Authentication and Key Distribution Protocols for Wired and Wireless Systems

N. Komninos, M. I. Samarakoon, B. Honary

Department of Communication Systems
Lancaster University
Lancaster LA1 4YR

ABSTRACT

With the emergence of E-Commerce communication security has become a very important issue. Two main considerations of secure communication systems are authentication, and key distribution. Authentication and key distribution may differ from one system to another due to the system parameters such as bandwidth, and available processing power at the end terminals. This paper focuses on end-to-end authentication and key management strategies in wireless and wired systems. Public and secret key encryption techniques are used to provide authentication and key distribution.

INTRODUCTION

As the use electronic communications play an ever-increasing role in business activities, security in communications through networking environments is becoming very important. It is very important to preserve the confidentiality, data integrity, and authenticity of sensitive information to prevent fraud. In cryptography all modern algorithms use a **key** to control encryption and decryption; a message can be decrypted only if the key matches the encryption key. The key used for decryption can be different from the encryption key, but for most algorithms they are the same.

There are two classes of key-based algorithms, **symmetric** (or **secret-key**) and **asymmetric** (or **public-key**) algorithms. The difference is that symmetric algorithms use the same key for encryption and decryption (or the decryption key is easily derived from the encryption key), whereas asymmetric algorithms use a different key for encryption and decryption, and the decryption key cannot be derived from the encryption key. The **public-key algorithms** permit the encryption key to be public, allowing anyone to encrypt with that key, whereas only the proper recipient (who knows the decryption key) can decrypt the message. The encryption key is called the **public key** and the decryption key the **private key** or **secret key** [1].

Public key methods require in complex computations compared to secret key methods. However, since the public key can be distributed through an insecure channel key management is easier in public key systems. Also, the public key schemes can be used to generate digital signatures for authentication. Therefore, the public key schemes are usually used for distribution of a session key, which will be used to encrypt general traffic (files, multimedia etc.). However, the complex computations are not suitable for systems with low processing power and the secret key schemes should be used for providing key distribution and authentication in real time.

In wired systems where computers posses high processing power, public key techniques can be used for key distribution. On the other hand, in wireless systems where mobile stations have low processing power secret key techniques should be used. In this paper we propose authentication and key distribution techniques for both wired and wireless systems.

AUTHENTICATION AND KEY DISTRIBUTION TECHNIQUES

The secret key schemes require the keys to be shared with a third party which is trusted by all users. The third party generates a session key for two users to communicate securely. The shared keys are used to distribute this session key to the respective users. Authentication is done through a challenge response scheme [2].

The public key schemes on the other hand uses the recipients public key to encrypt the session key. Authentication is done with the use of certificates. A certification authority (CA), which is assumed to be trusted by all users, is responsible for providing these certificates [3].

Certificate generation and verification processors are done using public key encryption techniques and one way hashing functions. For generating a certificate the user identity (i) and public key (Pi) are first hashed using one way

hashing function. Then the hashed value h(i,Pi) is encrypted using the private key S_{CA} of the Certification Authority (CA) to generate the certificate shown in Figure 1(a).

Since the public keys of the CA and the user (P_{CA} and Pi) can be published the validity of the certificate can be readily verified. For verification, first the certificate is decrypted using P_{CA} to obtain the hashed h(i,Pi). Using the user identity (i) and public key Pi the hashed value h'(i,Pi) can be locally calculated. The two-hashed values are then compared to authenticate the user as shown in Figure 1(b).

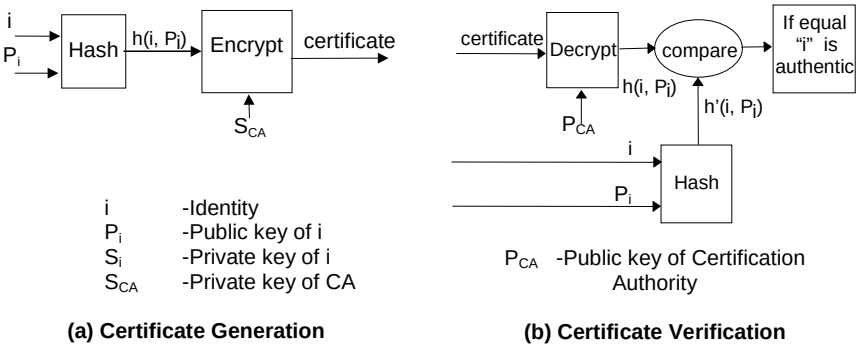

Figure 1 Certificate Generation and Verification Process

WIRELESS SYSTEMS

Managing keys through a control centre (CC) is considered for wireless systems. In this method a CC is assumed to be 'trusted' by all MS's. Each MS shares a secret key and a secret identification code with the CC. Therefore, the CC has to maintain a database of shared secret keys and shared secret identification codes of all the mobile stations supported by it.

