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 Research Laboratory
Säumerstrasse 4
CH-8803 Rüschlikon - Switzerland
gts@zurich.ibm.com

IEEE Network, *Special Issue on Mobile Communications Technologies,* Vol. 8, No. 2, March/April 1994.

Abstract

Internetworks of the future will allow and promote universal access. Users will be able to access the network at a multitude of access points separated by significant geographic distance and many administrative boundaries. Without a single central authority, a new set of inter-domain security mechanisms is needed to allow users to venture 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 user-related security information. In this paper, we propose a generic approach for authenticating mobile users in remote domains that satisfies the domain separation constraint. The protocols described herein can be applied in different mobile-user environments including wireless networks and mobile user services on traditional wireline networks.

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

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. User mobility and universal network access certainly exasperate certain security threats such as illegal access (fraud) and eavesdropping. In addition, one new factor brought about by 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 general approach 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 introduced in Section 5 and Section 6 concludes with the summary of the paper.

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

*Names appear in alphabetical order.

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-added service such as Universal Personal Telecommunication (UPT)[14]. 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[1] periodically *pops up* in a new, foreign domain. A user may be simply transiting a foreign domain or planning to linger about for some time. Regardless of his intentions or 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. In order to do that, he must first establish *temporary* residence in the foreign domain.

User mobility in the network environment is not unlike mobility in the real world, where a person travelling from one country to another must often engage in some 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 varies one country to another. The same can be expected of network domains.

At the first glance, it seems that the problem can be easily solved by requiring each mobile user to carry a universally-recognized credential, i.e., a passport. An electronic equivalent of a passport is called an *electronic certificate*. There are electronic certification schemes based on public key as well as conventional cryptography. One notable example is *Privacy-Enhanced Mail* (PEM)[10, 11, 12].

Given a certificate of a foreign user, any domain can verify the credential and confirm the identity of that user. However, there are several issues that can not be addressed with electronic certification.

- While a certificate's authenticity is readily verifiable, its current status is not. In other words, since a mobile user demands service – not a *free* commodity – the foreign domain may need to establish that a newly-arrived, visiting user is currently in good standing. This can not be accomplished without some interaction with the user's home domain, since only the home domain is able to comment on the user's current standing. For example, if a user obtains service while *abroad* and is later billed for it, the home domain may refuse to pay since the expenditures were not authorized.

- Some domains may desire to restrict the mobility of their constituent users; such constraints may be very difficult to encode in a universal certificate.

- Another problem with electronic certificates is the underlying assumption that the user has some means of carrying his certificate, e.g., within a cellular phone, a personal communicator or a smartcard. However, in the simplest mobile-user environment (e.g., UPT[14]), a user is "armed" with only a password or a PIN – something that a human brain can easily retain.

In summary, universal user certification is not sufficient or well-suited for establishing temporary residence in a foreign domain. In addition, the preceding discussion points to the need to contact the user's home domain. Even if access control is not an issue and if the user's ability to pay for services can be confirmed on-the-spot (e.g., with electronic cash[20, 21]), the user's home domain must be contacted if only for the reason of *tracking*. This is because a user is normally registered with his home domain on a long-term basis; anyone wanting to contact a user has to consult his home domain first. Therefore, it is natural for a domain to *track* the whereabouts of its constituent users.

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

[1] No gender implication.

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 in 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 typical 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 the home domain.

3 Existing Approaches

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

3.1 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 Visiting 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 called 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 the MS notifies the local MSC of its presence, the local VLR contacts the mobile unit's HLR and transmits his own identity, the mobile station's International MS Identity (IMSI) and position to the HLR. The HLR asks its local 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 forwarded back to the VLR and each triplet is used only once for the authentication of the MS.

Parameters $SRES$ and K_c are computed with proprietary algorithms A_3 and A_8 that implement one-way 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 proprietary 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.

