Key Distribution Protocol for Digital Mobile Communication Systems

Makoto Tatebayashi [1]
Natsume Matsuzaki
Matsushita Electric Industrial Co. Ltd.
Moriguchi, 570, JAPAN

and

David B. Newman, Jr.
The George Washington University
Washington, DC 20052

ABSTRACT. *A key distribution protocol is proposed for digital mobile communication systems. The protocol can be used with a star-type network. User terminals have a constraint of being hardware-limited.*

Security of the protocol is discussed. A countermeasure is proposed to cope with a possible active attack by a conspiracy of two opponents.

1 Introduction

Proposed digital mobile communication systems potentially offer means for communications security using encryption techniques. For a secure secret key cryptosystem, a key should be changed for each session and shared by both terminals of a communication link. Thus, we have to solve the problem of key distribution.

Mobile communication systems may be regarded as star-type networks. Each user terminal in the network communicates with another user via a network center. Restrictions on hardware and implementation cost of a secure mobile communication system are more strict in user terminals than in a network center.

[1] Visiting Research Scholar at The George Washington University, 1988–1990

G. Brassard (Ed.): Advances in Cryptology - CRYPTO '89, LNCS 435, pp. 324-334, 1990.

In this paper we propose a key distribution protocol suitable for digital mobile communication systems. A public key cryptosystem is employed for uplink channels (from a user terminal to a network center); it makes the mobile communication systems free from key management problems. A secret key cryptosystem is employed for downlink channels (from a network center to user terminals); It enables high speed performance at hardware-limited terminals. The security of the key distribution protocol is discussed.

The protocol is shown not to degenerate the level of security of the cryptoalgorithm employed, if an opponent makes a passive attack. The protocol may be unsafe, however, by a conspiracy of two opponents and their active attack. We propose a countermeasure in the protocol to cope with this attack.

2 Previous Key Distribution Schemes

In this section we review previous key distribution schemes and their problems when they are applied to mobile communication systems.

2.1 Centralized Key Distribution Protocol

This protocol [DEN83] assumes that a network has a centralized key distribution facility which distributes a session key to the requesting terminals. The session key is encrypted by the terminal's encryption key.

If a classical key cryptographic method is employed for the key-encryption, then the central facility should manage each user's private key.

If a public key cryptographic method is employed for the key-encryption, then the management problem is reduced. Decryption at a hardware-limited user terminal may, however, take an impractically long time.

2.2 Public Key Distribution Protocol

The public key distribution protocol, invented by Diffie and Hellman [DH76], enables direct key distribution between two user terminals in a system and eliminates the key management problem at a network center. This protocol requires computation in a finite field. For the scheme to be secure, the order of the finite field should be very large, making realization of this scheme impractical without using special hardware or high-speed digital signal processors (DSP's).

3 Proposed Key Distribution Protocol

The objectives of our key distribution protocol for a mobile communication system are:

(1) to remove the key management at a network center, and

(2) to enable hardware-limited user terminals to obtain a common secret key in a reasonable time.

When a first user at a first terminal desires to share a common key or secret message with a second user at a second terminal, the first user generates a random number r_1 as a first key-encryption-key. The first key-encryption-key signal is passed to the network center using a public key scheme. Using a public key scheme in this uplink enables each user terminal to keep only a public key of the network center. This type of scheme also allows hardware-limited users to perform the encryption in a reasonable time since a public-key-encryption scheme can be employed in first and second user terminal which requires only a small computation.

The network center, upon receiving the first key-encryption-key signal, generates a request signal and transmits the request signal to the second terminal. In response to receiving the request signal, the second terminal generates a second key-encryption-key signal r_2. The second key-encryption-key r_2 will become a common key or message between the first terminal and the second terminal. The second terminal encrypts the second key-encryption-key signal using a public key scheme. This encrypted signal is then passed to the network center over the communications channel.

The network center, in response to receiving the first ciphertext signal and the second ciphertext signal, decodes these as the first key-encryption-key signal and the second key-encryption-key, respectively, using a public-key-decoding device. Thus the network center has the first and second key-encryption-key signals r_1 and r_2. The network center then can encrypt the second key-encryption-key signal r_2 with the first key-encryption-key signal r_1 using the classical-key-encoding device, employing any type of classical encryption device. At this point, public key encryption concepts are not required.

