Comparison Between PKI (RSA-AES) and AEAD (AES-EAX PSK) Cryptography Systems For Use in SMS-based Secure Transmissions

Hao Wang, William Emmanuel Yu
Department of Information Systems and Computer Science
Ateneo de Manila University, Philiphines

ABSTACT: *In today's mobile communication systems, security offered by the network operator is often limited to the wireless link. This means that data delivered through mobile networks are not sufficiently protected. In the particular growing field of interest of machine-to-machine (M2M) communications, these applications typically require a mobile, secure and reliable means of data communication. This paper compared two (2) cryptographic mechanisms, the RSA-AES and the AES-EAX PSK which provide end-to-end security for SMS-based transmission. We implemented these two (2) mechanisms assuming the constraints of standard SMS network and measured their performance in terms of transaction time. Our study indicated that in terms of processing time, the Authenticated Encryption and Associate Data (AEAD) modes represented by EAX performed better even when the digital signature of the Public Key Infrastructure (PKI) mode represented by RSA was not included.*

Keywords: Cryptography, Encryption, RSA, EAX, GSM, SMS

Received: 2 March 2011, Revised 7 April 2011, Accepted 11 April 2011

1. Introduction

The Global System for Mobile Communications (GSM) is a common standard issued by the European Telecommunications Standards Institute (ETSI). Phase I of the GSM specification was published in 1990 and is currently the most widely used mobile phone system in the world. The Short Message Service (SMS) standard was first discussed in the early 1980s but the world's first commercial SMS service was not introduced until 1992. SMS was created as part of Phase I of the GSM standard. SMS is widely adopted with approximately one (1) billion SMS messages sent every day only in the Philippines[1].

Recently, a survey carried by the Internet Data Center (IDC) shows that more than 90% of mobile users prefer SMS as their main communication tool[3]. The report has concluded that with the statistic of 65% of the mobile users sending text messages every day, SMS will continue to play an important role as the most popular mobile data application for a few more years. This also goes to show that network operators have invested significantly in ensuring the optimal performance of their SMS networks.

With the rise of mobile communications and commerce and the increasingly wide use of machine-to machine (M2M) communication applications, such as the fields of Automatic Teller Machine (ATM) banking, telemetry and telematics, navigation, smart metering and many others, a mobile, secure and reliable means of data communication is a primary necessity. Currently, the SMS M2M networks have become a popular means of transmitting the sensitive information necessary for these applications. However, SMS security needs to be improved.

2. Statement of the Context

ABI Research estimates that the total number of cumulative global M2M connections rose from 46.78 million connections in

2007 to 71.09 million cumulative connections in 2009, and this number is still growing[21]. M2M market boosted by thriving technologies, and is currently being applied widely; some of the use cases involve financial, telemetry and telematics, navigation, logistics and voting systems. The most widely available data service in GSM networks today is SMS. This is why we focus on SMS for this study. However, the current GSM data transmission in some cases cannot provide a secure and stable environment. So its security has become an increasingly important issue. In particular, some specific M2M applications (such as ATM banking, POS machines, voting systems) need a higher level of security than currently provided by mobile networks.

When sensitive information is exchanged using SMS, it is crucial to protect the content from eavesdroppers. By default, SMS content is sent over the Global System for Mobile communications (GSM) network in clear text form, or in a predictable format[20]. The message sent from the mobile device will store at the message centre of associate network provider. The message will travel across different base station in unprotected manner. This means there is an opportunity to allow the middle man attack on those confidential messages. Moreover, this allows an attacker with the right equipment to eavesdrop on the information that is being sent. Another problem with SMS is that the originating address (OA) field in the SMS header can be forged, thus allowing masquerading and replay attacks. Therefore SMS is not totally secure and cannot always be trusted. For example, there has been at least one case in the UK where SMS information has been abused by the operator employees[20].

In some cases, SMS messages are encrypted using a family of cryptography algorithms collectively called A5. A5/1 is the "standard" encryption algorithm, which was used by about 130 million customers in Europe. While A5/2 is the "export" (weakened) algorithm, which was used by another 100 million customers in other markets. A5/3 is a new algorithm based on the UMTS/WCDMA algorithm Kasumi[13].

