Authentication of Mobile Users

Refik Molva Didier Samfat
EURECOM Institute
CICA, 2229 Route des Crêtes
F-06560 Valbonne - FRANCE
{molva,samfat}@eurecom.fr

Gene Tsudik
IBM Zurich Research Laboratory
Säumerstrasse 4
CH-8803 Rüschlikon - Switzerland
gts@zurich.ibm.com

August 20, 1993

Abstract

Internetworks of the future will allow and promote universal access. Network users will be able to access the network at a multitude of access points separated by significant geographic distance and many administrative boundaries. Lacking a single central authority, a new set of inter-domain security mechanisms is needed to allow users to migrate into remote domains while *inheriting* privileges from their home domain. Solutions addressing this issue must take into account a somewhat contradictory security constraint that calls for strict separation of security domains in order to avoid sharing sensitive security information among them. In this paper, we propose a solution for authenticating mobile users in remote domains that maintains the domain separation property. The protocols described herein can be applied in different mobile-user environments including wireless networks and mobile user services on traditional wireline networks.

Keywords:[1] mobility, internetworks, mobile users, network services, authentication, roaming, inter-domain protocols, network security, GSM, UPT, CDPD

1 Introduction

Recent emergence of network technology which supports user mobility has prompted new security requirements and concerns. This is mainly due to the lack of physical protection mechanisms as in traditional fixed-topology, static-user networks. In addition to the increasing motivation to prevent illegal access (fraud) and eavesdropping in mobile-user networks, one new problem involving mobility is the ever-increasing distance that can separate network access points. Since network access points are not necessarily under the control of the same administrative authority, a new set of inter-domain mechanisms is needed in order to allow users to perform security operations in visited domains. Potential solutions must take into account a somewhat contradictory security constraint that calls for strict separation of security domains in order to avoid sharing domain-specific security information. The goal of this paper is to propose a basic solution for the authentication of users in remote domains while maintaining strict separation of security domains.

This paper is organized as follows. We begin in section 2 by discussing security issues specific to user mobility and identifying authentication requirements. Section 3 summarizes authentication solutions in existing mobile-user environments. Section 4 presents our solution to authentication of mobile users. Some variations on the theme are presented in Section 5 and Section 6 concludes with the discussion of open issues.

[1]DRAFT: In submission to IEEE Network, August 1993.

2 User Mobility and Its Security Implications

User mobility is a feature that can be offered in different network environments. Some environments are, by definition, oriented towards mobile users. These include all types of wireless networks such as infrared and radio (cellular being the most popular.) Other environments can be adapted to support user mobility, i.e., a wireline network can be equipped to allow universal access through offering a value-add service such as Universal Personal Telecommunication (UPT)[12]. Furthermore, the current trend is towards mixing wireline and wireless access within an internetwork. To this end, the rest of the paper tries to play down the distinction between wireless and wireline access in order to keep the discussion fairly general.

2.1 Establishing Temporary Residence *Abroad*

We begin by stating the basic assumption of user mobility, that a user has but one *home*. A user's home is the administrative domain where he is registered on a long-term basis. Typically, it is also the place where accounting and billing information is accumulated. In some sense, a home domain bears some responsibility for its constituent users.

As a mobile user migrates throughout an internetwork, he[2] periodically *pops up* in a new, foreign domain. Regardless of the type of access (i.e., via a cellular phone or a fixed workstation) the goal of a mobile user is to obtain some service from the network. A user may be simply transiting a foreign domain or planning to linger about for some time. In either case, he must establish *temporary* residence in the foreign domain in order to obtain service locally.

This process is similar to mobility in real-world, where a person physically travelling from one country (or regions therein) to another country must engage in some form of a bureaucratic procedure the purpose of which is to establish temporary physical residence in the new location.

In the real world, the procedure of establishing a temporary residence can vary from country to country. The same can be said of network domains. In some cases, it may suffice for a mobile user to possess a universal credential (passport/public-key certificate.) Thus, a foreign domain can verify and accept the credential. However, while the credential's authenticity is verifiable, its current status is not. In other words, since a mobile user demands service, a commodity that is not *free*, it is naturally desirable for a foreign domain to ascertain that a newly-arrived, visiting user is currently in good standing. This can not be accomplished without some interaction (perhaps, not direct) with the user's home domain, since only the home domain is able to comment on the user's current standing. Therefore, a plain solution with global user certification, such as the one employed by Privacy-Enhanced Mail (PEM) [10], is not applicable to the general mobile-user environment. Certificates can be still used for identifying users. It is the authorization of users that can not be solved with certificates alone. Another problem with using certificates (both public- and shared-key based) is the implied assumption that a user has some means of carrying a certificate, e.g., within a cellular phone, a personal communicator or a smartcard. However, in the simplest mobile-user environment (e.g., UPT[12]), a user is "armed" with only a password or a PIN—something that a human brain can easily retain.

Even if the user's ability to pay for services can be confirmed without contacting the home domain (e.g., with electronic cash[18, 19]), there remain two unsolved issues: tracking and access control. A user is normally registered with a home domain on a long-term basis; anyone wanting to contact a certain user will have to consult his home domain first. Therefore, a home domain should *track* the whereabouts of its constituent users.

Moreover, if the home domain implements Mandatory Access Control (MAC), a user may be required to "call home" first, before establishing temporary residence in a foreign domain. For example, a domain administered by a military-related organization may prohibit its *native* users from establishing residence and obtaining service in domains administered by foreign entities in other countries.[3] Another way of presenting

[2] No gender implication.

