# Correspondence

## A Protocol for Establishing Secure Communication Channels in a Large Network

Lein Harn and David Huang

*Abstract*—**Knowledge exchange and information access in a truly distributed network often require transmitting of data through an open media. Consequently, data presented through such an environment are vulnerable to attacks. To minimize such vulnerability, data transformation or encryption/decryption techniques are often utilized among senders and receivers to achieve secure communication. Since data encryption/decryption requires sharing of a secret session key, finding an efficient way to distribute the session key in a large-scale, truly distributed network has been a nontrivial task. This paper presents a protocol for efficiently distributing session keys in such an environment to establish a secure channel. We assume the target network consists of many locally trusted centers, and each center has many users attached to it. The scheme incorporates the public-key distribution concept and the RSA encryption scheme as the basic mathematical tools, but eliminates the storage problem associated with huge public-key files. In addition, the proposed scheme has the added feature of providing the authenticate session key to the two parties in a secure communication.**

*Index Terms*—**Protocol, data security, RSA, public key, encryption.**

### I. Introduction

With rapid growth of modern technology, more and more people rely on computer networks to exchange knowledge, access information, and process data in a distributed environment. One major drawback of transmitting data through such an environment is that it opens more security holes to intruders and wiretappers. To meet the demand of people who seek secret communications, various kinds of data transformation, or encryption/decryption techniques, have been developed such that the data through open media become meaningless to intruders, but the original text can be recovered when data reach the destination. The encryption/decryption performed between the sender and the receiver requires to share a secret session key between these two parties. Traditionally, private couriers were utilized to distribute the secret session key. This approach was also used when a group of users were involved in a private conference. But with the explosion of secret communications over a distributed network environment, this approach has been rendered obsolete.

In 1976, Diffie and Hellman [1] introduced the concept of the public-key distribution system, which provides a realistic alternative to the traditional approach. One of the problems arising with this approach is that unauthorized users may attempt tampering with the public keys stored in the public-key file. To solve this problem, key authentication schemes are proposed to be added in conjunction with Diffie and Hellman's scheme to make their key distribution scheme fully secure. Two such schemes are the public-key distribution with certificate [2] and the public-key distribution with tree authentication [3]. Both of these schemes, however, being centralized approaches, have the weakness that they too require the maintenance of a large public-key file. Considering the case of a large and distributed environment with millions of users, managing a centralized file with that many entries clearly is impractical [4]. In 1984, Shamir [5] proposed the new idea of an identity-based cryptosystem for distribution of session keys. Since then, many algorithms [6]–[10] have been proposed to implement this identity-based cryptographic approach. However, the security of these algorithms is still being tested by the cryptographic community. There are some other methods for distributing secret session keys to network users using a centralized key generation and key distribution approach [11, 12]. These schemes have three major drawbacks in common: 1) a large key file needs be maintained at the key center, 2) the centralized key center is the focus of cryptanalytic attack, and 3) since all key generation and distribution go through this center, traffic congestion between the key center and other nodes in the distributed network could be severe.

Other efforts have been made towards finding more suitable solutions to the user's authentication problem in a large-scale network. One scheme proposed in 1986 by Birrell *et al.* [4] assumes the existence of a hierarchical global tree-structured name service in the network. Each principal (principals could be people, machines, organizations, printers, databases, file systems, etc.) must register into this name service at a directory, and all directories maintain a fixed logical relationship to all other directories in a tree hierarchy. When principle $i$ is registered under directory $j$, directory $j$ becomes the parent of principal $i$, and a secret key $K_i$ is shared by these two entities. Each directory shares a secret key with its parent directory (if any) and with each directory that is its child in the directory tree hierarchy. Thus, a secret session key transmitted from principle $i$ to principle $k$ is first passed to directory $j$, and then, using a relative name path through a sequence of directories, to principle $k$'s directory and so to principle $k$. The problem associated with this method is that a secure communication channel between principle $i$ and principle $k$ needs to be established via some indirect trusted intermediate name service directories. Thus, the shared key between $i$ and $k$ is also exposed to all these intermediate directories.