The following are examples of what might be used in the public key schemes. For example, consider the RSA cryptographic method used as the public key scheme. The modulus n is a product of p and q, where p and q are prime numbers. The encryption exponent e is chosen to be 3. The decryption exponent d is a number satisfying $ed = 1$ (modulo L) where L is the least common multiplier $p-1$ and $q-1$.

Additionally, consider a simple substitution cipher which may be used as a classical key encryption scheme. An example of a simple substitution cipher is the Vernam cipher. The encryption and decryption transformations are as follows:

Encryption: $E(x,k) = x \oplus k$
Decryption: $D(x,k) = x \oplus k$
Where "$\oplus$" denotes addition modulo 2 for each bit.

A second example of a simple substitution cipher is based on addition modulo n:

Encryption: $E(x,k) = x + k$ (modulo n).
Decryption: $D(x,k) = x - k$ (modulo n).
Where x and k are any element in the modulo ring.

The protocol for the key distribution, as illustratively shown in Figure 1, can be summarized as follows:

KEY DISTRIBUTION PROTOCOL 1 (KDP1)

1. First terminal, A, generates r_1 as a key-encryption key.
2. A encrypts r_1 with S's public key ($e = 3$) and sends ${r_1}^e \pmod n$ to S.
3. S decrypts ${r_1}^e \pmod n$ by its secret key d and gets $({r_1}^e \pmod n)^d \pmod n = r_1$.
4. S calls B.
5. B generates r_2 as a session key between A and B.
6. B encrypts r_2 with S's public key ($e = 3$) and sends ${r_2}^e \pmod n$ to S.
7. S decrypts ${r_2}^e \pmod n$ by its secret key d and gets $({r_2}^e \pmod n)^d \pmod n = r_2$.
8. S encrypts r_2 by a key-encryption key r_1 and sends $E(r_2, r_1)$ to A.
9. A decrypts $E(r_2, r_1)$ by its key-encryption key r_1 and gets $D(E(r_2, r_1), r_1) = r_2$ as a session key with B.

Choosing an RSA exponent e of 3 and the Vernam cipher enables the first terminal to be easily implemented.

The following discusses the security of the proposed method with an exponent $e = 3$ and modulo n classical encryption.

One might question whether revealing $r_1 + r_2 \pmod{n}$, as well as ${r_1}^3 \pmod{n}$ and ${r_2}^3 \pmod{n}$, degrades the security of the method. Under the assumption that an opponent has knowledge of only those parameters, we can show that the security of the method is not degraded as follows.

A cryptoanalyst, by knowing the transmitted ciphertexts, obtains the following simultaneous congruencies:

$$ {r_1}^3 = a \pmod{n} \tag{1} $$

$$ {r_2}^3 = b \pmod{n} \tag{2} $$

$$ r_1 + r_2 = c \pmod{n} \tag{3} $$

Where a, b, and c are known constants. From these congruencies one can yield a quadratic congruence of $r_1(r_2)$ in modulo n,

$$ {r_1}^2 - c r_1 + (1/3c)(c^3 - a - b) = 0 \pmod{n} \tag{4} $$

if $\gcd(3c, n) = 1$ holds.

Rabin [RAB79] showed that solving the quadratic congruence (4), without the knowledge of the factors of n, is as difficult as factorizing $n = pq$. Since the security of the RSA cryptography depends on the difficulty of factorization of n, we can conclude that revealing $r_1 + r_2 \pmod{n}$ in this protocol does not degrade the security of the protocol.

In this discussion we assume that an opponent makes only a passive attack; the cryptoanalyst only uses the knowledge of transmitted ciphertext and does not participate in the protocol.

In the next section, we will discuss the case of an active attack to the key distribution protocol.

4 An Active Attack to the Key Distribution Protocol

It was pointed out by G. J. Simmons [SIM89] that the method provided as KDP1 has a vulnerability. When legitimate first and second terminals communicate with each other to generate a common key signal, a first opponent may conspire with a second opponent to obtain the common key. As discussed herein, the common key is the key-encryption-key signal shared by the first and second terminals. The break-in protocol requires the first opponent to conspire with the second opponent in advance.

The first and second opponents agree that when either receives a request from the network center to establish a session key for communication with the other, that they will use a jointly known key, R.