[3] The term *foreign* is used to denote a network domain other than the home domain; *country* is used in the traditional, geopolitical sense.

this issue is from the point of view of charging for services. If a user obtains some service while *abroad* and the foreign domain(s) later bill for it, the user's home domain may refuse to pay since the expendatures were not authorized. This calls for involving the home domain in the process of establishing temporary residence abroad.

2.2 Migration Among Foreign Domains

There is a slight difference between a user appearing in a foreign domain and moving between two adjacent foreign domains. While a user makes his way from one foreign domain to another, his credibility must be confirmed with every crossing of domain boundaries. In general, we can not assume that the path taken by a mobile user is continuous. That is, a user may operate in one domain and several hours later pop up is another domain thousands of miles away, e.g., an 8-hour flight from Switzerland to the United States carries a user between two distant, non-adjacent domains.

Somewhat different dynamics take place in a wireless environment where, instead of simply popping up, a user may *wander* into a new domain. The difference is best illustrated by an example: a cellular telephone subscriber engaged in a conversation crosses the domain boundary in real-time while, say, driving a car. This type of movement can be classified as *continuous* as opposed to *discrete* where a subscriber simply turns on his cellular phone far away from home. (The latter case is essentially the same as in the wireline environment.) This type of real-time inter-domain transition demands real-time hand-over of user's state including current session activity data and authentication/authorization information. A simple case is when a user is migrating from his home domain to an adjacent one; the hand-over can be done in a trivial manner. In a more general scenario, a user migrates from one foreign domain to another. Both domains can be very far away from *home* and it is not obvious how to authenticate a user in a new domain (in real-time) without incurring the overhead of contacting his home domain.

3 Existing Approaches

In this section we briefly review how some of the current mobile-user environments reconcile authentication with user mobility.

3.1 Current State-Of-The-Art: GSM

Groupe Special Mobile (GSM)[1, 2] is the first mobile digital cellular network architecture to provide security services such as user authentication, traffic confidentiality and key distribution.

GSM subscribers (users) are traced during their intra- and inter-domain movements. Each Mobile Station (MS) informs the network of its position; this information is used to update the Visitors Location Register (VLR) and the Home Location Register (HLR). Furthermore, the establishment of communication is under control of the Authentication Center (AUc) which is often co-located with the local Message Switching Center (MSC) where most domain-wide policy is enforced.

For each active or passive MS, real-time identification of the visited domain and authentication of both caller and callee MS is performed in order to avoid fraud.

Every GSM subscriber has in his MS a smartcard (SIM) containing a secret key K_i known only by the HLR. When MS notifies the MSC of its activity, VLR activates the mobile unit's HLR and transmits his own identity, the mobile station's International MS Identity (IMSI) and position to the HLR. HLR asks AUc for a set of triplets containing: a challenge(random number $RAND$), a signed response ($SRES$), and a corresponding session key (K_c). The triplets are then sent to the VLR and each triplet is used only once for the authentication of the MS.

Parameters $SRES$ and K_c are computed with the classified algorithms A_3 and A_8 that implement one-way trapdoor functions:

- $SRES = A_3(K_i, RAND)$
- $K_c = A_8(K_i, RAND)$

Subsequently, privacy between the MS and the local MSC is achieved by enciphering data with K_c. A_5 is another classified algorithm used to encipher data, speech and signalling messages:

- $Ciphertext = A_5(K_c, Cleartext)$
- $Cleartext = A_5(K_c, Ciphertext)$

Figure 1 depicts the MS authentication protocol in GSM. Message flows between HLR and VLR perform the export of the subscriber's credentials from the home domain to the remote domain while the interaction between MS and VLR consists of challenge-based authentication of MS by VLR.

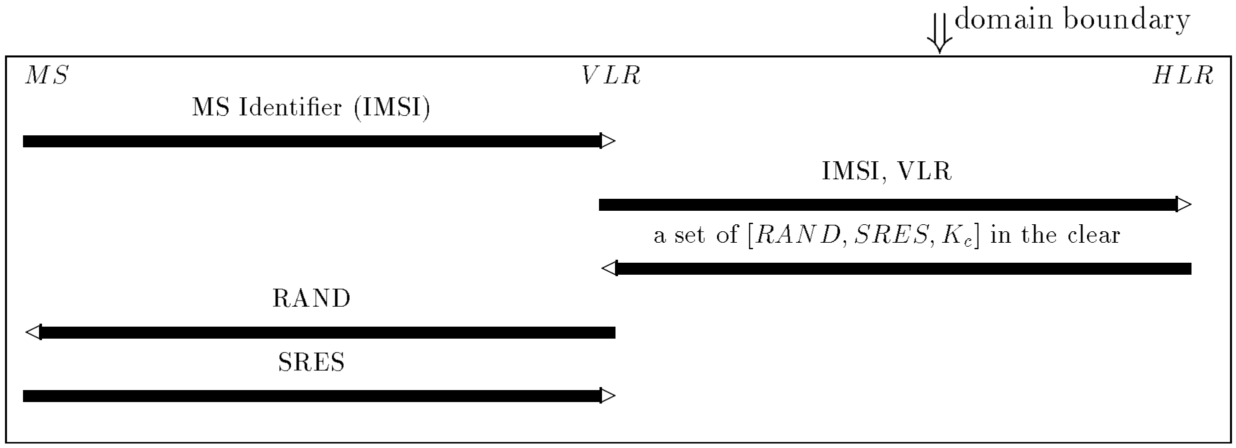

Figure 1: Mobile Station authentication in GSM

The overriding concern with the GSM authentication approach is the implicit reliance on the security of the internetwork that is traversed by the $VLR \leftrightarrow HLR$ communication. Even if this was a reasonable assumption for the signalling networks of today's mobile telephone systems, the same cannot be guaranteed in a large or global-scale, administratively heterogeneous network environment. What is needed is a security architecture with minimal assumptions about the security of intermediate transport networks.

