Application-Layer Security Mechanism for M2M communication over SMS

Nenad Gligoric, Tomislav Dmicic, Dejan Drajic, Member, IEEE, Srdjan Krc, Senior Member, IEEE, and Nhon Chu

Abstract — With an increasing number of M2M devices communicating over SMS, the need for security was raised because standard SMS messaging service does not provide any light-weight security mechanisms. In this paper we propose hybrid application-layer security with compression for M2M communication over SMS. The signature is calculated using device IMEI, secret key and payload; thus signature is not always the same, which prevents duplicated/stolen signature and replay attack and it provides a content integrity check. The security is implemented on android powered mobile devices and evaluated against non-secure payload for message response time and bandwidth in live GSM network. Evaluation shows that SMS delivery time is not influenced by this security mechanism. In addition compression can save bandwidth for more verbose and longer payloads.

Keywords — Application layer, Compression, GZIP, MD5, M2M, Security, SMS.

I. INTRODUCTION

According to GSMA, there are 9 billion M2M (machine-to-machine) devices today and there will be 20 billion M2M devices connected to the web by 2020 [1]. These devices require protocols to enable communication and services as well as security mechanisms to secure channels between participants in the communication.

As there is significant number of M2M devices operating over SMS, standardization bodies including ETSI, OMA, IETF, etc. are considering machine-type-communication (MTC) over Short Message Service (SMS). In addition, small amount of traffic and low cost and existing infrastructure are also motivation for leveraging SMS in the M2M context.

Open Mobile Alliance (OMA) has recognized the need for current mobile technology to evolve as follows [2]:

- Technology needs to evolve from traditional mobile devices networks to heterogeneous networks that support both mobile and M2M devices.
- Technology needs to support devices on heterogeneous networks through a Gateway
- Technology needs to support M2M devices as a gateway for other devices
- Support for IP (TCP, UDP) & Non-IP Transport (SMS, USSD, CSD)

In M2M communication it is necessary to employ certain mechanisms to provide minimum protection against spamming, DoS attacks and flooding, as well as authentication mechanism to avoid communication with unauthorized party. With the increasing use of SMS in M2M context, sensitive information should not be sent over regular SMS transport without any security. In this paper, secure SMS transport is considered for M2M communication, which simultaneously enables heterogeneous networks (mobile and sensor networks) to interact utilizing SMS Gateway interface, thus merging the gap between current mobile technologies and embedded low-powered devices.

The paper is structured as follows: previous work is described in section II. Application-layer security mechanism is described in section III. The proposed algorithm is evaluated in section IV. The paper is concluded with section V.

II. PREVIOUS WORK

Available SMS security mechanisms such as Security Parameter Index (SPI) and Ciphering Key Identifier (KIC) [3], are rarely used in today deployments. SPI is identification tag added to the header used in data transport that uniquely identifies the sender for this datagram.

There is also proposal to encrypt the payload with Symmetric Key encryption [4]; but this may introduce the security risk because it is not safe to send a new secret over the air that can be exposed to a third party.

The paper [5] shows that large key size algorithms are not suitable for SMS encryption due to small memory and low computational power of mobile phones.

Encrypted messages are usually larger than the original leading to excessive charges in sending SMS encrypted message. Therefore, compression should also be considered to lower the size of encrypted message.
There are many possible algorithms for compressing the message size: GZIP, Huffman encoding, Arithmetic coding, Lempel–Ziv–Welch (LZW), etc. In [6], compression of SMS using arithmetic coding is proposed with the aim to optimize the maximum character capacity of SMS body. Arithmetic coding converts a data message to a real code number between 0 and 1. It provides nearly optimal data compression, but it requires additional memory space in compressed data to save arithmetic coding probability table for decompressing the compressed-data.

GZIP is based on the Deflate algorithm which is used as a replacement for LZW. Deflate is combination of LZ77 and Huffman coding. As GZIP is widely available and provides efficient compression (by combining the best from other above mentioned algorithms), it is chosen as a compression technique in order to reduce payload size for proposed security mechanism. Different encryption methods have diverse memory and power consumption requirements that may impact SMS service delivery time; so it is essential to use solution that will provide the best relation of security and resource consumption for the certain scenario. In next section, proposal of a hybrid application layer security approach is given.

III. SMS SECURITY METHOD WITH SIZE REDUCTION

The following security approach with size reduction is proposed for M2M communication:

1. Each device can have an initial secret which is known by the devices itself (e.g. IMEI provided by the device manufacturing process), and optionally a group device secret.

2. The signature is calculated as follows:
   \[ S = H(DeviceIMEI + secret + (payload)), \]
   where H is MD5 or SHA1 hashing algorithm. Therefore, the signature is not always the same (prevents duplicated/stolen signature and replay attack) and it provides a content integrity check.

3. Sending device calculates the signature (2) and put it in the SMS and hash it again:
   \[ \text{signature} + \text{(compression}(payload)); \]
   where GZIP is used as a compression algorithm.

4. Upon receiving the full SMS message, the device applies the same signature calculation as (2). Message is authenticated and accepted if signatures (3) & (4) matches. The message signature can be calculated using MD5 or SHA1 hash. As we aim to reduce the payload size, we are using MD5 as it fixed-length hash value is always shorter than SHA1. New cryptographic attacks have discovered MD5 vulnerabilities [7], but as we calculate hash using the device IMEI and secret (which are never sent over the network) and a payload, this is not an issue.

Before payload can be sent over SMS, it must be encoded using Base64, because string representation of GZIP byte array cannot be transferred over 7-bit SMS encoded messages.

The reference payload size is XML formatted message of environmental sensor readings, with shortened tags to lower verbosity (Fig. 1). The spaces between namespace are displayed for readability, but the format of the message used in evaluation was without these inner spaces.