The communication protocol that we are going to present in this paper incorporates the public-key distribution concept [1] and the RSA public-key encryption scheme [13] as the basic mathematical tools, and is suitable for a large-scale distributed network. Any number of network users (or principals) can be added onto or deleted from the network freely without affecting the performance of this scheme. Even though public-key concepts are utilized, there is no need for each user to keep a copy of the global public-key file. We assume that the network is distributed in the sense that there does not exist a globally trusted key distribution center (KDC) or authentication center. However, we assume that the network consists of many local centers, and all network users must register under one of these centers. These local centers are *trusted* in the

Manuscript received December 6, 1988; revised September 1, 1990 and June 8, 1993.

L. Harn is with the Computer Science Telecommunications Program, University of Missouri—Kansas City, Kansas City, MO 64110.

D. Huang was with the Computer Science Telecommunications Program, University of Missouri—Kansas City, Kansas City, MO 64110. He is now with the Wollongong Group Inc.

IEEE Log Number 9212808.

sense that they are expected to handle requests from their users exactly according to the proposed protocol, and they have no ability to interpret secret conversations carried out among different users of the system.

Section II gives a detailed description of our scheme; Section III presents the security analysis of our scheme and discusses its advantages and tradeoffs; Section IV is the conclusion.

## II. The Proposed Scheme

Our proposed scheme requires a large-scale distributed network service consisting of many local centers, each local center having several or many users attached to it, as shown in Fig. 1. The network service may be in a computer network system, consisting of many hosts and users attached to hosts, or in a telephone system, consisting of many switching centers, and telephones attached to the centers, or in a networked banking association, consisting of many headquarters, and banks attached to headquarters, etc. Any user who wants to access the network service must register under one of the local centers (LC) when first joining the network, providing to LC some personal information. Each LC must follow certain procedures and protocols in order to make this scheme work.

### A. Initialization

Each $LC_i$ (local center $i$) needs to select two distinct large primes $p_i$ and $q_i$, and calculate

$$n_i = p_i * q_i$$

and

$$\phi(n_i) = (p_i - 1) * (q_i - 1)$$

where $\phi$ is called the Euler totient function [15, p. 41].

Each $LC_i$ needs to select signature and verification keys $e_i$ and $d_i$, respectively, such that

$$(e_i * d_i) \bmod \phi(n_i) = 1. \quad (1)$$

Just as in the RSA scheme [15, pp. 101–102], if (1) is satisfied, then the following verification operation (3) will restore the original plaintext message $M$ from its signature $C$:

$$\text{Signature Generation: } C = E_{e_i}(M) = M^{e_i} \bmod n_i \quad (2)$$

$$\text{Signature Verification: } M = D_{d_i}(C) = C^{d_i} \bmod n_i. \quad (3)$$

The two values $d_i$ and $n_i$ are recorded onto a verification table (also called the public-key table of the local centers), as shown in Fig. 2. This table is duplicated at each LC, and can be made public without threatening the system's security. Other information, such as $e_i$, $p_i$, $q_i$, and $\phi(n_i)$, should be kept secret and are known only to $LC_i$. When a new LC is created within the network and chooses its public and private keys, its public key is added to the table, and this update information should be broadcast to all LC's. On the other hand, when an old LC is deleted from the network, one entry in the public-key table of all LC's needs to be removed.

If user $i$ ($U_i$) wants to join the network, it is mandatory that $U_i$ does two things during registration.

1) Select a secret random integer $X_i$ and compute the corresponding public-key information $Y_i$ as

$$Y_i = a^{X_i} \bmod N$$

where $N$ is a large prime and $a$ is an integer within the range $[1, N - 1]$. Both $a$ and $N$ are publicly known throughout the network. $Y_i$ is $U_i$'s public key.

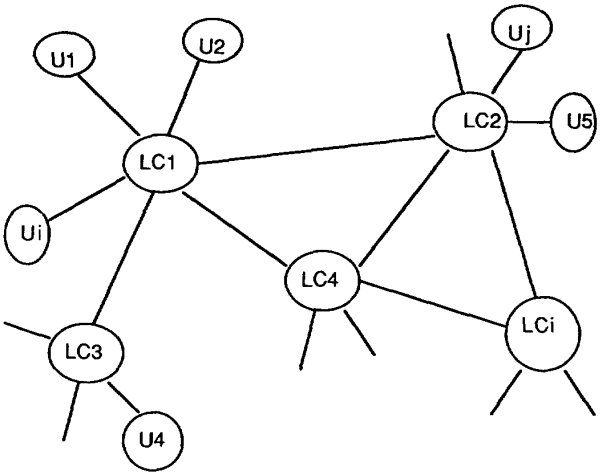


Fig. 1. A large, distributed network service.

