ELSEVIER

Computer Standards & Interfaces 21 (1999) 417–428

COMPUTER STANDARDS & INTERFACES

www.elsevier.com/locate/csi

Authentication protocols for the broadband ISDN billing system

Chi-Chun Lo *, Yi-Chun Yeh

Institute of Information Management, National Chiao-Tung University, 1001 Ta Hseuh Road, Hsinchu 300, Taiwan

Received 10 May 1999; received in revised form 25 June 1999; accepted 6 July 1999

Abstract

The broadband integrated services digital network (B-ISDN) offers a wide variety of services to its customers. From both provider's and user's perspective, a fair and flexible billing system is indispensable to the operation of B-ISDN. In this paper, we discuss different billing policies, and their corresponding authentication protocols, which can be used by the B-ISDN billing system. Flat rate and usage-based rate are two most frequently used billing policies. Because of the diversity of B-ISDN services, combining these two billing policies, along with priority pricing, is a feasible scheme for designing a B-ISDN billing system. The membership authentication protocol (MAP) and the personal authentication protocol (PAP) are proposed to support the authentication requirements of the flat rate and usage-based rate billing policies, respectively. The MAP applies the cipher block chaining (CBC) encryption in the two-party authentication protocol, while the PAP is a modified version of the station-to-station (STS) authentication protocol. Both protocols are designed in conjunction with a suggested key management method. Cryptanalysis shows that both the MAP and the PAP are very secure. Simulation results indicate that both protocols are efficient for providing mutual authentication. Together, these two protocols provide a good authentication mechanism to the B-ISDN billing system.

Keywords: Flat rate billing policy; Usage-based rate billing policy; Master key; Membership authentication protocol; Personal authentication protocol

1. Introduction

The broadband integrated services digital network (B-ISDN) offers a wide variety of services to its customers. Some customers want to be charged by flat rate while others prefer to pay their fees on the basis of the usage of the resources they consumed. Therefore, the B-ISDN billing scheme has to be fair and flexible. Another major concern of designing a B-ISDN billing system is user authentication. How to design an authentication framework, which can satisfy various requirements of B-ISDN services [13,14], is a major challenge.

Vecchi [11] described that the control structure of B-ISDN should be distributed to support broadband services as network applications become increasingly complex and diverse; hence, it can be modeled by the client/server paradigm. Billing functions are included in every individual server. In other words, every server should have its own billing scheme and security control. In the following discussions, the client/server model is assumed for the B-ISDN billing system. Under this model, a billing system is composed of three service modules: the directory service module, the authentication service module,

* Corresponding author. Tel.: +886-3-5731909; fax: +866-3-5723792; E-mail: cclo@cc.nctu.edu.tw

0920-5489/99/$ - see front matter
PII: S0920-5489(99)00024-0

and the usage service module. These three modules interact in the following way:

- Upon receiving a service request from a client, the server identifies this client via the directory service module and the authentication service module. Then, it negotiates service price and quality of services (QoSs) with the client.
- Once a connection is established, the usage service module is invoked to measure the resources consumed by the client until the connection is released.
- By using the authentication service module and the usage service module, the server can charge the client for a reasonable price.

In this paper, we analyze the requirements of the B-ISDN billing system and recommend a feasible billing scheme for it. A key management method is suggested. Authentication protocols are then designed in conjunction with the suggested key management method. From cryptanalysis and simulation analysis, we notice that the proposed authentication protocols are secure and efficient. The rest of this paper is organized as follows. Section 2 discusses the security issues of the B-ISDN billing system. The recommended billing scheme is presented in Section 3. Section 4 states the suggested key management method. Section 5 details the proposed authentication protocols. Cryptanalysis and simulation analysis are given in Sections 6 and 7, respectively. Section 8 concludes this paper with possible future research directions.

2. Security issues

2.1. Authentication protocol

The purpose of authentication is to identify communicating parties to ensure that no hostile third party can masquerade one of them. Authentication protocol also produces a session key for communicating parties. With this shared secret, communicating parties can secure their message exchanges in subsequent communications. Since the B-ISDN control structure is described by the client/server model rather than the trusted third-party model, we restrict our survey to the two-party-based authentication protocols.

2.1.1. The International Standard Organization (ISO) two-way authentication protocol [15]

Bird et al. [5] presented a simple two-way authentication protocol which provides mutual authentication using challenge–response. This protocol is simple. However, it suffers from interleaving attacks; e.g., the known plaintext attack, the chosen ciphertext attack, the oracle session attack, and the parallel session attack. The ISO two-way authentication protocol improves the security of the simple two-way authentication protocol. Nevertheless, it is still not completely free from the oracle session attack. An intruder can use one party as an oracle ''decryption server'' and successfully pretend as an authorized party.

