Secure SMS Communication Based on Quasigroup Transformations

Smile Markovski[1], Aleksandra Kuzmanovska[2] and Milivoj Simeonovski[3]

[1] Institute of Informatics, Faculty of Natural Sciences and Mathematics, UKIM-Skopje
[2] EOS Matrix DOO, Leninova 1 Gradski Stadion, 1000 Skopje
[3] NLB Tutunska Banka, Vodnjanska 1, 1000 Skopje
smile@ii.edu.mk,a.kuzmanovska@eos-matrix.com,milivojs@gmail.com

Abstract. By its nature, the SMS communication is insecure and the message information can be viewed of many interested parties. Here we propose a protocol for secure SMS end-to-end communication between two mobile devices. Our protocol is based on a symmetric encryption of the message context. The encryption is defined by using quasigroup transformations. We have considered several possible attacks on our secure protocol and we could conclude that it is resistant of them.

Keywords: Mobile communication, Secure SMS, Quasigroups, Quasigroup transformation

1 Introduction

Short Messaging Service (SMS) is a communication service, originally developed as part of the Global System for Mobile Communications (GSM). Today it is one of the most widely used mobile services, with million messages exchanged on a daily basis.

Their present-day uses are far different from the initial idea. SMS has now become a popular mean of communication by individuals and businesses. Banks worldwide are using SMS to conduct some of their banking services. People sometimes exchange confidential information such as passwords or sensitive data amongst each other. The mobile commerce is everyday growing.

SMS provides many conveniences in our everyday lives, but is it really secure?

SMS messages are sent via a store-and-forward mechanism to a Short Message Service Centre (SMSC), which will attempt to send the message to the recipient and possibly retry if the user is not reachable at a given moment. Transmission of the short messages between SMSC and phone is via the Signalling System Number 7 (SS7) within the unencrypted protocol allowing employees within the cellular provider's network to eavesdrop or modify SMS messages.

To protect our privacy, as well as to protect our confidential data, we propose a protocol that SMS messages from source to destination will be sent encrypted. If someone sniff the channel or wants to read the contents of the SMS messages directly from the base, he can only see the text with meaningless incomprehensible characters.

Most manufacturers of mobile phones include Java into their platform that can be used for software development. This enables portability of Java applications between devices from different manufacturers. Unfortunately, Java's Wireless Messaging API (WMA),

package that provides platform-independent access to wireless communication resources like SMS, does not support any security for SMS communication. These facts make the Java programming language a natural choice for implementation of our protocol.

In our protocol, a secure channel is established between two network-connectable mobile devices, using SMS as the transmission medium. To ensure that the SMS remains confidential, a symmetric-based cipher is used to encrypt the message's content. The encryption is defined by using quasigroup transformations of the messages. We emphasise that our paper is an extension of the ideas given in Hassinen and Markovski's paper [1]. Here we developed complete secure protocol that work autonomously and allows arbitrary many users. The communication is establishing online, provided that the users have installed the needed components of our protocol.

2 Quasigroup Transformations

A *quasigroup* $(Q,*)$ is a groupoid satisfying the law:

$$(\forall u,v\in Q)(\exists! x,y\in Q)(u*x=v \,\&\, y*u=v) \tag{1}$$

This implies that any quasigroup satisfies the cancelation law and the equations $x * a = b$ and $a * y = b$ have unique solutions x and y.

Given a quasigroup operation $*$, two new operation $\backslash$ and / can be defined by

$$x*y=z \Leftrightarrow y=x\backslash z \Leftrightarrow x=z/y \tag{2}$$

and then the algebra $(Q,*,\backslash,/)$ satisfy the laws

$$\begin{aligned} &x\backslash(x*y)=y, \qquad (x*y)/y=x,\\ &x*(x\backslash y)=y, \qquad (x/y)*y=x \end{aligned} \tag{3}$$