First, consider the following attack. The first opponent, listening the first terminal's communication with the network center, initiates the KDP1 method to have a session key with the second opponent, sending ${r_1}^3 \pmod n$, replaying the first terminal's message to the network center. The second opponent sends a prearranged number, R, to the first opponent. Then the first opponent can apparently obtain r_1 and thus the common key r_2. Thus, the method of KDP1 is vulnerable to a replay attack.

Second, even if the network center has a mechanism for protecting against a replay attack, the following break-in protocol enables the first opponent to obtain a common key r_2, avoiding the protect mechanism against a replay attack.

SIMMON'S BREAK-IN PROTOCOL AGAINST KDP1

1. The first opponent, C, chooses a random number r_3 and calculates ${r_3}^{-1} \pmod n$. He also calculates ${r_1}^3{r_3}^3 \pmod n$ and sends it with a request that the network center, S, set up a session key for him with the second opponent, D.

2. S decrypts ${r_1}^3{r_3}^3 \pmod n$ to obtain $r_1 r_3 \pmod n$.

3. S calls D.

4. D sends $R^3 \pmod n$ to S.

5. S decrypts $R^3 \pmod n$ to obtain R and computes $R + r_1 r_3 \pmod n$ which it sends to C.

6. C subtracts R and multiplies the result by ${r_3}^{-1}$ to recover r_1.

7. C observed $r_1 + r_2 \pmod n$, so he can subtract r_1 from it to recover r_2, that is, the session key being used by the first terminal, A, and the second terminal, B.

Since r_3 is unknown to the center, ${r_3}^3$ becomes a one-time key that the first opponent can use to conceal the fact that ${r_1}^3$ is involved in ${r_1}^3{r_3}^3$.

The source of weakness is the fact that ${r_1}^3{r_3}^3 = (r_1 r_3)^3 \pmod n$, i.e. that the RSA encryption commutes with modular multiplication.

5 A Countermeasure Against the Active Attack

5.1 A Structure in the Sending Data

A countermeasure against the attack may be obtained by adding structure with the first key-encryption-key signal.

In order to destroy the ability of the first and second opponents to make use of the multiplicative property of the RSA scheme, a certain predetermined structure should be provided in the data to be encrypted. One example of the structure in the data is restricting a random number, r, to be stored in the least significant 256 bit, keeping the significant 256 bit to be zero. The center should have a mechanism to check that the decrypted message is in the predetermined set, $\mathcal{M}$, of the message. If r_1 and r_2 are chosen randomly in $\mathcal{M}$, then the probability that $r_1 r_2 \pmod{n}$ is in $\mathcal{M}$ is negligibly small. Thus if an opponent sends ${r_1}^3 {r_2}^3 \pmod{n}$ to the network center, then the center can, with high probability, recognize that an illegal message is sent.

5.2 A Measure to Prevent a Replay Attack

Another measure to prevent a replay attack is generating a timestamp which can be generated at the first terminal, and concatenating it with the first key-encryption-key. The transmitting data from the first terminal to the center is now

$$(t_a \,||\, r_1)^3 \pmod{n}$$

where t_a denotes a timestamp, $||$ denotes a concatenation and r_1 denotes a random number.

The network center should have a mechanism for checking the timeliness of the timestamp. The timestamp may include a transmitted date and time and expiring date.

5.3 User Identity Verification

A mechanism for user identity verification should also be provided in the protocol, since the key distribution protocol KDP1 does not solve the user authentication problem. From our basic standpoint the identity verification should not require the center to manage secret information for each user.

We will describe a possible user verification scheme for the key distribution protocol.

The network center generates each user i's secret s_i from user i's identifier, IDi,

$$s_i = f(IDi)$$

where f is a polyrandom function which the center only knows. The network center distributes s_i to user i in secret, possibly in the form of a smart card.

The first terminal constructs a data signal, which is a concatenation of user i's secret, s_a, a random number, r_1, and other information. The first terminal encrypts the data signal and the network center decrypts the encrypted data signal and gets the user i's secret, s_a'. The network center calculates $f(IDa)$ and checks if it is the same as s_a' which he received. If they coincide, the network center verifies the sender. Otherwise the network center rejects the sender and quits the protocol.