Consider two mobile stations MS A and MS B that belong to two different groups. The MS A shares its secret key Ka and identification code C_a with the CC. Similarly, MS B shares its secret key Kb and identification code C_b with the CC. Time stamps are included in some messages to give them a limited lifetime of transmission to prevent replay attacks. Sensitive information such as identification codes, session keys, and time stamps are encrypted using session keys, which are derived by hashing the shared secret keys (Ka or Kb) with random numbers as shown in Figure 2.

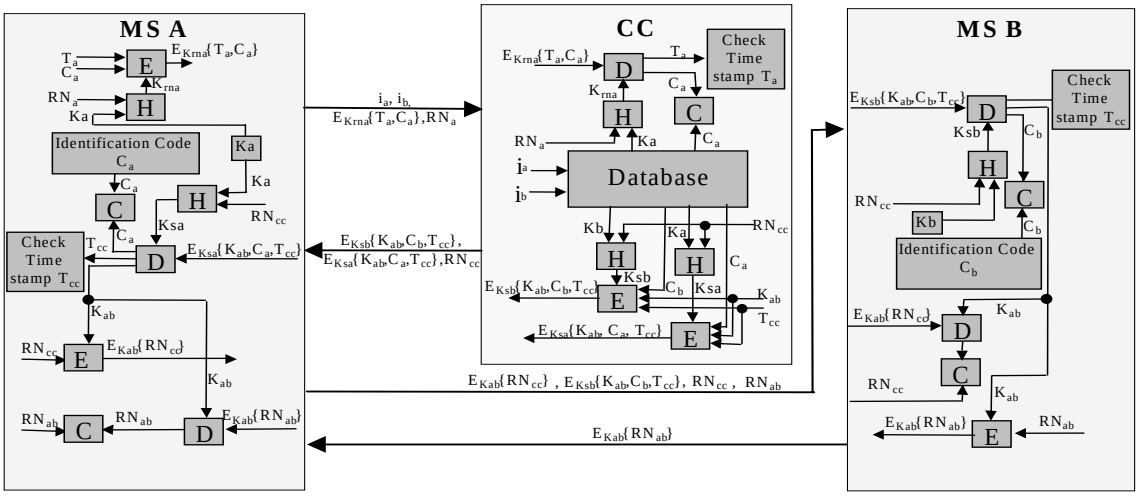

Figure 2: Key management through a Control Centre

When MS A wishes to communicate with MS B, it derives a session key Krna by hashing Ka with a random number RN_a and encrypts C_a and a time stamp T_a using the derived session key Krna. Then MS A sends its identity i_a, B's identity i_b, RN_a, and the encrypted message $E_{Krna}\{T_a,C_a\}$ to the CC. The CC looks i_a and i_b up in its database and obtains Ka, C_a, Kb, and C_b. Then the CC derives the session key using RN_a and Ka, and decrypts $E_{Krna}\{T_a,C_a\}$ to obtain T_a and C_a. The CC checks the validity of T_a and authenticates MS A by comparing the received C_a with the stored C_a. If MS A is authentic the CC generates a random number RN_{cc} and hashes it with Ka and Kb to derive two

session keys Ksa and Ksb to encrypt messages for MS A and MS B. The CC also generates another random session key K_{ab}, which will be used later as the session key for encrypting and decrypting traffic between MS A and MS B. Then the CC encrypts K_{ab}, C_a, and a time stamp T_{cc} using K_{sa} to obtain $E_{Ksa}\{K_{ab},C_a,T_{cc}\}$. The CC also encrypts K_{ab}, C_b, and T_{cc} using Ksb to obtain $E_{Ksb}\{K_{ab},C_b,T_{cc}\}$. The CC sends these two encrypted messages and RN_{cc} back to MS A.

MS A derives the session key Ksa using received RN_{cc} and its Ka and decrypts $E_{Ksa}\{K_{ab}.C_a,T_{cc}\}$ to obtain K_{ab},C_a, and T_{cc}. Then it checks the validity of T_{cc} and authenticates the CC by comparing the received C_a with the stored C_a. If CC is authentic MS A generates a random number RN_{ab} and encrypts the received RN_{cc} using Kab to obtain $E_{Kab}\{RN_{cc}\}$. Then it forwards $E_{Kab}\{RN_{cc}\}$, RN_{ab} and $E_{Ksb}\{K_{ab},C_b,T_{cc}\}$, and RN_{cc} to the MS B.