The main concern with the GSM authentication approach is its 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

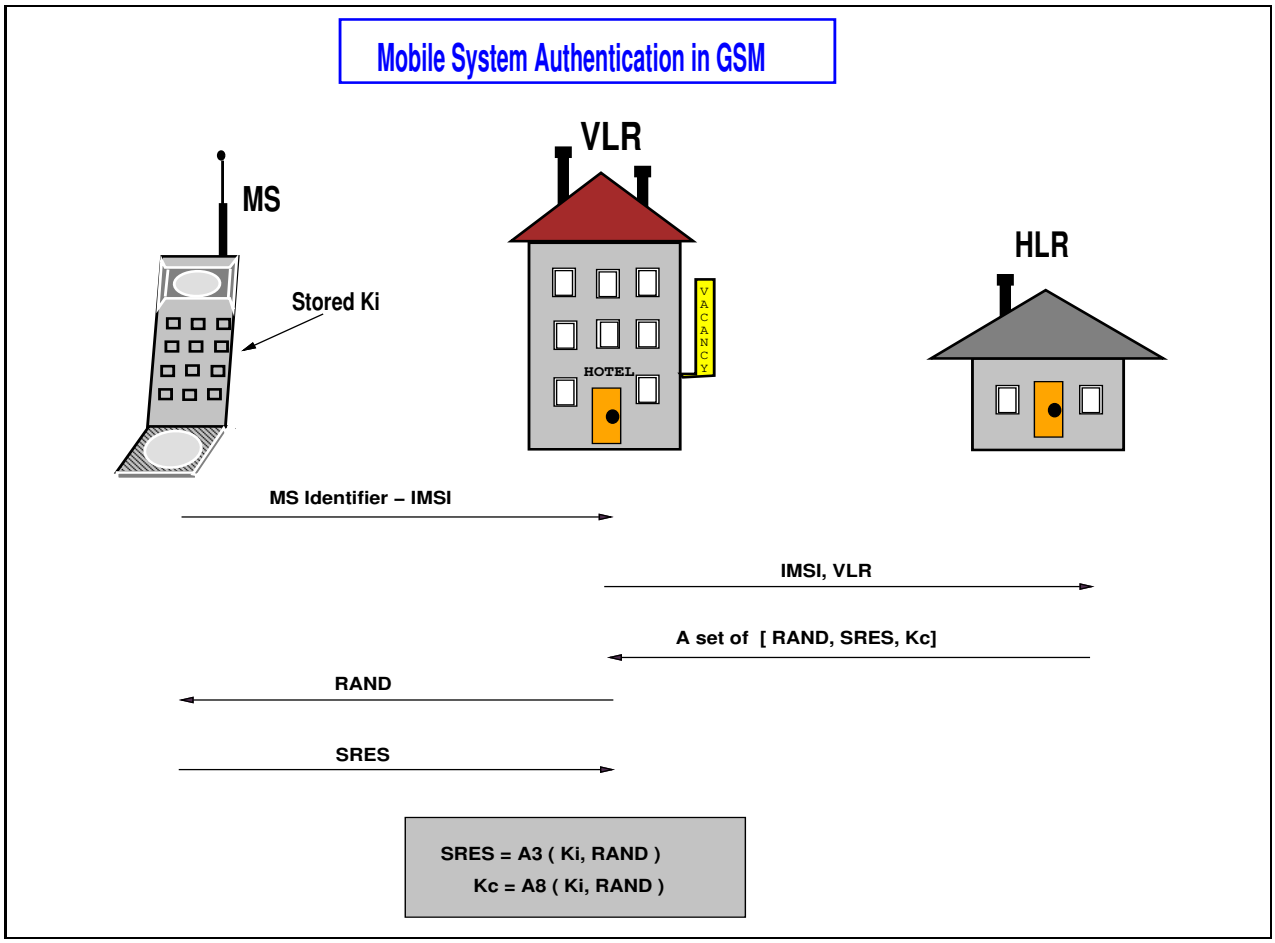

Figure 1: Mobile Station authentication in GSM

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 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 "home-grown" proprietary 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.[2]

