_i$ in the clear along with some protocol-dependent authentication information. The home authority – AS_x first decrypts $ALIAS_i$, identifies U_x and verifies the authentication information. Next, AS_x generates a new session key K, computes the new alias – $ALIAS_{i+1}$ – and encrypts it under K. Finally, AS_x sends to U_x: i) its own authentication (if applicable), ii) key distribution expression containing K, and iii) new, encrypted alias. The same procedure is repeated upon the next network access.

Our example in Figure 2 assumes that a new session key (K_i) is distributed by the home authority to the mobile user upon every network access. This is not mandatory. Instead, a mobile user (device) can share a long-term strong key with the home authority. In this case, the one-time alias (in the flow from AS_x to U_x) would be encrypted using this long-term key.

An important comment on the present approach is that it requires non-volatile memory on the user's device in order to store the current alias. This is both a blessing and a curse. On one hand, the added burden for the device amounts to nothing more than storing the alias (no computation whatsoever.) On the other hand, the alias has to be stored reliably. In the event that the current alias is somehow lost, a recovery scenario can be envisioned whereby the mobile defaults to, say, its serial number or some other *permanent* identity for just one emergency network access. It is, however, important to note that if a current alias is somehow revealed, it alone cannot be misused. Hence, storing the alias does not require the user's device to be tamper-resistant. This holds only because we are considering, for a general treatment, the `Key Distribution` and authentication processes as separated services (i.e. they could be achieved for a different purpose).

Another problem with this approach is its robustness. The protocol described above assumes that all messages are properly received. If the protocol does not complete successfully, the mobile user has to reuse the same alias, thus sacrificing unlinkability. This may be acceptable since the user did not communicate at all within the broken connection. A stronger condition may be provided by keeping several aliases in the mobile, so that if one is wasted (as a result of a broken connection), another may be used the next time.

[4] As opposed to solutions based on Public Key cryptography.

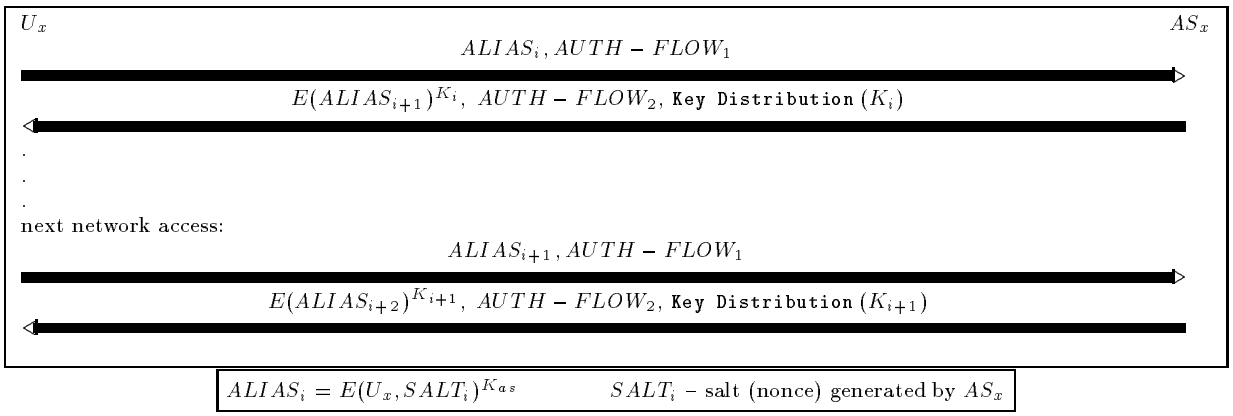

Figure 2: Home-centric Protocol Example

To summarize, this approach provides conditionally unlinkable anonymity, with a weak mobile device which must also be capable of keeping dynamic state.

4.3 Device-Based Anonymity: User-Centric Approach

As mentioned in the beginning of this section, many other device-based anonymity approaches are possible. Some rely on the mobile unit to generate aliases. This makes sense only with the aid of public key encryption. (It is easy to see that symmetric, conventional encryption would not work unless multiple mobiles share the same key with the home authority.) We refer to this approach as *user-centric* since the user is responsible for generating own aliases. An example protocol is illustrated in figure 3, where $SALT_i$ denotes a *fresh* generated random value.