2.1.2. The secure two-party authentication protocol [2]

To prevent oracle session attacks, a two-way authentication protocol should never perform cryptographic operations on inputs which may be altered by an attacker [2]. The cipher block chaining (CBC) encryption can be employed in the ISO two-way authentication protocol to ensure *pseudo-independence* of responses of different sessions; thus, making it unfeasible to compute the response of one session by interleaving other sessions.

The secure two-party authentication protocol proposed by Bird et al. uses CBC encryption. It is designed to defeat interleaving attacks.

2.1.3. The X.509 three-way exchange [16]

The X.509 three-way exchange, described in ITU-T Recommendations X.509, is a novel example of public-key-based authentication protocols. Key management of this protocol is easier than those of the protocols which use symmetric cryptography. However, the X.509 three-way exchange has some noticeable problems [12]. Since the X.509 three-way exchange performs encryption before signing, an attacker may remove the signature from the encrypted message and replace it with his own [1]. Furthermore, the X.509 three-way exchange does not provide *perfect forward secrecy* so that the disclosure of the private key may compromise the session key.

2.1.4. The station-to-station (STS) protocol [6]

The STS protocol is another example of public-key-based authentication protocols. The STS performs Diffie–Hellman [7] key exchange, followed by signature exchange. Diffie [6] claimed that the STS has the following desirable features: perfect forward secrecy, direct authentication, and signing before encrypting. However, Lin [9] presented several weaknesses of the STS. By replacing key exchange parameters, an attacker may guess the session key successfully.

2.2. Key management

In a multi-service network environment, different service providers may use different security mechanisms. A user may have to keep a number of keys for requesting services from different providers. However, most low-end network computers are not capable of storing and managing these keys. To solve this problem, Harn and Lin [8] proposed a key management method for accessing multi-service networks. This RSA-based key management method incorporates the master key concept [4]. It consists of three components: the smart card producing center (SCPC), users, and service centers. The SCPC manufactures smart cards for all entities which include users and service centers. Each smart card contains a unique public/private key pair. On the basis of this key pair, a user keeps a corresponding master key, and a service center stores a list of corresponding service tokens, in their smart cards, respectively. The SCPC should be trusted by all entities and is not allowed to involve in the on-line authentication process. Users are those who request different network services with their smart cards. A user is not supposed to reveal his/her secret key to any service center. Each service center manages a specific network service and is responsible for both user registration and user authentication. A service center maintains a list of registered users' identities and their corresponding public keys.

Harn and Lin's method supports two kinds of authentications, *user authentication* and *membership authentication.* By presenting his/her identity and password, a usage-based rate user can prove his validity to the service center during log-in process. This is the so-called *user (personal) authentication.* However, for a flat rate service, every user pays the same membership fee for getting the service. Service charge is based on whether the user subscribes the service or not. Upon requesting the service, a user should prove his validity by demonstrating the possession of a pre-distributed *token*. This is the so-called *membership authentication.*

Harn and Lin's key management method supports decentralized services and provides an efficient key management mechanism. However, their approach has several problems which may expose underline systems to various attacks. These problems are stated as follows.

- After computing the password with his/her master key, a user transmits his/her identity, along with the password, to the requested service center via an open channel. Since the channel is insecure, the password in plaintext could be wiretapped or compromised by an attacker.
- The SCPC generates key pairs with the same modulus for all entities; consequently, some entities may expose themselves to the common modulus attack.
- For a service center, its secret key (service token) is used as the shared secret between itself and its users, for membership authentication. The recommended key length for service token is 1024 bits. For symmetric cryptography, such a long key is not necessary, because the 56-bit key for the Data Encryption Standard (DES) is secure enough.
- For supporting priority pricing [5], members should be classified into different classes, with different priorities and prices assigned to each class. However, there is only one service class in Harn and Lin's method.

3. Recommended billing scheme

Flat rate and usage-based rate are two most frequently used billing policies. Flat rate is beneficial to high-utilization users while usage-based rate is beneficial to low-utilization users. For flat rate users, they pay a fixed fee on a per-day, per-month or per-year basis. For usage-based rate users, in addition to the initial connection cost, they pay extra fees in accordance with connection frequency and volume of data transferred. To ensure the balance between fairness

Table 1
Comparison between the flat rate billing policy and the usage-based rate billing policy

	Flat rate	Usage-based rate
User behavior	frequently use with high-volume data transferred	occasionally use with low-volume data transferred
Billing scheme	charge membership fee by day/month/year	charge usage fee of the resources consumed
Priority pricing	classify users into multiple service classes	negotiate price during connection establishment
Authentication mechanism	membership authentication	personal authentication
Authentication time	connection establishment	connection establishment and connection release

and efficiency, the two billing policies should coexist so that the most efficient use of network resources can be achieved. A sophisticated priority pricing scheme is also needed for providing differential pricing. Those performance-sensitive users should pay more money than those who are performance-insensitive [5]. In the study of Cocchi et al. [5], they compared two different pricing policies: *flat-per-byte fee* and *graduated fee*. Measuring user satisfaction as a function of both cost and QoSs received, they proved that better services can be provided with priority pricing than without it. Parris and Ferrari [10] showed that users tend to disperse their requests over a long period of time when using the peak load pricing policy. The distributed use of resources reduces the blocking probability and eases the congestion problem. According to the discussions aforementioned, we recommend that the B-ISDN billing scheme support both flat rate and usage-based rate billing policies, along with priority pricing.

