2013 International Conference on Advances in ICT for Emerging Regions (ICTer): 080 - 083

Extensive Compression of Text Messages in Interactive Mobile Communication

H.K. Salinda Premadasa[1], R.G.N. Meegama[2]

[1]*Centre for Computer Studies, Sabaragamuwa University of Sri Lanka*
Belihuloya, Sri Lanka
salinda@sab.ac.lk

[2]*Department of Statistics and Computer Science, Faculty of Applied Sciences, University of Sri Jayewardenepura*
Gangodawila, Nugegoda, Sri Lanka
rgn@sci.sjp.ac.lk

***Abstract*— An extensive text message increases the message payload size causing a standard text message to be stripped into several concatenated message. Sending such concatenated messages to transmit a single message is a major drawback considering the cost associated in sending out multiple messages. In this paper, we propose an extensive text message compression technique having message confidentiality, authenticity and integrity with cryptographic protection to enhance security. Initially, the extensive text message is compressed using the MD5 algorithm into a cipher text that consists of 32 characters. The encryption is performed by using an initialization vector and a secret key while extensive message compression is achieved simultaneously. Finally, this cipher-text is transmitted through the SMS gateway to the recipients who will decompress the cipher-texts into its original form. Results indicate that message delivery time is not affected by the proposed mechanism.**

***Keywords*—SMS, concatenated message, cryptography, message compression**

I. Introduction

Short Messaging Service (SMS) and Multimedia Message Service (MMS) have become perhaps the most popular method of communication in during last decade. In mobile communication, when it is required to send a long text message having more than 160 characters (often referred to as an extensive message), the message is transmitted in a series of concatenated multiple messages. When sending such extensive message, it is automatically segmented into the multiple messages and the receiving handset is then responsible for reassembling the segmented messages into a single message and presenting it to the user as single extensive message. Theoretically, an extensive message can be segmented up to 255 segments, conversely, 6-10 segment messages being the upper limit in practice. Hence, a user needs to pay to the service provider for the number of concatenated messages used in sending out an extensive message [1,2].

The proposed mechanism integrates a secure and open source messaging system into a learning management system (LMS) by compressing extensive text message with a secure encryption. At this point, we use MD5 fixed length message compression technique and Advanced Encryption Standard (AES) for message encryption (combination of initial vector and secret key). With the proposed mechanism, an extensive message can be compressed into a single standard text message and assurance can be given for protection from modification, eavesdropping and man-in-the-middle attacks with the AES encryption. Extensive text message compressio is accommodated to avoid a major drawback, such a concatenated messaging, in the mobile learning environment This assists us to minimize the cost associated with sendin bulk messages. Moreover, encryption facilitate confidentiality and integrity on such messaging system t transmit sensitive data.

To implement the proposed mechanism, we use an SM gateway, SMS daemon, LMS integrated database and combination of JAVA and PHP technologies. The reason fo using such emerging technologies is the facility offered t extensive message compression with a cryptographi protection and the ability to integrate this mechanism into widely used LMS such as Moodle.

II. Theoretical Framework

Compressing data allows us to reduce communication cost by using available bandwidth effectively. Data compressio algorithms can be divided into two main categories, namel Lossless algorithms and Lossy algorithms. We distinguis between Lossless algorithms, which can reconstruct th original message exactly from the compressed message, an Lossy algorithms, which can only reconstruct a approximation of the original message. Such Lossles algorithms are used for all kinds of text, scientific an statistical databases, medical and biological images wher recovery is needed to be precise. The Lossy compressio technique, on the other hand, can be used in normal imag compression and multimedia compression where a mode loss in resolution is often undetectable or at least acceptabl [3]. The message encryption can be taken into considerati as an added value in the message compression mechanism t transmit sensitive data. Message confidentiality can b retained with the encryption technique but other technique are also essential to protect the message integrity an authenticity [4]. Private-key and public-key encryption are th most popular encryption techniques used at present. Whe choosing encryption techniques, factors such as securit energy consumption, speed, etc have to be considered However, simultaneous implementation of a messag compression technique and security is a challenging task i SMS technology [5].

