Scientific Research and Essays Vol. 5(22), pp. 3455-3466, 18 November, 2010
Available online at http://www.academicjournals.org/SRE
ISSN 1992-2248 ©2010 Academic Journals

Full Length Research Paper

A novel peer-to-peer SMS security solution using a hybrid technique of NTRU and AES-Rijndael

Sameer Hasan Al-bakri* and M. L. Mat Kiah

Department of Computer System and Technology, Faculty of Computer Science and Information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia.

Accepted 11 October, 2010

Short message service (SMS) is a very popular and easy to use communications technology for mobile phone devices. Originally, this service was not designed to transmit secured data, so the security was not an important issue during its design. Yet today, it is sometimes used to exchange sensitive information between communicating parties. This paper proposes an alternative solution that provides a peer-to-peer SMS security that guarantees provision of confidentiality, authentication, integrity and non-repudiation security services. A hybrid cryptographic scheme has been used which combines the NTRU and AES-Rijndael algorithms to achieve more robust functionality. For implementation, a mobile information device application (MIDlet) has been developed in J2ME to introduce a required security services for SMS. The developed application is tested on real equipment such as a Nokia N70. It is able to achieve all the required cryptographic operations completely on the users' mobile phone in less than one second for each operation, and thus the mobile phone performance still remains effective.

Key words: Public key cryptography, NTRU, SMS security, peer to peer, confidentiality, authentication, integrity and non-repudiation.

INTRODUCTION

Various types of tools have been created to make human communications simpler and faster. The most significant communication tool is the modern telephone which was first invented by Sir Alexander Graham Bell in the 19th century. Since then, communication devices have evolved into very advanced and sophisticated tools. The demand for such devices is tremendous as in the second quarter of 2009, there were more than 4.3 billion mobile subscribers worldwide compared to 3 billion mobile subscribers in 2008 and 2.5 billion mobile subscribers in 2007 (GSMWorld, 2009b). The majority of them are sending and receiving not only casual greetings, but also important data such as social security numbers, bank account details, passwords, and so on and so forth. In some cases, this data may also include very private information reserved for the personal viewing of the legal recipient.

The message journey starts from the mobile phone to the BTS (base transceiver station) over the radio waves. It is transmitted in encrypted format by using A5 algorithm (GSMWorld, 2009a). Usually, the message transmits in unencrypted format in the mobile operator's network to the message center and then stores it until its delivered to the destination (Hassinen, 2006). In September 2003, a group of researchers introduced a practical cipher text-only cryptanalysis for GSM encrypted communication, and launched active attacks on the GSM protocols. They describe a cipher text-only attack on A5/2 that merely requires a few dozen milliseconds of encrypted off-the-air cellular conversation to find the correct key in less than a second on a personal computer (Barkan et al., 2008). It is clear that the transmitted data through the mobile networks is unsecured. Nowadays, the visibility of security applications is wide; the term security presented in the scholar papers side by side with terms such as confidentiality, integrity, authenticity, non-repudiation, privacy and data protection. Sometimes it goes beyond that to present the privacy statements for surveys and interviews. It may not be exaggerating if we say e-life equals to security, researchers state many words on the role played by security on the life. Haque et al. (2009)

*Corresponding author. E-mail: Sameer_hassn2002@yahoo.com. Tel: +60173540964.

says “only protected transactions, have significant impact on consumers’ perception about e-banking security (Hashim et al., 2010) says “Privacy and security are very important issues being discussed in the literature on the current use of ICT”. In addition to that, there are ethics and privacy statements recorded in the first page of surveys such as (Babalola et al., 2006; Findik et al., 2010; Gullu and Yilmaz, 2010; Shittu et al., 2007).

Many researchers have proposed solutions to secure the mobile phone communication by using public key cryptography such as Hassinen, 2006; Anuar et al., 2008; Narendiran et al., 2009; Jimale, 2008; Kuen, 2008. However all the solutions are based on the server architecture and the mobile operator or service provider controls the servers. The servers in such architecture are controlling the cryptographic key generations such as (Kuen, 2008; Zhao et al., 2008) key distributions and authenticate the users as well (Jimale, 2008; Kuen, 2008). One of the main reasons for not implementing public key cryptography in non-server architecture is the restricted resources (that is, computing power and storage capacity) in the mobile phone devices. The second important reason is the user’s authentication scheme. How the user can authenticate the sender’s entity in non-server architecture systems. We believe that we can overcome these problems either; by using the modern technologies in the mobile phone devices or by using additional security techniques.

Current SMS security solutions

In this review for SMS security solutions we focus on three aspects; firstly, the solution’s ability to provide peer-to-peer SMS security. Secondly, the security services which are provided by the solution, in this case we give attention to four main security services; confidentiality, integrity, authentication and non-repudiation. Thirdly, the solution’s independency, some solutions depend on the mobile network operator servers or the service provider servers to achieve some of its functions such as authentication. Marko and Smile (2003), proposed the cryptographic methods based on the theory of quasi groups to secure SMS messages. They used the same key for encryption and decryption. Therefore, we can consider a quasi group as a symmetric cryptography. Their solution provides peer-to-peer confidentiality but does not provide sender authentication and message integrity. Moreover, they did not propose any method for exchanging the secret key and assumed that users can handle those themselves before communicating.

Ratshinanga et al. (2004) proposed a protocol to secure SMS communication between a client and a server using the WMA package. They used asymmetric cryptography (RSA) and symmetric cryptography (AES/CTR) and password authentication strategy to ensure the confidentiality and integrity of the SMS communication. Their solution is designed to secure the communication between the user (that is, mobile phone device) and the server (that is, the mobile network operator servers). If the server is compromised by a hacker, the whole protocol will fail because the hacker can disguise himself as the legitimate party using the stolen information. So it is not fully secure. Moreover, it provides the confidentiality and integrity, but the solution can’t provide a non-repudiation security service. (Croft and Olivier, 2005) used one-time pads using shared information between the communicating peers and the GSM network servers. The keys generated from this shared information using hashing techniques, is sufficiently random for use in their approximated one-time pad. These keys should be unique for every SMS, assuming that GSM server will change the Temporary Mobile Subscriber Identity (TMSI) for every SMS. The SMS has to decrypt at Mobile Switching Center (MSC) where it resides as a plain text. The solution depends on the mobile network operator for some of their solution events such as authenticating the users and the change of the TMSI after using it in an encryption process. Moreover, it did not provide peer-to-peer encryption because the message has to be decrypted in the mobile network and encrypted again then sent to the recipient. If an attacker gains the access to the mobile operator network, the confidentiality will be lost because the SMS is in plain text format at MSC.