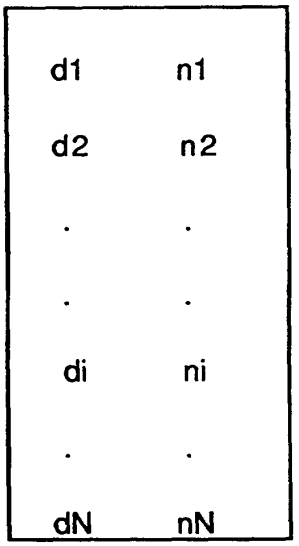


Fig. 2. Public-key table for local service centers.

2) Register under some LC. Registration is similar to applying for a credit card or having a new telephone line installed; it may involve verification of the user's identity and the checking of the user's references. More specifically, if $U_i$ wants to register under $LC_1$, then the registration procedure should include the following steps.

a) $U_i$ submits his/her name, public key $Y_i$, and some personal information such as social security number, hair and eye colors, etc., to $LC_1$.

b) After verifying the correctness of the submitted information, $LC_1$ appends its own $ID$, $ID_{LC_1}$, to the user's supplied information, and makes a new personal $ID$, $ID_i$, for $U_i$. In other words, this new $ID_i$ is formed by concatenating several segments of data, and each segment contains specific information such as $U_i$'s name, $U_i$'s SSN, $Y_i$, $ID_{LC_1}$, etc.

c) $LC_1$ calculates $S_i = E_{e_1}(ID_i)$ and issues both $S_i$ and $ID_i$ to $U_i$. Since both $S_i$ and $ID_i$ are large integers, it is difficult for $U_i$ to memorize them. $S_i$ and $ID_i$ will need to be entered into the system

from a physically interconnected point to the network each time that $U_i$ wants to authenticate himself/herself to a remote entity. To type in large, hard-to-memorize character strings may not be a very pleasant experience. Thus, we recommend the use of some local storage device, e.g., a PC diskette or magnetic card or smart card. In other words, at registration time, the values of $S_i$ and $ID_i$ should be recorded onto such a device and then issued to $U_i$.

### B. Authentication

Suppose $U_i$ in Fig. 1, attached to $LC_1$, wants to create a secure channel (to share a secret session key) with $U_j$ attached to $LC_2$. The authentication procedure works as follows:

$U_i$ sends $S_i$, $ID_i$ as plaintext through the network to $U_j$.

Upon receiving $S_i$ and $ID_i$, $U_j$ extracts $ID_{LC_1}$ from $ID_i$. Next, $U_j$ obtains $d_1$ and $n_1$ from the verification table of his/her local center $LC_2$. Then, $U_j$ computes a user identification for $U_i$ as follows:

$$ID_i'' = D_{d_1}(S_i).$$

If $ID_i'' = ID_i$, the authentication procedure will continue; otherwise, the procedure is terminated.

It is to be noted that users of the network need to be able to identify each other by user name and to access local center public keys from the local center to which they are attached. Of course, a user $U_i$ could submit the other user's name $U_j$ to a name server and obtain, from the name server, along with the network address of user $U_j$, the public keys for the local center to which $U_j$ is attached.