3.2 CDPD

Cellular Digital Packet Data (CDPD) architecture[13] has been recently developed by a consortium of several US-based companies. As the name suggests, it is oriented towards *data*, and not voice, traffic. CDPD takes advantage of free slots in cellular voice communication and uses them to transport data. Like UPT, it provides for network access through either mobile or fixed end-systems. However, it is not simply a value-

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

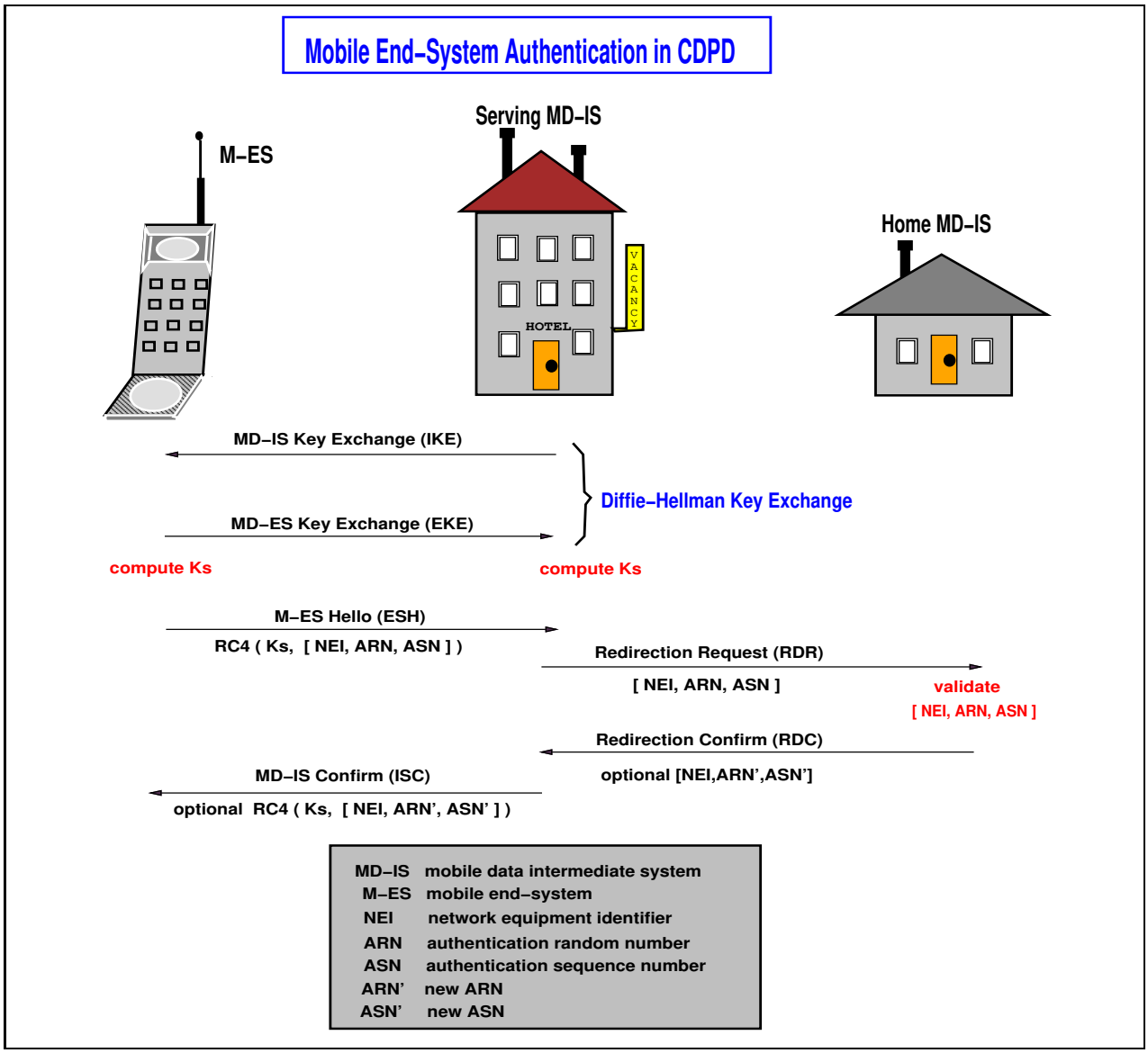

Figure 2: Mobile End-System Authentication in CDPD

added service but a complete architecture including, among other things, a MAC layer. The architecture supports several network layer protocols including IP[17] and CLNP[16].