Lisonek and Drahanský (2008) used asymmetric cryptography (that is, RSA) to provide confidentiality, integrity, authentication and non-repudiation. They assumed that the certificate authority will handle the generating of the certificates, and the users have to download the certificate from the certificate authority server on their own through the internet. In general, their solution can provide the peer-to-peer SMS security, as well as guarantee the confidentiality, integrity, authentication and non-repudiation security services. However, the dependence on a certificate authority to generate the certificate will complicate the solution implementation for the individuals. Anuar et al. (2008) proposed the SMS/MMS encryption solution. They used both of symmetric cryptography (that is, AES) and asymmetric cryptography (that is, RSA) to provide two levels of secure mobile communication, internal SMS (symmetric encryption: AES), and confidential SMS (asymmetric encryption: RSA). For the asymmetric cryptography, users have to apply for a digital certificate with a certificate authority server before starting to use their solution. After the certificate is issued by the authority server, users must browse the public key cryptography directory on the server to download the public key. This directory will provide the information about all users’ certificate and its status. However, their solution provides the confidentiality, integrity, authentication and non-repudiation, but it depends on the certificate authority server to generate the cryptographic

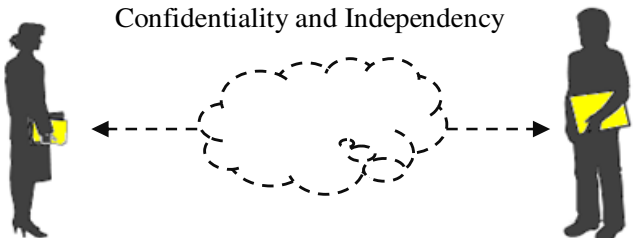

Figure 1. Symmetric cryptography with non-server architecture.

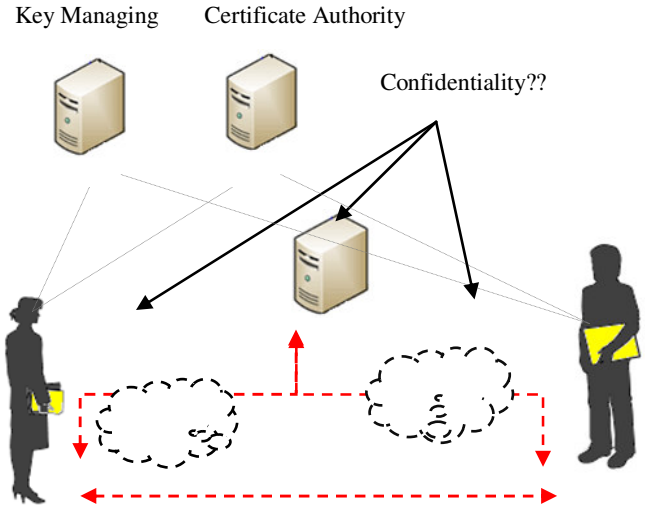

Figure 2. Asymmetric cryptography with server architecture.

keys and to authenticate the users. Toorani and Shirazi (2008) introduced a new secure application layer protocol called SSMS. This protocol can be used to embed the desired security attributes in the SMS messages. It acts as a secure bearer in the m-payment systems. SSMS embeds the confidentiality, integrity, authentication, and non-repudiation in the SMS messages. It provides an elliptic curve-based public key (ECC) solution that uses public keys for the secret key establishment of a symmetric encryption. However, it relies on the third party servers (that is, KGS (Key Generating Server), OCSP (Online Certificate Status Protocol) server) as part of the solution. This dependency demands a third party approval in its implementation.

Zhao et al. (2008) proposed a new solution for a secure messaging channel using identity-based cryptography. This solution provides peer-to-peer security from service provider to mobile users, and between mobile users. Identity-based cryptography specifies a cryptosystem in which both public and private keys are based on the identities of the users. Such a scheme has the property that a user's public key is an easily calculated function of his identity, while a user's private key can be calculated for him by a trusted authority, called private key generator (PKG). Identity- based cryptography needs a setup phase in which system parameters are distributed to its users. These parameters include system public key, master key, private key of each user, and algorithms to be used for encryption and decryption as well as hashing (Zhao et al., 2008). This scheme provides integrity, confidentiality and authentication of SMS by binding a message with a private key. However, it does not provide non-repudiation since the service provider provides the private key to a user, thus the user can simply deny having sent a message signed with his private key. Moreover, the solution depends on the mobile network operator which makes its implementation for individual users more difficult.

Wu and Tan (2009b) proposed a high security communication protocol for SMS. They used the RSA-1024 asymmetric algorithm and AES and 3DES symmetric algorithms to provide a peer-to-peer secure channel between server-side and mobile terminal. They used MD5 and SHA1 algorithms to check messages integrity. Due to their using the asymmetric cryptographic algorithms, symmetric cryptographic algorithms and hashing algorithms they were able to ensure confidentiality, integrity and non-repudiation of SMS messages. However, their solution is server-based solution which can be implemented only by the mobile network operator. From the literature we found many research papers is concerning SMS encryption. Some of them have proposed symmetric cryptography as a solution that can provide confidentiality for SMS communication. These proposals are arguable since symmetric key cryptography is not able to provide the sender authentication, non-repudiation and the message integrity. In Figure 1, we can see the symmetric cryptography in non-server architecture that can guarantee provision confidentiality and independency but not sender authentication and non-repudiation. On the other hand, some others researchers chose to use the asymmetric cryptography as a solution to provide the confidentiality, authentication, integrity and non-repudiation services of SMS communication. However, this is also not the best solution and that because, asymmetric cryptography requires a lot of computational and storage resources to calculate and store public and private keys. Figure 2 shows the use of the asymmetric cryptography usually implemented in server-architecture security systems due to its high computing resources requirements. However, if the server is compromised, the whole security system will fail.