Table 1 compares the flat rate billing policy with the usage-based rate billing policy.

4. Suggested key management method

The design of the suggested key management method follows the management structure proposed by Harn and Lin [8]. For supporting personal authentication, every entity needs a unique public/private key pair. Key pairs must be generated from different moduli, since the common modulus attack is a serious problem in Harn's and Lin's key management method. As for membership authentication, master key is used to generate multi-class service tokens. In principle, the suggested key management method is designed to be immune from the common modulus attack and be able to classify members into multiple classes.

4.1. System components

The suggested key management method consists of four components: the SCPC, users, service centers, and certificate authorities (CAs).

4.1.1. SCPC

The SCPC should be trusted by all entities and is not allowed to involve in the on-line authentication process. It has the following responsibilities.

- Generate a public/private key pair for every entity (user or service center) with different moduli.
- Generate multiple public/private prime pairs for every service center, with the same modulus. If a service center provides a multi-class service, it will receive multiple prime pairs for that service.
- Generate service tokens for every service center. A service center which provides a multi-class service will receive multiple service tokens for that service.
- Manufacture a smart card for every user. In addition to the personal public/private key pair, the smart card contains a master key which is used to derive needed service tokens.

4.1.2. Users

A user requests different network services with his/her smart card.

4.1.3. Service centers

Every service center provides a specific network service and has its own billing scheme. A service center is responsible for user registration, user authentication, and user billing.

4.1.4. CAs

For a usage-based user, he/she should be able to verify the signature of the requested service center; therefore, CAs are required for signing public-keys of service centers.

4.2. Key generation

Key pair, prime pairs, service tokens, and master key are generated as follows.

4.2.1. Public / private key pair

Every entity gets a unique public/private key pair which is generated by modular arithmetic with different moduli for different entities:

- modulus: c — product of two primes, p and q (p and q must be kept in secret);
- public key: e — relatively prime to $(p-1)(q-1)$; and
- private key: d — $d \equiv e^{-1} \bmod [(p-1)(q-1)]$.

4.2.2. Public / private prime pairs

The *hierarchical keying* concept proposed by Chick and Tavares [4] is used to classify users. On the basis of this concept, each membership-based service will have multiple prime pairs assigned. Pairs are generated from a common modulus and are kept in secret by SCPC. For preventing the common modulus attack, a service center is not allowed to access its private primes:

- number of service classes: l;
- common modulus: w — product of two primes, u and v (w can be made public while u and v must be kept in secret);
- public primes: $r_1, r_2, r_3, \ldots, r_l$ — relatively prime to $(u-1)(v-1)$;
- private primes: $s_1, s_2, s_3, \ldots, s_l$ — $s_i \equiv r_i^{-1} \bmod [(u-1)(v-1)]$;
- class-i public prime: $R_i = \Pi(r_i)$; and
- class-i private prime: $S_i = \Pi(s_i)$.

4.2.3. Service tokens

Multi-class service tokens are computed with service center's private primes by SCPC, and then manually distributed to the service center.

- generator: $a \in [1, n-1]$, where n is a positive integer; and
- class-i service token: $K_i = a^{S_i} \bmod n$.

The length of service token can be shortened by using a small modulus; e.g., n is equal to 64 instead of 1024. Communicating parties can negotiate n and adjust the key length accordingly.

4.2.4. Master key

Master key is a compact representation of a set of service tokens. After subscribing a set of services, a user receives his/her master key from SCPC. Upon requesting service from the service center, a user can generate needed service tokens with his master key and public primes of all registered service centers:

- the number of registered services: m;
- the highest priority for accessing service center j: P_j; and
- master key: $K_{\text{master}} = a^{\Pi(S_{P_j})} \bmod n$ for $j = 1, 2, \ldots, m$.

