Key Management for Decentralized Computer Network Services

Lein Harn and Hung-Yu Lin

Abstract— **When a computer network grows larger and adds more services offered by different providers, each user may have to keep many keys for different services. Incorporating smart card technologies and master key concept, this letter proposes an efficient scheme to solve the key management problem in such multiservice environments.**

I. Introduction

COMPUTER and Communication (C&C) technologies have been developed together to encourage a dramatic increase in the volume and speed of information processing and distribution. As a result, different services—e.g., news or stock information services, electronic secretary services, electronic banking and shopping services, etc.—can be provided through computer networks. Organizations in both the public and private sectors are becoming increasingly interested in implementing more services on these networks. Without appropriate protection, these services are susceptible to unauthorized access.

User authentication is probably the most commonly used protection mechanism for this purpose. In this approach, every service center keeps a list of registered user identifications with their secret passwords. By displaying one's ID and demonstrating the possession of its corresponding secret (i.e., password), each registered user can prove his/her validity to the service center in the login process. All of the operations requested later are associated with this unique ID. So this approach is particularly appropriate for services which require access control, audit, or accounting for each service requested.

However, for some other services, the above requirements may not be necessary. Instead, every user has the same privilege to access the service. The service charge is fixed and is based on whether the user subscribes to the service or not, with no consideration of how he/she consumes the resources of the service. More importantly, users do not want to reveal their identities either when they subscribe to the service or when they request services from service centers. For any of these fixed-rate services, every user pays the same membership fee and gets a token when he/she subscribes to the service. Later, the user can prove his/her validity to the service center by demonstrating the possession of the token. This kind of protection mechanism is called membership authentication. Obviously, in order not to reveal each individual's identity, the token given to each member must be the same so that the service center cannot figure out who is requesting the service. Here comes the problem: how to prevent the members from revealing the token to other users who do not pay the membership fees. Even when user authentication is required but the service charge is membership-based, how do we prevent members from requesting services for nonmembers? This problem has moral implications and seems to be unsolvable by current techniques. Even so, there are still many membership-based services in the real world. In this letter, we will refer to services in which the above problems can be tolerated.

When the computer network is growing larger and more services provided by different providers are added into the network, each user may have to keep many secrets (i.e., keys, passwords, tokens, etc.) for different services. The storage requirement and the possible exposure of these secrets become important issues for the users. Administration and accounting for these services will also become very complicated for centralized service control. Many schemes have been proposed to solve either user authentication [1], [2] or membership authentication [3], [4] for a single service or for multiple services managed by a centralized center. None of them is designed for multiservice environments, where one user may access services provided by different service providers. Here, an integrated scheme is proposed to solve the problem of managing keys for services which require either user authentication or membership authentication in multiservice environments with the incorporation of smart card technology.

In this letter, system components for the proposed scheme and detailed operations are given in Sections II and III, respectively. The advantages of this scheme are described in Section IV, and some implementation and security issues are discussed in Section V. Finally, Section VI gives a brief conclusion.

II. System Components

In this proposed scheme, the network has the following components.

1) A Smart Card Producing Center (SCPC): This center is responsible for manufacturing the smart cards and assigning one unique pair consisting of a public prime and a corresponding secret key (password, token, etc.) to each smart card and membership-based service. The SCPC should be trusted by all service providers and users not to reveal these secret keys, but the SCPC never gets involved in any real-time authentication process. Before subscribing to any service in the network, every user has first to purchase a smart card from this center.

2) Users: In a more general scenario, the user can be defined as an individual subscriber with his/her own PC, a system

Paper approved by H. Rubin, the Editor for Network System Service of the IEEE Communications Society. Manuscript revised April 2, 1991 and June 5, 1991. This work was supported by the Center for Advanced Technology in Telecommunication and Computer Networking, University of Missouri, Kansas City, and the Missouri Department of Economic Development.

The authors are with the Computer Science Telecommunications Program, University of Missouri-Kansas City, Kansas City, MO 64110.

IEEE Log Number 9213541.

with several workstations (i.e., a site license holder), or even a cluster of small systems in a network. After a user gets a smart card from the SCPC, he/she can start to subscribe to different services. Note that users do not reveal their secret keys to any service, but they may get secret keys (passwords, tokens) from membership-based services.

3) Service Centers: Each service center manages a specific service. It is responsible for the user registration and user/membership authentication. For any service which requires user authentication, the service center keeps—at the least—each registered user's ID and its associated public prime on the card for user authentication in the login process. So, a new user can be added by storing this new use's ID and public prime number, and an old user can be deleted by removing that user's ID entry from storage.