Security services in CDPD are composed of data confidentiality, key distribution and mobile unit authentication. CDPD requires a logically-distinct entity – Authentication Server (AS) – to be present in every CDPD domain (area). An AS is typically co-located with the Mobile Data Intermediate System (MD-IS) in a service provider's domain. Mobile unit (M-ES, in CDPD parlance) authentication always involves contacting the AS in the unit's home domain.

As shown in Figure 2, the authentication process begins with the Diffie-Hellman key exchange protocol [5]. As a result, the M-ES and the serving MD-IS come to share a secret key, K_s.[3] Armed with K_s, M-ES submits its credentials (encrypted with K_s) for authentication. The M-ES credentials consist of a triple: $[NEI, ARN, ASN]$. Exactly how ARN is generated is not specified. Presumably, it is an unpredictable random number produced by the home AS. However, ARN is not a true *nonce*[4] since the same value of

[3] There are actually two keys: K_0 and K_1, both derived from K_s. They are used for MD-ES←MD-IS and MD-ES→MD-IS communication, respectively.

[4] An unpredictable, **used-only-once** random number [9].

ARN may be used multiple times.

The serving MD-IS decrypts the credentials and forwards them to the home MD-IS *in the clear*. The home MD-IS then validates the credentials and optionally issues a new ARN. M-ES authentication completes when the serving MD-IS receives a positive confirmation from the home MD-IS and signals to M-ES (enclosing the new ARN is applicable.) The M-ES authentication is unidirectional, i.e., the serving MD-IS is not authenticated to the M-ES. This means that an intruder can masquerade as the serving MD-IS and discover the M-ES credentials.[5]

Like GSM, CDPD makes an assumption that the "fixed" network is secure. Therefore, communication between the serving MD-IS and the home MD-IS is conducted in the clear. Since this includes M-ES credentials $[NEI, ARN, ASN]$, CDPD is susceptible to the same attacks as GSM.

In fact, GSM has a slight advantage over CDPD since, in GSM, the mobile station's long-term key (K_i) is never revealed outside the station. Thus, anyone intercepting $VLR \leftrightarrow HLR$ traffic can gain at most several authentication triplets: $[RAND, SRES, K_c]$ and impersonate the mobile station as many times as there are triplets in the intercepted message.

In contrast, CDPD does not require a long-term M-ES key. If an intruder intercepts M-ES credentials only once, he can impersonate the victim M-ES *ad infinitum*. This is because, the authenticators (ARN values) in CDPD are computed by the home AS (MD-IS) and not by the M-ES; thus, mere possession of just one currently valid triplet $[NEI, ARN, ASN]$ is enough to obtain subsequent triplets and continue impersonating the M-ES.

CDPD, like GSM, uses an unpublished encryption function. As mentioned above, the initial key exchange is performed via Diffie-Hellman protocol. Subsequent traffic is encrypted using the (proprietary) RC4 encryption function [22].

3.3 UPT

The upcoming European value-added service, Universal Personal Telecommunications (UPT), is aimed primarily at the European market. It is designed to provide universal user access and to support both fixed and mobile 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$\leftrightarrow$UPT-equipment) using smartcards[14, 15]. Unlike GSM, UPT has not matured to a state where specific security solutions have been proposed.