Let Q be a finite set, i.e., an alphabet, and let $*$ be a quasigroup operation on Q. Denote by $Q^+=\{x_1x_2...x_t \mid x_i \in Q,\ t\geq 2\}$ the set of all finite strings over Q. For a fixed $l \in Q$, called leader, the quasigroup string transformations of type e and d are defined in [2] as follows:

$$\begin{aligned} &e_{*,l}(x_1,x_2,...,x_n)=y_1y_2\ldots y_n \Leftrightarrow y_{i+1}=y_i*x_{i+1},\\ &e'_{*,l}(x_1,x_2,...,x_n)=y_1y_2\ldots y_n \Leftrightarrow y_{i+1}=x_{i+1}*y_i,\\ &d_{*,l}(x_1,x_2,...,x_n)=y_1y_2\ldots y_n \Leftrightarrow y_{i+1}=x_i*y_{i+1},\\ &d'_{*,l}(x_1,x_2,...,x_n)=y_1y_2\ldots y_n \Leftrightarrow y_{i+1}=x_{i+1}*x_i, \end{aligned} \tag{4}$$

for each $i = 0, 1, ..., n-1$, where $y_0 = l$.

By using compositions of e- or d-transformations with given leaders $l_1, l_2, \ldots, l_s \in Q$ and given quasigroup operations $*_1, *_2, \ldots, *_s$, new composite E, E' and D, D' transformations can be defined by

$$\begin{aligned} &E=E_l=e_{*_1,l_1}\circ e_{*_2,l_2}\circ\ldots\circ e_{*_s,l_s},\quad D=D_l=d_{*_1,l_1}\circ d_{*_2,l_2}\circ\ldots\circ d_{*_s,l_s}\\ &E'=E'_l=e'_{*_1,l_1}\circ e'_{*_2,l_2}\circ\ldots\circ e'_{*_s,l_s},\quad D'=D'_l=d'_{*_1,l_1}\circ d'_{*_2,l_2}\circ\ldots\circ d'_{*_s,l_s} \end{aligned} \tag{5}$$

Note that $(e_{*_1,l_1} \circ e_{*_2,l_2} \circ \ldots \circ e_{*_s,l_s}) \circ (d_{*_s,l_s} \circ d_{*_{s-1},l_{s-1}} \circ \ldots \circ d_{*_1,l_1})$ is the identity mapping on Q^+, and for each of the above transformations there exists an inverse one. So, the following property is true.

Proposition 1 The transformations E, E', D and D' are permutations on Q^+.

We note that some quasigroups produce exponential increasing of the periods, and they are called exponential quasigroups. By Proposition 1, the transformations E, E', D and D' can be used suitable encryption and decryption functions to be defined. In our secure SMS protocol we are using these types of transformations for encrypting and decrypting purposes.

2.1 Encryption and decryption functions

We use quasigroups of order 4, i.e., $Q = \{0, 1, 2, 3\}$. There are 576 quasigroup operations on Q, but we are using only 192 of them, that are exponential quasigroups (sometimes called non-fractal quasigroups [6]). The algorithm, according to the key, chooses 16 quasigroup operations and 16 leaders, and the encryption and decryption of the messages is produced by them.

Note that there are $192^{16} > 2^{121}$ possible 16-tuple of quasigroup operations and $4^{16} = 2^{32}$ possible 16-tuples of leaders. Altogether, there are more than 2^{153} possible choices of operations and leaders.

Let $*_0$, $*_1$,…, $*_{15}$ be the chosen operations and let l_0, l_1, …, l_{15} be the chosen leaders. Our encryption function is defined by

$$F_E = e_{*_0,l_0} \circ e'_{*_1,l_1} \circ e_{*_2,l_2} \circ e'_{*_3,l_3} \circ \ldots \circ e_{*_{14},l_{14}} \circ e'_{*_{15},l_{15}} \quad (6)$$

and the decryption function by

$$F_D = d'_{*_{15},l_{15}} \circ d_{*_{14},l_{14}} \circ d'_{*_{13},l_{13}} \circ d_{*_{12},l_{12}} \circ \ldots \circ d'_{*_1,l_4} \circ d_{*_0,l_0} \quad (7)$$

On Fig. 1 the encryption and decryption process of message $m = m_0m_1m_2\ldots$ by the transformation e_{*_1,l_1} are illustrated:

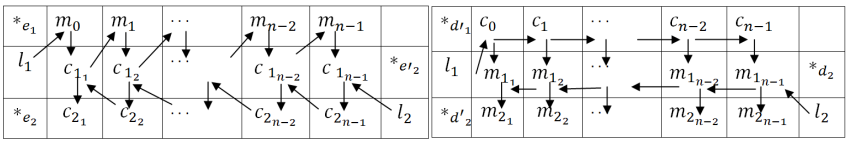

Fig. 1. Encryption and Decryption process

3 Protocol for Secure SMS Communication

The protocol for secure SMS communication is based on the quasigroup transformation algorithm for design of an encryption and a decryption function, as defined in Section 2.