Another point of contention with GSM is the manner of distributing the user authentication information. The home domain is expected to generate on-the-fly a set of challenge/response pairs that the foreign domain is then supposed to use in successive authentication flows with the end-user. This solution is inefficient in terms of both bandwidth consumption and the overhead incurred at the home domain. In addition, since only a (presumably) small number of such challenge/response pairs is communicated, their supply can eventually be depleted and the foreign domain would have to contact the home domain for a fresh batch.

A final remark on GSM has to do with the use of "classified" algorithms A_3, A_5 and A_8 to obtain authentication and secrecy. Hiding the algorithm is certainly contrary to the open-systems philosophy. Furthermore, the time-tried *security-by-obscurity* principle has not proven to be effective in preventing hostile attacks succeeding.[4]

3.2 Other Mobile-User Environments

The upcoming value-add service, Universal Personal Telecommunications (UPT), is a service aimed primarily at the European market. It will provide universal user access and will support both fixed and mobile

[4] If the recent Clipper proposal is any indication, the GSM "secret" solution also fails to give the informed end-user a comfortable feeling of security.

end-systems. The UPT design anticipates many types of fraudulent use and suggests some general solutions ranging from simple PIN-based authentication to more involved two-way authentication (user↔UPT-equipment) using smartcards[12, 13]. However, unlike GSM, UPT has not matured to a state where specific security solutions have been proposed.

Cellular Digital Packet Data (CDPD)[11] architecture has been developed by a consortium of (mostly, US) companies. Like UPT, it envisions a network accessible through either mobile or fixed end-systems. However, instead of being a value-add service, it is a complete architecture encompassing, among other things, a complete MAC layer. The architecture supports several network layer protocols including IP[15] and CLNP[14].

CDPD specifications are not as vague as UPT about security requirements. In particular, CDPD requires a logically-separate entity, Authentication Server to be present in every service provider's domain. User authentication involving a Mobile Data Intermediate System (MD-IS) must refer to the local Authentication Server. However, even though CDPD provides a detailed description and packet formats for inter-domain location update (see Mobile Network Location Protocol, [11] sections 500-501), no details as to the format of authentication information are provided.

4 Authentication of Mobile Users

In this section we present a solution for the authentication of mobile users. In doing so we try to avoid the drawbacks of GSM and, furthermore, make no assumptions about the trustworthiness of the intermediate networks. In addition, some usability concerns are taken into account in order to offer a transparent user interface in remote domains.

4.1 Initial Assumptions

When accessing the network in the *home* domain, the user is authenticated with a traditional server-based authentication mechanism such as Kerberos [3] or KryptoKnight [4]. Users of every network domain are registered with that domain's Authentication Server (AS). The AS of a domain can be replicated or partitioned within the domain but the set of all partitioned and duplicated ASs represent a single domain-level authority.

An important characteristic of mobile environments is the speed at which users move across domains in the network. We assume that the inter-domain travel has a relatively low frequency, i.e., for a typical user, the intra-domain migrations (be it within a home or a remote domain) will be more frequent and last longer than the inter-domain migrations.

4.2 Design Criteria

In addition to avoiding the aforementioned drawbacks of existing systems like GSM, the solution must take into account the following design criteria:

- *Domain Separation:* Domain-specific secret or sensitive information such as the user's secret key or password should not be propagated from the home domain to a foreign domain or between foreign domains.
- *Transparency to Users:* Authentication in foreign domains should have minimal impact on the user interface with respect to authentication in the home domain.
- *User Identity Confidentiality:* It is often desirable to keep both the movements and the current whereabouts of mobile users secret. For this reason, all user identification information must be protected from disclosure.

- *Minimal Overhead:* The distance between the home and the foreign domain may be very large. Hence, the number of messages exchanged between the home domain and the remote domain for the purpose of authentication should be kept minimal.

4.3 Protocol Building Blocks

We base our design on top of existing two- and three-party authentication and key distribution protocols. These protocols are borrowed from *KryptoKnight*, an authentication and key distribution system prototype developed by IBM Research[4]. The reason for choosing KryptoKnight is twofold:

- The existing KryptoKnight protocols have a number of qualities that make them attractive to build on. These include: message compactness, usage of nonces (as opposed to timestamps), formal assurance of security, use of strong one-way hash functions (as opposed to encryption), protocol flexibility with respect to different network configurations, and amenability to small hardware-based implementations.[5]
- The resultant protocol(s) will be incorporated into the existing KryptoKnight protocol family (which currently lacks any provisions for user mobility.)

