A Message Encryption System Architecture for MeeGo Mobile OS

Anton Ovseenko
State University of
Aerospace Instrumentation
St. Petersburg, Russia
aovseenko@vu.spb.ru

Vitaly Petrov
Tampere University
of Technology
Tampere, Finland
vitaly.petrov@tut.fi

Abstract

In this paper the issues of existing security mechanisms of mobile operation systems are highlighted. The architecture of message encryption system for MeeGo mobile OS, which deeply integrates into the Qt Message Framework (default MeeGo message framework), is described. Finally, the fast key search algorithm, that does not require the preliminary decryption of all the messages, is proposed. The combination of these solutions comes to a fast, scalable and user-friendly message encryption service.

Index Terms: Encryption, Key word search, QMF, MeeGo.

I. Introduction

According to the Infowatch security report [1], the average number of personal data records influenced by a leak grew from 405000 in 2008 up to 754000 in 2009. Moreover, about 20% of all the confidential data leaks were associated with laptop, PDA or mobile phone stealing (see fig. 1). Therefore, the security mechanisms for the mobile devices should be rapidly improved in order to resist such a threat.

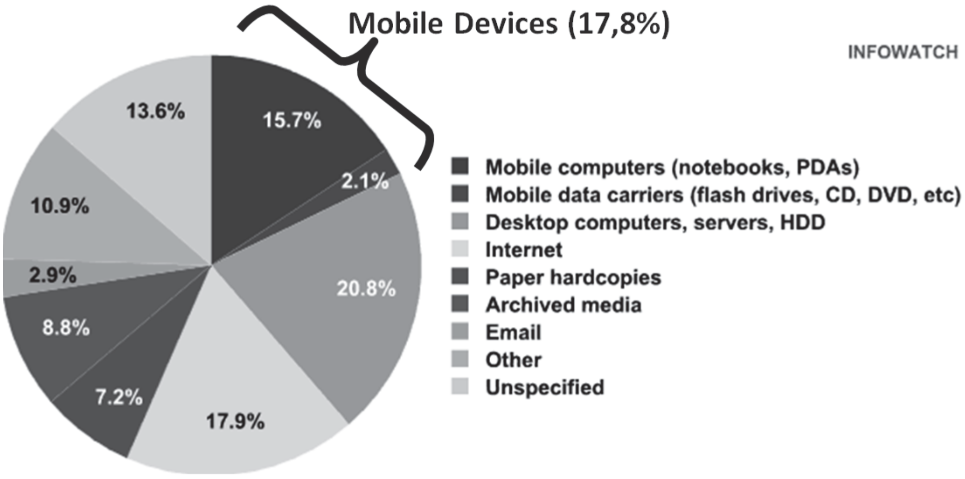

Fig. 1. International leaks distribution by leak channels [1]

The new generation smartphones with powerful CPUs and Office, Skype and e-mail client installed, can now assist in comprehensive part of business activities, including involvement of private data. Comparing with a laptop, the advantages of smartphones are small size and low weight, as well as much longer battery life. So the percentage of managers, preferring a smartphone to a laptop for short trips, grows from year to year. But in order to let this happen, several technical problems have to be solved. And one of them is significantly low security level of the data, stored on a smartphone. The overwhelming majority of business laptops provides lots of security mechanisms, including transparent data encryption, anti-virus and firewall systems, fingerprint scanners for user authentication, etc. Moreover, they could be easily installed and maintained even by the company security engineer. On the contrary, there are very few such applications and hardware modules for smartphones and tablet PCs. This comes from the operation system difference: laptops and desktop PCs usually have the traditional multiuser OS installed, while smartphones consider the current user as the only one. Despite the presence of security boot features and access control systems in some OS (e. g. Symbian [2], Maemo [3] or MeeGo [4]), most part of the job should be done by the additional software.

Despite the fact, there are several well-made applications for confidential data concealment (e. g. *ProtectedSMS* [5] and *SMS-PRO* [6]), all of them have the common problem: they are not integrated to the device OS. As the result, the user interface is different, and usually it takes more time to decrypt the message via one application and to answer it using another. Moreover, due to the encryption and decryption algorithm complexity, all the operations with the encrypted data take more time. Considering the limited processor speed, it comes to substantial delays, especially during the key word search — one of the common operation with the stored data. The problem is that traditional key word search requires the preliminary decryption of all the data chunks — a very laborious operation.