If $ID_i''$ and $ID_i$ match correctly, then $U_j$ transmits $S_j$ and $ID_j$ to $U_i$. Upon receiving $S_j$ and $ID_j$ from $U_j$, $U_i$ computes

$$ID_j'' = D_{d_2}(S_j)$$

in the same manner as $U_j$ computed $ID_i''$ and compares $ID_j''$ with the received $ID_j$. If these two values do not match, $U_i$ may either disconnect from the network or retry by resubmitting his/her $S_i$ and $ID_i$ to $U_j$. If these values do match, $U_i$ extracts $Y_j$ from $ID_j$ and computes the secret common session key $K_{ij}$ shared by $U_i$ and $U_j$, according to the public-key distribution scheme proposed by Diffie and Hellman [1], as follows:

$$K_{ij} = Y_j^{X_i} \bmod N$$

where $X_i$ is the secret information known only to $U_i$ and $N$ is a publicly available large prime chosen by the global service. This computed $K_{ij}$ is the secret session key that will be shared by $U_i$ and $U_j$ for establishing a secure channel.

In order to verify that $U_j$ has successfully obtained the same $K_{ij}$, and to avoid playback attack from a third party, $U_i$ could use $K_{ij}$ to encrypt a small paragraph of text which contains a time stamp of current time and a sentence, such as

$$C = E_{K_{ij}}(\text{"I am } U_i \text{ want to talk to } U_j, \text{ current time"})$$

and sends this ciphertext to $U_j$. $U_j$ can extract $Y_i$ from $ID_i$, computes

$$K_{ij} = Y_i^{X_j} \bmod N$$

where $X_j$ is the secret information known only to $U_j$, and then uses $K_{ij}$ to decrypt the ciphertext to

$$D_{K_{ij}}(C) = \text{"I am } U_i, \text{ want to talk to } U_j, \text{ current time"}$$

to thus verify the correctness of $K_{ij}$. If the decrypted sentence is correct, then a secure channel has been successfully established between $U_i$ and $U_j$.

## III. Security Analysis

There are four possible attacks that we need to discuss in this section. They are the following.

*(Attack 1):* If an imposter wants to pretend to be $U_i$ and establish a "secret channel" with $U_j$, then he/she can take the following approach. Since $a$ and $N$ are publicly known information, select a random number $X'$ and generate the corresponding $Y'$ according to the equation

$$Y' = a^{X'} \bmod N.$$

Because $S_i$ and $ID_i$ are transmitted in plaintext form, and can be intercepted by wiretappers, the imposter can intercept them, replace $Y_i$ in $ID_i$ by $Y'$ to obtain $ID_i'$, and then send $S_i$ and $ID_i'$ to $U_j$. But, upon receiving $S_i$ and $ID_i'$, $U_j$ first verifies whether $S_i$ and $ID_i$ are correct by computing

$$ID_i'' = D_{d_1}(S_i)$$

and comparing the received $ID_i'$ with the computed $ID_i''$. Since $ID_i'$ is different from $ID_i''$, the comparison fails, and the request to establish a "secret channel" between the imposter and $U_j$ is rejected. Thus, this attack fails.

*(Attack 2):* An imposter may modify $ID_i$ to $ID_i'$ in exactly the same way as in (Attack 1) above, and also modify $S_i$ to $S_i'$ such that $ID_i' = D_{d_1}(S_i')$. If so, the comparison between $ID_i'$ and $ID_i''$ will succeed. However, since $S_i$ is calculated from the equation

$$S_i = E_{e_1}(ID_i)$$

and $e_1$ is known to $LC_1$ only, computing $S_i'$ from $ID_i'$ would be extremely difficult without knowledge of $e_1$.

*(Attack 3):* Since $S_i$ and $ID_i$ can be intercepted by an intruder during transmission, people other than $U_i$ who have obtained $S_i$ and $ID_i$ can also send these two values to someone else, claiming to be $U_i$. However, since they lack knowledge of the personal secret key $X_i$, they still cannot obtain the correct session key.

*(Attack 4):* If an imposter wants to pretend to be $U_i$, and attempts to establish a "secret channel" with $U_j$, he/she could intercept not only $S_i$ and $ID_i$, but also the encrypted paragraph

$$E_{K_{ij}}(\text{"I am } U_i, \text{ want to talk to } U_j, \text{ current time"})$$

and play it back whenever he/she chooses. But because of the time stamp associated with the ciphertext and the lack of knowledge of $K_{ij}$, the ciphertext containing a correct time cannot be calculated correctly by the imposter, thus this attack will fail.

## IV. Summary and Conclusion

A new, authenticated public-key distribution scheme for a distributed computer network is proposed. We believe that this scheme has the following merits.