However, a number of attacks on A5 have been published[14][12][22]. Some require an expensive pre-processing stage after which the cipher can be attacked in minutes or seconds. Until 2000, the weaknesses have been passive attacks using the known plaintext assumption. In 2003, more serious weaknesses were identified which can be exploited in the ciphertext-only scenario, or by an active attacker. In 2006, Elad Barkan, Eli Biham and Nathan Keller demonstrated attacks against A5/1, A5/3, or even GPRS that allow attackers to tap GSM mobile phone conversations and decrypt them either in real-time, or at any later time. It follows that the current GSM network does not provide end-toend security services even with A5[17]. This requires system to provide external privacy guarantees.

3. Statement of the Objectives

The objective of this study is to implement and compare the performance of a PKI and an AEAD encryption system for securing SMS-based transmission networks[26] in terms of transaction time. We first introduce a PKI-based mechanism on the Rivest, Shamir and Adleman (RSA) algorithm[23]. Followed by describing an Authenticated Encryption and Associate Data (AEAD) mechanism called EAX (AES-EAX PSK)[10]. Both these systems are used to provide privacy/confidentiality, integrity and authenticity as security guarantees. Then, we describe the implementation of both mechanisms. Finally, we evaluate and compare the performance in terms of transaction time between these two mechanisms.

4. Scope and Limitation of the Study

Cryptography does not "solve" computer security. Security is always relative. It's hard to say that any cryptography algorithm is always safe. With the hardware and network development, or there is a probability that the current encryption algorithms used are likely to be cracked sooner or later, then we have to use a longer key or more advanced algorithms to ensure data security. These cryptography algorithms, therefore, still need to be constantly developed and improved, providing greater strength and speed.

In this study, the computing system used and platforms are controlled, the payload for both was also controlled. Key sizes used were based on equivalent strength provided. In order for these security mechanisms to be used in the SMSbased transmission network, the final payload must be broken into fragments 140 bytes which is the maximum amount of data an SMS can carry[6].

5. Security and Mobile Networks

In this section, we present an overview of the required security guarantees and current state of mobile network security.

5.1 Security Guarantees

In the current scheme of information security practice, there are some specific security guarantees that we require to consider a service secure. The ISO 17799[2] names the following guarantees:

1. Authentication: The process of guaranteeing the identity of the message sender.

2. Privacy/confidentiality: Ensuring that no one can read the message except the intended receiver.

3. Integrity: Assuring the receiver that the received message has not been altered in any way from the original.

For this study, we used these three (3) security guarantees as the baseline for designing our security mechanisms.

5.2 State of Mobile Network Security

SMS is a highly suitable bearer given its pervasiveness, reliability and integrity. The payload is small, with only 160 ASCII characters or 140 bytes for binaryencoded messages, which results in a highly efficient means for transmitting short bursts of data[6]. SMS is globally available, and requires no further external protocols or provisioning since it is a complete, two-way delivery system native to the GSM protocol. SMS is delivered to a GSM network that will further the message to the necessary recipient or service[15].

However, the most pervasively deployed GSM encryption algorithm, A5, is now considered ineffective. Some solutions exist that only require an expensive pre-processing stage after which the cipher can be attacked in minutes or seconds[13][16][8]. There are a number of studies that cover the safety aspect of SMS transmission. In discussing the weaknessess of the SMS, Lo et al [19] suggested a PKI-based approach to overcome SMS communications security problems. This is an over-the-top approach, the principles of which can be implemented in other network packages or mechanisms. To ensure the secure transport of keys, PKI, particularly RSA, is employed as the key exchange mechanism. In addition, AES in CTR mode and HMAC with SHA256 are used for integrity and privacy, which, according to Bellare et al [11], is an example of a non-composite scheme.

In this study however, key exchange was not considered; the use of preshared keys was assumed; and a composite authenticated encryption scheme was employed.

