AN AUTHENTICATION PROTOCOL FOR MOBILE USERS

Refik Molva, Didier Samfat and Gene Tsudik. †

Abstract. Mobile networks need additionnal security functions in contrast to traditional fixed-topology static-user networks. In fact, a new problem involving mobility is that users are able to access the network at multiple points which can be separated by significant geographic distances and many different administrative boundaries. As these access points are not necessarily under the control of a single administrative authority, a new set of inter-domain mechanisms is needed in order to allow users to perform security operations in visited domains, providing they obtain an agreement from their home domain. Even if this requirement is obvious, the corresponding solutions should however take into account a somewhat contradictory security constraint that calls for strict partitioning of security domains in order to avoid sharing domain-specific security information among several domains. In this paper, we suggest a generic solution for the authentication of users in visited domains that maintains the domain separation property. The advantage of the protocols described herein is they may be adapted to both wireless networks and traditional wireline networks supporting mobility.

Introduction. With the advent of mobile networks, new security requirements arise due to the lack of physical protection mechanisms akin to the traditional fixed-topology, static-user networks. In addition to the need for security functions preventing illegal access (fraud) and eavesdropping in mobile networks, a new problem involving mobility is the ever-increasing distance that can separate network access points. Since all network access points are unlikely to be 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. We begin by presenting the drawbacks of authentication solutions in existing mobile-user environments. Then, we presents our solution and an application to the wireless environment.

Review of Existing Approaches. User mobility is a feature that can be offered in different network environments such as wireline, infrared, radio and cellular. In the case of mobile digital cellular network, the *Global System for Mobile* (GSM) [1, 2] architecture is the first to provide security services such as user authentication, traffic confidentiality and key distribution.The main concern with the GSM authentication approach (and also for *Cellular Digital Data* (CDPD) approach [11]) is it relies on the assumption that the "fixed network" is secured.

†
R. Molva, D. Samfat
EURECOM Institute
2229, Route des Crêtes
BP 193 Sophia Antipolis Cedex - France
{molva,samfat}@eurecom.fr

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

Therefore, messages are transmitted in a clear text form between Mobile Switching Centers (MSC) which trust each other. However, the same assumption cannot be made for large heterogeneous network environment managed by different administrative authorities. Thus, a security architecture with minimal assumptions about the security of intermediate transport networks is needed.

Another point of contention with GSM is the manner of distributing user authentication information. The home domain is expected to generate a set of challenge/response pairs on-the-fly 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 in 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 non-published algorithms A_3, A_5, A_8 that are used for authentication and privacy. The principle of hiding the algorithm has not proven to be effective in preventing hostile attacks. Furthermore, even if the GSM solution is well-designed, "secret" or "unknown" algorithms always fail to give an informed end-user a comfortable feeling of security.

Other environments can be adapted to support user mobility, i.e., a wireline network can be equipped to allow universal access by offering a value-added service such as Universal Personal Telecommunication (UPT) [12]. The UPT design anticipates many types of fraudulent use and suggests some general solutions [12, 13]. Unlike GSM, UPT has not matured to a state where specific security solutions have been proposed.

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.

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.

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]. 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, protocol flexibility with respect to different network configurations, and amenability to small hardware-based implementations.[1] The resultant protocol(s) will be incorporated into the existing KryptoKnight protocol family (which currently lacks any support for user mobility).

Basic Protocol. The basic protocol is depicted in figure 1. It enables a travelling user to establish a temporary residence in a visiting dommain by requesting the transfer of location-dependent authentication information from the authentication server of his home domain to its peer in the remote domain. The following notation is used in the protocol and throughout the rest of this paper:

- *SUid*– Identification of the end-user U in different protocol flows.
- 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.[2]
- 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
- $AUTH_K(X,Y,Z)$ – Equivalent to $E_K(Z \oplus E_K(X \oplus E(Y)))$. Authentication token computed over a triple (X,Y,Z), under key K using E as a strong cryptographic algorithm like DES [6]
- $TICK_{\hat{K}}(A,\boxed{K},B,C)$ – 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 is discussed below.

The basic protocol can be adapted in the least "sophisticated" case where the user relies only on his password or PIN and has no special hardware, or in a cellular environment where personal device (portable phone, smartcard or token card) is usually used. We now turn on the details of the protocol:

[1] Refer to [4] and [16] for details.