III. The Proposed Scheme

A. Key Generation

The SCPC selects two large secret primes, p and q, and makes their product n $(= p \times q)$ public. It also selects a generator, $\alpha \in [1, n-1]$. For each card produced, a distinct public prime e_i is assigned by the SCPC. This publicly known prime and its corresponding password w_i, calculated by SCPC as $w_i = \alpha^{d_i} \bmod n$, where $d_i \times e_i \bmod \varnothing(n) = 1$ and $\varnothing(n) = (p-1)(q-1)$ is the Euler's totient function [5], are stored in the smart card. Note that even though e_i and n are publicly known, without knowing the factoring of n, it is computationally infeasible to compute d_i.

In the same manner, each membership-based service will also be assigned a public prime and a secret token, r_j and k_j with $k_j = \alpha^{s_j} \bmod n$ and $s_j \times r_j \bmod \varnothing(n) = 1$.

B. User Registration

For a service which requires user authentication, the user must first register with the corresponding service center, providing to that center a personal identification (ID) and the public prime, e_i, on his/her card. If he/she is qualified to access this service, the service center will record that user's ID and e_i in a verification table maintained by the center.

For a service which requires membership authentication, after paying the membership fee if required, the user will receive a secret token k_j from the corresponding service center.

C. Key Management

When the number of services one has to access increases, key management becomes a serious problem due to the fact that each membership-based service has a different secret token and the memory space on each card is very limited. Suppose that a user with public prime e_i and secret password w_i already has access privileges to m different membership-based services. He/she will have to keep his/her own secret password w_i and m secret tokens k_j, $j = 1.2.\cdots, m.$ for these services. Therefore, an algorithm which can create a master key for these secrets is quite necessary. According to the following theorem, a master key algorithm does exist if all of the secret tokens and the password are generated by the SCPC and the secret tokens and password can be easily derived from the master key with public information.

Theorem: If $k_1 = \alpha^{d_1} \pmod n$ and $k_2 = \alpha^{d_2} \pmod n$ are two secrets generated by the SCPC associated with public primes e_1, and e_2, respectively, then $k = \alpha^{d_1 d_2} \pmod n$ is the master key for k_1 and k_2 and k can be easily derived from k_1 and k_2.

Proof: Let's first prove $k = \alpha^{d_1 d_2} \pmod n$ is the master key for k_1 and k_2. By computing $k_i = k^{T/e_i} \pmod n$, where $T = e_1 e_2$, one can easily derive k_i, $i = 1$ or 2. So k is the master key of k_1 and k_2. Then we prove the master key can be easily generated from k_1 and k_2. Since $\gcd(e_1, e_2)$ must be 1 and $k^{e_1} = \alpha^{d_2} = k_2 \pmod n$ and $k^{e_2} = \alpha^{d_1} = k_1 \pmod n$, one can find one pair of (a, b) from Euclid algorithm [6] such that $a * e_1 + b * e_2 = 1$ and then compute k as

$$(k_1)^b * (k_2)^a = (k)^{b*e_2 + a*e_1} = k(\bmod\ n)$$

Q.E.D.

By iteratively applying the same method, one can create a master key for a large number of secret keys if he/she subscribes to many membership-based services.

According to the above theorem, the master key for user i who has access to m membership-based services can be computed as

$$k_{\text{master}} = \alpha^{d_i \Pi(s_j)} \bmod n. \qquad \text{for } j = 1, 2, \cdots, m.$$

Note that k_{master} can be used to derive one's own password and those secret keys. If a service is removed from this user's access list, say, the service with public prime r_v, a new master key can be computed as

$$k'_{\text{master}} = k_{\text{master}}^{r_v} \bmod n\,.$$

D. Authentication

Now if the user wants to prove that he/she is an authorized user for a service which requires user authentication, he/she needs to compute his/her secret password from the master key stored on the card:

$$w_i = (k_{\text{master}})^{t_i} = \alpha^{d_i} (\bmod\ n). \qquad \text{where } t_i = \Pi(r_j)$$

and transmit his/her ID_i to the service center. The service center will first check the validity of the submitted ID_i against the verification table. If it is found there and the corresponding prime is e_i, the user will then be asked to prove that he does possess the secret $w_i = \alpha^{d_i} (\bmod\ n)$. In most situations where communication channels are insecure, the proof of the possession of w_i can be achieved through a challenge-response process which convinces the center of one's possession of w_i without revealing w_i to any other. Shamir's zero-knowledge proof protocol [7] is a good example of such a process.

