Remote Locking of Android Phone through the SMS and Wipe System
Kiran Baravkar1 Tushar Dimble2 Ashwini Jamdade3 Rahul Samant4
1,2,3Student 3HR & Team Leader 4Assistant Professor
1,2,4Department of Information Technology
1,2,4NBN Sinhgad School of Engineering, Pune, India 3 Amdos, Pune, India

Abstract—Android offers hundreds of thousands of applications via application markets and users can readily install these apps. However, this growth in smartphone usage and the ability to install third-party applications has given rise to several security concerns. In this paper, we present smartphone security mechanisms.

Key words: Remote Lock and Wipe System, Smartphone’s, SMS, Message Authentication Code

I. INTRODUCTION

Smartphone’s are becoming more and more popular due to the increase in their process power, personal nature. Android is one of the most popular and fully customizable open source mobile platforms that come with a complete software. One of the main reasons behind the rapid growth in adoption of Smartphone is their capability to facilitate users with third-party applications. Presently issue regarding to the android phone with the it can’t lock with the present of the user. When user is absent any person can operate their phone. Hence they want complete security in absent of user. This locking of the android phone through the SMS is used to overcome the entire problem which they are facing currently and making the complete secure your android phone from anywhere and anytime.

A. The Remote Lock and Wipe System:
The remote lock and wipe system consists of a remote control module on a server and a command handling module on a smartphone.

Fig. 1: Remote Lock and Wipe System
For example, when the users send a lock command to the smartphone via the remote control module, the remote handling module enables the password locking function to lock the smartphone. Similarly, by sending a wipe command, all personal data such as smart wallet, smart keys, contacts, SMS, E-mail, photos, and movie clips are remotely deleted.

II. RELATED WORK

Existing paper proposes a model to secure smart phones from theft as well as provides options to access a smart phone through other smart phone via Short Message Service. They provide option to track and secure the mobile by locking. It also provides facilities to receive the incoming call and SMS information to the remotely connected device and enables the remote user to control the mobile through SMS. Zohaib has put forth the major challenges faced in designing a ubiquit application. Android operating system is suggested as a best tool for designing applications. It can locate the mobile and track it, using the proposed approach. Listen incoming calls and read incoming SMS and give automatic reply. Access and change GPS, WIFI and profile settings through SMS. The application uses Google map API to show location on map. An Android application called "mySpy" was built to allow concerned party to view videos and photos stored device, see the phone owner's list of applications and software updates, open the calendar, notes and tasks. Even get hold of the phone's unique IMEI number which means owner might not have any access to its own device. Antone Gonsalves, on his research on “Kindsight: application to turn Android device to Agent 007 tool” stated that installing and running Kindsight's component on a device, the user would have to find a way to bypass Android's built in security features. By default, application does not have the permission needed to perform operations impacting other apps or the device in general. Such permission would have to be granted by the user. In contrast to existing models our model provides options to access a smart phone through multimedia phone. It doesn’t allow locking the client smart phone and ignores controlling the client smart phone by concerned party, considering it as unauthorized access. It also facilitates location updates whenever a coded SMS is received. Our application uses location service-network provider for location details, rather than Google map API. It also maintains a log file containing the incoming call and message details which are mailed to the concerned party every day and then it updates itself. This log file is out of scope of user of second party.

III. THE PROPOSED SYSTEM

We propose the remote locking and wipe system using SMS push notification with integrity checking of the commands without sacrificing security level. To this end, we employ password-based key derived function (PBKDF) in PKCS#5 which requires users to put only a password in and outputs 20 bytes long authentication code. The basic idea in our proposal is that Hash-based Message Authentication Code (HMAC) for authenticating SMS command message and the secret key to HMAC is derived from the passwords keyed in from the user. The HMAC is based on SHA-1 algorithm.
A. Generating Message Authentication Code:
When the SMS command notification is sent, the remote control module first creates a secret key from the password using PBKDF. Using HMAC function with the secret key, the message authentication code (MAC) is generated on the command message along with the timestamp which is added to protect against the well-known reply attack. Then, the command message is sent with the MAC to the designated smartphone. The sending message consists of command (4-6 bytes), timestamp (8 bytes), and MAC (20 bytes) and thus the total size of message is 32 or 34 bytes. To send the message in form of SMS, the sending message should be encoded with Base-64.

B. Verifying Message Authentication Code:
Upon receiving the authenticated SMS notification, the command handling module first decodes it with Base-64, and check the MAC. The MAC verification mechanism is as follows: the user needs to set up the initial passwords in the smartphone. Next, the command handling module computes a secret key from the passwords using PBKDF and generates a MAC on the decoded command and timestamp. Then, it compares the regenerated MAC with the decoded MAC. If they are the same, the command is executed. If not, the command is ignored because it is considered as the command is compromised while transmission over the network.

IV. IMPLEMENTATION
We implemented the proposed system in Java 1.6.0 and specifically used the Java extended cryptography package (javax.crypto) for cryptographic functions. The command handling module for smartphone was implemented on top of Android 2.2. The remote control module was tested using Tomcat web server 6.0 running on Windows 2003 and the command handling module was tested on Google Nexus-1.