Theorem 1: *A user can generate the class-i service token of service center C (K_i^c) with his master key ($a^{\Pi(S_{P_j})} \bmod n$) and public primes of all registered service centers.*

$$K_i^c = \left(K_{\text{master}}\right)^{\Pi(R_{P_j})/R_i^c} \bmod n.$$

Proof: According to Euler's generalization of Fermat's little theorem, we have:

$$K_{\text{master}}{}^{\Pi(R_{P_j})/R_i^c} \bmod n = \left(a^{\Pi(S_{P_j})} \bmod n\right)^{\Pi(R_{P_j})/R_i^c} \bmod n;$$

$$K_{\text{master}}{}^{\Pi(R_{P_j})/R_i^c} \bmod n = \left(\left(a^{\Pi(S_{P_j})*\Pi(R_{P_j})} \bmod n\right)^{(R_i^c)^{-1}}\right) \bmod n;$$

$$K_{\text{master}}{}^{\Pi(R_{P_j})/R_i^c} \bmod n = \left(a^{(R_i^c)^{-1}} \bmod n\right);$$

$$K_{\text{master}}{}^{\Pi(R_{P_j})/R_i^c} \bmod n = \left(\left(a^{(R_i^c)^{-1}} \bmod n\right)^{R_i^c S_i^c}\right) \bmod n;$$

$$K_{\text{master}}{}^{\Pi(R_{P_j})/R_i^c} \bmod n = a^{S_i^c} \bmod n \qquad \text{QED.}$$

5. Proposed authentication protocols

The *membership authentication protocol* (MAP) and the *personal authentication protocol* (PAP) are

proposed for supporting the flat rate and usage-based rate billing policies, respectively.

We assume that the cryptosystem is protected by a trusted machine. The underlying cryptographic mechanisms; e.g., RSA, DES, are not vulnerable with regards to data privacy and data integrity.

5.1. MAP

5.1.1. Design rationale

Since a user is capable of computing needed service tokens, membership authentication can use the pre-assigned service token as the shared secret between a user and the requested service center. Therefore, the design of the MAP is based on symmetric cryptography.

With the flat rate billing policy, a user need not to reveal his identity; instead, he is only required to prove his membership. This implies that user privacy should be guaranteed. Also, entity authentication during connection establishment should be provided, and the integrity of the service request message should be preserved.

According to the discussions aforementioned, the MAP is referred to the *secure two-party authentication protocol* [2].

5.1.2. Protocol design

The MAP includes three steps which are illustrated in Fig. 1.

5.1.2.1. Step 1. Member A requests service from service center B. The service request message contains service center identifier and the requested service class (i). The service request message is sent, along with a non-repeated random number (nonce), N_m, where N_m is a challenge to service center B.

5.1.2.2. Step 2. Service center B uses class-i service token (K_i^B) to perform CBC encryption on service center identifier, the requested service class (i), and nonce N_m, where N_m is the response to member A (B is authenticated); uses K_i^B to encrypt service center identifier, the requested service class, and the session key ($K_{session}$), where $K_{session}$ is used to secure subsequent data transfer; sends the two encrypted messages along with nonce N_s, where N_s is a challenge to member A.

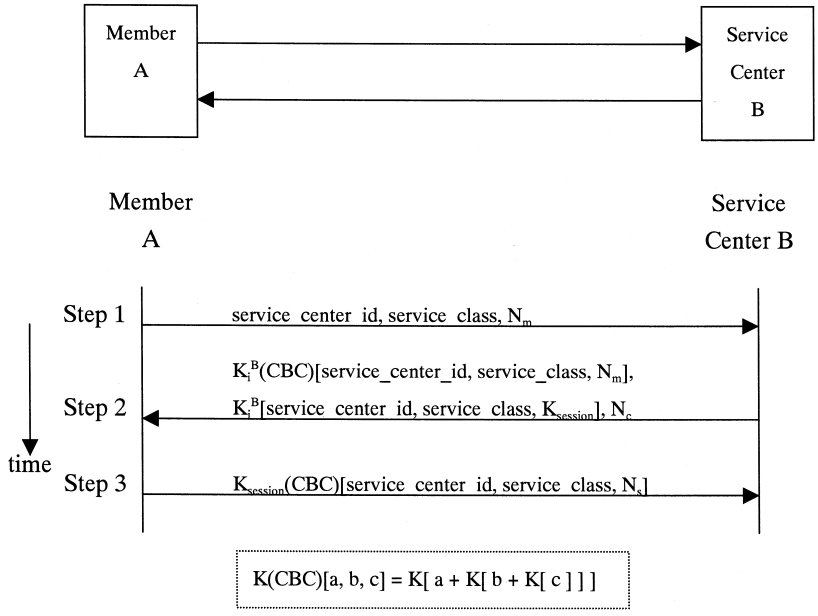


Fig. 1. The MAP.

5.1.2.3. Step 3. Member A derives class-*i* service token (K_i^B) from his/her master key; uses K_i^B to perform CBC encryption on service center identifier, the requested service class, and nonce N_m, and then compares this encrypted message with the CBC-encrypted message received; uses K_i^B to decrypt the session key ($K_{session}$); uses $K_{session}$ to perform CBC encryption on service center identifier, the requested service class, and nonce N_s, where N_s is the response to B (A is authenticated.); sends the encrypted message to B.