A. Advanced Encryption Standard (AES)

With the AES symmetric block cipher algorithm, the tex message (plain-text) is converted into an unintelligible form

(i cipher-text) and cipher-text is converted into its original form f the message with decryption. The AES algorithm is capable f using cryptographic keys of 128, 192, and 256 bits to crypt and decrypt data in blocks of 128 bits [6]. AES has a xed block size of 128-bits and a key size of 128, 192, or 256-its. But the Rijndael can be specified with any key and block zes in a multiple of 32-bits, with a minimum of 128-bits and maximum of 256-bits. In a JAVA platform, libraries are sed to implement AES-128 bits key for encryption. Hence, e Rijndael-128 bit algorithm is most suitable for encryption nd decryption in a JAVA environment [7].

The Initialization Vector (IV) is a random number used in ombination with a Secret Key (S_k) to encrypt data. This mber is sometimes referred to as a "number occurring once" an encryption program uses it only once per session. The ndom IV is used to avoid repetitions during message cryption and makes it impossible for intruders who use ctionary attack to decrypt the exchanged encrypted message v discovering a pattern. The point is that the attacker does ot know what the IV is and hence, the attacker must compute very possible IV for a given cipher-text to find out the iginal plain-text. This IV is not being considered as secret d it can be transmitted in plaintext with the message. owever, the secret key must be kept confidential from authorized users. This would not be a crisis because the IV anges randomly at each session and S_k is held as a private y [8].

The AES-Rijndael 128 bit algorithm can be used in the pher Block Chaining (CBC) mode with the combination of itial vector and secret key in a PHP environment [9]. The cryption of plain-text and decryption of cipher-text with tialization vector and secret key by the block cipher in the BC mode is described in Figure 1 [10].

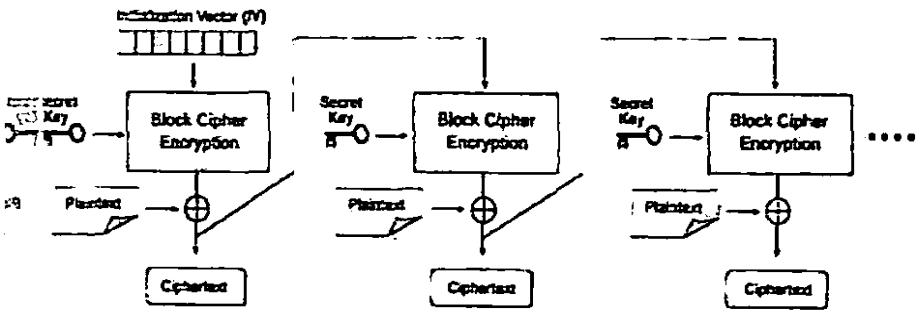

Fig 1. Message encryption with initialization vector and secret key in CBC mode

Message Digest-5 algorithm (MD5)

The MD5 hash algorithm is much more secure as well as a faster than the previous MD4 algorithm [11]. This MD5 been the most widely used cryptographic hashing gorithm with message authentication and integrity capabilities. A text message with an arbitrary character length can be taken as the input to the algorithm which produces a 128-bit "fingerprint" or "message digests" as an output. The MD5 hash is typically expressed in 32-digits hexadecimal constant number. A completely different message digest output is obtained when we consider two messages which differ only by a single character. Hence, the MD5 hash function is a powerful mechanism for detecting very small changes in very large files or messages and can be used to compress as a fixed length message securely [12].

New cryptographic attacks such as brute force, rainbow tables, dictionary attacks and hybrid attacks have been discovered in MD5 vulnerabilities. Hence, in some cases, such as Secure Socket Layer (SSL) certificates or digital signatures, the MD5 is not suitable for message compression. However, it is often used in digital signatures with many variations. The combination of MD5 hash and the (IV can be used to avoid a dictionary attack [13].