III. SMS SECURITY METHOD WITH SIZE REDUCTION

The reference payload is presented in the figure below.

\[
\text{Fig. 1. The reference payload}
\]

After compression with GZIP and encoding with Base64 the compressed payload is 13.5% smaller than the reference payload (Fig. 2).

\[
\text{Fig. 2. The payload after compression (GZIP) and Base64 encoding}
\]

The message signature calculated from device IMEI, secret and payload is presented in the figure below.

\[
\text{Fig. 3. The example of the hash calculated from the device IMEI, secret and the payload}
\]

The messages from Fig. 1 and Fig. 2 are used for evaluating the proposed mechanism efficiency and results of the evaluation are given in section IV.
the message, other device sends response message (Fig. 4), also signed using previously described mechanism.

`:xml version='1.0' encoding='ISO-8859-1'?>
<response>200 OK</response>

Fig. 4. Response SMS message before compression and encoding

Sending small encrypted payload can result in actual increasing payload size, because signature can be longer than a payload itself. Nevertheless, as each 7-bit encoded SMS is 160 characters long, the compression ratio is considered from this length perspective. This means that for smaller payloads (one SMS) there is no compression gain and no loss as well. But for longer and more verbose payloads then one SMS, compression ratio of proposed mechanism is positive compared to a native payload size (Table 1).

**TABLE 1: THE PAYLOAD SIZE COMPARISON BEFORE AND AFTER COMPRESSION AND SIGNING**

<table>
<thead>
<tr>
<th>SMS</th>
<th>Size [bytes]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference payload</td>
<td>297</td>
</tr>
<tr>
<td>Secure payload compressed and encoded with Base64</td>
<td>257</td>
</tr>
<tr>
<td>Response payload</td>
<td>70</td>
</tr>
<tr>
<td>Secured response payload</td>
<td>144</td>
</tr>
<tr>
<td>Compression ratio for reference payload</td>
<td>0.87</td>
</tr>
<tr>
<td>Compression ratio for response payload</td>
<td>-0.49</td>
</tr>
</tbody>
</table>

**IV. EVALUATION**

In order to evaluate proposed security approach, a large number of SMS messages are sent between two devices. Response times and bandwidth are compared for messages sent using proposed hybrid security approach and for messages sent without employing any security and compression. The methodology used for calculating response time was next: after each response from the device that receives the message the new message is sent. The average response times were 20.5s and 20.3s for secured and regular payload respectively.

The t-test showed no significant statistical difference between two approaches (p = 0.8). Median and standard deviation were similar for both cases (Table 2). This provides evidence to accept the null hypothesis of equal means for both regular and secured payload delivery time.

**TABLE 2: RESPONSE TIME FOR SECURED COMPRESSED AND REGULAR PAYLOAD**

<table>
<thead>
<tr>
<th>PAYLOAD</th>
<th>Average response time [s]</th>
<th>Response time standard dev. [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secure compressed</td>
<td>20.5</td>
<td>15.2</td>
</tr>
<tr>
<td>Regular payload</td>
<td>20.3</td>
<td>10.2</td>
</tr>
</tbody>
</table>

Therefore, it is showed that proposed hybrid security mechanism for SMS is not influencing the delivery time, and that is possible to send longer messages over secured SMS transport without any influence on the efficiency of the SMS service.

In the case of the M2M communication over SMS payload size can vary from few tens to few hundred of bytes depending on the scenario. If device sends SMS messages with payload from Fig. 1 by utilizing proposed mechanism, the other devices that receive payload can authenticate the sender. The message will also be 13.5% shorter, which have no impact on the SMS in this case because both payloads fits in 2 concatenated messages, but for other scenarios this method can save one or more messages depending on the payloads size and content. In spite of already shortened reference payload namespaces (Fig. 1), the GZIP showed very good compression ratio. In addition, as stated earlier, for smaller payloads there is no compression gain, but as one 7-bit encoded short message is 160 characters long, there is no loss of compression as well.

Each phone number (MSISDN) in a network has limited number of SMS messages that can be sent in a certain time period. This limit can be increased by agreeing over SLA with Service Provider that will enable sender to achieve higher SMS traffic rate in a short intervals. To efficiently utilize this high traffic rates, it is suggested to employ the
interface between the SMSC (Short Message Service Center) and the SMS gateway.

This interface is not part of the GSM standard and connections are very loosely protected since the content is delivered not encrypted. The authentication of the gateway is done using a header that contains login and password in plain text format. In addition, many SMSC protocols identify the sender using specific field of the short message. Therefore, MSISDN can be easily spoofed, and since the sender is not authenticated, a message can appear to come from any phone number that an attacker defines. Since there is no authentication, the attacker can also pretend to be the real SMSC by using the SMSC simulator. The proposed lightweight security mechanism can easily be used to identify if the messages is really coming from the MSISDN that is indicated in the message.

V. CONCLUSION

In this paper, the signature of the short message is based on a hash calculated from the device IMEI, its secret and a payload, which increases size of a short message. To avoid this issue, the payload is further compressed with the GZIP and the final output is encoded with Base64 to enable transport of unicode string over 7-bit SMS. The evaluation of the implemented security mechanism in live GSM network between two Android devices showed that proposed secured communication has no influence on SMS service delivery time compared to a regular M2M communication over SMS. In addition, the compression ratio can provide better efficiency by sending shorter payload in a communication where sender can be authenticated.

ACKNOWLEDGMENT

This paper describes work undertaken in the context of the EXALTED project - Expanding LTE for Devices (http://www.ict-exalted.eu). The research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreement n° ICT-5-258512.

REFERENCES