Non-server applications have been successfully implemented and tested to secure peer-to-peer communications, the success factor was the independency of the application from the operator server or any other server. However, this success is not fully acceptable since, this kind of solution has not provided the integrity, authentication and non-repudiation. In addition to that, implementation of non-server application using public key cryptography is a challenge with the current methods

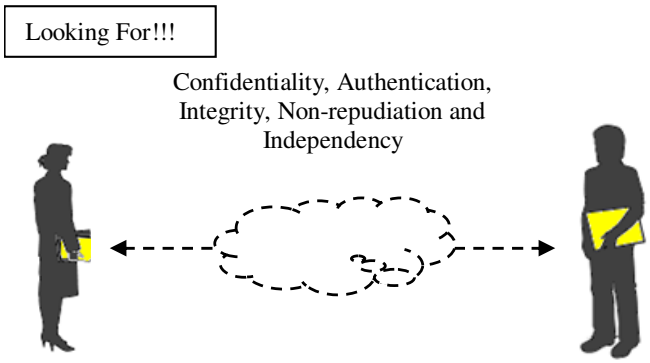

Figure 3. Non-server architecture using public key cryptography.

(that is, RSA, ECC and ElGamal), these algorithms can provide solutions with the integrity, authentication and non-repudiation, however, it cannot be fully implemented without servers. Figure 3 depicts the final aim for this research; which is providing a complete security solution for peer-to-peer SMS communication that guarantee provision of confidentiality, authentication, integrity, non-repudiation and independency security services.

SMS encryption

One of the most successful solutions for the security weakness on the mobile network is cryptography. Cryptography can be defined as the conversion of data into a scrambled code and then sending it to the recipient; the scrambled code can be decrypted to retrieve the original data once it is received. It has two main forms for encrypting data; symmetric and asymmetric encryption. The first is symmetric key cryptography, which is also called secret key cryptography. It is a type of cryptography where the same key is used to encrypt and decrypt the message. While the second is asymmetric key cryptography which is referred to as public-key cryptography, it uses two keys. One is the private key that must be kept private and only known by its owner and the second is the public key that can be made known to all communication participants. Data encrypted with the public key can be decrypted only by the private key, and data signed with the private key can only be verified with the public key. In general, the implementation of cryptography does not require any modification to the mobile network structure or any installation for additional hardware.

Nevertheless, both the two forms of cryptography have some advantages as well as some drawbacks. Symmetric key algorithms are generally much less computationally intensive, unlike asymmetric key algorithms calculations that take longer than symmetric key calculations. This is because, they involve the use of exponentiation of very large numbers which in turn take longer to compute and need more memory to do the calculations and store the keys. Moreover, symmetric key algorithms presume that the communicating parties have agreed on a key and are able to exchange that key in a secure manner before communicating. While, in asymmetric key algorithms there is no need to exchange keys in a secret manner. Another difference is symmetric key algorithms cannot provide authentication, non-repudiation and the message originality. Yet, asymmetric key algorithms can provide authentication, non-repudiation and the message integrity by signing the message with the private key while verifying with the public key. However, a combination of the asymmetric cryptography and symmetric cryptography can achieve more robust functionality.

Server architecture vs. non-server architecture

Usually, the mobile communications security solutions that are based on public key cryptography rely on the mobile phone network operator or service provider as part of the proposed solutions. Generally, the server architecture solutions need for additional hardware (that is, servers) and as result to a qualified staff to maintain the servers. Moreover, server architecture mobile security systems user has to get the mobile network operator or the service provider approval because it depends on their servers. Besides the overhead cost of communication is increased due to users need to access to the servers in many cases such as uploading and downloading the cryptographic keys. We do not expect that the mobile operators will provide security services to the transmitted data through the SMS service for individuals, at least not in the near future.

On the other hand, non-server architecture mobile communications security solutions are implementable for individuals due to its independency from the mobile phone network operator or service provider. Thus the user does not need to make any agreement with the mobile phone network operator or service provider. As a result, all the cryptographic operations are achieved on the user's mobile phone. Moreover, the overhead costs of communication are less than server architecture system. This is because the communication between the user and the server for authenticating purposes are not required. Most of the current non-server mobile security systems are based on symmetric cryptographic algorithms. For example, CryptoGraf messaging software is used to encrypt messages with AES algorithm within the mobile phones without requiring any servers (Wu and Tan, 2009b).

Research objectives

1. To study the existing SMS encryption techniques.
2. To identify the appropriate technique for peer-to-peer

Table 1. Preliminary experimental results (Wu and Tan, 2009b).

	NTRU-251 (ms)	RSA-1024 (ms)
Key generation time	9617	2090509
Encryption time	515	1505
Decryption time	1132	35102

SMS security.
3. To propose an alternative solution for securing SMS.
4. To develop and test the proposed technique in regards to confidentiality, authentication, integrity and non-repudiation services.

Research questions

1. What are the current end-to-end SMS communication security solutions weaknesses?
2. How we can develop the end-to-end SMS security solution which guarantees the confidentiality, integrity, authentication and non-repudiation security services?
3. How we can provide the end-to-end SMS security without depending on mobile network operator or third party?
4. How we can develop a solution for end-to-end SMS security that is implementable by individuals as well as by the commercial ones?

Proposed solution

The public key cryptography is able to provide the confidentiality, authentication, integrity and non-repudiation security services needed to secure SMS. But it needs a high computing power, so it is usually used in server architecture mobile security systems. Although the implementation of public key cryptography in server architecture mobile security systems provides high level of security, the risk of the server penetration by hackers is also possible (Ratshinanga et al., 2004) and then the whole system security will fail. In non-server architecture mobile security systems, all the cryptographic operations will be achieved in the user mobile phone device. Therefore, most of the existing non-server architecture mobile phone security solutions are based on symmetric cryptographic algorithms due to its low demands for computing power, but such algorithms are unable to provide all the required security services (Wu and Tan, 2009a). Unlike the symmetric cryptographic algorithms, the asymmetric cryptographic algorithms are able to provide all the required security services, but they need higher computing power. Thus, to implement asymmetric cryptographic algorithms in non-server architecture mobile security systems, we must select an asymmetric cryptographic algorithm with low computing power demand.