A good overview of the built-in security infrastructure of today's mobile networks is given by Schmidt [24] who faults current security mechanisms such as A5 and A3 as potentially weak and untrustworthy. Existing crypto-system, he notes, does not provide some security guarantees such as non-repudiation. However, this can be supplemented by an additional security infrastructure such as TLS/SSL, he recommended, which is a PKI-based approach.

Another research by Abidalrahman et al [7] compared the Secure Hash Algorithm (SHA) family and provided estimates of the amount of increase in energy, area and time consumption. After reviewed the standard SHA family members' designs, the results and the compatibility of the SHA algorithms for Wireless Sensor Networks (WSNs) were implemented on hardwares. The author indicated the feasibility of SHA-2 family algorithms as a replacement of the broken SHA-1 and Message-Digest 5 (MD-5) algorithms for WSNs. SHA-256 is shown as the better energy consumption per block.

6. Framework

The study compared the performance of a PKI and AEAD cryptography systems for use in SMS-based secure transmissions in terms of transaction time. AES algorithm with 128-bit keys was used to serve as the baseline block cipher. A comparison between these two mechanisms was attempted in terms of security guarantees and performance.

6.1 RSA-AES Mechanism

The most popular PKI algorithm used is RSA. The algorithm generates two keys, a public key and a private key, by manipulating two prime numbers with a series of computations. The public key distributed publicly and the private key can be kept secretly by the user. This is to ensure that the secure message, which was encrypted using the recipient's public key, will be read by the targeted person, with the private key to decrypt the encryption. Furthermore, the public key can be used to verify the digital signature which is signed with the sender's private key. The RSA scheme is a block cipher in which the original nonciphered text and cipher text are integers between 0 and n-1 for some 'n'. That is, the block size of RSA is determined by the bit length of the integer 'n' and regarded as the key size of the RSA scheme[23].

Decryption: For a given cipher text C, the original non-ciphered text is computed by M = C d mod n.

As RSA requires very large prime numbers, it is impractical to use it to encrypt the entire payload. So RSA cryptography is used to encrypt the keys. In this study, 128 bit AES cryptography is used first to encrypt the data, specifically the Cipher Block Chaining (CBC); then RSA is used to signature and encrypt the transaction key of AES. This mechanism is similar to the one used by Transport Layer Security and Secure Sockets Layer (TLS/SSL)[18].

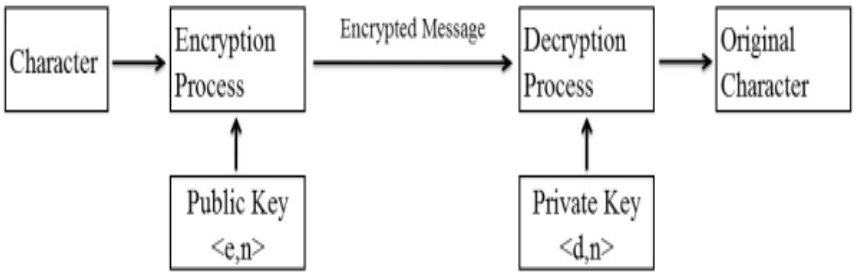

Figure 1. The RSA procedure for sending an encrypted short message

6.2 AES-EAX PSK Mechanism

EAX is an n-bit mode of operation. This allows the mode to operate on AES with 128 bits; or Secure Hash Algorithm (SHACAL-2) with its 256 bit block size. EAX is online, meaning the data does not need to be known in advance; it can be streamed into the object though there are some practical implementation constraints. The AES-EAX PSK scheme is an Authenticated Encryption with Associated Data (AEAD) algorithm designed to simultaneously provide both authentication and privacy of the message (Authenticated encryption) with a two-pass scheme, one pass for achieving privacy and one for authenticity for each block[10]. EAX is also shown to provide all three (3) required security guarantees: privacy, integrity and authentication[9].

6.3 Overview Of Cryptography Mechanisms

In the RSA-AES mechanism, RSA is used to encrypt the transaction keys. AES in CBC mode is then used to encrypt the data with the transaction key for transmission. Finally, RSA is then used to sign the payload. In the AES-EAX PSK mechanism, EAX is used with the AES block cipher using a pre-shared key (PSK).