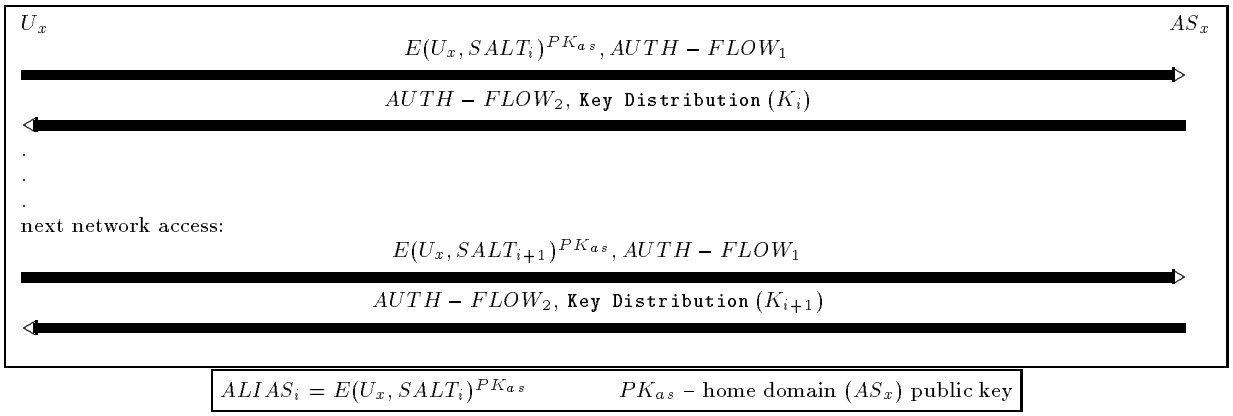

Figure 3: User-Centric Protocol Example

The use of public key encryption solves some problems, e.g., it no longer requires non-volatile storage on the device. Also, robustness is less of an issue because alias computation is done entirely by the mobile. However, all this is commensurate with certain drawbacks:

- **Higher computational overhead:** public key encryption costs significantly more than conventional encryption.

- **Increased message size:** since public keys tend to be longer (e.g, at least a 512-bit modulus for RSA) message size grows accordingly.
- **Random number generators (RNGs):** each device needs to be equipped with a RNG and each RNG must be securely seeded with a distinct, random initial value.
- **Public key change complexity:** in case of the home domain authority changing its public key, an additional (highly-secure) protocol is required.

We do not discuss this approach in detail here, however, a thorough treatment can be found in the work of Samfat et al. [5].

4.4 Contributory Aliases

Thus far, we treated authentication and key distribution as services orthogonal to anonymity. We now switch gears and integrate the key distribution and authentication services with alias-based anonymity service. In general, we observe that a one-time alias can be thought of as a session key shared by the user U_x and its home domain authority AS_x.

Hence, our goal is to construct an efficient key distribution protocol providing implicit key authentication and key confirmation. Implicit key authentication protects against active attacks such as the well-known *man-in-the-middle* attack [19]. Key confirmation, on the other hand, assures one party that its peer is in possession of the correct key. Note that, without key (i.e., alias) confirmation, the mobile user can compute an incorrect key (alias) which, in turn, can greatly complicate the next run of the protocol. In addition, all this must be done while avoiding encryption so as to stay clear of export restrictions.

Any number of protocols having the above features can be envisaged. One concrete example is illustrated in figure 4. In it, we assume that U_x and AS_x share a long-term secret key K and both parties use K to derive two distinct sub-keys: K_1 and K_2.

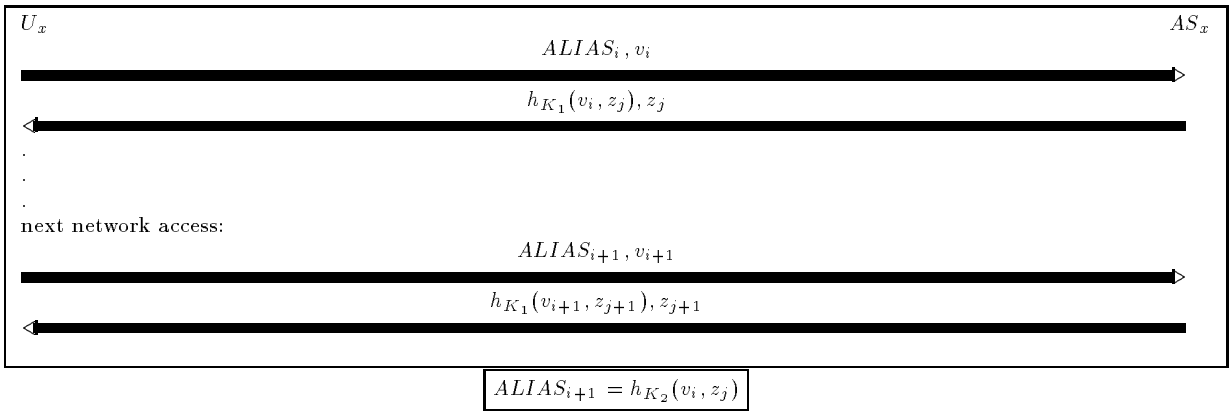

Figure 4: Contributory Alias Protocol

Let $h_k()$ denote a keyed MAC[5] with key k and n-bit output. A keyed MAC can be viewed as a hash function which, given two inputs: a message and a secret key, produces an n-bit result. One popular keyed MAC implementation is the so-called HMAC [22]. HMAC and other keyed MACs based on hash functions provide performance very close to that of unkeyed hash functions.