Selected public key algorithm

In 1996, Jeffrey Hoffstein, Jill Pipher and Joseph H. Silverman introduced a new public key cryptographic algorithm based on the shortest vector problem in a lattice. The main advantage of this algorithm is that it runs much faster with lower memory requirement than conventional public key algorithms such as RSA. The security of the NTRU cryptosystem comes from the interaction of the polynomial mixing system with the independence of reduction modulo two relatively prime integers p and q (Hoffstein et al., 1998).

NTRU performance

NTRU is a collection of mathematical algorithms based on manipulating lists of very small integers and polynomials. This allows NTRU to achieve high speeds with the use of minimal computing power. Despite the RSA cryptosystem being the most popular public key system, it requires more computing power than other public key systems such as ECC and NTRU. Most of the smart cards are unable to process RSA 1024 key lengths due to the high computing power required. Thus, there is an increasing need for another public key system which is able to provide the same level of security but at the same time needs less computing power at the same time. The most suitable alternative is NTRU cryptosystem, due to its low requirement of computing power and its ability to provide an equivalent level of security to RSA 1024 (Challa and Pradhan, 2007).

Challa and Pradhan in 2007, published a comparison between RSA and NTRU in an experimental study. Their study shows that NTRU requires approximately only one third of the time that RSA requires in the encryption process and also, it requires only one seventh of the time that RSA needed for the decryption process. Shen et al. in 2009 made an experiment on the performance of enhanced NTRU-251 and compared it with RSA-1024 in the mobile java emulator. The results of the experiment are shown in Table 1. The results show that NTRU is 217 times faster than RSA in key generation and NTRU is also three times faster than RSA in the encryption process. Moreover, NTRU is 31 times faster than RSA in

Table 2. NTRU plaintext and cipher text block sizes (NTRUCryptosystems, 2003).

Key strength	Cipher text size	Padding mode		
		SVES-1 Padding	SVES-2 Padding	SVES-3 Padding
NTRU-251	251	21	20	20
NTRU-347	347	29	28	N/A
NTRU-503	503	42	41	N/A

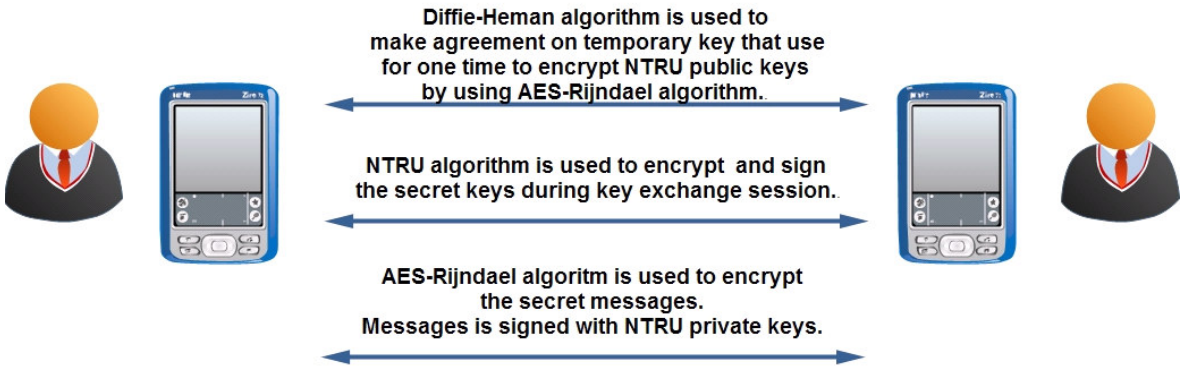

Figure 4. NTRU and AES usage in the system.

the decryption process.

NTRU limitations

Although, the NTRU algorithm is much faster than RSA and ECC and requires less computing power, it also has some limitations, such as: (1) The NTRU algorithms include an encryption algorithm NTRUEncrypt which is used for encryption and decryption and a signature algorithm NTRUSign which is used for digital signature purpose. This means we have to use NTRUEncrypt key pair for encryption or decryption processes and another NTRUSign key pair for digital signature processes. The keys in NTRU are not interchangeable, we can encrypt only with the public key and decrypt only with the private key. Consequently, NTRU user has to use two pairs of keys, one for encryption or decryption and another for the digital signature. (2) The security level of asymmetric cryptography algorithms does not only depend on the difficulty of the problem but also on how the plain text is padded so as to prevent attacks. The padding mode affects the size of the plaintext that can be encrypted in a single block. The common padding technique that is used with NTRU is SVES-3. The following table shows some of the padding techniques. It also shows the plain text block size and the cipher text block size. With regards to mobile text message size, padding will increase the size of encrypted message (NTRUCryptosystems, 2003). The Table 2 shows the cipher text size according to which passing mode is used and which key strength is required.

NTRU algorithm is fast enough to achieve all cryptographic operations on the mobile phone devices, but it has one drawback, it is the cipher text size. NTRU cipher text is large, such that the size of the encrypted SMS with NTRU also results in a large output. We used the NTRU keys with strength 251, which provide high level of security equivalent to RSA 1024 key strength for our solution. The use for NTRU 251 key strength with SVES-3 padding mode will produce cipher text with size 251 byte for each block with size 20 byte. This means that an SMS with a size of 140 byte will break down into seven blocks to encrypt and the cipher text will be 251 multiplied by 7 and that is equal to 1757 byte. Therefore, we used NTRU for securing the key exchange sessions, and we used symmetric algorithm AES-Rijndael to secure the ordinary messages. Figure 4 shows the usage of the NTRU and AES-Rijndael algorithms in the proposed system.