4 Authentication of Mobile Users

In this section we develop a generic solution for the authentication of mobile users. In doing so we try to avoid the drawbacks of GSM and CDPD. In particular, we make no assumptions about the security of the intermediate, fixed networks. In addition, we take into account some usability concerns in order to minimize the burden on the user and provide a transparent user interface.

4.1 Initial Assumptions

We assume that, when accessing the network in the *home* domain, the mobile 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.

[5] GSM is slightly better off since after the initial MS authentication, the VLR must be able to decrypt incoming traffic with K_c; which can only be obtained from the real HLR of the mobile unit.

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 service developed at IBM Research[4]. The reason for choosing KryptoKnight is twofold:

- 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.[6]
- The resultant protocol(s) will be incorporated into the existing KryptoKnight protocol family (which currently lacks any support for user mobility.)

In order to illustrate some KryptoKnight concepts, figure 3a 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 by the responding party (B in our case). This expression is computed as a result of applying a strong encryption function E, e.g., DES[6], with K_{ab} as the encryption key, over three inputs: two nonces (N_{ab}, N_{ba}) and the name of the message originator. One example of $AUTH$ is:[7]

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

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 3b.) 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.

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

[6] Refer to [4] and [18] for details.

[7] All encryption is performed with K_{ab}.

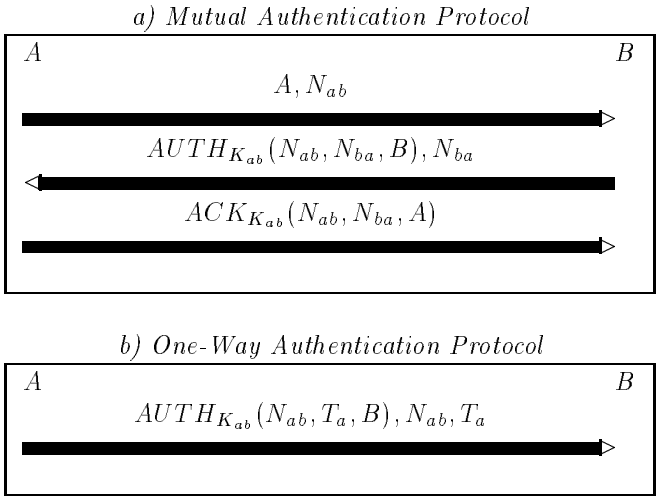

Figure 3: KryptoKnight Two-Party Authentication Protocols

- $SUid$, $\overline{SUid}$ – 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} – Long-term key shared by AS_r and AS_h. We assume that K_{rh} is installed either out-of-band or using a secure key distribution procedure involving a mutually-trusted third party.[8]
- 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 over a triple (X, Y, Z), under key K (as described in Section 4.3.)
- $TICK_{\hat{K}}(A, \boxed{\text{K}}, B, C)$ – 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 is 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

[8] This can be done with a hierarchy of ASs; see, for example, [23].