Table 1 shows the comparison between RSA-AES and AES-EAX PSK according to protection guarantees they provide.

7. Methodology

7.1 Tools

Python is an object-oriented, literal-style computer programming language, and has a history of more than ten years of development, maturity and stability[5]. Python has a very large library, they can be quickly adopted by most common tasks, such as: string processing (regular expressions, Unicode, calculating differences between files), Internet protocols (HTTP, FTP, SMTP, XML-RPC, POP, IMAP, CGI programming), software engineering (unit testing, logging, profiling, parsing Python code), and operating system interfaces (system calls, file systems, TCP/IP sockets).

Botan is a BSD-licensed cryptographic library written in C++ and with Python bindings[4]. It is one of the few libraries which can provide complete and functional AES-CBC, AES-EAX and RSA cryptographic algorithms. In this paper, we have used a development build of Botan with Python bindings and the RSA-PrivateKey fix for Fedora 13[25].

7.2 Methodology

First of all, the composite encryption modes were used in this study, and the process for data splitting, combination, encryption and decryption was coded by Python and the Botan cryptography libraries. The target environment is described in Table 2.

Second, there are two different modes of transmitter flows. Figure 2 shows the two different kinds of transmitter flow. Based on these two modes, four (4) different schemes were compared as follows:

	RSA-AES	AES-EAX PSK
Authentication	RSA	ASE-EAX
Privacy/confidenrtiality	ASE-CBC	ASE-EAX
Non-repudiation	RSA	ASE-EAX
Key exchanging support	RSA	PSK
key exchanging size	1024 bit	128 bit
Payload key size	128 bit	128 bit

Table 1. Comparison between RSA-AES and AES-EAX PSK according to protection guarantees they provide

Operating System	Linux Fedora 13
Program Language	Python 2.6
Cryptography Library	Botan 1.9.3 with RSA-Private Key Fix
Computer CPU	Intel Core Duo T7300 2.00GHZ
Computer Memory	1024MB

Table 2. Experiment environment

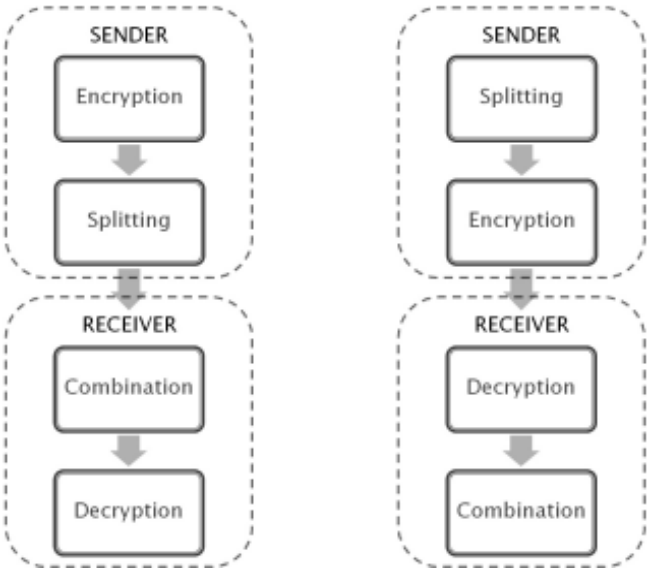

Figure 2. Two Kinds of Transmitter Flow

1. For the sender, encrypt the entire data using AES-EAX first, then split it into block size; In the receiver's side, combine the encrypted block files together, then do the decryption. In the following experiments, "eax-es" can be used to refer to this set of data;

2. For the sender, split the data into block size first, then encrypt each block file using AES-EAX; In the receiver's side, decrypt each encrypted block file, then combine the decrypted block files together. In the following experiments, "eax-se" can be used to refer to this set of data;

3. For the sender, encrypt the entire data using RSA-AES first, then split it into block size; In the receiver's side, combine the encrypted block files together, then do the decryption. In the following experiments, "rsa-es" can be used to refer to this set of data;

4. For the sender, split the data into block size first, then encrypt each block file using RSA-AES; In the receiver's side, decrypt each encrypted block file, then combine the decrypted block files together. In the following experiments, "rsa-se" can be used to refer to this set of data.