Security services

The most important security services that must be provided by mobile security solutions are: confidentiality, authentication, integrity and non-repudiation. In this part of the paper we will discuss how the proposed solution will achieve these services on the mobile phone devices without adding any additional hardware or relying on third party such as mobile network operator, at the same time the solution must be achieved without negative effects on

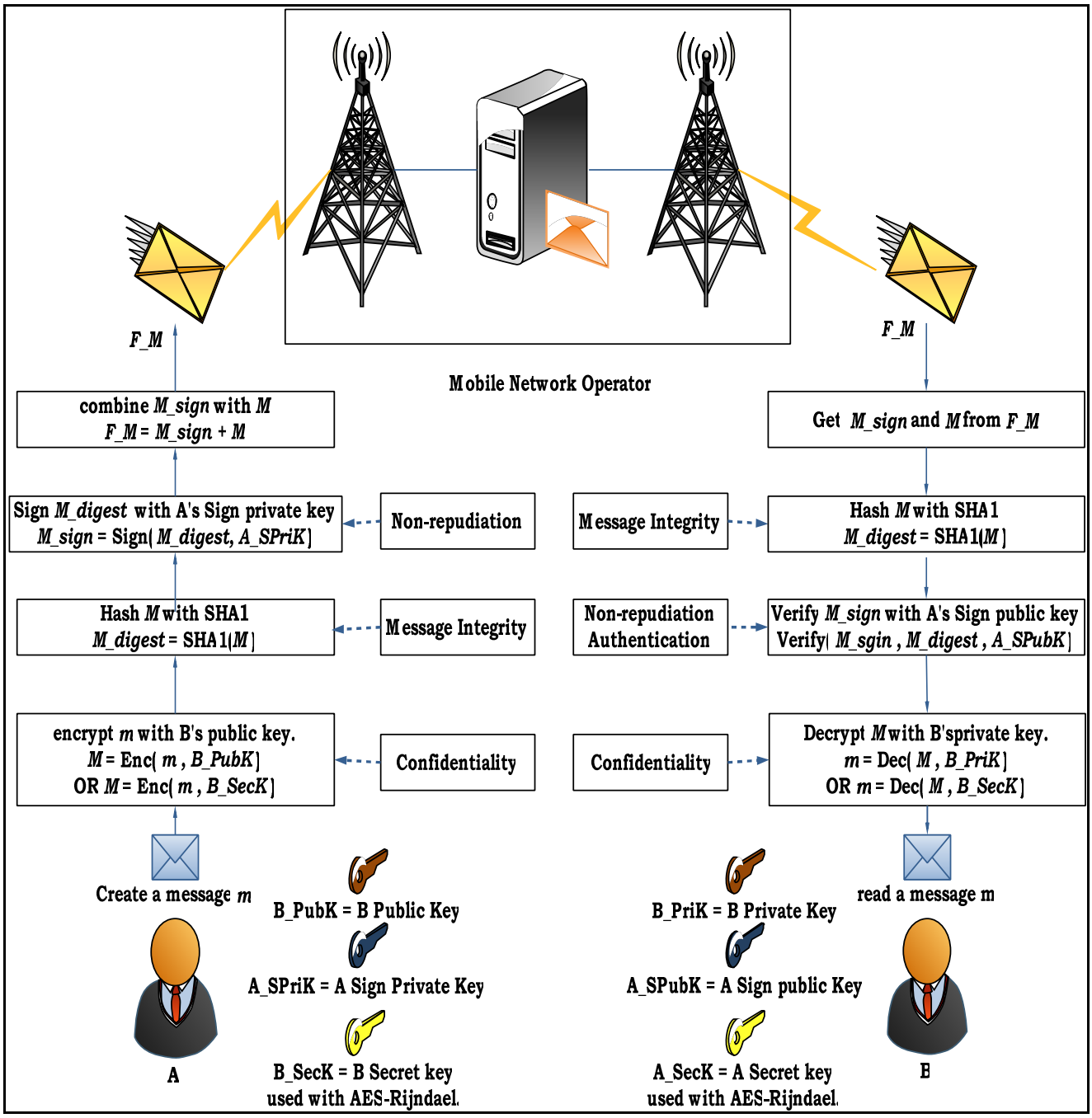

Figure 5. Security services.

the mobile phone's performance. Figure 5 illustrates how the solution achieves all the security services. Some abbreviations that are used in Figure 5 are:

m: the original message in plain text format.
M: the encrypted message in cipher text format.
F_M: the final encrypted message which contain the encrypted message and which is signed by the sender.
M_digest: the hashing value for encrypted message M with SHA1 hashing algorithm.
M_sign: the signed value for M_digest value with A_SPriK by using NTRUSign algorithm.
B_PubK: the user B NTRUEncrypt public key.
B_PriK: the user B NTRUEncrypt private key.
A_SPriK: the user A NTRUSign private key.
A_SPubK: the user A NTRUSign public key.

Confidentiality

Confidentiality is the security service which ensures that

the data will NOT be disclosed for unauthorized parties during its transmission. It is the main security service which is an essential component in any security system. Encryption is the most common technique used to provide confidentiality security service. We use the NTRU public/private keys to encrypt the key exchange sessions, and we use the AES-Rijndael to encrypt the messages.

Integrity

Integrity is the security service which ensures that data is not changed during its transmission from the sender to the receiver. Usually the integrity can be achieved by hashing the encrypted message and encrypting the message hashing then send it with the message to the receiver. Once the receiver receives the message, he will decrypt the encrypted message hashing and compare it with his own hashing on the received message. If the receiver's message hashing equals to the sender's message hashing, then the message has sound integrity, otherwise the message has been modified. We achieve the integrity in our solution by using NTRUSign. NTRUSign has a pair of keys; one for signing the message and another for verifying the signed message. The SHA1 algorithm is used to hash the message then the message digest can be signed with the NTRUSign private key. Once the message is received the NTRUSign public key can be used for verifying the received message.