5.1.3. Comments

Differences between the MAP and the secure two-party protocol are stated as follows.

(1) The service request message is encrypted along with nonce in the second and third steps. It is worth noticing that the size of the service request message is usually very small. Since the contents of the service request message; e.g., service center identifier and the requested service class, are limited and easy to predict, an attacker can find *probabilistic encryption* by guessing enough strings. To foil such

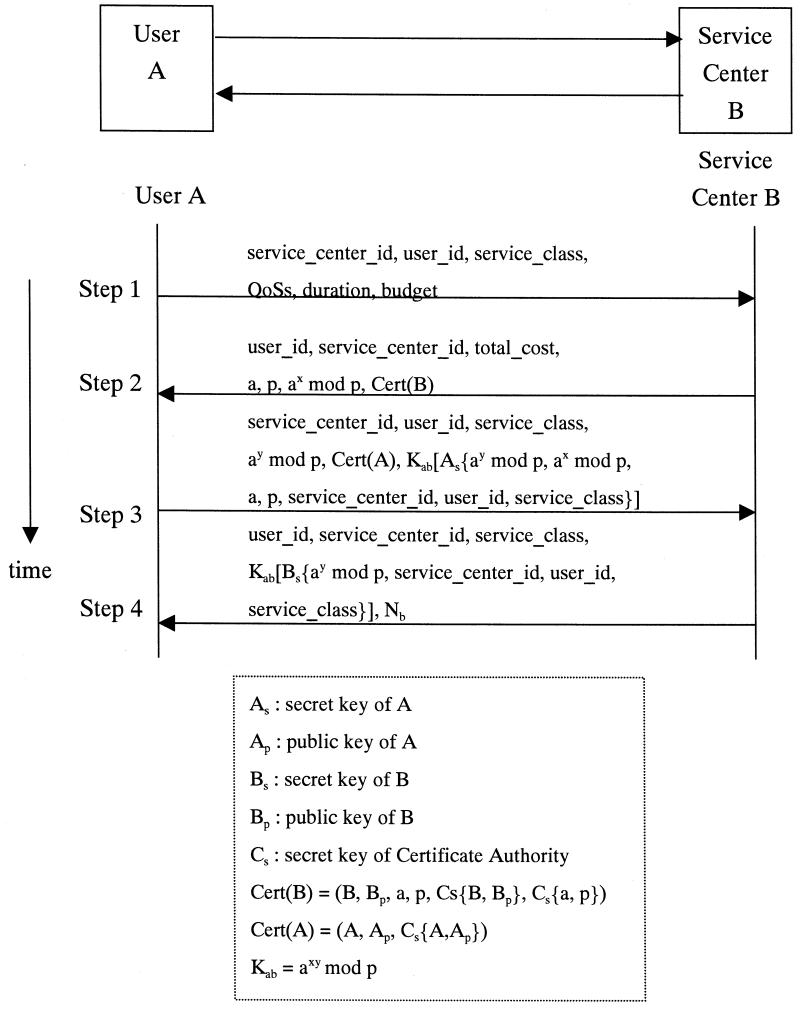


Fig. 2. Connection phase of the PAP.

an attack, it is practical to encrypt the service request message along with an unpredictable random number (nonce) [2].

(2) The session key ($K_{session}$) is exchanged in the second step for securing subsequent data transfer.

5.2. PAP

5.2.1. Design rationale

Unlike membership authentication, a user should prove his/her identity to the requested service center. Because user and service center do not share any ''secret'', e.g., service token used in the MAP, they cannot authenticate each other using symmetric cryptography. Therefore, the design of the PAP is based on public-key cryptograph.

With the usage-based rate billing policy, a user is charged for the resources he consumed. Authentication is required not only when a connection is being established but also when the connection is being released. Since a user is charged afterwards, the service center should be able to prevent him from denying the service he received. This implies that non-repudiation is needed. Furthermore, the integrity of the service request message should be preserved.

According to the discussions aforementioned, the PAP is referred to the STS authentication protocol. In this design, CAs are required to certify the authenticity of public-keys and Diffie–Hellman key exchange parameters.

5.2.2. Protocol design

The PAP has three phases: the connection phase, the data transfer phase, and the release phase.

5.2.2.1. Connection phase. The connection phase includes three steps. Fig. 2 depicts the connection phase of PAP.

Step 1. User A requests service form service center B. The service request message contains service center identifier, user identifier, the requested service class, QoS parameters, the estimated service duration, and the budget.

Step 2. Service center B calculates the estimated cost based on A's request; selects a large integer, x, and generates a, p, a^x mod p, where a and p are Diffie–Hellman key exchange parameters and a^x mod p is a challenge to A; sends the response to A with which B's certificate (Cert(B)) is included [note that B, B's public key, a, and p are signed inside Cert(B) by CA].