Measurements were taken according to the transaction time consumed for each step. The measure of the performance was the transaction time of encryption and decryption for each scheme. Splitting and combining times were recorded as well. All the experiments conducted were under the same environment to ensure a fair comparison. In the RSA mechanism, the digital signature was not included in the computation as it would significantly skew the results if each part was computed for a digital signature.

The experiments were carried out on four different sizes of files, which are 1KB, 10KB, 100KB and 1MB. The block size for each experiment was the same, 140 bytes, which is the maximum SMS payload. Each experiment mode was repeated 100 times in order to guarantee the accuracy of data as possible.

8. Results

Through the discussion of the previous chapter, we have developed the system in Python. There are four (4) sets of data for each experiment, which is "eax-es", "eax-se", "rsa-es", "rsa-se". The format of each output result is shown in Table 3.

Protocol	Action	File Name	Creation Time	File size (bytes)	Block Size (bytes)	Part	Repeat Number	Time (milliseconds)

Table 3. Preliminary results format

The "Action" field can be "Splitting", "Combination", "Encryption" or "Decryption". "File Name" refers to the name of the file which is going to be encrypted. "Creation Time" can be accurate to the second. Both "File Size" and "Block Size" are in bytes. The values of the "Part" represent the number of files which original document can be divided into. The "Repeat Number" refers to the current number of repetitions. "Time" is in milliseconds.

Action	File Name	Creation Time	File size (bytes)	Block Size (bytes)	Part	Repeat Number	Time (milliseconds)
Encrption	test 10kb.bak	2010/9/30 8:33	10240	140	74	8	22.87197113
Splitting	test 10kb.bak	2010/9/30 8:33	10240	140	74	8	10.96510887

Table 4. Samples of preliminary results

In table 4, this output refers to the 8th repetition, encrypt the "test10kb.bak" file first, then split the encrypted file into 140 bytes, which becomes 74 parts, and the encryption time is 22.87197113 ms, the splitting time is 10.96510887 ms. A box plot is used to reflect the results of the experiment.

The following is the box plot for a preliminary experimental data. Obviously, in the 1kb file encryption, both eax-es and eax-se, were much faster than rsa-es and rsa-se. But in the decryption, their time difference was not large, this is because the original file was only 1kb, divided into 140 bytes, the number of block files was very small. Let us move forward to 10kb file. In the 10kb file encryption, eax-es and eax-se was still shown very obvious advantages, and eax-es seems more faster. In the decryption, the "se" and the "es" began to have some time gaps. The advantage of "eax-es" was clearly reflected in 100kb file, both in encryption and decryption.