[5]MAC - Message Authentication Code.

The first alias, $ALIAS_1$, is part of the initial shared information. The party U_x selects a n-bit random string v_i and sends AS_x the message $\{ALIAS_i, v_i\}$. Upon receipt, AS_x selects a random n-bit string z_j and sends U_x the message $\{h_{K_1}(v_i, z_j), z_j\}$. If the protocol ends correctly, U_x and AS_x share the session key $h_{K_2}(v_i, z_j)$.

The present protocol has certain advantages:

- Messages are relatively short (in practice, no more than 320 bits using, e.g., SHA-1 [17] based MAC function);
- Protocol avoids encryption which is an advantage with respect to export restrictions;
- Key agreement and authentication techniques are fully integrated into the protocol and are efficiently implemented.
- Each alias is equally contributed to by U_x and AS_x. This results in stronger (more random) aliases if AS_x is equipped with a strong random (or pseudorandom) number generator. Then, U_x can be equipped with an efficient, yet structurally simple, pseudorandom number generator. (Note that the randomness of the entire $U_x \rightarrow AS_x$ message directly relies on the randomness of an alias.)

  For example, we can use the well-known Secure Hash Algorithm (SHA) to generate pseudorandom bit sequences by computing $SHA(s), SHA(s+1), SHA(s+2), \ldots$ starting with some secret, device-specific seed s. Although this method has not been proven to be cryptographically secure, it seems sufficient for this application. Alternatively, the FIPS 186 pseudorandom number generator [18] might be used. In addition to being fairly efficient, it allows the seed s to be augmented with user-supplied input such as a pass-phrase.

In the protocol above we assumed that all messages are properly received. In order to avoid reusing the same alias in case of a broken connection (sacrificing unlinkability), $ALIAS_{i+1}$ should be defined as $h_{K_2}(v_i, z_j, \ell)$, where ℓ is an index ranging in a specific set. Hence, several aliases may be used by the user U_x just incrementing ℓ (note that only AS_x may need to store all the aliases for any proper value of ℓ).

Remark: There are many possible attacks on MAC and keyed hash functions; see, for example, [19]. The objective of an attack is: given one or more pairs $[x_i, h_K(x_i)]$, compute a new pair $[x, h_K(x)]$ with $x \neq x_i$ and without knowledge of the key K. In some case, the adversary is able to choose or somehow influence the plaintext values.[6]

In order to make this kind of attack difficult, we can modify the protocol shown in figure 4 as follows: the user U_x sends AS_x the message:

$$\{ALIAS_i, h_{K_1}(ALIAS_i) \oplus v_i\}$$

and, in turn, receives from AS_x:

$$\{h_{K_2}(v_i) \oplus z_j\,, h_{K_2}(z_j)\}$$

(where $\oplus$ is the XOR operation.) The new alias is then defined as $ALIAS_{i+1} = h_{K_3}(v_i, z_j)$. Note that, only MAC results of unknown plaintexts with unknown keys are revealed. Moreover, the particular design of the protocol allows to hide the partial contributions v_i and z_j.

4.5 Summary

Table 1 summarizes and compares the essential features of the four schemes. Note that, aside the Contributory approach, all features are analyzed according to anonymity service, i.e. considering `Key Distribution` and authentication as separated services.

5 Conclusions

This paper discussed the anonymity problem in mobile networks and offered a range of possible solutions. The proposed techniques apply to many contexts where user mobility is supported. This includes the emerging satellite-based mobile communication networks. In such networks satellites will be orbiting cellular base stations with which

[6] These attacks are called known-plaintext, chosen-plaintext and adaptive chosen-plaintext attack, respectively.