Non-repudiation

Non-repudiation is the security service that prevents the sender and the receiver from denying their participating in message transmission. Thus, when a message is sent, the receiver can prove that the alleged sender in fact sent the message. Similarly, when a message is received, the sender can prove that the alleged receiver in fact received the message (Stallings, 2005). The most common technique to provide non-repudiation security service is the digital signature. In our solution a sender will sign the encrypted message hashing with his NTRUSign private key, and the receiver can verify that signature with the sender's NTRUSign public key.

Authentication

Authentication is the assurance that the communicating entity is the one that it claims to be (Stallings, 2005). This service provides a system with the capability to verify that a user is the actual one he or she claims to be based on what the user knows or have. For authentication purposes users have to sign the messages that they are going to send with their NTRUSign private key, the receiver will be able to verify that signature with the NTRUSign public key which they have already received during a key exchange session. Thus users will be able to make certain of the identity of the sender within their mobiles phone, without the need to access to the third party severs to check the sender's authenticity.

The problem is in the first contact; before the users exchange the keys among them they can't communicate or authenticate each other. For the first time contact users can't authenticate the sender, because they didn't have the sender's NTRUSign public key. So the difficulty is how we can authenticate the sender without his digital signature. To solve this problem, in our proposed solution the developed application follows some techniques to exchange the cryptographic keys and for the first time, these techniques are as follows:

1. Before the user starts any connection, one must add the other party's information to the application contacts list, such as name, group name and phone number in international format. Therefore any request from any party that does not exist in the application contacts list will be ignored immediately.
2. The application uses a sequence number for each incoming and outgoing messages from each contact in the contacts list. Therefore, any message with a wrong sequence number will be immediately ignored by the application.
3. The application uses the Diffie-Hellman algorithm to make agreement on a temporary secret key which is used with the AES algorithm to encrypt the public key (NTRUEncrypt public keys and NTRUSign public keys) and then exchange it in an encrypted format in session keys exchange prior to the first contact.

Key exchange session protocol

Figure 6 illustrates the key exchange session steps. User X can start the key exchange session immediately after he generates his public keys and adds Y contact information to his contacts list. He can start a key exchange session by calculating the value of A depending on the secretly generated value a, and the share secret parameters g and p, and send the value of A with request to start a key exchange session. User Y can reject the session if not ready to go through the key exchange session steps, he can also accept the request and as result he must calculate the value of B and send it back to X with accept message.

User X will be able to calculate the value of K once he receives the value of B from the user Y. user Y also will be able to calculate same key K depending on the value of A which is already received from user X in request message. Thus, the users X and Y will obtain the same secret key which they can use for one time only to encrypt their public key and exchange it. For next key

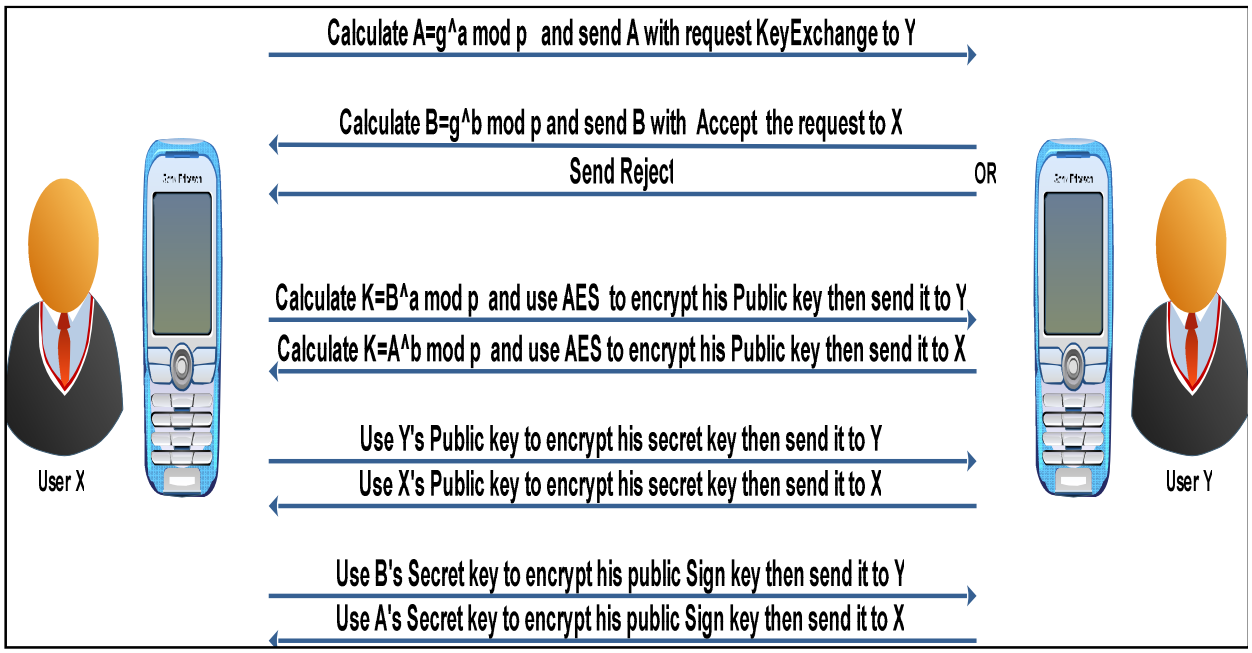

Figure 6. Key exchange session.

exchange session users can use NTRU public keys to encrypt and sign the new cryptographic keys before exchanging them.

Proposed solution implementation

Java is an object-oriented programming language that compiles the program to byte code (.class files) that runs on a virtual machine. The compiled byte code is then ready to be executed within a special virtual environment known as the Java Virtual Machine (JVM). The JVM acts as a consistent layer between byte code and the actual machine instructions. The goal of the virtual machine layer is to protect the underlying device from the damage executable code might cause. Byte code instructions are translated into machine-specific instructions by the JVM at runtime. This enables programmers to write one program and run it on different operating systems. The expression often used to sum up the work of the JVM is “Write once, run anywhere” (Flynt and Wells, 2008).