Combining these three mechanisms the key distribution protocol, as illustratively shown in Figure 2, can be summarized as follows:

KEY DISTRIBUTION PROTOCOL 2 (KDP2)

1. The first terminal, A, generates r_1 as a key-encryption key.
2. A sends to the network center, S, IDa and $(t_a \| s_a \| r_1)^3 \pmod n$.
3. S decrypts the encrypted data signal and gets $(t_a \| s_a \| r_1)$. S extracts t_a, s_a and r_1 from the decrypted data. S checks the validity of the timestamp t_a. S verifies A.
4. S calls the second terminal, B.
5. B generates r_2 as a session key between A and B.
6. B sends to S, IDb, $(t_b \| s_b \| r_2)^3 \pmod n$.
7. S decrypts the encrypted data and gets $(t_b \| s_b \| r_2)$. S extracts t_b, s_b, r_2 from the decrypted data. S checks the validity of the timestamp t_b. S verifies B.
8. S sends A, $r_1 + r_2 \pmod n$.
9. A subtracts r_1 and gets r_2 as a session key with B.

In this protocol, a structure in the transmitted data prevents an enemy from utilizing the distribution property of the RSA cryptography. A timestamp mechanism prevents a replay attack. A mechanism for identity verification prevents masquerading.

In the protocol, the second terminal can obtain $(t_a \| s_a \| r_1)^3 \pmod n$ and r_1. The second terminal can easily guess a timestamp t_a. If the second terminal can get the first terminal's secret s_a, this cryptosystem is unsafe.

In general this problem is considered to find a plaintext, with a part of which being known to a cryptoanalyst. We do not know any successful attack at present. In order to avoid an exhaustive search for s_a, a field length for user secret should be long. We believe that 200 bits, for example, are sufficiently long.

This protocol may be exposed to a "low exponent protocol failure" [MOO88], since we restricted ourselves to a case where the RSA exponent is a small number. As long as we consider a case where only one network center exists, this protocol is safe. But if we extend our scheme to a case with multiple network centers, we have to be careful about the inherent weakness of the low exponent scheme.

Since the data transmitted from user terminals to a network center have a structure, mathematical analysis of the security is no more possible on the assumption that the opponent only has the information known by himself. But we believe that the previous results on KDP1 give us a lower bound on the security of KDP2 under the same assumption, since the opponent in KDP2 obtains less information than in KDP1.

6 Conclusions

In this paper we proposed a key distribution protocol for mobile communication systems. In the protocol a public key cryptography is employed in uplinks and a secret key cryptography is employed in downlinks. We focused our discussions on a special case using the RSA scheme with encryption key $e = 3$ for uplinks, and of simple substitution ciphers for downlinks. The protocol makes a network center free from key management problems and enables hardware-limited user terminals to operate in a reasonable time to get a common key.

We introduced a structure in the transmitted data and a mechanism checking a replay attack in order to avoid a protocol failure based on the multiplicative property of the RSA cryptography.

Acknowledgements

The authors thank Dr. G. J. Simmons for showing the active attack in Section 4. The authors thank Dr. J. K. Omura for very helpful discussions.

References

[DEN83] D. E. Denning, "Cryptography and Data Security", Addison–Wesley, 1983.

[DH76] W. Diffie and M. Hellman, "New Directions in Cryptography", IEEE Trans. on Info. Theory, Vol. IT–22, No. 6, pp. 644–654, 1976.

[JC85] W. de Jonge and D. Chaum, "Attacks on Some RSA Signatures", Advances in Cryptology: Proceedings of Crypto'85, Springer–Verlag, pp. 12–27, 1986.

[MOO88] J. H. Moore, "Protocol Failures in Cryptosystems", Proc. of IEEE, Vol. 76, No. 5, pp. 594–602.

[RAB79] M. O. Rabin, "Digitalized Signatures and Public-Key Functions as Intractable as Factorization", MIT/LCS/TR–212, MIT Lab for Computer Science, Cambridge, Massachusetts, January 1979.

[SIM87] G. J. Simmons, "An Impersonation-proof Identity Verification Scheme", Advances in Cryptology: Proceedings of Crypto'87, Springer–Verlag, pp. 211–215, 1988.

[SIM89] G. J. Simmons, private letter, January 1989.