In this paper, the new data encryption system architecture, that transparently integrates into the existing environment and provides fast key word search without the preliminary decryption, is proposed on the example of MeeGo OS [4]. Compairing with the existing solutions, it results in faster and more user-friendly message encryption service. The paper has the following structure. In section II there is a short description of the Qt Message Framework (default MeeGo message framework), those API will be used to perform encryption and decryption operations on-the-fly, when necessary. In section III the system architecture is proposed and integration problems are considered. In section IV the novel key word search algorithm is described and its efficiency is estimated. The paper ends with some conclusion remarks.

II. Qt Message Framework description

The Qt Messaging Framework, QMF, consists of C++ library and a daemon server. The QMF is a common framework for all the device message clients, including e-mail, SMS and instant messaging services. The framework divides the message application functionality into two parts (see fig. 2).

The *message server* is a long lived process, performing synchronisation with external mail servers, which are normally located on the Internet or company network server. The second part of QMF application is a *message client*. It uses the QMF library to interact with the message server. The server and client QMF libraries interacts via internal Inter-Process Communication (IPC), which depends on the operation system.

All the messages are stored in the light-weight SQLite [7] database, and the current architecture can be extended by the protocol plugins. The detailed service description can be found from the official Nokia documentation [8].

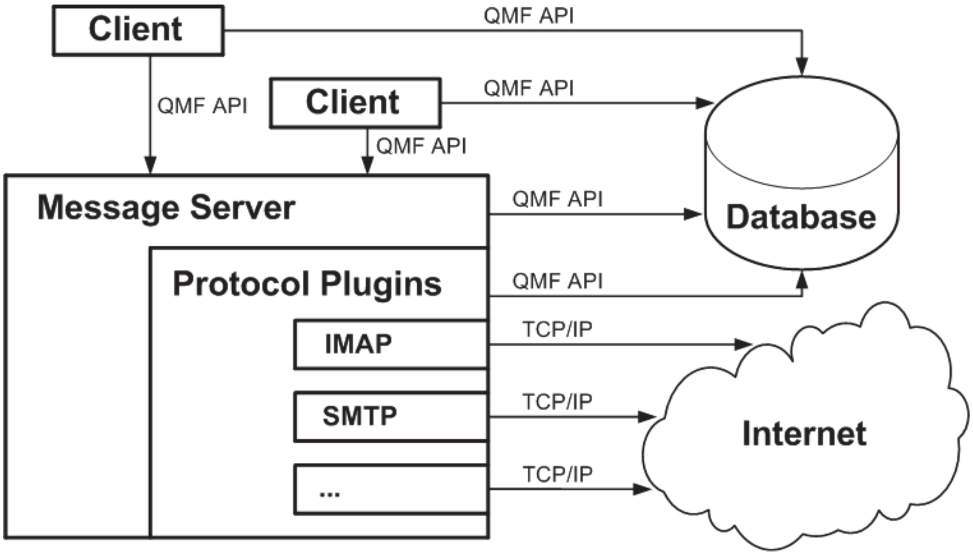

Fig. 2. The QMF common architecture [8]

III. Security system architecture

The proposed security system structure is represented in the fig. 3. The main idea is not to develop an own application GUI, but to use the existing one. So when user starts any communicating application, including E-mail, SMS or instant messaging client, Contacts or Notes, the security system gets the signal via the current IPC, decrypts all the necessary information and provides it to the appropriate QMF client (see fig. 3).

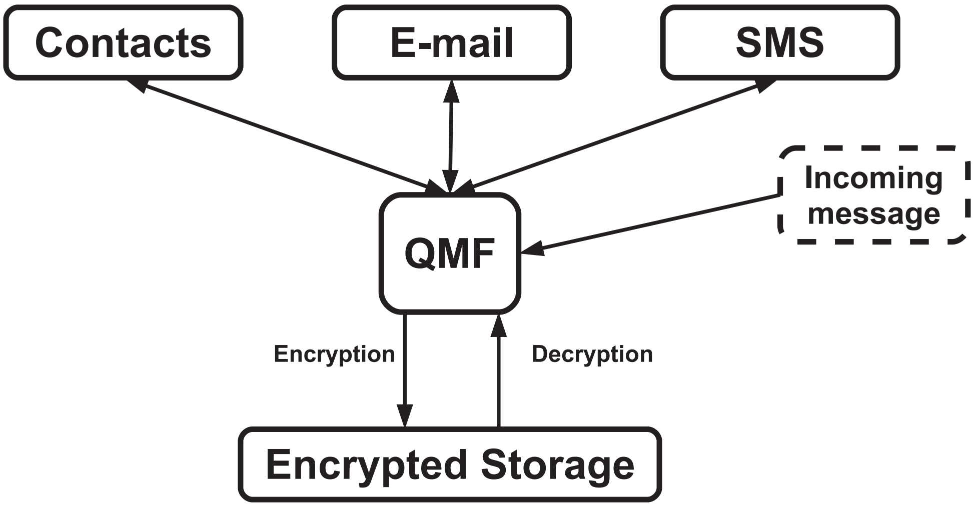

Fig. 3. The proposed security system architecture

The proposed system does not interrupt the communication between the QMF client and QMF server, but integrates into Server-to-Database channel. There are two cases, when the existing algorithm is modified. The system adds two operations to the standard communication protocol between the QMF client and QMF server.