In contrast, the MD5 cryptographic hashing algorithm is used in the proposed mechanism to perform extensive message compression. Further, the IV is combined with the MD5 hash algorithm to provide cryptographic protection while obtaining message compression to avoid a dictionary attack in MD5 vulnerabilities. An arbitrary number of characters contained in the extensive text message are compressed using the MD5 algorithm into a 32-digit hexadecimal fixed length text. This fixed length text can be inserted into the OUTBOX table in the LMS database as a standard single text message. This allows us to mitigate the costs involved in sending out concatenated messages. PHP and JAVA languages are used in the implementation phase at the teacher's end (LMS server) and the student's end (mobile device), respectively, while the operating system of the mobile device is Android. Also, the Rijndael AES-128 bit algorithm with secret key performs well in the encryption and decryption processes in PHP and JAVA environments. The secret key can be exchanged between a teacher and a student securely. We can denote the cipher text M (containing 32 digits)

$$M=\{H_{MD5}(IV) \oplus S_k(T_p)\}$$

Where T_p = plain-text, IV = initialization vector, $\oplus$ = XOR operator, S_k = secret key and H_{MD5} = hashing algorithm for message compression. Finally, this encryption provides message authenticity, integrity and confidentiality for the proposed mechanism as an added value.

III. System Function and Architecture

The theoretical concept behind this system functionality is built upon the convention of extensive text message compression into a single standard text message with a cryptographic protection. This enables us to avoid concatenation of several messages by replacing it with extensive text message compression technique to generate a standard single text message with security assurance.

As illustrated in Figure 2, the proposed mechanism enables a teacher to create extensive text messages containing arbitrary number of characters via a mobile browser interface through an Internet-enabled mobile device. Once a teacher logs in to the system relevant to a particular course unit, he/she can create an extensive text message and send brief lecture summaries, assessment schedules or feedbacks, detailed news and notices to students as wells as confidential data such as exam results.

When a teacher creates and ready to insert an extensive text, the front-end software compresses this extensive text by using the MD5 hash algorithm with an initialization vector (IV) and a secret key (S_k) into a 32-digit fixed length payload (cipher-text). The IV is randomly defined in each session and wraps it with plain-text, encrypt with S_k (key should enter by the teacher while creating extensive text) and then compresses with MD5 hash algorithm into a cipher-text.

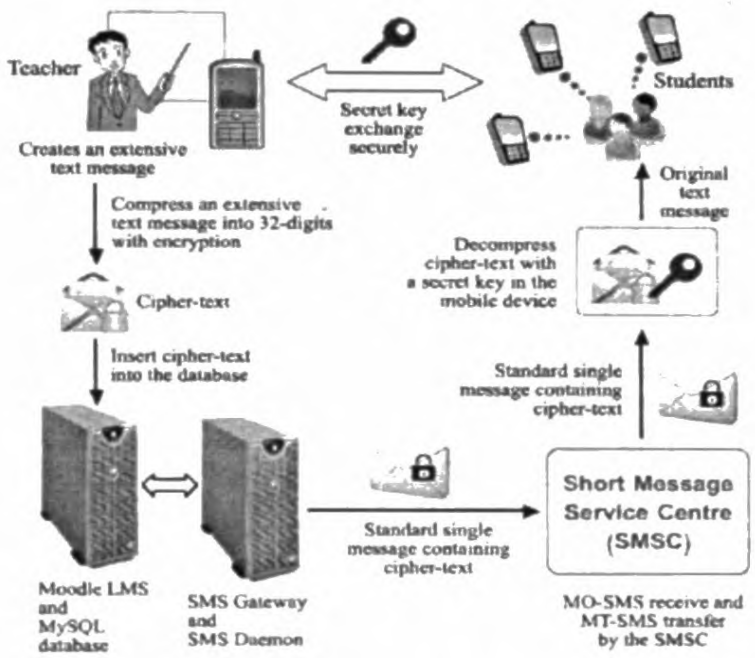

Fig 2. System function and architecture

Thereafter, this cipher-text is inserted into a database where an SMS daemon checks the database periodically and sends out stored messages through an SMS gateway to a GSM modem. Using 'AT' commands, the modem in turn sends out these cipher text as a standard single text messages to all the recipients having access to that particular course unit through the SMSC. Finally, this cipher-text is received by the recipient's mobile device in which the software decompresses that cipher text using S_k (key should receive by the student from the teacher securely) allowing the message to be viewed in the standard INBOX.

The front-end design of the system provides the teacher the ability to select a single or multi-recipient phone numbers in addition to students' mobile phones numbers previously added. In order to mitigate system abuse, restrictions can be enforced to limit the number of messages that can be sent by each teacher for a given month. System logs allow a teacher to view delivery information such as recipient phone numbers, delivered times, message contents about each text message.