When MS B receives the message from MS A, it derives K_{ab} by hashing the received RNcc and its shared key Kb. Then it decrypts $E_{Ksb}\{K_{ab},C_b,T_{cc}\}$ and checks the validity of the time stamp and authenticates the CC by comparing the received C_b with the stored C_b. Then it decrypts $E_{Kab}\{RN_{cc}\}$, and compares it with the received RN_{cc}. The received RN_{cc} and the decrypted RN_{cc} will be equal only if MS A has the correct session key Kab. Therefore, MS B can authenticate MS A by the comparison. If MS A is authentic MS B encrypts the received RN_{ab} using the derived session key K_{ab} and sends this encrypted message $E_{Kab}\{RN_{ab}\}$ back to MS A. MS A decrypts the received encrypted message $E_{Kab}\{RN_{ab}\}$ to obtain RN_{ab} and compares it with the previously generated RN_{ab}. If they are equal MS B is authentic. Then MS A and MS B use K_{ab} to secure traffic between them.

It is important to note that, both MS A and MS B authenticate the CC. Additionally, MS A and MS B mutually authenticate each other during the process of key distribution. This is very important to prevent security breaches of the system. This is the main reason for generating the session key K_{ab} in the CC rather than by the MS's.

An important consideration in this technique is to keep the number of call set-ups to a minimum to reduce the overall key distribution time. In this technique only two call set-ups are required. This system is useful for mobile stations with low processing power, which cannot carryout complex public-key computations in real time. For mobiles with high processing power, public key techniques can be used for key distribution.

WIRED SYSTEMS

Availability of high bandwidth and high processing power of the computing devices allow public key techniques to be used for authentication and key distribution in wired systems.

In public key protocols two parties which require secure communication exchanges the certificates obtained from a trusted certification authority (CA) to authenticate each other. Public key schemes are also used to distribute session keys. Here, both communicating parties involve in the generation of the session key. More specifically, each party generates a random session key and securely distributes it to the other end. Then the two keys are combined to obtain the session key that is used to secure traffic between the two parties. Involvement of both parties in the generation of the session key avoids replay attacks by a third party. In addition, this process enables one party to check whether other party posses the private key corresponding to the public key it receives from the other party.

The proposed protocol procedure is shown in Figure 3. When station-A wants to establish a secure communication with the station-B, it sends its certificate C_A, identity ID_A, and public key P_A to station-B. Station-B verifies this certificate and if it is valid, sends its certificate C_B, identity ID_B, and public key P_B to station-A. Station-A then verifies the certificate of station-B.

Even though both certificates may be valid, still there is an uncertainty about whether the two parties possess the proper private keys. This is because the transmitted information can be recorded by a third party and can be used for authentication. In this case, the certificate will appear as a valid one to a recipient even though it is not sent by the legitimate user. Therefore, further checks are carried out to authenticate the communicating parties, station-A and station-B.

Station-A generates locally a random number x and hashes it using a one-way hashing function. Then the hashed value is encrypted using the private key S_A of station-A to create a signature $S[H(x)]$. The random number x and the signature $S[H(x)]$ are then encrypted using the public key P_B of station-B and this encrypted message $E[S[H(x)], x]$ is sent to station-B. When station-B receives this message, it obtains x by decrypting it and verifies the signature as shown in figure 3.

Similarly, station-B generates a random number y and a signature. Then it encrypts the signature $S[H(y)]$, the numbers y, and x using the public key P_A of station-A and sends this message $E[S[H(y)], y, x]$ to station-A. When station-A

receives this message it obtains *x*, and *y* by decrypting it and verifies the signature *S[H(y)]*. Then it compares the received *x* with the previously generated *x*, to check whether station-B has the proper private key. If the two *x*'s are the same, station B is considered to be authentic. Once station-B is authenticated, station-A encrypts the number *y* using the public key P_B of station-B and sends it to station-B. Station-B compares the received *y* with the previously generated *y* to authenticate station A.

After both stations are authenticated the two parties hash the numbers *x* and *y* to create a session key to encrypt traffic between them.