J2ME

The main goal of the Java 2 Mobile Edition (J2ME) is to provide an extensible yet highly portable, minimum-footprint, Java implementation that can run on a wide variety of network devices with constant or intermittent network connectivity. J2ME consists of a light-weight of JVM known as Kauai virtual machine (KVM) capable of providing a secure and clean execution environment in resource-constrained mobile devices such as mobile phone devices.

RESULTS OF THE PROPOSED SOLUTION TEST

We selected four of Nokia devices to test the NTRU speed; Nokia N70, Nokia N73, Nokia N93 and Nokia 5800 Xpress Music; the Nokia N70 belongs to the second generation of Nokia mobile devices, and it operates with symbian operating system v8.1a. It has ARM9 CPU with a 220 MHz clock rate. It has 22MB internal memory and it supports extension memory from MMC type. Next, is the Nokia N73 is from the third generation and operates with the developed version of symbian operating system, which is symbian OS v9.1. This model has Dual ARM 9 CPU with 220 MHz clock rate. It has 42 MB internal memory and 2 GB Mini SD as extended memory. Followed by the Nokia N93 which belongs to the third generation, yet it has Dual ARM 11 CPU with 332 MHz clock rate. It also operates with the symbian operating system v9.1. This model has 50MB internal memory and 2 GB Mini SD as extended memory. Finally: Nokia 5800 Xpress Music it the most modern one among the rest. It belongs to the fifth generation, and operates with the Symbian OS v9.4. This model has ARM 11 CPU with 434 MHz clock rate. It also has 81MB internal memory with 16GB Mini SD as extended memory (Nokia, 2009).

We performed tests on key generation, encryption and decryption, as well as signing and verifying operations for one hundred times and then calculated the averages. Table 3 shows the average of elapsed time on the real

Table 3. NTRU tests on real equipment.

Cryptographic operation	Nokia N70	Nokia N73	Nokia N93	Nokia 5800 Express	
Public keys generation	142	77	53	29	
Encryption	14	12	6	3	Data size = 1 block =20 Byte.
Decryption	32	13	10	5	
Sign	38	42	24	14	Data size = 20 byte.
Verify	28	23	17	8	

Total tests: 100 for each. Key Strength: NTRU251. Time Measurement: milliseconds.

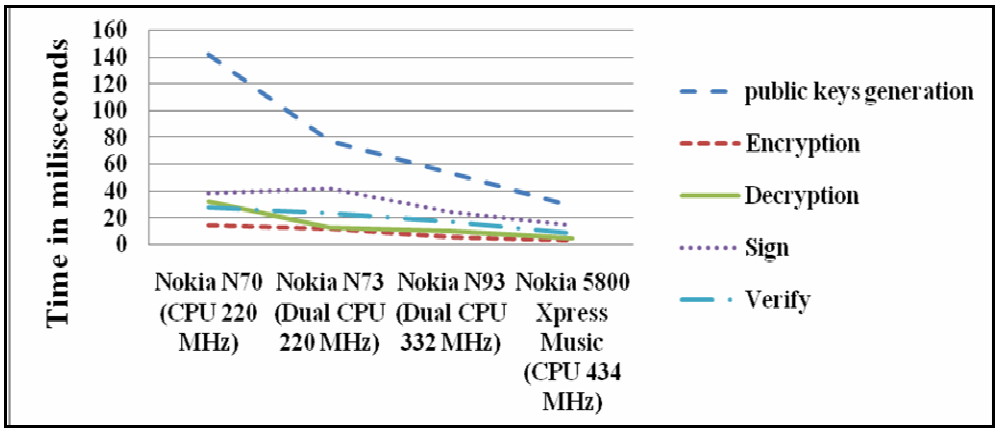

Figure 7. Time taken for cryptographic operation vs. CPU speed of a mobile device.

equipment for each model. The results in Table 3 that shows that the NTRU algorithm performed very well on the mobile devices and there are no negative effects on the mobile devices' performance. Figure 7 shows that the time will decrease when the CPU clock rates increased. From the results above we notice that NTRU does not require high computing ability, which makes it the best alternatives for mobile devices. Moreover, the NTRU algorithm will not slow the mobile phone performance.

Security strength

The various stages of the proposed solution and the potential risks are discussed here. Subsequently, we will highlight on how to protect the users from the risk. The proposed scheme has two main stages: the exchange of cryptographic keys and the exchange encrypted messages safely.

Keys exchange session

The potential risk lies in assuming that the attacker is able to capture the exchanged messages during the keys exchange session. To analyse this case, we will focus on the keys exchange between two users X and Y and the potential risks of interception of the messages by the attacker Z. The first message is the request message for the keys exchange. The first message is sent from X to Y and it holds the value of A. Assuming that the attacker Z manages to capture this message, he/she will unable to obtain the value of key K by only depending on the value of A. The second message is the reply message which is sent from Y to X. This message is to accept the exchange of keys and it contains the value of B. Assuming that the attacker Z manages to capture this message, he/she will not be able to obtain the value of key K, because the value a is kept secret; this value is only known by user X, as well as the value of b which is kept secret by user Y. In addition, the lack of knowledge of Diffie-Hellman algorithm parameters g and p will make the calculating of the key K value impossible.

The third and fourth messages are the encrypted messages using the key K and AES-Rijndael algorithm, which hold the users' NTRU public keys. Even if the attacker could capture the messages; he/she will fail to decrypt them and will not know the users' NTRU public keys because of lack of knowledge of the value of key K. Moreover, the attacker Z will fail to start a key exchange session because of lack of knowledge of the Diffie-Hellman parameters, and because of lack of knowledge

of the port number, also the solution will reject his request because his contact numbers are not in the contacts list. Therefore, the user will be confident after the completion of the keys exchange session that the process has been made with the right person. In addition, even if the attacker Z successfully impersonated one of the parties, he/she will fail to complete a successful keys exchange with the other user due to lack of knowledge of the Diffie-Hellman parameters that are needed to complete the process of making agreement on a shred secret key with the other user.