Again, if the user wants to prove that he/she is an authorized user for a membership-based service, he/she needs to compute the corresponding secret service key from the master key stored on the card. Suppose the public prime associated with this service is r_v. The corresponding service key is computed

as

$$k_c = (K_{\text{master}})^{e_i \times t_i / r_c} = \alpha^{s_c} (\text{mod } n).$$
$$\text{where } t_i = \Pi(r_j).$$

The user then proves to the center his/her possession of k_c.

IV. Advantages

In comparison to the traditional approach, in which users have to keep many passwords or tokens in order to access multiple services which require either user authentication or membership authentication, this proposed scheme has the following advantages:

1) Efficiency of Key Management on the User Side: When a user subscribes to many different services, he/she has to keep many secrets for authentication. By carefully arranging key generation, these secrets can be regenerated from a single master key which is stored in the smart card. Since each smart card has its own processing power and limited memory space, all secret keys can be regenerated within the card later in a secure manner. In addition, this master key can be updated by the user without third-party intervention when he/she subscribes to a new service, or when an old service is deleted from his/her current access list.

2) Enhanced Security: For services which require user authentication, the corresponding service centers do not have to keep their users' passwords. Therefore, the possibility of the exposure of passwords can be reduced.

3) Decentralized Service Management: Each service has its own administration and accounting. If a user is no longer eligible for a particular service which requires user authentication, the corresponding service administrator can take away his/her authority simply by removing this user's ID from the verification table.

V. Implementation And Security Issues

The cost to implement a smart card is not expensive since it contains much less hardware than a small PC, and the manufacturing technology is currently available. In fact, there are already many smart card applications in the world. A major concern about our approach is the processing speed of the smart card, since password or token derivation from a master key requires modular exponentiations which are quite time-consuming. However, this special form of integer operation also allows us to design specialized processors and algorithms to enhance the performance. It should be noted that smart card technologies have advanced in recent years, allowing the computation of X^E mod M, with 512 b operands, in less than 1.5 s [8]. Hence, smart card processing speed for this scheme may be a resolvable issue.

The security of the proposed scheme relies ultimately on the secret numbers, p and q, of the smart card producing center. Based on the fact that factoring n into p and q is difficult, any user who tries to access a membership-based service which he/she is not entitled to would have to get at least one secret value, s_j. However, if p and q are both at least 256 b long and are selected as in the well-known RSA scheme [9], even if r_j is a public value, the computing of s_j without knowing p and q is computationally infeasible. On the other hand, if any user tries to impersonate any other, he/she has to get at least one secret value, d_i. For the same reason, this is computationally infeasible.

In real applications, each card should be associated with a PIN number (perhaps four characters) to prevent it from being used illegally in case the card is lost or stolen. Failing to enter the correct PIN number within two or three tries will invoke a built-in function in the card to erase the secrets inside.

VI. Conclusion

Incorporating smart card technology and master key concept, we propose an efficient scheme for key management in multiservice environments over a large-scale network for both users and service providers. This scheme has the following features.

1) Every service can handle its own authentication and administration.

2) Every user has a smart card with a single master key. The service keys can be regenerated within the card in a very secure manner. Users also can update their master keys by themselves without third-party intervention.

3) Without storing any user password in the service center (except, perhaps, for a user PIN number), security is greatly enhanced.

References

[1] G. B. Purdy, "A high security log-in procedure," *Commun. ACM*, vol. 17, no. 8, pp. 442–445, Aug. 1974.
[2] L. Harn, D. Huang, and C. S. Laih, "Password authentication using public-key cryptography," *Comput. Math. Appli.*, vol. 18, no. 12, pp. 1001–1017, 1989.
[3] D. Chaum, "Showing credentials without identification: Signature transferred between unconditionally unlinkable pseudonyms," in *Advances in Cryptology, Eurocrypt '85.* New York: Springer-Verlag, pp. 241–244.
[4] K. Ohta and T. Okamoto, "Membership authentication for hierarchical multigroups using a master secret key," *Trans. IEICE*, vol. E 73, no. 7, pp. 1107–1110, July 1990.
[5] D. Denning, *Cryptography and Data Security.* Reading, MA: Addison Wesley, 1982, p. 41.
[6] U. Dudley, *Elementary Number Theory.* San Francisco, CA: Freeman, 1979, p. 8.
[7] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification an signature problem," in *Proc. CRYPTO '86,* Springer LNCS 263, 1987, pp. 186–194.
[8] D. de Waleffe and J. J. Quisquater, "CORSAIR: A smart card for public key cryptosystems" (extended abstract), in *Proc. CRYPTO '90,* Santa Barbara, CA, August 1990.
[9] R. L. Rivest, A. Shamir, and L. Adelman, "A method for obtaining digital signatures and public-key cryptosystem," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978.