Step 3. User A validates a and p; selects a large integer, y, and computes the session key (K_{ab}) and a^y mod p; signs a, p, a^x mod p, a^y mod p, service center identifier, user identifier, and the requested service class, where a^x mod p is the response to B (A is authenticated); uses K_{ab} to encrypt the signature; sends the response to B with which A's certificate (Cert(A)) and a^y mod p are included, where a^y mod p is a challenge to B.

Step 4. Service center B computes the session key (K_{ab}); uses K_{ab} to decrypt the received encrypted message; uses A's public key to decrypt A's signa-

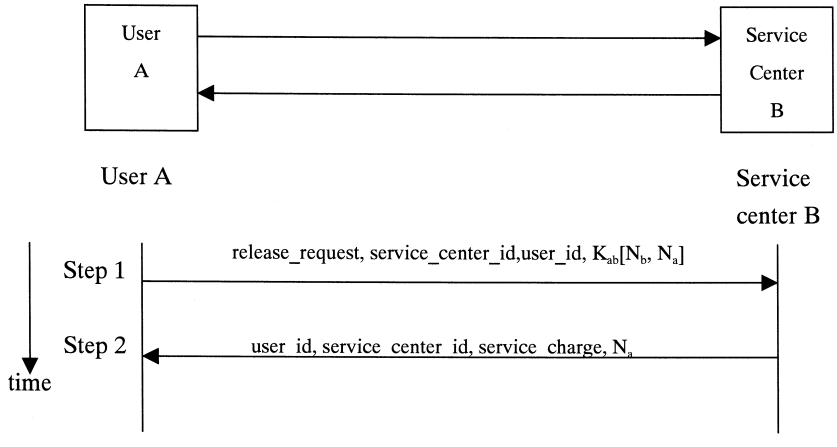


Fig. 3. Release phase of the PAP.

ture; signs a^y mod p, service center identifier, user identifier, and the requested service class, where a^y mod p is the response to A (B is authenticated); uses K_{ab} to encrypt the signature; sends the response to A with which nonce N_b is included, where N_b will be used in the release phase.

5.2.2.2. Data transfer phase. Data encryption/decryption are performed by symmetric cryptography using the session key (K_{ab}).

5.2.2.3. Release phase. The release phase includes two steps. Fig. 3 depicts the release phase of PAP.

Step 1. User A uses K_{ab} to encrypt nonces, N_a and N_b, where N_a is a challenge to service center B and N_b is the response to service center B (A is authenticated); sends the encrypted message, along with release request, service center identifier, and user identifier, to service center B.

Step 2. Service center B calculates service charge; sends service charge, service center identifier, user identifier, and nonce N_a, to user A, where N_a is the response to A (B is authenticated).

5.2.3. Comments

The differences between the PAP and the STS are highlighted as follows.

(1) In the second step of the connection phase, service center B's certificate (Cert(B)) is used to certify Diffie–Hellman parameters a and p. By doing this, B can ensure that user A will receive the correct a and p.

(2) In the third step of the connection phase, user A signs a, p, a^x mod p, and a^y mod p while the STS only signs a^x mod p and a^y mod p. By doing this, service center B is sure that A uses the same a and p as those of step 2 of the connection phase to generate a^y mod p.

(3) In the third and fourth steps of the connection phase, the service request message is encrypted along with nonce.

(4) In the fourth step of the connection phase, nonce N_b is used as a response in the release phase. By doing this, the number of message exchanges in the release phase is reduced to two, which is one less than the minimum number of message exchanges required by a challenge–response-based authentication protocol [3].

6. Cryptanalysis

We discuss the security of MAP and PAP with regards to their strength of preventing attacks.