Front-end design of the teacher's end, as given in Figure 3, is implemented by using the combination of PHP, XHTML-MP and CSS and the recipient's end is implemented by using JAVA on the Android platform.

Fig 3. Front-end interfaces of the proposed system of the mobile phone with, (a) teacher inserts extensive text message with secret key, (b) student receives message and inserts secret-key, (c) student views original message

IV. Results and Discussion

The proposed algorithm is tested with 100 messages sent by the system to a series of smart phones on Android platform having connectivity to several GSM networks (GSM provider I – 36 SMSs, GSM provider II – 26 SMSs, GSM provider III – 23 SMSs and GSM provider IV – 15 SMSs). Compressed text messages (as cipher-texts) are inserted into the database with 32-digit fixed length size. The entire process of message delivery contains the combination of two partial processes namely from submitting the message via mobile device by the teacher to inserting into the database and transmitting the message to the recipient. The message compression with encryption carried out in the first stage of the entire process.

Initially, different character lengths of extensive messages (i.e. 144, 221, 372, 466 and 637) were submitted by the teacher to evaluate the response time. The compression ratio (CR) can be measured by:

$$CR = Compressed\ file\ size / Source\ file\ size$$

A comparison of compression ratios with proposed mechanism is illustrated in Table I.

TABLE I
COMPRESSION RATIOS WITH THE PROPOSED MECHANISM

Reference payload size (characters)	Compressed payload size (characters)		Compression ratio
144 (1 message)	32	Required only single standard message	0.22
221(2 messages)	32		0.14
372 (3 messages)	32		0.09
466 (4 messages)	32		0.07
637 (5 messages)	32		0.05

The response time obtained in the proposed mechanism with extensive messages having an average character length of 368 is illustrated in Figure 4. The mean values of the response time in the proposed mechanism and the standard mechanism were measured to be 3.129 and 3.040 seconds, respectively.

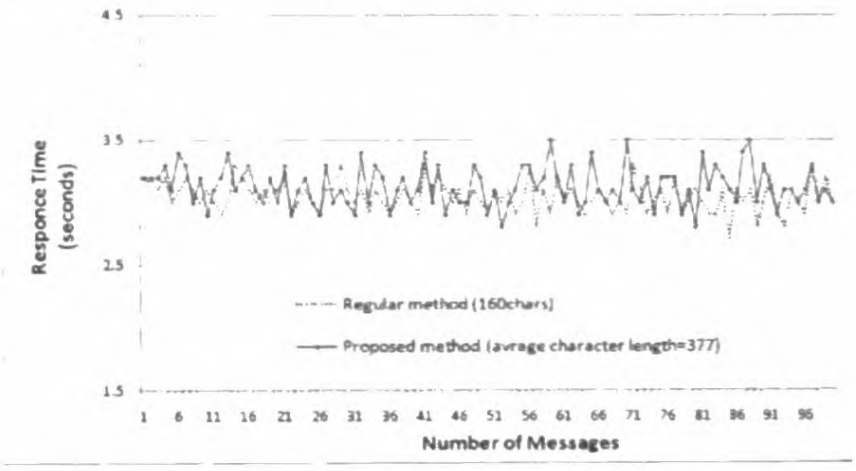

Fig 4. Response time for proposed and regular method

A paired sample t-test (parametric) was applied after defining a null hypothesis (H_0) to compare these two response times. H_0 was accepted by considering the p-value of the test (i.e. if p-value<0.05 then reject H_0 or accept otherwise). The null hypothesis is defined as

H_0 = the proposed mechanism does not influence message delivery time

Table II gives statistical results obtained after applying the paired sample t-test. It shows that the statistical difference between the two approaches and the p-value was 0.99 (> 0.05). This result provides evidence for accepting the null hypothesis within the significant level. In the second stage of the process, all compressed text messages were delivered after

inserting them into the database continuously within 329 seconds through the SMS gateway without any significant time variations and interruptions. This partial process consumed the same time intervals for both the standard mechanism as well as the proposed mechanism.