In order to illustrate some KryptoKnight concepts, figure 2 illustrates the two-party mutual authentication protocol. $AUTH_{K_{ab}}$ denotes an authentication expression based on the shared key K_{ab}. It is always generated

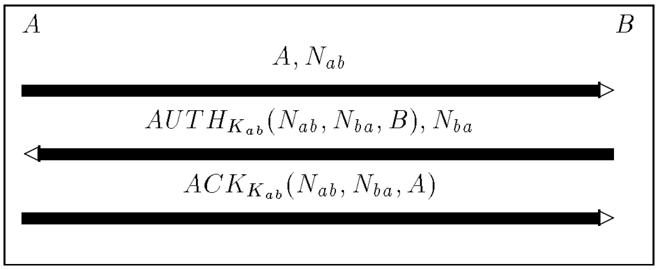

Figure 2: Secure Two-Party Authentication Protocol (2PAP)

by the responding party (B in our case). This expression is computed over three inputs: two nonces generated by each party and the name of the message originator. One example of $AUTH$ is:[6]

$$E(B \bigoplus E(N_{ba} \bigoplus E(N_{ab})))$$

Similarly, $ACK_{K_{ab}}$ is the authentication expression that the initiating party computes in order to complete two-way authentication. It is computed over the same inputs except for the message originator's name (A in this case.) An example of an ACK expression is:

$$E(N_{ba} \bigoplus E(N_{ab}))$$

Where mutual authentication is not desired, a one-way version of 2PAP can be used (see figure 3.) The protocol is reduced to just one flow and one of the nonces in the authentication expression is replaced by the current timestamp. As described in the next section, elements of both mutual and one-way protocols are used in the design of the mobile user authentication protocol.

4.4 Basic Protocol

The basic protocol is depicted in figure 4. It enables a user presently located in a foreign domain to request the transfer of location-dependent authentication information from the authentication server of his home domain to its peer in the remote domain. In other words, the protocol's purpose is to establish a temporary foreign residence for a travelling user.

[5] Refer to [4] and [16] for details.
[6] All encryption is performed with K_{ab}.

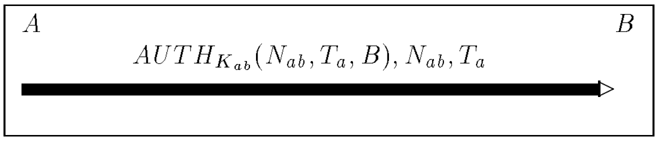

Figure 3: One-Way Authentication Protocol

The following notation is used in the protocol and throughout the rest of this paper:

- $SUid$, $\overline{SUid}$ Identification of the end-user U in different protocol flows. The exact format is discussed in section 4.6.3 below.
- AS_h Authentication Server of the home domain
- AS_r Authentication Server of the remote domain
- K_u Key shared between U and AS_h
- K_{ur} Location-dependent key computed for the remote domain, i.e., a key that user U can use only in domain R. The particulars of K_{ur} computation are discussed below.
- K_{rh} Key shared by AS_r and AS_h
- K_s Short-term session key to be shared by AS_r and U (generated by AS_r.
- N_u Nonce issued by user or by the access device on user's behalf
- T_u Timestamp issued by the user or by the access device on his behalf
- N_r^1 Nonce generated by AS_r
- $AUTH_K(X, Y, Z)$ Authentication token computed (e.g., using DES-based MAC [5]) over a tuple (X, Y, Z), under key K.
- $TICK_{\hat{K}}(A, \boxed{K}, B, C)$ A ticket (i.e., a certificate) issued by A, sealed with the key $\hat{K}$, containing a secret quantity (typically, a key) K to be used by B for communicating with C. The exact format of a ticket will be discussed below.

Many variations of the basic protocol are possible. The main distinguishing factor of this particular version of the protocol is the computation of the domain-specific key K_{ur}. It is computed as a strong one-way function over the foreign domain's identity, user's identity, and the permanent credential of the user which he uses *at home* (e.g., a password). This allows us to accommodate the least "sophisticated" user who relies only on his password or PIN and uses no special hardware to access the network, i.e., no *smart* cellular phone, smartcard or token card. (However, users equipped with personal trusted hardware can still use the same key computation method.) In the *weakest* case, a user not having any special trusted hardware of his own, is forced to rely on whatever public access equipment is available, e.g., a workstation or a simple terminal. In this case, all computation attributed to the user is actually performed by the access equipment on the user's behalf. Of course, a user in possession of a trusted personal device can benefit from reduced exposure by having all or part of the computation take place on the device.

We now turn to the details of the protocol:

1. The user begins by generating a nonce N_u, a timestamp T_u and computing K_{ur}. Next, he computes the authentication token $AUTH_{K_{ur}}(N_u, T_u, U)$ and sends it to the local AS, AS_r.

   There are many ways of constructing K_{ur} is a manner that makes it dependent on the user's current location. For example, K_{ur} can be computed as $E_{K_u}[F(AS_r, U)]$ where F is a strong one-way hash

