SECURE COMMUNICATION MECHANISMS FOR GSM NETWORKS

Chi-Chun Lo and Yu-Jen Chen
Institute of Information Management
National Chiao-Tung University
1001 Ta Hsueh Road
Hsinchu, Taiwan 300, ROC
Email: cclo@cc.nctu.edu.tw

ABSTRACT

With the advance of wireless communications technology, mobile communications has become more convenient than ever. However, because of the openness of wireless communications, how to protect the privacy between communicating parties is becoming a very important issue. In this paper, we focus on the security of the Global System for Mobile communication (GSM) networks. A secure communication architecture for the GSM network is proposed. In the proposed architecture, we use public-key cryptography for user authentication and stream cipher for message encryption and decryption. An authentication protocol and a key generation method are presented in conjunction with the proposed architecture. Cryptanalysis and operational analysis show that the authentication protocol is secure and efficient. Simulation results indicate that the key generation method can always produce key strings of evenly distributed 0's and 1's and with infinite period.

Keywords: GSM Networks, Wireless Communications, Mobile Communications, Block Cipher, Stream Cipher, Authentication

1. INTRODUCTION

Mobile communications has become more popular and easier for the past few years. Nowadays, people can communicate with each other on any place at any time. However, the openness of wireless communications poses serious security threats to communicating parties. How to provide secure communication channels is essential to the success of a mobile communication network. The Global System for Mobile communications (GSM) is the standard for digital mobile communications. It has a comprehensive set of security features [6]. However, these features have several limitations. In this paper, we propose a secure communication architecture for GSM networks. In the proposed architecture, public-key cryptography and stream cipher are used for user authentication and message encryption/ decryption, respectively. An authentication protocol and a key generation method are presented in conjunction with the proposed architecture. Cryptanalysis and operational analysis show that the authentication protocol is secure and efficient. Simulation results indicate that the key generation method can always produce key strings of evenly distributed 0's and 1's and with infinite period. In the following section, literature review is presented. Section 3 first describes the proposed architecture, and then details the authentication protocol and the key generation method. Analysis is given in Section 4. Section 5 concludes this paper with possible future research directions.

2. LITERATURE REVIEW

2.1 GSM [6][18]

The GSM is the first mobile communication system which has comprehensive security features. It is gaining tremendous supports from the telecommunication industry. The security architecture of GSM is intended to prevent unauthorized network access, disallow subscriber impersonation, protect confidentiality, and provide privacy. GSM's security services include anonymity, authentication, signaling protection, and user data protection.

Fig. 1 depicts the security architecture of GSM. It is composed of three tiers. The first tier; namely, the A3 algorithm, uses the challenge-response method [1] for user authentication. The second tier; namely, the A8 algorithm, uses the output from the A3 algorithm to generate the key string for the A5 algorithm. The last tier; namely, the A5 algorithm, uses a proprietary algorithm for message encryption/decryption. This architecture has the following problems: challenge-response is a simple authentication method, but is not very secure [4]; the length of the key generated by the A8 algorithm is fixed (114 bits), hence this key might be disclosed by the brute-force attack [1]; the A3, A8 and A5 algorithms are proprietary, thus their security can not be easily verified; and the A5/1 algorithm is subject to export control.

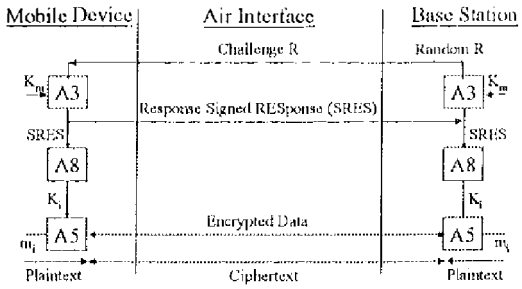

Fig. 1. Security Architecture of GSM

2.2 Stream Cipher [20]

Symmetric cryptosystems are further classified into

Contributed Paper
Manuscript received June 4, 1999

block cipher and stream cipher. Block cipher divides the plaintext into blocks and encrypts each block independently. On the other hand, stream cipher encrypts the plaintext on a bit-by-bit (or byte-by-byte) basis. In essence, GSM networks carry mostly voice, which is one type of continuous data. Since stream cipher is based on the eXclusive OR (XOR) operation, to encrypt/decrypt voice data bit-by-bit (or byte-by-byte) using stream cipher is very efficient. In general, stream cipher is much simpler and faster than block cipher [1][20]. Therefore, stream cipher is a good choice for message encryption/decryption for GSM networks.