Quasigroups and leaders are basic elements that should be provided on both sides for algorithm to be used. The security of protocol needs the following two types of keys to be used:

- Initial Session Key, used two parties to establish initial secure communication
- Working Session Key, used for sending and receiving secure SMS messages

For any two users, the protocol creates pairs of their phone numbers and the secret working session keys. Each user has own password and it is used for access to the protocol data stored into user mobile phone, allowing security if the mobile device is misused by an eavesdropper.

3.1 Initial Session Key

Initial Session Key is in fact a secret Initial Vector (*IV*) used for starting the secure communication between two parties. One of the users, after agreement of establishment of secure communication between them, sends the IV to the other one. The IV is an SMS message with a minimum length of 20 bytes. It is transferred to the other party by using strong Diffie-Hellman algorithm with Digital Signature (D-H DSA).

Both sides, by using this *IV*, allocate 16 quasigroups and 16 leaders. For quasigroups, the first 16 bytes of *IV* are used. The last 4 bytes of *IV* are used leaders to be defined. Since 192 quasigroups are on disposal, the first 16 bytes are taken modulo 192 and they define which quasigroup can be used as first, second, third,… in the transformation algorithm. From the last four bytes, we generate in reverse order 16 nibbles for leaders. The quasigroups and leaders are stored in phone memory.

Exchanged *IV* will be stored in phone memory on both sides as follows:
User *A* stores a pair of the phone number of user *B* and *IV*, and user *B* stores a pair of the phone number of user *A* and *IV*. This *IV* will be used only the first sent message from *A* to *B* (or from *B* to *A*) to be encrypted, i.e., decrypted. After that, the *IV* will be replaced by the working session key and the allocated quasigroups and leaders will be erased.

3.2 Working Session Key

After first successfully exchanged message, the new session key, called a working session key is generated from the last exchanged encrypted message. The generation of a working session key is as follows.

Let the user *A* sent an encrypted message to user *B*. Then the working session key for *A* will be generated by the protocol, immediately after sending the encrypted message. The protocol will generated the working session key for *B* immediately after decrypting of the received encrypted message from A. The original SMS message, after padding, is transformed by using an *E*-transformation defined by the allocated 16 quasigroups and 16 leaders, stored in the phone memory of both users, taken in the reverse order. From such transformed string, the first 16 and the last 4 bytes are used new quasigroups and leaders to be allocated, as it was explained in Subsection 3.1.

We need to pad the original messages, since it can happens the messages to be shorter than 20 chars, and in the same time we increase the security of the protocol. The original SMS message will be padded up to 160 chars. The padding is done such that the original message of $n < 160$ chars is concatenated with the char of the ASCII code 199, and after

that with 159 – *n* randomly generated chars. The user *A* will send to *B* the padded message of 160 chars. After decrypting, the user *B* will not see the random chars, because the application will not display them. (We choose the ASCII code 199 because it can not be typed from the mobile phone keypad, so the user *A* cannot type it by accident.)

After completion of this phase, both users have a new temporary working session key, and it will be changed when the next message will be exchanged between them. This working session key will be used for allocation of the necessary quasigroups and leaders for encryption/decryption of the next message only.

3.3 Authenticated access

We introduce password in our protocol to provide protection for all secret data in the phone memory from unauthorized use. The password is stored in the phone memory as hash value.

Before any secure communication the password should be entered. Three unsuccessfully hitting of password will produce deleting of all components of the secure protocol.

In the case of deleting of the protocol components of user *A*, the user *A* should upload the needed algorithms in his/her mobile device, and after that to start again by sending *IV* to user *B*. The protocol is made in such a way to check if the phone memory of user *B* contains the pair of the phone number of *A* and a working session key. If that is a case, the working session key of *A* will be changed with the new initial session key.