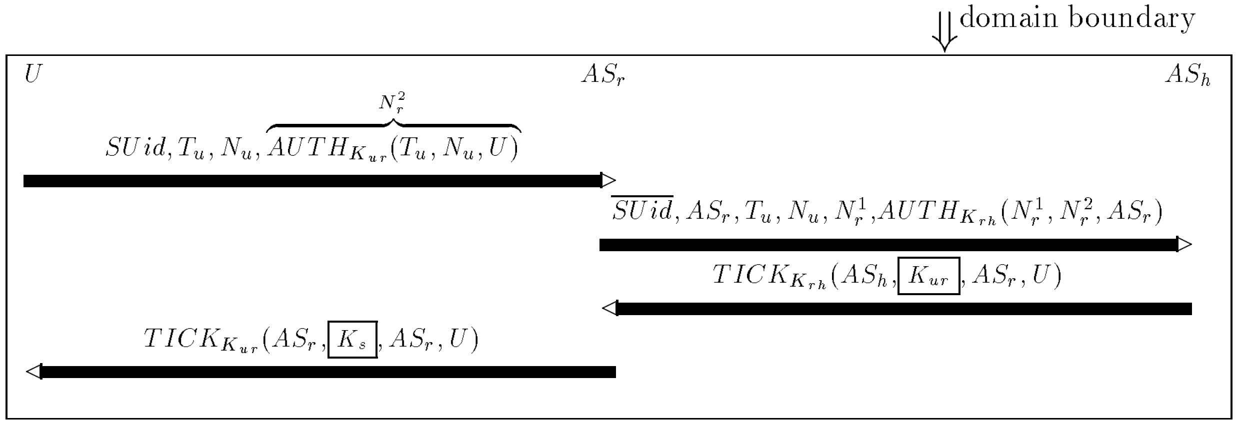

Figure 4: Basic Mobile User Authentication Protocol

function such as MD5[17] and E is a strong encryption functions such as DES[5]. We can even avoid using encryption by computing K_{ur} as $F(AS_r, U, K_u)$.

Constructing K_{ur} as described above has an important consequence that the user visting the same domain on two different occasions will continue using the same K_{ur}. At the first glance, this may seem like a serious weakness. However, recall that most users change (or are, at least, expected to) their passwords and PINs. This can provide the necessary separation between different visits to the same foreign domain.

If K_u is a weak key, e.g., a password or a PIN, the authentication token (regardless of its construction details) is subject to verifiable-plaintext attack [7, 6]. Little can be done to avoid this unless the access equipment maintains a secure channel to the local AS.[7] Ideally, however, K_u is a strong key which, instead of being memorized by the user, is kept in secure storage on the user's smartcard or some other type of personal device. For the purpose of the ensuing discussion, we assume that, the user and his/her device constitute a single *entity*.

2. Upon receipt of the initial message, AS_r recognizes that the user is a foreign one. Not having any means of authenticating/identifying this user, AS_r needs to request a proof of the user's identity from the claimed home location. The request must also authenticate AS_r to AS_h and U to AS_h. The latter serves as evidence of U being in the remote domain.

   One obvious solution is to send two separate authentication tokens. However, the protocol must accommodate the environment where the user's secret is weak and, as mentioned above, susceptible to guessing attacks. If a token computed with such a weak secret traverses a wide-area backbone comprised of potentially untrusted areas, the possibility of hostile attacks increases dramatically.

   For this reason, the user's token is not actually sent in flow 2. Instead, one of the two nonces used in the computation of the AS_r token is the user token itself. This *token chaining* technique prevents guessing attacks on the user authentication token. This also has an advantage of reducing the size of the message in flow 2.

3. When AS_h receives the message in flow 2, it proceeds as follows:

   (a) Looks up the record for the user U and obtains K_u.

   (b) Validates T_u by comparing it to the current clock reading. Obviously, some skew is to be expected and a maximum clock difference must be defined accordingly. T_u is also compared to the last

[7] One could envision, for example, a physically secure, public access workstation that maintains a strong key in secure storage and uses that key to secure all communication with the nearest AS.

timestamp recorded in the user's record. If the new timestamp is not greater than the recorded one, the request is rejected.

(c) Given U, K_u and AS_r recomputes K_{ur}.

(d) Using K_{ur}, recomputes $N_r^2 = AUTH_{K_{ur}}(N_u, T_u, U)$.

(e) Using N_r^2, N_r^1 and AS_r, recomputes $AUTH_{K_{rh}}(N_r^1, N_r^2, AS_r)$ and compares it to the corresponding token that arrived in flow 2.

A match in the last step successfully authenticates both U and AS_r to AS_h. At this point, AS_h issues a ticket (in flow 3) that confirms U's identity and allows U to operate in the *realm* of AS_r. A KryptoKnight-style ticket contains the following logical information:

- ticket issuer - AS_h
- issuer nonce - N_h
- ticket reader - AS_r
- reader nonce - N_r
- ticket third party - U
- ticket key - K_{ur}
- expiration time[8] and/or other metering information, e.g., a spending limit.

Not all of the above needs to be sent explicitly. In particular: AS_h, AS_r, U and N_r can be inferred from the context of the message, i.e., the recipient (AS_r) knows them.

All ticket fields appear in cleartext with the exception of the key, which is secured within a key token. A KryptoKnight-style key token[4], in turn, is computed as: $AUTH_{K_{rh}}(N_r, N_h, U) \bigoplus K_{ur}$. This construction insures that the key is protected from unauthorized disclosure. Note that the integrity of the key is not guaranteed within the ticket, i.e., the key token can be modified in transit. However, this does not present a problem since AS_r confirms the *goodness* of K_{ur} in the next protocol step (see below).

In total, all that is actually sent in flow 3 is N_h and the key token, i.e.:

$$TICK_{K_{rh}}(AS_h, \boxed{K_{ur}}, AS_r, U) \;=\; \left[N_h,\ AUTH_{K_{rh}}(N_r, N_h, U) \bigoplus K_{ur}\right]$$

4. Upon receiving the message in flow 3, AS_r:

   (a) Recomputes $AUTH_{K_{rh}}(N_r, N_h, U)$

   (b) Extracts the key K_{ur}

   (c) Recomputes $AUTH_{K_{ur}}(N_u, T_u, U)$ and, finally, compares it to the corresponding token received in flow 1 from U. This comparison is needed to check the integrity of K_{ur} extracted from the ticket.

   A match in the last step is crucial. It completes the protocol cycle by authenticating to AS_r both AS_h and U simultaneously. Next, AS_r proceeds to install the user's temporary credentials in the subscriber/user database. We assume that a separate database of visiting users will be maintained (a VLR, in cellular parlance). Of course, before installing the credentials, AS_r has to be satisfied with the conditions accompanying the credentials, i.e., validity interval, payment limit, etc.