1) The shared, secret key can be authenticated by both users and be distributed successfully without the help of a globally trusted centralized KDC. This is advantageous because: a) for a truly distributed, large-scale network, such a KDC may not exist, and b) the security of a centralized key distribution scheme totally depends on the security of the centralized KDC; compromise of the KDC endangers all network security [3].

2) Even though the proposed scheme utilizes public-key cryptography as the basic mathematical tool, it effectively eliminates the large public-key file management problem. There is no need for each network user to keep a copy of the global public-key file. Instead, each individual local center needs to keep a public-key file for all local centers for authentication purposes. We also require a

user registration process and encourage the use of a local storage device.

3) A "secure channel" is said to be established between users $A$ and $B$ if these two users have successfully obtained the common secret session key. With our scheme, such a key can be obtained *only* by $A$ and $B$ (thus not exposing it to other users or to local centers). We believe this feature is more advantageous compared with Birrell's scheme in which this secret key is also exposed to certain intermediate trusted directories.

4) Similar to Simmon's scheme [14], in our scheme, each user holds a secret number, and this number is never transmitted directly through the network. Not even the local center has any knowledge of this number. Thus, the probability of having it exposed is kept to a minimum.

5) With this scheme, users can be added onto or deleted from the network freely without degrading the performance of the scheme. Only when there is a new local center added onto or an old local center withdrawn from the network must some update work be done. For example, when a new LC is created in the network, all local center verification tables need to be updated; this can be accomplished by broadcasting the public key of this new LC to all the other LC's. Finding an efficient way for authenticated key distribution in a large-scale, truly distributed network is a difficult task. Our proposed scheme is one of many efforts towards finding a possible solution. We hope to see more constructive work in this field in the near future.

## REFERENCES

[1] W. Diffie and M. Hellman, "New directions in crytography," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 644-654, Nov. 1976.

[2] L. M. Kohnfelder, "Towards a practical public-key cryptosystem," B.S., M.I.T., Cambridge.

[3] R. C. Merkle, "Protocols for public key cryptosystem," in *Proc. IEEE Symp. Security and Privacy*, 1980, pp. 122-134.

[4] A. D. Birrell, B. W. Lampson, R. M. Needham, and M. D. Schroeder, "A global authentication service without global trust," in *Proc. IEEE Symp. Security and Privacy*, 1986, pp. 223-230.

[5] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Adv. in Cryptol.—Crypto '84*, Santa Barbara, CA, Aug. 1984, pp. 47-53.

[6] E. Okamoto, "Proposal for identity-based key distribution systems," *Electron. Lett.*, vol. 22, pp. 1283-1284, 1986.

[7] K. Koyama and K. Ohta, "Identity-based key conference key distribution systems," in *Adv. in Cryptol.—Crypto '87*. New York: Springer-Verlag, 1987, pp. 175-184.

[8] E. Okamoto, "Key distribution systems based on identification information," in *Adv. in Cryptol.—Crypto '87*, Santa Barbara, CA, May 1987, pp. 194-202.

[9] S. Tsujii and T. Itoh, "An ID-based cryptosystem based on the discrete logarithm problem," *IEEE J. Select. Areas Commun.*, vol. 7, pp. 467-473, May 1989.

[10] E. Okamoto and T. Tanaka, "Key distribution system based on identification information," *IEEE J. Select Areas Commun.*, vol. 7, pp. 481-485, May 1989.

[11] G. J. Popek and C. S. Kline, "Encryption protocols, public key algorithms, and digital signatures in computer networks," in *Foundations of Secure Computation*, pp. 133-153.

[12] R. M. Needham and M. D. Schroeder, "Using encryption for authentication in large networks of computers," *Commun. ACM.*, vol. 21, pp. 993-999, Dec. 1978.

[13] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM.*, vol. 21, pp. 120-126, Feb. 1978.

[14] G. J. Simmons, "An impersonation-proof identity verification scheme," in *Adv. In Cryptol.—Crypto '87*, Aug. 1987, pp. 211-215.

[15] D. E. R. Denning, *Cryptography and Data Security*. Reading, MA: Addison-Wesley, 1982.