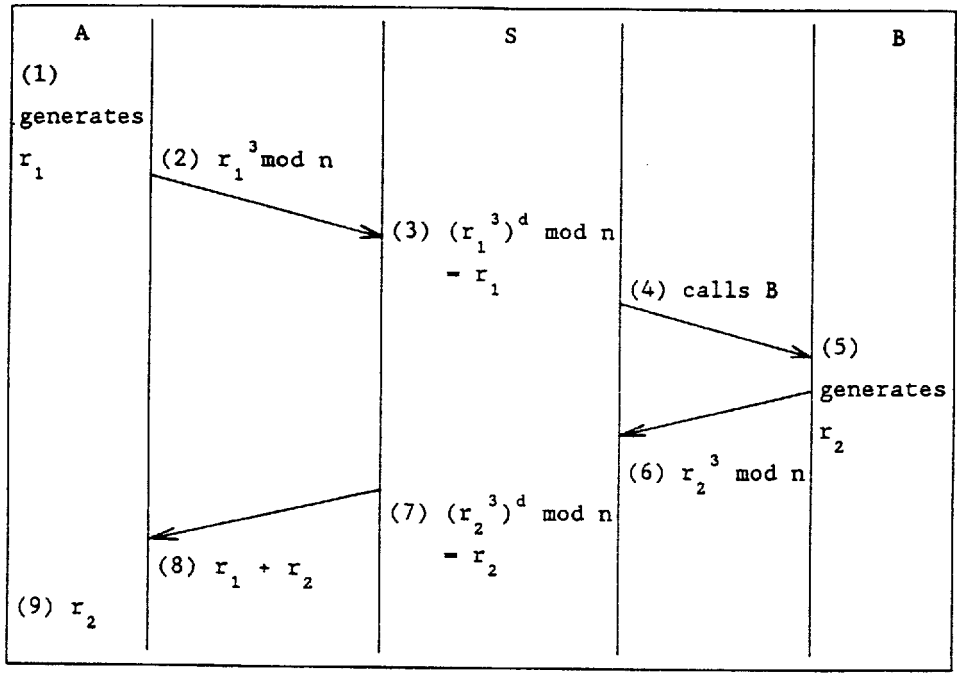

Figure 1. Key Distribution Protocol 1 (KDP1)

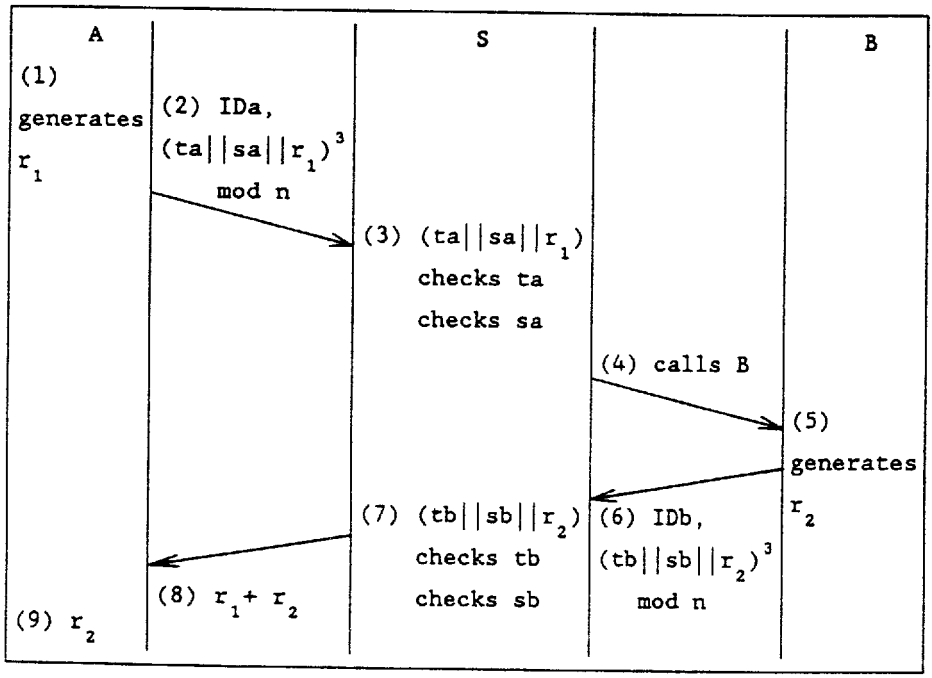

Figure 2. Key Distribution Protocol 2 (KDP2)