The major problem of stream cipher cryptography is the difficulty of generating a long unpredictable bit pattern (keystream). In the one-time pad of stream cipher, a keystream is a sequence of randomly generated bits, and the probability of one bit to be 1, independent of other bits, is equal to one half. An ideal keystream in one-time pad is purely random and has infinite length. The keystream can not be generated by the receiving end, and can not be distributed to the receiving end either. The pseudorandom bit generator [20] has been widely used to construct the keystream. It generates a fixed-length pseudorandom noise as the keystream. How to increase the length of the keystream while still maintaining its randomness is important to the security of stream cipher.

2.3 The X.509 Three-way Exchange [21]

The X.509 three-way exchange, described in ITU-T Recommendations X.509, is a novel example of a public-key based authentication protocol. Key management of this protocol is securer than those of the protocols using symmetric cryptography. However, the X.509 three-way exchange has some noticeable problems. Since the X.509 three way exchange performs encryption before signing, the attacker may remove the signature from the encrypted message and replace it with his own [2]. 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.

2.4 Security Attacks

Security mechanisms are subject to different attacks. Following is a discussion on some most noticeable attacks.

2.4.1 Replay Attack [1]

An attacker can use this attack to capture legitimate messages and retransmit them for illegal purpose. To defeat such an attack, non-repeated random numbers are often used to ensure that all replayed messages will be detected. Time stamp, sequence number, and challenge-response are three different types of nonces. Each has its own limitations. As for time stamp, it is very difficult and expensive to maintain clock-synchronization among all parties at all time; as for sequence number, it requires that each party has to maintain state information of other parties, which is impractical in an open system [8]; as for challenge-response, its security depends on the randomness of nonce, and it needs an extra message exchange to complete mutual authentication.

2.4.2 Guessing Attack [1]

Authentication using password is widely used by many security systems. However, password is vulnerable under the dictionary attack by which an attacker can guess the password successfully. Public-key cryptography provides a means for preventing the guessing attack.

2.4.3 Interleaving Attack [5]

If an attacker could collect information from different executions of a security protocol, he/she might be able to break the protocol. This is the so-called interleaving attack. Based on the paper given by Ray Bird *et. al* [5], interleaving attacks include the known plaintext attack [5], the chosen ciphertext attack [5], the oracle session attack [4] and the parallel session attack [4]. The cipher block chaining (CBC) encryption and public-key cryptography are often used to prevent this attack [5].

2.4.4 Man-in-the-Middle Attack [1]

An attacker can use the man-in-the-middle attack to intervene between two communicating parties and masquerade as one to communicate with another bidirectionally. Public-key cryptosystem using certificate often provides a solution for preventing this attack.

3. SECURE COMMUNICATION MECHANISMS

The following notation is defined.

Notation:

$Identity_X$	X's identity
$K_{priv,X}$	X's private key
$K_{pub,X}$	X's public key
$S_X\{...\}$	X's signature followed by data
$S'_X\{...\}$	X's signature
$E_{Kpub,X}()$	message encrypted by X's public key
Cert(X)	X's certificate; it contains X's identity, X's public key, and certificate authority's signature on X's identity and X's public key

3.1 The Proposed Architecture

The proposed architecture, shown in Fig. 2, still follows the three-tier structure as described in GSM standards. In this architecture, the C3, C8 and C5 algorithms are suggested to replace the A3, A8 and A5 algorithms, respectively. The following sections detail the C3, C8 and C5 algorithms.

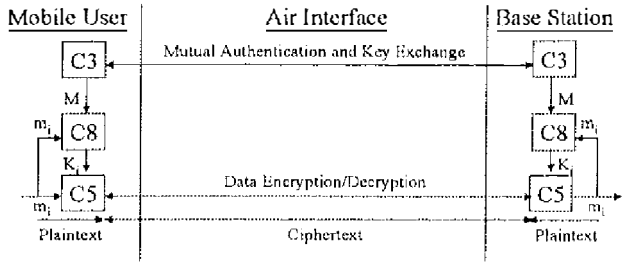

Fig. 2. The Proposed Architecture

3.2 Authentication Protocol (C3)

3.2.1 Design Rationale

Upon requesting service from a base station, a mobile user has to prove his/her identity to the base station. After completing the service, he/she is charged for the service he received. Authentication is required not only when a connection is being established but also when the connection is being released. Since a mobile user is charged afterwards, the base station 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. Although public-key cryptography is computationally intensive; however, it does meet the authentication requirements of GSM networks. According to the discussions aforementioned, the design of C3 refers to the X.509 three-way exchange. In this design, certificates are required.