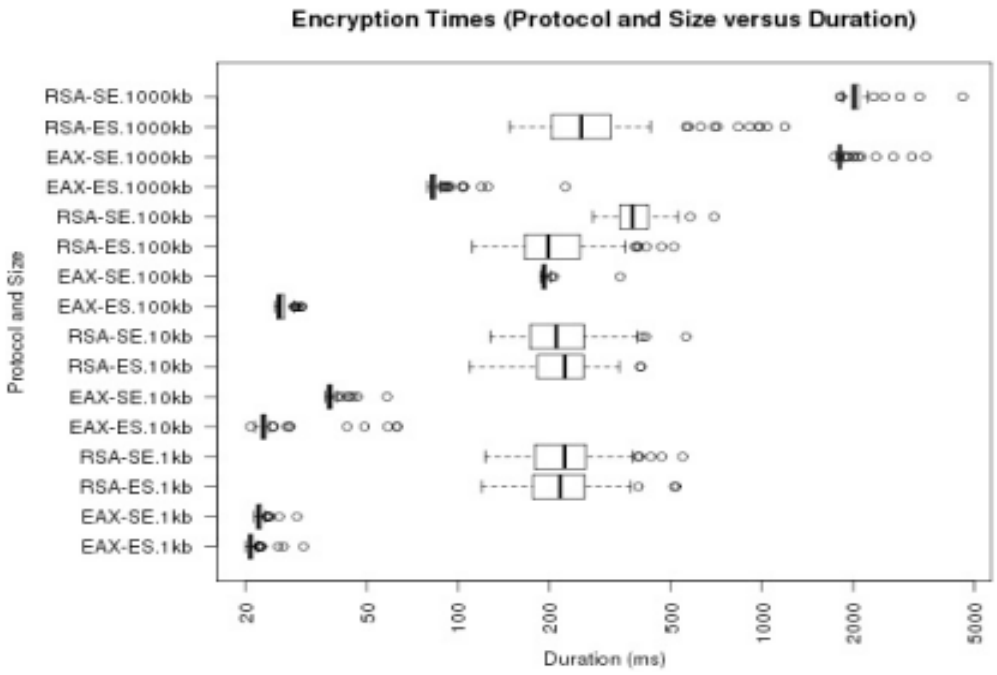

Figure 3. Box plot for encryption

In the experiments of 1mb file, which was split 1MB file to 140 Bytes, it generated more than 10,000 files. In this case, the speed of EAX and RSA encryption and decryption were not the main factor; the limitation of the hardware became an issue. Obviously, encrypting the entire data then splitting into block size, was much faster than splitting the data into small pieces then encrypting every small block data. As a next step, the researchers will try to re-implement the system without using files to avoid hitting this I/O bottleneck.

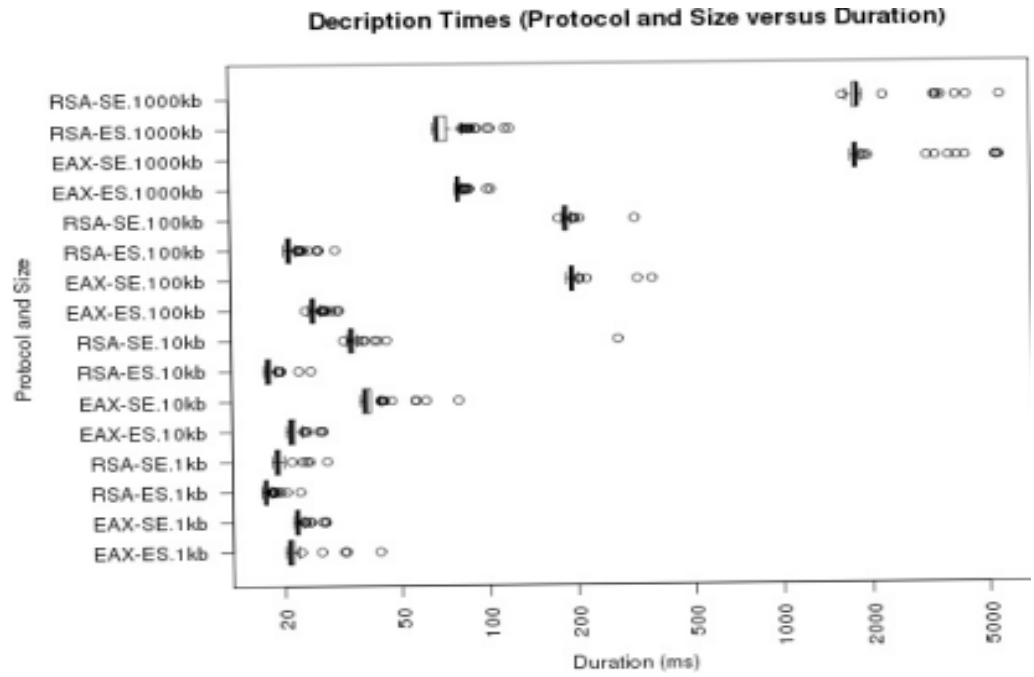

Figure 4. Box plot for decryption

9. Conclusion

In summary, this paper shows an comparison of RSA-AES and AES-EAX PSK cryptography mode of operation for use in SMS-based secure transmissions. Considering the security guarantees and the speed of encryption and decryption, under the same hardware and software platform, which are suitable for some M2M applications. This study pointed out the AES-EAX PSK mechanism and the RSA-AES mechanism have the same security guarantees but AES-EAX PSK mechanism performed better, and was more instantaneous. It was noted that despite the non-inclusion of the digital signatures on the RSA-AES mechanism, the AEAD mechanism still performed better. This paper also indicated that applying encryption before splitting the data is better than encryption after.