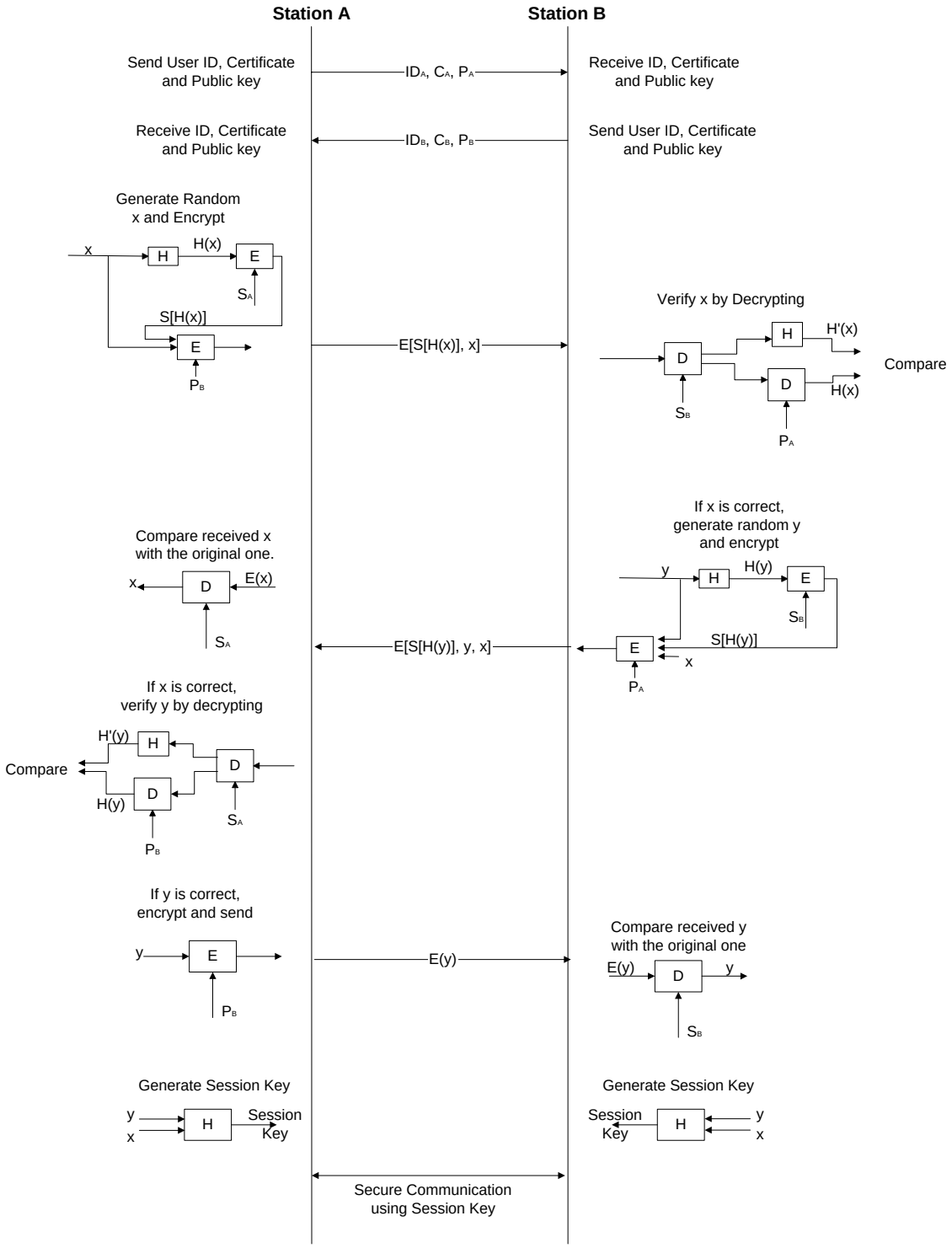

Figure 3 Authentication and Key distribution

CONCLUSIONS

In this paper authentication and key distribution strategies for wireless and wired systems were presented. Two protocols were proposed in this paper. Both protocols have been designed to suite the processing power available in the end devices (i.e. mobile and computing devices).

For wireless systems secret key schemes were used because the mobile stations do not possess enough processing power to perform the complex public key calculations in real time. Therefore, the proposed protocol uses a control centre to generate session keys for two users who require secure communication. The control centre requires a database to store all the shared secret information.

The availability of high processing power in the wired systems allows the public key schemes to be used for authentication and key distribution. The proposed protocol for wired systems performs additional operations to verify whether the parties involved possess the proper private keys. Also, both parties involve in the session key generation process which enhances the security of the system. In addition, the authentication protocol for wired systems overcomes the drawbacks of shared secret key protocols (i.e. reflection attack).

REFERENCES

[1] B.Schneier, “APPLIED CRYPTOGRAPHY ”, Published by John Wiley & Sons Inc., 1996.

[2] M. I. Samarakoon, B. Honary, “Novel authentication and key agreement protocol for low power and system resource requirements in portable communications systems”, IEE Colloquium on Novel DSP Algorithms and Architectures for Radio Systems, September 1999.

[3] D. Brown, “Techniques for Privacy and Authentication in Personal Communication Systems”, IEEE Personal Communications, August 1995.