3.2.2 Protocol Design

The C3 protocol has two phases: the connection phase and the release phase.

Connection Phase

The connection phase includes three steps. Fig. 3 depicts the connection phase of C3. For mobile user (MU) m, his/her identity ($Identity_m$), private key ($K_{priv,m}$), public key ($K_{pub,m}$) and certificate (Cert(m)) are issued by certificate authority (CA), and are saved inside the Subscriber Interface Module (SIM) of his/her mobile device. For base station (BS) s, its identity ($Identity_s$), private key ($K_{priv,s}$), public key ($K_{pub,s}$) and certificate (Cert(s)) are also issued by CA. Authentication is performed according to the following procedure:

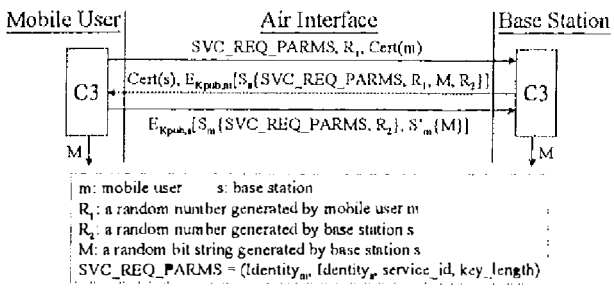

Fig. 3. Connection Phase of the C3 Protocol

Step 1: The MU requests service from the BS. Service request parameters (SVC_REQ_PARMS) include $Identity_m$, $Identity_s$, service_id (the requested service identifier), and key_length (the length of the initial keystream). SVC_REQ_PARMS, along with nonce R_1 and MU's certificate (Cert(m)), are sent to the BS, where R_1 is a challenge to the BS.

Step 2: The BS uses Cert(m) to verify MU's identity; generates a random bit string M of length "key_length" and a random number R_2, where R_2 is a challenge to the MU; signs SVC_REQ_PARMS, R_1, M, and R_2, where R_1 is the response to the MU (The BS is authenticated); uses $K_{pub,m}$ to encrypt the signed message; sends the encrypted message, along with its certificate (Cert(s)), to the MU.

Step 3: The MU uses Cert(s) to verify BS's identity; uses $K_{priv,m}$ to decrypt the encrypted message received; uses $K_{pub,s}$ to verify the signature of the BS; checks whether SVC_REQ_PARMS received is the same as the one he/she sent in Step 1; signs SVC_REQ_PARMS and R_2, where R_2 is the response to the BS (The MU is authenticated); signs M; uses $K_{pub,s}$ to encrypt the two signed messages; sends the encrypted message to the BS.

Note that the bit string M is a shared secret between the base station and the mobile user, and is used as the initial keystream of the C8 algorithm. The base station and the mobile user should negotiate the length of M (key_length) in advance. M's length can be very long; for example, 1 ~ 20 KBytes which is much longer than the normal key length; e.g., 56 bits, 128 bits, and 512 bits.

Release Phase

The release phase, shown in Fig. 4, includes two steps.

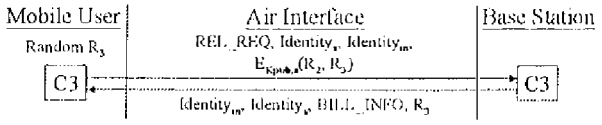

Fig. 4. Release Phase of the C3 Protocol

Step 1: The MU generates a random number R_3, where R_3 is a challenge to the BS; uses $K_{pub,s}$ to encrypt R_2 and R_3, where R_2 is the response to the BS (The MU is authenticated); sends the encrypted message, along with release request (REL_REQ), $Identity_s$, and $Identity_m$, to the BS.