5. The last flow (4) is strictly optional. It can be used to perform single sign-on for the user based on the information received from AS_h, i.e., to establich a working session key between AS_r and U. The ticket is computed under the newly-acquired K_{ur} and contains the strong session key (K_s) that the user can utilize immediately.

[8] The ticket lifetime should be chosen to be sufficiently long to cover the duration of a typical visit in a remote domain.

4.5 Subsequent Network Access

For subsequent network access in the same domain, the foreign user can be authenticated via ordinary single sign-on protocol (e.g., borrowed from KryptoKnight or Kerberos) using the same K_{ur}.[9] Of course, this can only happen within the limits of K_{ur} lifetime. The only difference with respect to normal network access at home is that, while the user will continue utilizing his permanent secret K_u (as if he is in the home domain), the actual key used in the single sign-on protocol will be K_{ur}. This means that the user's personal device or workstation must be programmed to calculate K_{ur} accordingly.

4.6 Protocol Evaluation

In this section we evaluate the proposed protocol along the lines of the previously stated design goals.

4.6.1 Domain Separation

The protocol maintains domain separation by allowing a foreign domain to obtain a temporary key for a visiting user. The key is valid only within the domain it was issued for. Under no circumstances is the actual user's key disclosed to the foreign domain. Of course, this argument becomes rather thin in the case of a user armed with only a weak password. (A malicious AS_r can easily break K_u with a known-plaintext attack on K_{ur}.)

4.6.2 User Transparency

Provided that the end-user equipment is programmed accordingly, the fact that the authentication protocol is executed in a foreign domain makes very little difference from the user's perspective. In the case of the *unequipped* user accessing the network through, say, a public workstation, the only difference for the user is having to enter:

UserName@HomeDomain instead of the usual UserName

The difference is even smaller for users equipped with personal devices. The device (e.g., a cellular phone) can sense whenever it is no longer in a home domain and obtain the name of the local domain in the *background*, before the authentication protocol is initiated. It can similarly supply the name of its home domain; thus, the user does not even have to enter it.

4.6.3 User Identity Confidentiality

Until now we have sidestepped the issue of protecting the user's identity. This isuue is very important but can not be discussed thoroughly in this paper given that our main concern is with the authentication of mobile users. However, it is worth noting that many variables influence the way of protecting the user's identity.

In particular, a user armed with a personal device can benefit from the device's ability to store arbitrary information. This can include, for example, a list of one-time pre-computed one-time userids that a user can supply in place of his real identification.

On the other hand, a user armed with only a password or a PIN has little recourse in safeguarding his identity. One possible solution is for each user to select a unique travelling alias, U_t. (U_t can be assigned to a user by the home AS as well.) An alias can be similar in structure to a userid, e.g., a user known as "john.smith" at home can select a U_t "julius.caesar" for use while travelling. The mapping between aliases and real user idenitites ($U \leftrightarrow T_u$) will be kept secret by the home AS. However, aliases will not be secret. A user can be free to change his U_t but only while he is *at home*.[10]

[9] Essentially, this means that only protocol flows 1 and 4 will be executed.

[10] In fact, a user should be encouraged to change his U_t reasonably often; perhaps as often as his password.

Our discussion so far applies only to the first time a user pops up in a given foreign domain. Moreover, it applies only to the flow 1 of the authentication protocol where $SUid$ field denotes the user's identity. In the subsequent protocol flows, the obvious solution is to use the same $SUid$ construction as in flow 1. However, since AS_r and AS_h are assumed to share a strong secret key, K_{rh}, the protection of the $SUid$ field can be strengthened further. (This is completely independent of the way $SUid$ is computed in Flow 1.) One possibility is to set:

$$\overline{SUid} = E_{K_{rh}}(SUid)$$

where $\overline{SUid}$ denotes the user identification field in flows 2 and 3. However, this implies that AS_h must decrypt $\overline{SUid}$ on receipt of message 2. Since we would like to avoid explicit decryption operations, an alternative is:

$$\overline{SUid} = E_{K_{rh}}(AUTH_{K_{rh}}(N_r^1, N_r^2, AS_r)) \bigoplus SUid$$

Once AS_r authenticates the user (and AS_h) in flow 3 and establishes a temporary record in its VLR, it can also assign a temporary identity to the user. This is different from the travelling alias referred to before. A temporary identity is issues by the local AS based on its own policy. It serves to idenitify a particular visit of a given user in a foreign domain and remains valid as long as the user's entry in the VLR does not expire.

4.6.4 Minimal Overhead

Given the initial requirement that the home domain be contacted to affirm the mobile user's provenance (and solvency), it is impossible to design a protocol with less than two flows traversing inter-domain boundaries, i.e., $AS_r \rightarrow AS_h$ and $AS_r \leftarrow AS_h$. In addition, at least one flow is necessary for the user to make initial contact with the local AS. This makes a minimum of three flows which is the number of flows in our protocol (the fourth being optional and local.)

A closely-related issue is the size of each protocol flow. Of particular interest are the sizes of flows 2 and 3 since both cross domain boundaries and potentially traverse significant distances.

The second flow ($AS_r \rightarrow AS_h$) is composed of six fields. If a personal smartcard-like device is involved, it is possible to maintain (perhaps, loosely) synchronized clocks between the device and AS_h. This would allow AS_r to omit the timestamp in flow 2. (However, no clock synchronization between the device and AS_r should be assumed.)