[2] This can be done with a hierarchy of ASs; see, for example, [20].

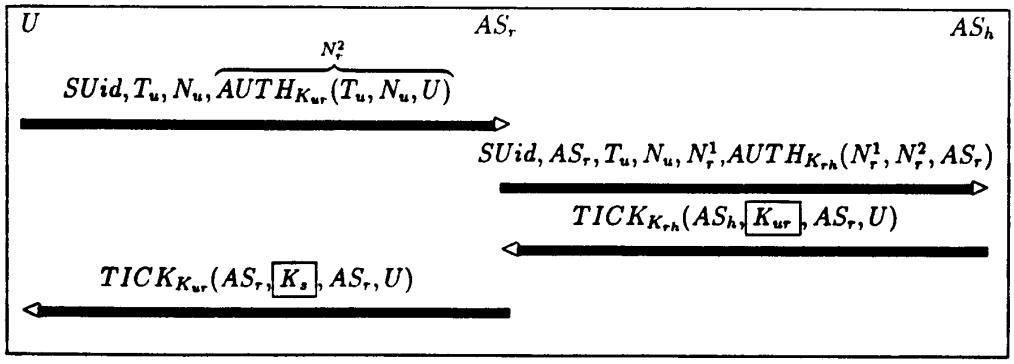

Figure 1: Basic Mobile User Authentication Protocol

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 [17]. 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}.

   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]. Little can be done to avoid this unless the access equipment maintains a secure channel to the local AS.[3] 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. The user's token is not sent in flow 2; as it may be computed with a weak key, the user's secret is susceptible to guessing attack when traversing networks composed of untrusted areas. In order to avoid this weakness, we uses a ***token chaining*** technique: one of the two nonces used in the computation of the AS_r token is the user token itself. 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.

[3] 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.

(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. 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: $AUTH_{K_{rh}}(N_r, N_h, U) \oplus K_{ur}$. This construction insures that the key is protected from unauthorized disclosure:

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

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.

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.

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}.[4]

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. Therefore, in GSM the home domain supplies the foreign domain with a set of challenge/response

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

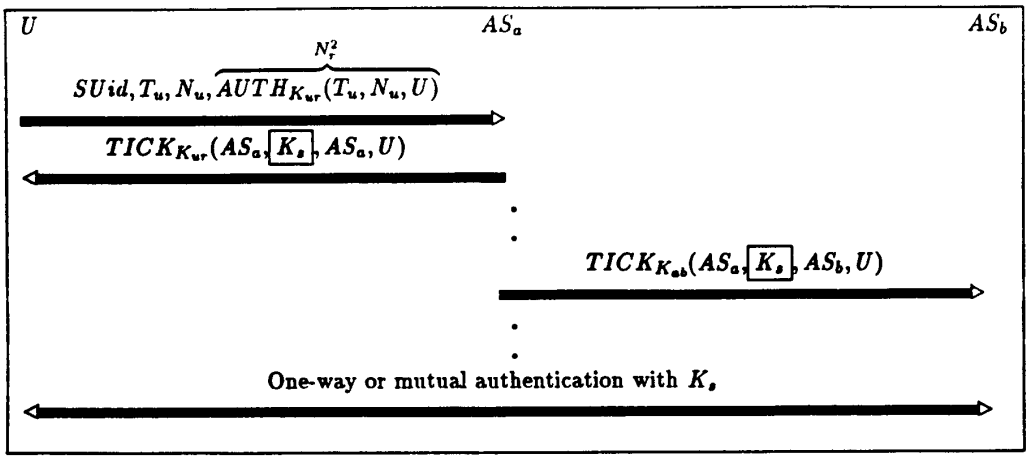

Figure 2: Fast *Hand-over* of Authentication Information

pairs. 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. So, 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 2 depicts a fast hand-over protocol avoiding a new visited domain (AS_b) to contact the home domain (AS_h) during a communication. 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.[5]

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.

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.

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

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, to appear in Spring 1993.

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

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

[11] *Cellular Digital Packet Data (CDPD) System Specification*, Release 1.0, July 19, 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] CCITT, *The Directory Authentication Framework*, CCITT Recommendation X.509, 1988.

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

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