Step 2: The BS uses $K_{priv,s}$ to decrypt the encrypted message received; sends R_3, along with $Identity_m$, $Identity_s$, and billing information (BILL_INFO), to the MU, where R_3 is the response to the MU (The BS is authenticated).

3.2.3 Comments

The security features of C3 are highlighted as follows:

- In the first and second steps of the connection phase, certificates, Cert(m) and Cert(s), are used to verify mobile user's and base station's identities, respectively.

- In the second and third steps of the connection phase, nonces, R1 and R2, are used to provide direct mutual authentication.
- In the first and second steps of the release phase, mutual authentication is provided via R_2 and R_3. Note that R_2 is reused in the release phase with which the number of message exchanges in the release phase is reduced by one.
- In the second step of the connection phase, the initial keystream (M) of the C8 algorithm, instead of a session key, is exchanged.
- In the second and third steps of the connection phase, signing is performed before encryption.
- In the second and third steps of the connection phase, the service request message is encrypted along with nonce. It is worthy of noticing that the size of the service request message is usually very small. Since the contents of the service request message are limited and easy to predict, an attacker can find probabilistic encryption by guessing enough strings. To foil such an attack, it is practical to encrypt the service request message along with an unpredictable random number (nonce) [5].
- In the third step of the connection phase, the service request message is signed by the mobile user; thus, non-repudiation is guaranteed.
- The C3 protocol can provide *perfect forward secrecy* since the initial keystream M is not repeated for every service request.

3.3 Key Generation (C8)

The C8 algorithm processes input data on a byte-by-byte basis. For an initial keystream of length k bytes, (M_0, M_1, ... , M_j, ... , M_{k-1}), and an input message of length n bytes, (m_0, m_1, ... , m_i, ... , m_{n-1}), the keystream of length n bytes, (K_0, K_1, ... , K_i, ... , K_{n-1}), is generated according to the following procedure:

Step 1: Let i = 0, j = 0, N = 170, cc = 0;

Step 2: $M_j = M_j + cc$;

If j = 0 then $M_j = M_j$ XOR N; else $M_j = M_j$ XOR M_{j-1};

Step 3: $K_i = M_j$; $N = (N + m_i) \bmod 256$;

i = i + 1; j = j + 1; cc = cc + 1;

If j = k then reset j to 0;

If cc = 256 then reset cc to 0;

If i = n, then exit; else goto Step 2.

where N, an eight-bit string, is used by the XOR operation to change M_0 at the beginning of each cycle of M; cc is a number added to M_j (j = 0 to k-1) so as to increase the randomness of M_j.

Note that the initial value of N can be any number between 0 and 255 ($2^8 - 1$). In C8, we use the number 170 (10101010_2) as the default value of N.

3.4 Message Encryption/Decryption (C5)

The C5 algorithm uses stream cipher for encryption/decryption. In our design, the simplest stream cipher using only the XOR operation [20] is chosen.

4. ANALYSIS

4.1 Analysis of C3

4.1.1 Cryptanalysis

- Keystream (M) revelation

  In the second step of the connection phase, since the initial keystream M is signed by the BS, it is impossible for an attacker to obtain M. In the third step of the connection phase, M is signed by the MU; thus, it is free from revelation.
- Interleaving attack

  (1) mobile user attack

  In the second step of the connection phase, if an attacker pretended as the base station, he/she could not sign R_1 without base station's private key. In the second step of the release phase, since an attacker can not decrypt nonce R_3 without base station's private key, he/she is not able to masquerade as the base station.

  (2) base station attack

  In the first step of the connection phase, if an attacker pretended as the mobile user, he/she could not modify CA's signature on mobile user's identity In the third step of the connection phase, since an attacker can not sign R_2 without mobile user's private key, he/she is not able to pretend as the mobile user. In the first step of the release phase, an attacker can not obtain nonce R_2; therefore, it is impossible for him/her to pretend as the mobile user.
- Replay attack

  The replay attack is not possible since the challenge-response mechanism is used.
- Guessing attack

  The guessing attack is not possible since service request parameters (SVC_REQ_PARMS) are encrypted along with nonce.
- Man-in-the-middle attack

  Both the mobile user and the base station can verify each other's identity by using each other's certificate. Also, since signing is performed before encryption in the C3 protocol, an attacker can not remove the signature from the encrypted message and replace it with its own. Therefore, the man-in-the-middle attack is prevented.