Exchanging encrypted messages

The key exchange stage is only a temporary stage and needed only in the first contact between the users. Once the key exchange has accomplished successfully the next stage will start, which is exchanging encrypted messages. This stage is the permanent and fixed stage. At this stage, users will be able to send and received the encrypted and signed messages. They also will be able to exchange the new updates for the current keys in encrypted and signed messages. As a result, they will able to verify the identity of the sender of any message and they can ignore any spurious message. Since the attacker fails to benefit from any of the captured messages during the keys exchange session in the process of violation of the privacy of any party to the communication, he also will not be able to decrypt the captured encrypted messages later. Thus we can say that the proposed non-server security scheme is capable to provide a high level of security for users.

Conclusion

Using a hybrid cryptographic scheme of NTRU and AES-Rijndael allows us to provide peer-to-peer SMS security solution which can be implementable in non-server architecture mobile security systems. It provides all the necessary security services such as confidentiality, integrity, authentication and non-repudiation of the user. The developed solution is not only for commercial and governmental use but for the average individuals as well. The developed solution runs fast enough so it has no slowdown on the mobile device's response, as well as it does not need the addition of extra hardware. Moreover, it is completely independent from the mobile network operator or any other third party As a result it is implementable by the individuals as well as commercially.

ACKNOWLEDGEMENT

This project has been funded in part by University of Malaya with grant number UMRG- RG029/09ICT. The author would like to acknowledge the people who helped or contributed in any way.

REFERENCES

Anuar N, Kuen L, Zakaria O, Gani A, Wahab A (2008). GSM mobile SMS/MMS using public key infrastructure: m-PKI. WSEAS Transactions on Computers, 7(8): 1219-1229.

Babalola O, Shittu L, Adesanya O, Jewo I, Oyewopo O, Ashiru O (2006). Pregnancy Outcome Following Swim Up Preparation Of Both Fresh and Cryopreserved Spermatozoa. Sci. Res. Essays, 1(3): 103-107.

Barkan E, Biham E, Keller N (2008). Instant Ciphertext-Only Cryptanalysis of GSM Encrypted Communication. J. Cryptol., 21(3): 392-429.

Challa N, Pradhan J (2007). Performance Analysis of Public key Cryptographic Systems RSA and NTRU. IJCSNS Int. J. Comput. Sci. Network Security, 7: 87-96.

Croft NJ, Olivier MS (2005). Using approximate one-time pad to secure short messaging Service (SMS), pp. 71-76.

Findik KT, Tasdemir S, Sahin I (2010). The use of artificial neural network for prediction of grain size of 17-4 pH stainless steel powders. Sci. Res. Essays, 5(11): 1274-1283.

Flynt JP, Wells MJ (2008). Java ME Game Programming.

GSMWorld (2009a). GSM Security Algorithms. Retrieved 2-9-2009, from http://www.gsmworld.com/our-work/programmes-and-initiatives/fraud-and-security/gsm_security_algorithms.htm.

GSMWorld (2009b). Market Data Summary (Q2 2009). Retrieved 2-9-2009, from http://www.gsmworld.com/newsroom/market-data/market_data_summary.htm

Gullu M, Yilmaz I (2010). Outlier detection for geodetic nets using ADALINE learning algorithm. Sci. Res. Essays, 5(5): 440-447.

Haque A, Tarofder A, Rahman S, Raquib M (2009). Electronic transaction of internet banking and its perception of Malaysian online customers. Afr. J. Bus. Manage., 3(6): 248-259.

Hashim F, Alam G, Siraj S (2010). Information and communication technology for participatory based decision-making-E-management for administrative efficiency in Higher Education. Int. J. Phys. Sci., 5(4): 383-392.

Hassinen M, Markovski S (2003). Secure SMS messaging using Quasigroup encryption and Java SMS API. In: SPLST'03, Finland.

Hassinen M (2006). Java based Public Key Infrastructure for SMS Messaging. Information and Communication Technologies, 2006. ICTTA'06. 2nd, 1.

Hoffstein J, Pipher J, Silverman JH (1998). NTRU: A Ring Based Public Key Cryptosystem. Lecture Notes in Computer Science 1423: 267-288.

Jimale MA (2008). Securing Mobile Communications Using Public Key Infrastructure for Multimedia Messaging Service (MMS). University Malaya, p. 70.

Kuen LN (2008). Mobile Messaging Using Public Key Infrastructure: M-PKI. University Malaya.

Shittu LAJ, Zachariah MP, Ajayi G, Oguntola JA, Izegbu MC, Ashiru OA (2007). Knowledge and perception of health workers towards tele-medicine application in a new teaching hospital in Lagos. Sci. Res. Essays, 2(1): 016-019.

Lisoněk D, Drahanský M (2008). SMS Encryption for Mobile Communication. Security Technology, 2008. SECTECH '08. International Conference on.., pp. 198-201.

Narendiran C, Albert Rabara S, Rajendran N (2009). Public key infrastructure for mobile banking security. Paper presented at the Global Mobile Congress.

Nokia (2009). Device comparison from http://www.forum.nokia.com/Tools_Docs_and_Code/deviceComparison.xhtml?dev=[N70,N93i,N73,5800_XpressMusic] <http://www.forum.nokia.com/Tools_Docs_and_Code/deviceComparison.xhtml?dev=%5bN70,N93i,N73,5800_XpressMusic%5d>

NTRUCryptosystems (2003). NTRU Neo for Java Programming Interface, Version 3.51.

Ratshinanga H, Lo J, Bishop J (2004). A Security Mechanism for

Secure SMS Communication.
Stallings W (2005). Cryptography and Network Security Principles and Practices.
Toorani M, Shirazi AAB (2008). SSMS - A Secure SMS Messaging Protocol for the M-Payment Systems.
Wu S, Tan C (2009a). High Security Communication Protocol for SMS, 2: 53-56
Wu S, Tan C (2009b). A High Security Framework for SMS. Paper presented at the Biomedical Engineering and Informatics, 2009. BMEI '09. 2nd International Conference, pp. 1 - 6.
Zhao S, Aggarwal A, Liu S (2008). Building Secure User-to-user Messaging in Mobile Telecommunication Networks. Wireless Telecommunications Symposium, WTS 2008, pp. 151-157.