4 Attacks of our crypto system

Our crypto system for secure SMS messages contains three types of secret keys: *IV*, working session key and a password. The *IV* is used only ones, the password is permanent, while the working session key is temporary and it is changed after each exchanged message. Each user chooses its own password and the attack on the password is protected by allowing three unsuccessfully hitting only. After that all protocol components will be erased and the phone device will be not available for our secure communication, until reloading of the components is realized.

The initial session key, *IV*, is most vulnerable and special attention for its security is given. It should be exchanged between two parties by public channel and that is why the strong Diffie-Helman Algorithm with Digital Signature Algorithm is used. Hence, the security of *IV* exchange is based on the security of D-H DSA. So, we have security of men in the middle attacks on *IV*. The working session keys are generated from encrypted messages that are also sent through insecure public channel. To generate the working session key, the attacker has to know the previously generated working session key, since at first he/she should decrypt the encrypted message. This means that the attacker should know the *IV*.

The length of our session keys is 20 bytes, so there are 2^{80} possible session keys. Since there are more than 2^{153} possible choices of 16 quasigroups (out of 192) and 16 leaders, the best brut force attack is on the session keys. We find that 2^{80} possible session keys are enough to achieve a security against the brut force attacks. We have chosen 192 exponential quasigroups to prevent statistical kinds of attacks. Quasigroup string transformations defined by these quasigroups produce almost random sequences, with uniform distribution

of the letters, pair of letters, …, up to 16-tuples of letters, and with periods that are growing exponentially. We pad the SMS messages to 160 chars, the maximal allowed length of an SMS message, in order to achieve protection of statistical attacks as well. Namely, from longer strings better pseudo random strings are produced.

The definitions of the encryption and the decryption functions F_E and F_D by (7) and (8) spread each bit information from the input string to every bit of the output string. We find that in such a way a protection against linear and differential attacks is achieved.

All secret data of our protocol are stored in the user's phone memory. If an attacker somehow can get the data from the phone memory of a user *A*, then the secret SMS messages of *A* (sent or received) are broken. That will not affect the security of the other users of our secure protocol (except of the communication with *A*). This means that our secure protocol can be only locally broken.

5 Conclusion

With this implementation of quasigroup transformation for encrypt and decrypt SMS messages, we introduce a new protocol of secure mobile communication. The SMS communication is widely used and it is a service that can be used in a combination with other services. Services like mobile payment, identity checks, digital signatures, one-time passwords, etc. can be implemented using SMS services in combination with our protocol.

We have made a real application that used this protocol for exchanging SMS messages. An application using our protocol is working in real time on standard mobile devices. It can be easily upgraded to become a protocol that can offer other mobile services based on SMS messages.

Our protocol is resistant to the main kinds of the attacks to secure mobile communications. Its security is based on the lower local level, i.e., on the security access of owner's phone memory. Even in the case of breakage of one of the users, the others can still communicate securely between themselves.

References

1. Hassinen, M., Markovski, S.: Secure SMS messaging using Quasigroup encryption and Java SMS API. SPLST'03, Kuopio, Finland, p.187 (2003)
2. Markovski S., Gligoroski D., Andova S.: Using Quasigroups for one-one secure encoding. Proc. VIII Conf. Logic and Computer Science "Lira `97", Novi Sad, 157-162 (1997)
3. Markovski S., Gligoroski D., Kocarev Lj.:Unbiased Random Sequences from Quasigroups String Transformations. H. Gilbert and H. Handschuh(Eds.):FES 2005, LNCS 3557, pp. 163-180 (2005)
4. Markovski, S., Gligoroski, D., and Bakeva, V.: Quasigroup String Processing:Part 1. Contributions, Sec. Math. Tech. Sci., MANU, XX 1-2 pp. 13-28 (1999)
5. Markovski, S., Kusakatov, V.: Quasigroup String Processing: Part 2. Contributions, Sec. math. Tech.Sci., MANU, XXI, 1-2 pp.15-32 (2000)
6. Dimitrova, V., Markovski, S.: Classification of quasigroups by image patterns. Proc. of the Fifth International Conference for Informatics and Information Technology, 21-25 Jan. 2007, Bitola, Macedonia, pp. 152 – 160 (2007)