4.1.2 Operational Analysis

Number of message exchanges

Nonce R_2 is used both in the connection phase and the release phase. By doing this, the number of message exchanges in the release phase is reduced to 2, which is one less than the minimum number of message exchanges required for a challenge-response based authentication protocol [9]. Consequently, the total number of message exchanges of the C3 protocol is 5 instead of 6.

Cryptographic operation

Cryptographic operation includes

encryption/decryption and signing/verification. For comparison, the X.509 three-way exchange is also considered. Table 1 presents the comparison.

Table 1. Comparison between the X.509 three-way exchange and the C3 protocol

		Public-key cryptosystem		Signature	
		Encryption	Decryption	Signing	Verification
X.509 three-way exchange	Mobile user	1	2	1	2
	Base station	2	1	2	1
C3	Mobile user	2	1	1	1
	Base station	1	2	1	1

For mobile users, the total number of encryption/decryption of the X.509 three-way exchange is the same as that of the C3 protocol. However, the C3 protocol performs one more encryption than the X.509 three-way exchange. For base stations, the total number of encryption/decryption of the X.509 three-way exchange is the same as that of the C3 protocol. The X.509 three-way exchange performs one more encryption operation than the C3 protocol. Consequently, base stations using the C3 protocol are more efficient than those using the X.509 three-way exchange.

For mobile users, the number of verification of the C3 protocol is one less than that of the X.509 three-way exchange. For base stations, the number of signing of the C3 protocol is one less than that of the X.509 three-way exchange. With regard to signing/verification, the C3 protocol is more efficient than the X.509 three-way exchange.

4.2 Analysis of C8

4.2.1 Security Analysis

The security of the C8 algorithm is evaluated by simulation. The simulator is coded in the C language and is running on an Intel Pentium II-266MHz PC with 64 MB RAM. Two metrics, randomness and period, are selected for security evaluation. **Randomness** is defined to be the probability of being 0 or 1. For a purely random keystream, the probability of one bit to be 1, independent of other bits, is equal to one half. **Period** is defined to be the number of bytes in the repeated pattern of a keystream; for example, if a keystream of n bytes has no repeated pattern, then its period is equal to n.

In the following simulations, we consider four different types of initial keystreams (Ms). Type-1 M is purely random, Type-2 M is of all 0's, Type-3 M is of all 1's, and Type-4 M consists of alternate 8-bit 0's and 8-bit 1's. Four different types of input messages (ms) are also considered. Type-1 m is purely random, Type-2 m is of all 0's, Type-3 m is of all 1's, and Type-4 m consists of alternate 8-bit 0's and 8-bit 1's. There are total 16 (4 * 4) combinations (cases) with respect to Ms and ms.

Randomness

Two different sizes of Ms, 20480 bytes (M1) and 20479 bytes (M2), are assumed. Note that 20480 is multiples of 256 (2^8) where 20479 is not. The size of the input message is 2000 KBytes. Each test case is run 100 times, and then the average is computed.

Tables 2(a) and 2(b) show the simulation results of M1 and M2, respectively.

Table 2(a). Simulation results of M1

M1 \ m	Type-1	Type-2
Type-1	0.500033 / 0.499967	0.500003 / 0.499997
Type-2	0.497006 / 0.502994	0.503482 / 0.496518
Type-3	0.498209 / 0.501791	0.485105 / 0.514895
Type-4	0.497070 / 0.502930	0.509102 / 0.490898
M1 \ m	**Type-3**	**Type-4**
Type-1	0.500003 / 0.499997	0.500003 / 0.499997
Type-2	0.503482 / 0.496518	0.503482 / 0.496518
Type-3	0.485105 / 0.514895	0.485105 / 0.514895
Type-4	0.509102 / 0.490898	0.509102 / 0.490898

Legend: a / b, where a is the probability of being 0, and b is the probability of being 1

Table 2(b). Simulation results of M2

M2 \ m	Type-1	Type-2
Type-1	0.500010 / 0.499990	0.499985 / 0.500015
Type-2	0.499929 / 0.500071	0.508430 / 0.491571
Type-3	0.500348 / 0.499652	0.490608 / 0.509392
Type-4	0.500144 / 0.499856	0.497414 / 0.502586
M2 \ m	**Type-3**	**Type-4**
Type-1	0.499984 / 0.500016	0.499987 / 0.500013
Type-2	0.496909 / 0.503091	0.500527 / 0.499473
Type-3	0.487560 / 0.512440	0.487418 / 0.512582
Type-4	0.511040 / 0.488960	0.497268 / 0.502732