The third flow ($AS_r \leftarrow AS_h$) contains a ticket which is logically comprised of six fields: AS_h, N_h, AS_r, N_r, $\overline{SUid}$ and K_{ur}. (This is excludes optional metering information, e.g., ticket lifetime.) Of the six, AS_r, N_r and AS_h can be recovered from the context of the protocol run. In other words, if AS_r retained state from flows 1 and 2, it can recover these values. Hence, as few as three fields are actually necessary in the third flow.

5 Variations on the Theme

5.1 Doing Away with State

One of the drawbacks of the basic protocol is the requirement that AS_r retain state during the time between sending flow 2 and receiving flow 3. There are some well-known benefits of stateless design; code simplicity, storage conservation and increased fault-tolerance are chief among them.

On the other hand, statefulness can result in shorter protocol messages, since when state is not retained it must be "exported" thus making messages longer. Moreover, it can be beneficial for AS_r to retain state and actually create a database entry in a VLR before receiving confirmation from AS_h in flow 3. This could save time when flow 3 is finally received by AS_r.

In spite of all the pros and cons of state retention, we consider a stateless version of the protocol if only for the reason of completeness.

As shown in figure 4 the user begins by sending a message to AS_r. As before, AS_r constructs the message in flow 2, forwards it to AS_h but **does not** keep any record of request in flow 1.

The exact contents of the message in flow 2 depend on the level of trust between the two domains. It is possible to leave the contents of flow 2 unchanged. In this case, when AS_h replies in flow 3, AS_r is not able to authenticate the user directly (since all knowledge of flow 1 is lost.) Instead, AS_r can go on with the protocol flow 4, distributing a strong session key K_s and subsequently authenticate the user on the basis of K_s. Alternatively, it can ask the user to re-generate flow 1.

With a lower level of inter-domain trust AS_r can "piggyback" the necessary state on top of flow 2. This would entail including the entire contents of flow 1. However, flow 2 already includes most of the components of flow 1. The only information to be piggybacked is the user's authentication token from flow 1 super-encrypted under a key K_r which is known **only** to AS_r. The same token (as well as T_u and N_u) must also be included in flow 3 as shown in figure 5.

When AS_r receives flow 3 it proceeds as before with one exception: when reconstructing the user's authentication token with K_{ur}, AS_r must super-encrypt it with K_r and only then compare to $E_{K_r}(N_r^2)$ received from AS_h. This procedure still preserves the *no decryption* property of KryptoKnight protocols.

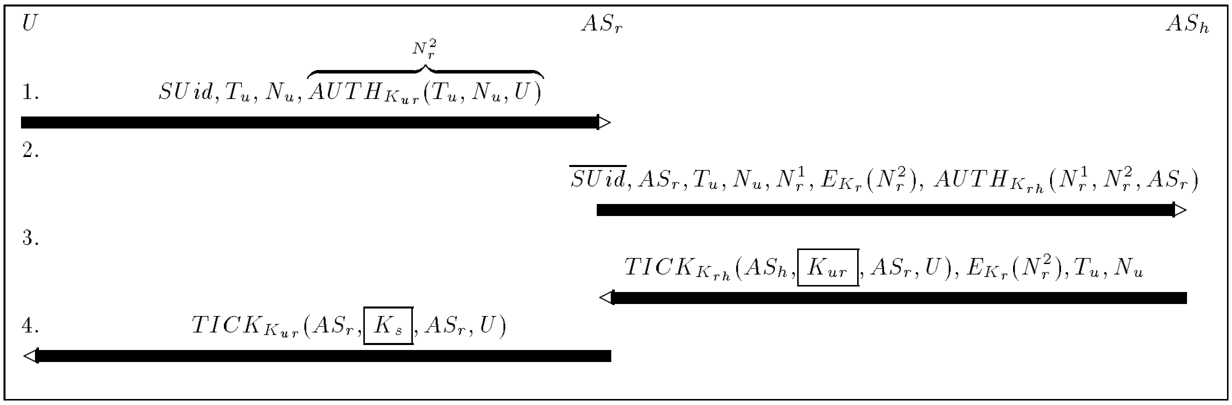

Figure 5: *Stateless* authentication protocol

5.2 Wireless/Cellular Considerations

In a highly-dynamic wireless environment where users frequently cross domain boundaries in the middle of communication, it is crucial to transfer the necessary state between domains in a manner transparent to the user. The same problem also occurs when users migrate among different cells within the same domain. However, in the latter case, authentication is not an issue.

GSM, for example, makes provisions for very fast transfer of user's authentication information between domains. Recall that (as described in section 3) in GSM the home domain supplies the foreign domain with a set of challenge/response pairs.[11] Each pair is good for one-time authentication of the user. Whenever the user moves from foreign domain A to foreign domain B, AS_a is allowed to forward the unused challenge/response pairs to AS_b. This allows AS_b to authenticate the user (or, rather, his SIM) immediately without having to get in touch with the home domain.

The GSM approach is reasonable considering that the authentication information is shared only between adjacent domains. (Non-adjacent domains are less likely to trust one another.)

In general, however, the GSM approach is neither secure nor flexible. Since GSM assumes that the fixed network is secure, communication between MSCs (ASs) is not protected. This certainly does not hold in all mobile-user environments. Also, if the entire original set of challenge/response pairs has been used, there is no way for the new domain (AS_b) to authenticate the user without contacting the home domain.

[11] In message flow 3, figure 1.