References

[1] SMS (Short Message Service) http://www.gsmworld.com/yechnology/sms

[2] Information Technology-Security Techniques-Code of Practice for Information Security Management (2005), geneva.

[3] SMS is top service for Asian mobile phone users (March 2006).

[4] Botan cryptography library (2010), http://botan.randombit.net/

[5] Python Programming Language (2010), http://www.python.org/

[6] 3rd Generation Partnership Project. GSM 03.40. Digital cellular telecommunications system (Phase 2+). Technical realization of the Short Message Service (SMS) (2001)

[7] Abidalrahman Moh'd, Nauman Aslam, H.M.L.T. (2010). Hardware Implementations of Secure Hashing Functions on FPGAs for WSNs, *Journal of Networking Technology* 1(1), 1–43 (March).

[8] Barkan, E., Eli, B. (2005). Conditional Estimators: An Effective Attack on A5/1 p. 1–19.

[9] Bellare, M., Namprempre, C. (2008): Authenticated encryption: Relations among notions and analysis of the generic composition paradigm, *Journal of Cryptology* 21(4) 469–491.

[10] Bellare, M., Rogaway, P., Wagner, D. (2004). The EAX mode of operation. *In*: Fast Software Encryption. p. 389–407. Springer.

[11] Bellare, M., N.C. (2008). Authenticated encryption. Relations among notions and analysis of the generic composition paradigm. Journal of Cryptology 21(4) 469–491.

[12] Biham, E., Orr, D. (2000). Cryptanalysis of the A5/1 GSM Stream Cipher p. 43–51.

[13] Biham, E., Orr, D. (2000). Cryptanalysis of the A5/1 GSM Stream Cipher. Indocrypt.

[14] Biryukov, A., Adi, S., Wagner, D. (2000). Real Time Cryptanalysis of A5/1 on a PC, encryption-FSE: 1-18.

[15] Dye, M.S. End-to-End M2M (Sample/Excerpts Copy only - Not Full Report).

[16] Ekdahl, P., Thomas, J. (2003). Another attack on A5/1. IEEE Transactions On Information Theory 49(1) 284–289.

[17] Elad, B., Biham, E., Keller, N. (2006). Instant Ciphertext-Only Cryptanalysis of GSM Encrypted Communication by Barkan and Biham of Technion (Full Version).

[18] Elgamal, T., Hickman, K. (1997). Secure socket layer application program apparatus and method (Aug 12), uS Patent 5,657,390.

[19] Lo, J., Binshop, J., Eloff, J. (2008). SMSSec: an end-to-end protocol for secure SMS. Computers and Security 27(5–6), 154–167.

[20] LORD, S. (2003). Trouble at Telco: When GSM Goes Bad , issue 1, 10-12.

[21] Lucero, S. (2010). Maximizing Mobile Operator Opportunities in M2M.

[22] Patrik, E., Johansson, T. (2003). Another attack on A5/1. *IEEE Transactions on Information Theory* 49 (1) 284-89.

[23] Rivest, R., Shamir, A., Adleman, L. (1978). A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM 21.

[24] Schmidt, M. (2001). Consistent M-Commerce Security on Top of GSM-based Data Protocols-A security Analysis.

[25] Yu, W. New Botan, the C++ Crypto Library, built for Fedora 13 with Python Bindings Enabled and the RSA-PrivateKey fix (August 2010), http://hip2b2.yutivo.org/2010/08/23/botan-patch/

[26] Yu, W., Tagle, P. (2010). Development of an Over-the-Top Network Protocol for Pervasive, Secure and Reliable Data Transmission over GSM Short Messaging Service. *In*: To be presented at the 2010 *International Conference on Computer and Software Modeling* (ICCSM 2010). p. 1–7. IACSIT.