Legend: a / b, where a is the probability of being 0, and b is the probability of being 1

By examining Table 2(a), we notice that the probability of being 0 and that of being 1 are the same, for Type-2 m, Type-3 m, and Type-4 m. This phenomenon indicates that the length of the initial keystream (M) should not be multiples of 256, since the C8 algorithm processes data on a byte-by-byte basis. By observing Table 2(b), we find that the probability of being 0 and that of being 1 are close to one half, for all test cases. This phenomenon indicates that the C8 algorithm does produce key strings of evenly distributed 0's and 1's, regardless of input patterns.

Period

We consider four different pairs of (l_M, l_m), where l_M and l_m represent the length (the number of bytes) of the initial keystream M and the input message m, respectively. Those four pairs are (1200, 20480), (2400, 20480), (1200, 102400), and (2400, 102400). For Type-1 M and m, their period is equal to l_M and l_m, respectively. For Type-2 M and

m, their period is equal to 1. For Type-3 M and m, their period is equal to 1. For Type-4 M and m, their period is equal to 2.

Tables 3(a) to 3(d) present the simulation results of pairs 1 to 4, respectively.

Table 3(a). Keystream periods with respect to pair 1 (1200, 20480)

M \ m	Type-1	Type-2	Type-3	Type-4
Type-1	20480	20480	20480	20480
Type-2	20480	20480	20480	20480
Type-3	20480	20480	20480	20480
Type-4	20480	20480	20480	20480

Table 3(b). Keystream periods with respect to pair 2 (2400, 20480)

M \ m	Type-1	Type-2	Type-3	Type-4
Type-1	20480	20480	20480	20480
Type-2	20480	20480	20480	20480
Type-3	20480	20480	20480	20480
Type-4	20480	20480	20480	20480

Table 3(c). Keystream periods with respect to pair 3 (1200, 102400)

M \ m	Type-1	Type-2	Type-3	Type-4
Type-1	102400	102400	102400	102400
Type-2	102400	102400	102400	102400
Type-3	102400	102400	102400	102400
Type-4	102400	102400	102400	102400

Table 3(d). Keystream periods with respect to pair 4 (2400, 102400)

M \ m	Type-1	Type-2	Type-3	Type-4
Type-1	102400	102400	102400	102400
Type-2	102400	102400	102400	102400
Type-3	102400	102400	102400	102400
Type-4	102400	102400	102400	102400

By examining Tables 3(a) to 3(d), we notice that keystreams generated by the C8 algorithm always maintain the maximal period, regardless of input patterns. This phenomenon indicates that the C8 algorithm is able to produce key strings with infinite period.

4.2.2 Efficiency Analysis

The C8 algorithm using only the XOR, INCREASE and ADDITION operations is very efficient.

4.3 Analysis of C5

The C5 algorithm is based on stream cipher. In our design, the simplest stream cipher using only the XOR operation [20] is recommended; therefore, the C5 algorithm is very efficient. As to the security of C5, it is very secure since the keystream (K) generated by C8 has infinite period.

4.4 Comparison

Table 4 compares the architecture of today's GSM network with that of the proposed GSM network.

The current architecture uses the challenge-response mechanism (A3) for user authentication and key exchange. In GSM standards, symmetric cryptography is recommended for supporting this mechanism. Symmetric cryptography is faster than, but less secure than, public-key cryptography. If the A3 algorithm is revealed, the known-plaintext attack could be attempted during user authentication. The public-key-based C3 protocol is more secure, but slower than the A3 algorithm.

The period of the keystream of the C5 algorithm can be very long, which is in sharp contrast to the fixed-period keystream of the A5 algorithm; the longer the period of the keystream, the securer the stream cipher. Although it takes more time to exchange a longer keystream (M) during connection establishment; however, stream cipher using only the XOR operation reduces the overheads during messages transfer. The C5 algorithm can use any publicly available stream cipher whereas the A5 algorithm is proprietary.