Figure 6 depicts a protocol that avoids the pitfalls of GSM. The first two flows represent a normal user sign-on in the foreign domain A. These flows are identical to flows 1 and 4 in the basic protocol. Subsequently, the user is crossing the boundary into the adjacent foreign domain B. Instead of immediately contacting AS_h (which is potentially very far away), AS_a forwards to AS_b a ticket containing the very same key K_s that was distributed to the user in flow 2 of the last network sign-on in A. Knowing K_s allows AS_b to authenticate the user immediately and directly.[12]

However, this protocol is only useful as a temporary measure. It is expected that the next time the user attempts to access the network in domain B, a full-fledged authentication procedure involving the home domain will take place.

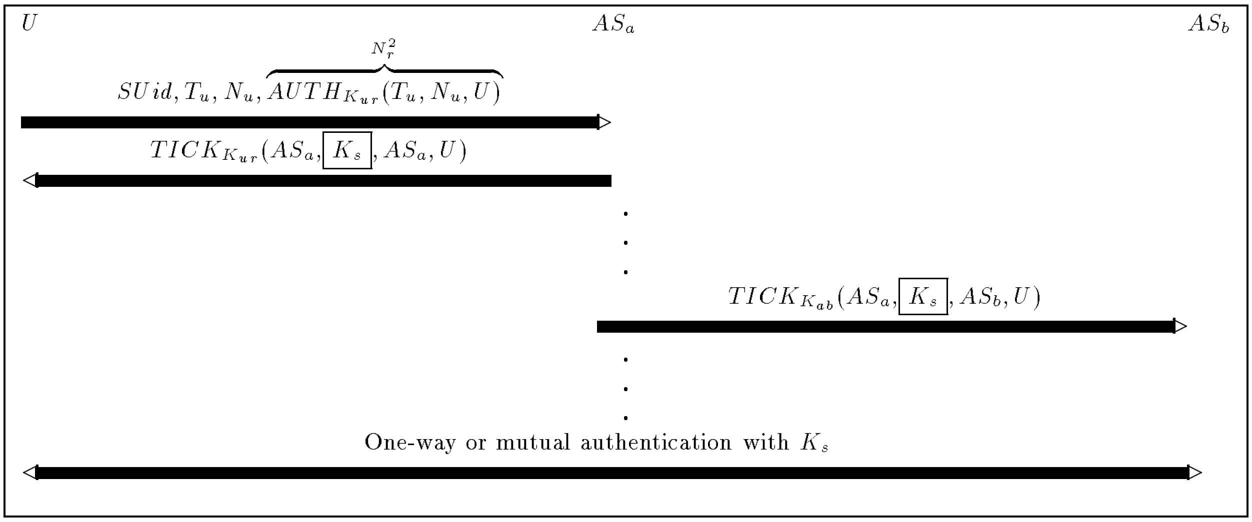

Figure 6: Fast *Hand-over* of Authentication Information

6 Summary

Authentication protocols described in this paper have been implemented as part of KryptoKnight Network Security Server prototype. The current operating environment supports supports user mobility via traditional, wireline access. Future work includes experimentation with wireless, including radio and, perhaps, infrared access.

In conclusion, this paper discussed security implications of user mobility in both wireline and wireless network environments. Existing approaches to mobile user authentication were found to lack in both flexibility and generality. After formulating a number of important design goals, we presented a new protocol for cross-domain authentication of mobile users. The flexibility of the proposed solution allows its implementation (irrespective of the network design particulars) in any environment that supports user mobility.

References

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

[2] B. Mallinder *An Overview of the GSM System*, Proceedings of Digital Cellular Radio Conference, October 1988.

[12] The ticket forwarded to AS_b must have a very short lifetime.

[3] J. Steiner, C. Neuman, J. Schiller, *Kerberos: An Authentication Service for Open Network Systems*, Proceedings of USENIX Winter Conference, February 1988.

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

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

[6] T. Lomas, L. Gong, J. Saltzer, R. Needham, *Reducing Risks from Poorly Chosen Keys*, Proceedings of ACM Symposium on Operating System Principles, 1989.

[7] L. Gong, T. Lomas, R. Needham, J. Saltzer, *Protecting Poorly-Chosen Secrets from Guessing Attacks*, IEEE Journal on Selected Areas in Communications, to appear in Spring 1993.

[8] R. Needham and M. Schroeder, *Using Encryption for Authentication in Large Networks of Computers*, Communications of the ACM, December 1978.

[9] R. Rivest, A. Shamir and L. Adleman, *A Method for Obtaining Digital Signatures and Public Key Cryptosystems*, Communications of the ACM, February 1978.

[10] J. Linn, *Privacy Enhancement for Internet Electronic Mail. Parts I, II and III*, **RFC 1113-1115, SRI Network Information Center**, January 1989.

[11] *Cellular Digital Packet Data System Specification*, **Preliminary Release V. 0.8**, March 1993.

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

[13] European Telecommunications Standards Institute, *Universal Personal Telecommunications*, **ETSI NA7 TS 02.03**, January 1992.

[14] International Standards Organization, *Information Processing Systems – Open Systems Interconnection – Basic Reference Model*, **ISO 7498**, 1977

[15] J. Postel, *Internet Protocol*, **RFC 791, SRI Network Information Center**, September 1981.

[16] R. Bird, I. Gopal, A. Herzberg, P. Janson, S. Kutten, R. Molva, M. Yung, *Systematic Design of Two-Party Authentication Protocols*, Proceedings of Crypto'91, August 1991.

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

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

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

[20] CCITT, *The Directory Authentication Framework*, **CCITT Recommendation X.509**, 1988.