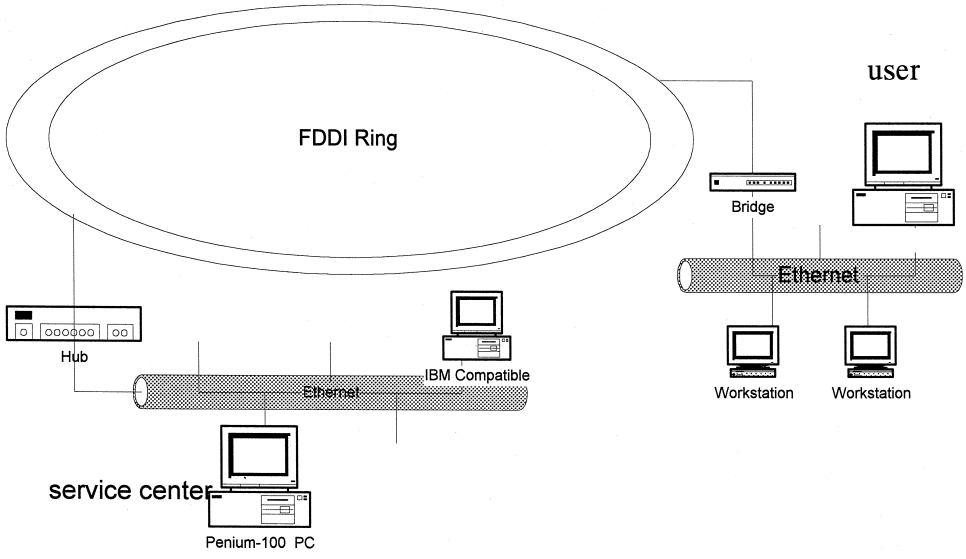


Fig. 4. Simulation environment.

Table 2
Connection time of telnet with/without MAP

	Telnet without MAP (a)	Telnet with MAP (b)	$b-a$	$(b-a)/a$ [%]
Time (ms)	294.23	326.44	32.21	10.95

6.1. MAP

6.1.1. Session key revelation

The session key ($K_{session}$) is used in the second and the third steps. In the second step, since we assume that service token (K_i^B) is stored in a trusted machine, attackers are not able to get it; therefore, it is impossible for an attacker to acquire the session key. In the third step, the session key is used to perform CBC encryption. According to the proof given in Ref. [2], CBC encryption is very secure.

6.1.2. Member attack

In the second step, if an attacker pretended as the requested service center, he/she would need service token (K_i^B). However, K_i^B is stored in a trusted machine.

6.1.3. Service center attack

In the third step, if an attacker pretended as the member, he/she would need the session key ($K_{session}$). However, $K_{session}$ is secure.

6.1.4. Other attacks

The replay attack is not possible since challenge–response is used in MAP. The guessing attack is not possible since the service request message is encrypted along with nonce. Furthermore, the MAP can resist oracle attacks due to CBC encryption.

6.2. PAP

6.2.1. Session key revelation

In the second step of the connection phase, since a and p are included in the certificate of the requested service center, a user can verify the correctness of a and p; thus, it is impossible for an attacker to modify a and p (note that the attack on the STS [9] is prevented). In the third step of the connection phase, since a user signs a, p, a^x mod p, and a^y mod p, the requested service center can prove that the received a^y mod p was not modified by an attacker. From the above discussions, we have shown that the session key (K_{ab}) is secure.

6.2.2. User attack

In the fourth step of the connection phase, if an attacker pretended as the requested service center, he/she could not sign a^y mod p without service center's private key. In the second step of the release phase, since an attacker cannot decrypt nonce N_a without the session key K_{ab}, he/she is not able to masquerade as the requested service center.

6.2.3. Service center attack

In the third step of the connection phase, if an attacker pretended as the user, he/she could not sign Diffie–Hellman key exchange parameters without user's private-key. In the first step of the release phase, since an attacker cannot encrypt nonces N_b and N_a without the session key K_{ab}, he/she is not able to pretend as a legitimate user.

Table 3
File transfer time of PAP vs. that of the X.509 three-way exchange

	Connection phase (a)	Data transfer phase (b)	Release phase (c)	Total latency ($a+b+c$)
Time using X.509 three-way exchange (ms)	9588.33	26,857.56	1744	38,189.89
Time using PAP (ms)	8134.33	27,475.33	995.33	36,604.99

6.2.4. Other attacks

The replay attack is impossible since challenge–response is used in PAP; the guessing attack is not possible since the service request message is encrypted along with nonce.

7. Simulation analysis

We evaluate the efficiency of MAP and PAP via simulation. Both client and server programs are coded in the C language, and are running on two Intel Pentium-100 MHz PCs with 32 MB RAM each. Fig. 4 depicts the simulation environment.

7.1. The MAP

Since the telnet of the Internet is an interactive application, its authentication can be best described by membership authentication; thus, the telnet is simulated to evaluate the efficiency of MAP. In this simulation, we assume that service tokens are pre-assigned (No extra time is needed for the derivation of service tokens.).

We further assume that there are no real data transferred (only the connection request is evaluated). The 56-bit DES is used for CBC encryption.

Table 2 presents simulation results. By examining Table 2, we notice that the overhead introduced by MAP is only 10.95%.

7.2. The PAP

The authentication of the FTP, a transaction-oriented application of the Internet, can be modeled by personal authentication. Hence, we choose the FTP to evaluate the efficiency of PAP. For comparison, we also simulate the X.509 three-way exchange. In this simulation, we assume that certificates are per-assigned (CAs are not needed). We further assume that files are transmitted in plaintext (only the connection and release phases are evaluated). The RSA using 1024-bit public key is assumed. An image file of size 1123 kbytes is transmitted.