Table 4. Comparison between the current architecture and the proposed architecture

	The Current Architecture	The Proposed Architecture
Complexity	Authentication is fast Key exchange is fast Message encryption is fast	Authentication is slow Key exchange depends on the key length Message encryption is fast
Security	Authentication is not secure enough Only the mobile user is authenticated Period is fixed and is not very long	Authentication is very secure Both the mobile user and the base station are authenticated (mutual authentication) Period can be infinite
Flexibility	A3, A8 and A5 are proprietary The SIM stores user's personal information and the A3 algorithm	C1, C8 and C5 are publicly available The SIM only stores user's personal information

5. CONCLUSIONS

5.1 Summary

In this paper, we focus on the security of the Global

System for Mobile communication (GSM) networks. Secure communication mechanisms for the GSM network are proposed. In the proposed architecture, we use public-key cryptography for user authentication, and stream cipher for message encryption and decryption. Cryptanalysis and operational analysis show that the C3 protocol is secure and efficient. Simulation results indicate that the key generation method can always produce key strings of evenly distributed 0's and 1's and with infinite period. Those security mechanisms significantly improve the security of today's GSM network.

5.2 Future Works

A very important part of the proposed architecture is the C5 algorithm. Although any publicly available stream cipher can be used; however, the effectiveness and efficiency of the selected stream cipher significantly affect the performance and the security of GSM networks. In the future research, we will survey existing stream ciphers; for example, the pseudorandom bit generator with bilateral step control [20], and design a more effective and efficient stream cipher for GSM networks.

REFERENCE

[1] Bruce Schneier, "Applied cryptography: Protocols, algorithms, and source code in C", Wiley

[2] R. Anderson and R. Needham, "Robustness Principles for Public Key Protocols", Research Notes

[3] David G.W. Birch and Ian J. Shaw, "Mobile Communications Security - Private or Public", IEE, June 1994

[4] R. Bird et al., "Systematic Design of a Family if Attack-Resistant Authentication Protocols", IEEE Journal on Selected Areas in Communications, Vol.11, No.5, June 1993

[5] R. Bird et al., "Systematic Design of Two-Party Authentication Protocols", Advances in Cryptology -- CRYPTO '91, 1991, pp.44-61

[6] Charles Brookson, "GSM Security: A description of the reasons for security and the techniques", IEE, 1994

[7] J.C. Cooke and R.L. Brewster, "Cryptographic security techniques for digital mobile telephones", Wireless Communications, 1992

[8] D.E. Denning and G.M. Sacco, "Timestamps in Key Distribution Protocols", Communications of the ACM, Vol.24, No.8, pp.533-536, Dec.1981

[9] L. Gong, "Efficient Network Authentication Protocols: lower bounds and optimal implementations", Distributed Computing, July 1995, pp.131-145

[10] Charle Kaufman, Radia Perlamn and Mike Speciner, "Network Security: PRIVATE Communication in a PUBLIC World", Pentice Hall, 1995

[11] Yi-Bing Lin, "No wires attached", IEEE Potentials, Vol. 14, Oct. 1995

[12] Jianwei Liu and Yumin Wang, "A user authentication protocol for digital mobile communication network", PIMRC'95, Vol. 2, 1995

[13] Jianwei Liu and Yumin Wang, "Authentication of mobile users in personal communication system", PIMRC'96, Vol. 3, 1996

[14] Chi-Chun Lo and Yi-Chun Yeh, "Authentication Protocols for the Broadband ISDN Billing System", submitted to Computer Standards & Interfaces

[15] A. Mehrotra and L.S. Golding, "Mobility and security management in the GSM system and some proposed future improvements", Proceedings of the IEEE, Vol. 86, Issue 7, July 1998

[16] M. Mouly and M.-B. Pautet, "GSM Protocol Architecture: Radio Sub-system Signalling", IEEE 41st Vehicular Technology Conference, 1991

[17] M. Mouly and M.-B. Pautet, "The GSM System for Mobile Communications", 1992

[18] John Scourias, "A Brief Overview of GSM", http://kbs.cs.tu-berlin.de/~jutta/gsm/js-intro.html, 1994

[19] S.J. Shepherd, "Public key stream ciphers", IEE, 1994

[20] Kencheng Zeng et al, "Pseudorandom Bit Generators in Stream-Cipher Cryptography", IEEE Computer, 1991

[21] 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