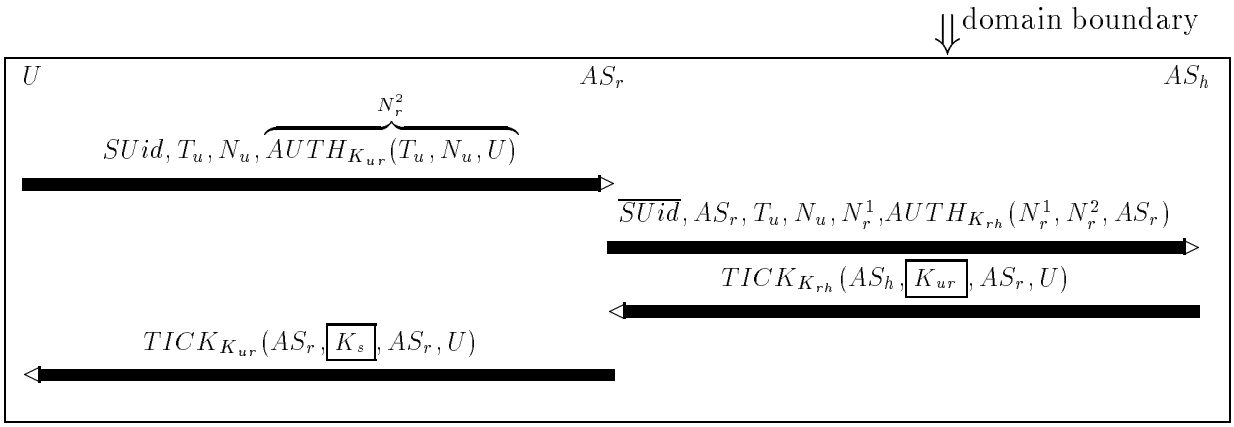

Figure 4: Basic Mobile User Authentication Protocol

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} in 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 function such as MD5[19]. We can even avoid using encryption by computing K_{ur} as $F(AS_r, U, K_u)$. The use of encryption or a secret one-way function in the formation of K_{ur} is essential in order to satisfy the domain separation constraint by preventing the derivation of K_u from K_{ur}.

   Constructing K_{ur} as described above has an important consequence that the user visiting the same domain on two different occasions will continue using the same K_{ur}. At the first glance, this seems like a serious weakness. However, most users regularly change (or, at least, are expected to change) their passwords and PINs. A change of K_u in the home domain will invalidate any previously used K_{ur}.

   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 [8, 7]. Little can be done to avoid this unless the access equipment maintains a secure channel to the local AS.[9] 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.

[9] 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.

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 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, AS_h 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[10] 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.

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

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 establish 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.

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}.[11] 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 issue 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 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 identities ($U \leftrightarrow T_u$) will be kept secret by the home AS. Users can be allowed and encouraged to change their aliases at regular intervals; perhaps as often as passwords or PINs.

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

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 issued by the local AS based on its own policy. It serves to identify 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 Low 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 flow is 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 the user has a personal smartcard-like device, it is possible to maintain 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 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.

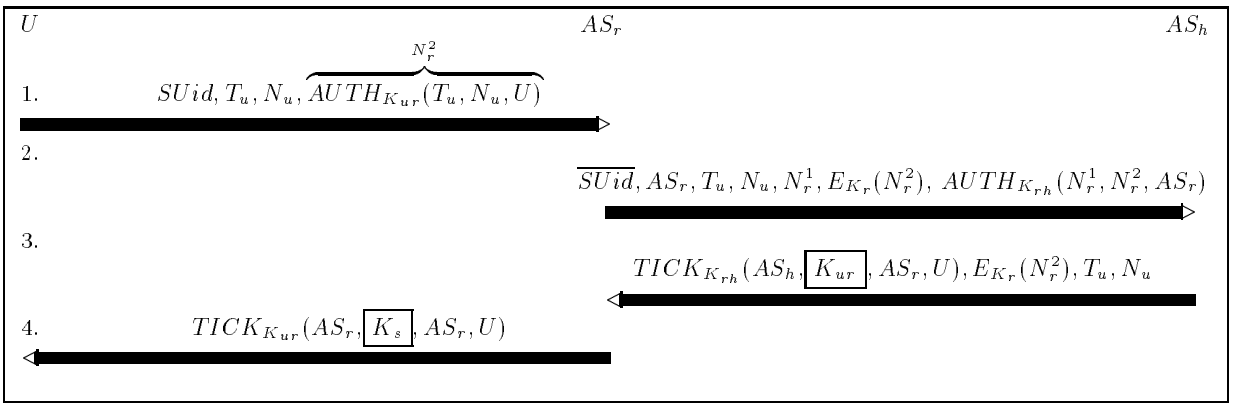

Figure 5: *Stateless* authentication protocol

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 users' 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.[12] 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, the user's SIM) immediately without having to contact the home domain.