Table 3 shows simulation results. From simulation results, we notice that the PAP is more efficient than the X.509 three-way exchange. Comparing to the original FTP (the FTP without authentication mechanism), both protocols cause large overheads due to public-key cryptography. However, for large size files, e.g., video files, the overheads are tolerable.

8. Conclusions

8.1. Summary

In this paper, we recommend a billing scheme for designing the B-ISDN billing system. This scheme provides both the flat rate and the usage-based rate billing policies. Both policies support priority pricing. Two authentication protocols, the MAP and the PAP, are proposed to satisfy the authentication requirements of the flat rate and usage-based rate billing policies, respectively. Both protocols are designed in conjunction with a suggested key management method. From cryptanalysis and simulation analysis, we notice that these two protocols together provide a good authentication mechanism to the B-ISDN billing system.

8.2. Future works

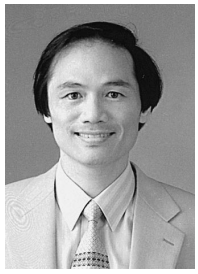
Here, we would like to mention the following areas of investigation which may merit further study:

- A more sophisticate key management framework is needed to manage those keys produced by the proposed key management method;
- New algorithms are needed to simplify the derivation of service token, since the derivation process is computationally intensive; and
- Certificate management should be integrated into the public key infrastructure (PKI).

References

[1] R. Anderson, R. Needham, Robustness principles for public key protocols, Research notes.
[2] R. Bird et al., Systematic design of two-party authentication protocols, Advances in Cryptology — CRYPTO '91, 1991, pp. 44–61.
[3] R. Bird et al., Systematic design of a family if attack-resistant authentication protocols, IEEE Journal on Selected Areas in Communications 11 (5) (1993) .
[4] G.C. Chick, S.E. Tavares, Flexible access control with mas-

ter keys, in: Proc. CRYPTO '89, Santa Barbara, CA, August 1989.
[5] R. Cocchi et al., A study of priority pricing in multiple service class networks, Proceedings of ACM SIGCOMM 1991 Conference, pp. 123–130.
[6] W. Diffie, Authentication and authenticated key exchanges, Designs, Codes and Cryptography 2 (1992) 107–125.
[7] W. Diffie, M. Hellman, New directions in cryptography, IEEE Transactions on Information Theory IT 22 (6) (1976) 644–654.
[8] L. Harn, H.Y. Lin, Management for decentralized computer network services, IEEE Transactions on Communications 41 (12) (1993) 43–46.
[9] Y.M. Lin, A study authentication and key distribution protocols, Master's Degree Thesis, National Chiao-Tung University, Institution of Computer and Information Science, 1996.
[10] C. Paris, D. Ferrari, A resource-based pricing policy for realtime channels in a packet-switching network, Technical report, International Computer Science Institute, Berkeley, CA, 1992.
[11] M.P. Vecchi, Broadband networks and services: architecture and control, IEEE Communications Magazine (August 1995) 24–32.
[12] Y. Zheng, J. Seberry, Immunizing public key cryptosystems against chosen ciphertext attacks, IEEE Journal on Selected Areas in Communications 11 (5) (1993) 715–723.
[13] RFC 1633, Integrated services in the internet architecture: an overview.
[14] RFC1272, Internet accounting: background.
[15] Entity authentication using symmetric techniques, ISO-IEC Jtc1.27.02.2(20.03.1.2), June 1990.
[16] ITU Recommendation X.509(1993E)|ISO/IEC 9594-8, Information technology–open systems interconnection — the directory: authentication framework, ITU-T SG7|ISO/IEC JTC1/SC21/WG4, 1993.



Chi-Chun Lo was born in Taipei, Taiwan, Republic of China, on August 22, 1951. He received the BS degree in Mathematics from the National Central University, Taiwan in 1974, the MS degree in Computer Science from the Memphis State University, Memphis, TN, in 1978, and the PhD degree in Computer Science from the Polytechnic University, Brooklyn, NY, in 1987. From 1981 to 1986, he was employed by Bell Laboratories, Holmdel, NJ, as a Member of Technical Staff. From 1986 to 1990, he worked for the Bell Communications Research as a Member of Technical Staff. Since 1990, he has been with the Institute of Information Management, National Chaio-Tung University, Taiwan, and is now an associate professor. He was the director of the institute from 1994 to 1996. His major current research interests include network design algorithm, network management, network security, network architecture, and multimedia system.

Yi-Chun Yeh was born in Taipei, Taiwan, Republic of China, on August 9, 1972. She received the BS degree in Information Management from the National Taiwan University, Taiwan in 1995, the MS degree in Information Management from the National Chaio-Tung University, Taiwan in 1997. Currently, she is working for American Express, Taipei, Taiwan, as a software engineer.