1) If the new message appears on the QMF server, it is first encrypted by the security

module and only after stored in the database.

2) If user starts a QMF client (Contacts, SMS, etc.) and the QMF client sends a data request, the QMF server decrypts the results before sending them to the client.

From the technical point of view, the security module is a QMF plugins modification encrypting the values of *InitializeQuery()* and *Execute()* functions and decrypting the statement, returned by *FetchNextRow()* and *CompleteQuery()* requests.

IV. Key word search algorithm

In this section the major aspects of the key word search algorithm, suitable for the suggested security system, are described. In order to complete the search without the preliminary decryption of all the messages, the default symmetric encryption procedure should be replaced. The proposed algorithm is the modification of the Symmetric Searchable Encryption (SSE) [9], mainly used for the private cloud storage. To perform this, the ID table should be added to the database (see fig. 2).

Let the $ID^i = ID_{i_1}, ..., ID_{i_n}$ be the array of links to the encrypted messages, containing the key word i, let the *encrypt(m)* function return the ID of encrypted message m and let the *divide(m)* function return $ID(m)$ the array of all the key words, that message m contains.

Then, if the new message m_{new} appears, it is divided to single words by the *divide()* function, and all the ID arrays are updated with the following algorithm.

```
for all $k$ do
    $flag \leftarrow 0$
    for all $i$ do

        if $i = ID(m_{new})_k$ then
            $ID^i \leftarrow ID^i + ID_{m_{new}}$
            $flag \leftarrow 1$
        end if
    end for
    if $flag = 0$ then
        create $ID^k = ID_{m_{new}}$
    end if
end for
```

So when the QMF client initiates a search for the word w, system just decrypts and returns messages from the ID^w array, if ID^w exists. Otherwise, the NULL string returns.

The advantage of such an algorithm is the key word search speed. Considering the decryption complexity to be much higher compared to the search, the proposed solution is faster, because only decryption of the messages containing the key word is required. However, the complexity of the update algorithm rapidly increases from $O(1)$ to $O(n)$. One of the possible solutions, that reduces it, is a hash table usage for storing IDs instead of a linked list. This is one of the future research directions, in addition to the proposed solution speed estimation and comparison with the speed of existing one.

V. Conclusions

The market of mobile devices, such as smartphones and tablet PCs, as far as the amount of stored confidential data, grows rapidly from year to year. However, the available security systems do not cover all the required user's scenarious, especially in communication area.

In this paper the new architecture of message encryption system for MeeGo mobile OS, which deeply integrates in the Qt Message Framework, is proposed. The advantage of the system, comparing with the existing solutions is transparency for the user and novel key word search algorithm that does not require the preliminary decryption of all the messages. This helps user perform the common operations faster and saves the mobile device battery life.

References

[1] N, Fedotov, ”Global Research on Data Leaks in 2009,” http://www.securelist.com/en/analysis/204792108/Global_Research_on_Data_Leaks_in_2009.

[2] Symbian Foundation, ”Symbian Mobile OS,” http://licensing.symbian.org, 2009.

[3] Maemo Community, ”Maemo OS Community Homepage,” http://maemo.org, 2005.

[4] MeeGo Community, ”MeeGo OS Community Homepage,” meego.com, 2010.

[5] Protected Mobility, ”ProtectedSMS official web-site,” http://www.protectedmobility.com/protected-sms/overview.

[6] ”SMS-PRO official web page,” http://sms-pro.smartcode.com.

[7] D. R. Hipp, ”SQLite homepage,” http://www.sqlite.org, 2000.

[8] Nokia Corporation, ”Introducing QMF an advanced mobile messaging framework,” http://labs.qt.nokia.com/2009/09/21/introducing-qmf-an-advanced-mobile-messaging-framework, 2009.

[9] S. Kamara, K. Lauter, ”Cryptographic Cloud Storage,” *Microsoft Research White Paper*, 2010.