TABLE II

RESULTS OF PAIRED SAMPLE T-TEST FOR PROPOSED MECHANISM

	N	*Mean(s)*	*StDev(s)*
Standard mechanism	100	3.04000	0.12309
Proposed mechanism	100	3.12900	0.16224
Difference	100	-0.089000	0.17860
p-value		0.99	

Hence, it can be seen that the time is independent of both systems in this second stage of the entire process. Finally, the results indicate that the proposed mechanism does not influence message delivery time (H_0 *accepted*). This confirms that the proposed mechanism is a competent short messaging service for communicating text messages with compression. Hence, the proposed mechanism, when implemented in an J2ME/Android environment, enables us to perform academic and administrative activities without worrying the cost per message. Moreover, the proposed mechanism offers security during transmission of sensitive data. Statistical data confirms the extensive usage of mobile phones for texting [14] in J2ME/Android environments [15].

V. CONCLUSION

The main advantage of the proposed technique is that it decreases the message payload size into a single standard message while at the same time, providing message confidentiality, integrity and authenticity for both the sender and the receiver. This technique can be easily integrated into a J2ME SMS and can be implemented in the Android operating system at the recipient's end. Hence the proposed mechanism will certainly provide an ideal opportunity not only in the education sector but also in other entities where extensive text messaging to a larger user base is required. Further work is required to apply the proposed mechanism in other mobile operating systems and services.

REFERENCES

[1] [1] Wikipedia. (2008) Short Message Service. [Online]. http://en.wikipedia.org/wiki/Short_Message_Service

[2] [2] M Ryan and McMinnville, System and Method for Electronic Messaging with Group Sending, Receiving and Replying Capabilities, Jul 2011, United States Patent Application Publication, Pub no: US 2011/0165895 A1.

[3] [3] A S Mohammad-Arif, A Mahamud, and R Islam, "An Enhanced Static Data Compression Sceme of Bengali Short Message," *International journal of Computer Science and Information Security*, vol. 4, no. 1, pp. 97-103, 2009.

[4] [4] A K Nanda and L K Awasthi, "SMS Security Using NTRU Cryptosystem for M-Commerce," in *Institute for Development & Research in Banking Technology*, India, 2011.

[5] [5] Wikipedia. (2013, March) Encryption. [Online]. http://en.wikipedia.org/wiki/Encryption

[6] [6] E Conrad. (2001) Advanced Encryption Standars. [Online]. http://www.giac.org/cissp-papers/42.pdf

[7] K Singh, S Maheshwari, S Verma, and R Dekar, "Peer to Peer Secure Communication in Mobile Environment: A Novel Approach," *International Journal of Computer Applications*, vol. 52, no. 9, pp. 24-29, 2012.

[8] Stackoverflow. (2008) Should I use an initialization vector (IV) along with my encryption? [Online]. http://stackoverflow.com/questions/65879/should-i-use-an-initialization-vector-iv-along-with-my-encryption

[9] Wikipedia. (2013, March) Block cipher modes of operation. [Online]. http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation#Cipher-block_chaining_.28CBC.29

[10] V Klima and T Rosa, "Strengthened Encryption in the CBC Mode," *Cryptology ePrint Archive: Report 2002*, vol. 61, 2002.

[11] R Rivest, "The MD5 Message-Digest Algorithm," in *MIT Laboratory for Computer Science and RSA Data Security, Inc.*, Cambridge, April 1992.

[12] J Deepakumara, H M Heys, and R Venkatesan, "FPGA implementation of MD5 hash algorithm ," in *Electrical and Computer Engineering, 2001. Canadian Conference on*, vol. 2, Toronto, 2001, pp. 919 - 924.

[13] R Grundmanis, "How to crack MD5 algorithm using advanced brute force ," in *Applied Information and Communication Technologies*, Latvia, 2010, pp. 256-262.

[14] Nielsen. (2012, April) The Digital Revolution. [Online]. http://na.ad-tech.com/sf/wp-content/uploads/DigitalConsumer.pdf

[15] S Abhishek. (2013, January) Worldwide Smartphone OS Market Share And Penetration 2012. [Online]. http://www.dazeinfo.com/2013/01/07/worldwide-smartphone-os-market-share-penetration-2012/