On the other hand, as discussed in Section 3.1, GSM assumes that the fixed network is secure and communication between the MSCs (ASs) is conducted in the clear. This assumption is valid only in homogenous mobile-user environments such as GSM or CDPD. Also, when the entire set of authentication triplets has been depleted, there is no way for the visited domain (AS_b) to authenticate the visiting user without contacting the home domain again.

Figure 6 depicts a fast hand-over protocol. 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.[13]

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.

[12] In message flow 3, figure 1.

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

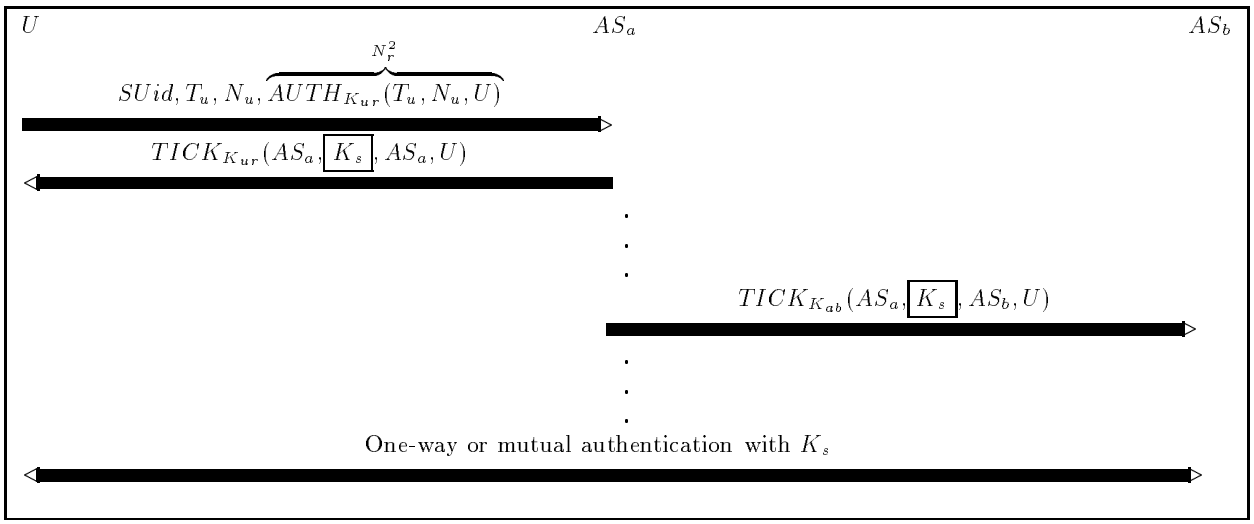

Figure 6: Fast *Hand-over* of Authentication Information

6 Summary

In conclusion, this paper discussed security implications of user mobility in both wireline and wireless network environments. Existing approaches such as GSM and CDPD were found to be ill-suited for general mobile user authentication. 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 underlying network design specifics) in any environment that supports user mobility.

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

7 Acknowledgements

The authors are grateful to Els Van Herreweghen, Ralf Hauser, Phil Janson and Liba Svobodova for many useful comments and suggestions. We are also indebted to the referees (who shall remain nameless) for their valuable input.

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.

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

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

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

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

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

[10] J. Linn *Privacy Enhancement for Internet Electronic Mail. Parts I: Message Encryption and Authentication Procedures*, RFC-1421, SRI Network Information Center, February 1993.

[11] S. Kent *Privacy Enhancement for Internet Electronic Mail. Parts II: Certificate-Based Key Management*, RFC-1422, SRI Network Information Center, February 1993.

[12] D. Balenson *Privacy Enhancement for Internet Electronic Mail. Parts III: Algorithms, Modes, and Identifiers*, RFC-1423, SRI Network Information Center, February 1993.

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

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

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

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

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

[18] 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.

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

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

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

[22] RSA Data Security Inc., *The RC4 Encryption Algorithm*, Document No. 003-013005-100-000-000, March 12, 1992.

[23] K. Sollings, *Cascaded Authentication*, Proceedings of 1987 IEEE Symposium on Security and Privacy, April 1988.