	Method			
Feature	Time-based	Home-centric	User-centric	Contributory
Public key	NO	NO	YES	NO
Secret key	YES	YES	NO	YES
RNG req-t	NO	NO	YES	YES
Clock req-t	YES	NO	NO	NO
Addl. storage at home domain	YES	YES	NO	YES
State on device	N/A	YES	N/A	YES
Device uniqueness	NO	NO	YES	YES
Comp. overhead at home domain	Table search	Decryption	Decryption	MAC computation
Home domain key change	Transparent	Transparent	Hard	Transparent
Encryption type change	Hard	Transparent	Hard	N/A
Protocol robustness	YES	NO	YES	NO

Table 1: Comparison of Time-bases, Home-centric, User-centric and Contributory anonymity approaches

end-user devices will communicate directly. The satellites will be put in orbits that are much closer to Earth than geostationary orbit. In some cases, (e.g., Motorola's Iridium), satellites will not only communicate with terrestrial stations but also with one another. This complicated design (requiring stored routing tables) will permit global service with only a dozen gateway terrestrial stations. For example, the Iridium system [24] is already operational with only about 66 active satellites. This means that, very soon, it will be possible to call everyone from essentially anywhere on the planet using a handheld terminal similar to a cellular phone. In this context, the anonymity problem will be a central issue. In fact, it would be enough to intercept communication of just one terrestrial station or satellite to accumulate a large amount of data about all kinds of transactions and conversations.

References

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

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

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

[4] R. Molva, D. Samfat and G. Tsudik, *Authentication of Mobile Users*, IEEE Network, Special Issue on Mobile Communications, Spring 1994.

[5] D. Samfat, R. Molva and N. Asokan, *Anonymity and Untraceability in Mobile Networks*, ACM International Conference on Mobile Computing and Networking, November 1995.

[6] A. Fasbender, D. Kesdogan, O. Kubitz, *Analysis of Security and Privacy in Mobile-IP*, 4th International Conference on Telecommunication Systems Modeling and Analysis, March 1996.

[7] D. Kesdogan, H. Federrath, A. Jerichow, and A. Pfitzmann, *Location Management Strategies increasing Privacy in Mobile Communication Systems* IFIP SEC '96, Proceedings of the IFIP TC11, Chapman & Hall, London 1996, 39-48.

[8] M. Beller, L. Chang and Y. Yacobi, *Privacy and Authentication on a Portable Communications System*, IEEE JSAC, Special Issue on Wireless Personal Communications, August 1993.

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

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

[11] Y. Frankel, A. Herzberg, P. Karger, H. Krawczyk, C. Kunzinger, and M. Yung, *Security Issues in a CDPD Wireless Network*, IEEE Personal Communications, Vol. 2, No. 4, August 1995, pp. 16–27.

[12] N. Haller, *The S/Key One-Time Password System*, ISOC Symposium on Network and Distributed Systems Security, February 1994.

[13] C. Perkins and D. Johnson, *Mobility Support in IPv6*, INTERNET DARFT, Work in Progress, ¡draft-ietf-mobileip-ipv6-07.txt¿, November 1998.

[14] S. Low, N. Maxemchuk and S. Paul, *Anonymous Credit Cards*, 2nd ACM Conference on Computer and Communication Security, November 1994.

[15] D. Chaum, *Security without Identification: Transaction Systems to make Big Brother Obsolete,* Communications of the ACM, 28/10, 1985, pp. 1030–1044.

[16] C. Gülcü and G. Tsudik, *Mixing Email with* Babel, 1996 Symposium on Network and Distributed Systems Security, February 1996.

[17] FIPS 180-1, *Secure Hash Standard*, Federal Information Processing Standards Publication 180-1, U.S. Department of Commerce/N.I.S.T., National Technical Information Service, Springfield, Virginia, 1994.

[18] FIPS 186, *Digital Signature Standard*, Federal Information Processing Standards Publication 186, U.S. Department of Commerce/N.I.S.T., National Technical Information Service, Springfield, Virginia, 1994.

[19] A. Menezes, P. van Oorschot and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996,

[20] L. Cottrell. *Mixmaster and Remailer Attacks,*
http://obscura.obscura.com/~loki/remailer/remailer-essay.html

[21] The Anonymizer Web Site, *http://www.anonymizer.com*

[22] H. Krawczyk, M. Bellare and R. Canetti, HMAC: Keyed-Hashing for Message Authentication, Internet RFC 2104, February 1997.

[23] A. Herzberg, H. Krawczyk and G. Tsudik, *On Traveling* Incognito, 1994 IEEE Workshop on Mobile Computing Systems and Applications.

[24] Iridium LLC Web Site, *http:*
www.iridium.com

[25] M. Reiter and A. Rubin, *Anonymity Loves Company: Anonymous Web Transactions with Crowds,* Communications of the ACM, February 1999.