Secure Storage of Data on Android Based Devices

Poonguzhali P., Prajyot Dhanokar, M. K. Chaithanya, and Mahesh U. Patil

***Abstract*—With the rapid evolvement in the mobile landscape, mobile devices has become one of the integral part of day-to-day activities. It has also become a storage place for all kind of information which includes personal information, photos, videos, work data, and many more. Moreover, the mobile devices are being widely used in financial transactions and people tend to store the financial credentials also on the mobile device. This inturn aids the user to easily carry and perform the work from anywhere and anytime. But, with this flexibility, there are high chances of user data getting compromised due to the varied mobile threats like device loss/theft, malwares. In order to prevent data compromise and to protect users' against data theft, there arise the need to secure the data stored on the device. Android provides a secure means of storing passwords/PIN as part of the platform, namely Android KeyStore and also supports device encryption that encrypts the complete device and there is no other means for storing other sensitive data securely. In order to address this issue, a secure means of storage for Android platform is developed namely Secure Storage. In this paper, we highlight some of the mobile device security threats, the importance of securing data and brief the secure storage component.**

***Index Terms*—Android, data compromise, keystore, mobile threats, secure storage.**

I. Introduction

The rapid proliferation of mobile devices has made a impact on everyone's day-to-day life. The outrage growth of smartphone industry has changed the way of organizing ones' life. This is because, it is very handy, simple, easy to use in nature helpful in performing day to day activities. Unlike the conventional feature phones used mainly for making calls and messaging (SMS), smartphones are used for various purposes apart from making calls/SMS like a camera, web browser, for online payments, data storage and so on. Though many varied smartphone platforms like iPhone, Blackberry, Windows, etc are available in the market, Android, with over 255 million units shipped and nearly 85% of the market share in the second quarter of 2014 continues to dominate the global smartphone market [1]. With this highest market share, Android continues to be the most targeted mobile operating system [2], [3].

The attacks could be anything that would compromise users' privacy. The possible attacks on mobile device but not limited to, are unauthorized access to mobile devices, stealing of sensitive data, or running up bills by automatically sending premium-rate text messages. The increased portability of smartphones has made it more susceptible to loss/theft of sensitive data, a primary risk which is prevalent today. Most of the security solutions that exists in the market for securely storing data in smartphones are cloud based or remote server based [4]. This introduces a requirement to the user, a working data connection. This makes the accessibility of the stored contents dependent on data connection. Also, there is a possibility of user data getting compromised, if the cloud storage or the remote server database gets compromised. This threat can be addressed enabling the storage of data securely locally on device, that removes the worry about data breaches. Also, the data can be accessed anywhere and any-time.

Though, there do exists solutions for securing data on the device, most of it are either hardware backed or with modifications in Android framework or in OS [5]-[8]. This paper presents a secure storage application for Android based devices. It is built as an indigenous solution at the application level with no extra requirement. This provides a flexible and more secure mechanism to store user's data locally on device.

The rest of the paper is structured as follows. Threats pertaining to mobile devices are presented in Section II. Data protection mechanisms available in Android are presented in Section III. Secure Storage is briefed in Section IV. Section V and Section VI presents the working and usage of secure storage application respectively. Section VIII presents the experimentation results of secure storage application.

II. Threats to Mobile Devices

Threat, is a possible vulnerability that may cause the applications on the device to malfunction or crash are spread across each and every aspect of mobile devices requiring high protection. Any system connected to Internet is prone to get affected by virus, malwares, spywares and so on. This applies to all from traditional desktop computers to the current smartphones. Smartphones are also being targeted for attacks along with traditional desktop computers as it have become so valued for a variety of personal information stored on it. Threats to smartphones can be categorized as application based, web based, network based or physical. Malwares/Spywares can be classified under application-based threats, as they are applications with malicious behavior. Network based or Web based threats are more prevalent on devices that are connected to internet [9]. The portability of the device has made the smartphones more vulnerable for physical attacks. Below mentioned are some of the mobile device security threats and the risks they pose.

A. Device Loss or Theft

Smartphones with small form factor with PC-grade processing power and storage, tend the users to store

Manuscript received November 15, 2014; revised February 3, 2015. This work is supported by Department of Electronics and Information Technology (DeitY), Ministry of Communications & IT, Government of India.

The authors are with C-DAC, Hyderabad, Telangana 500005, India (e-mail: poonguzhalip@cdac.in, padhanokar@cdac.in, mkchaithanya@cdac.in, maheshp@cdac.in)

DOI: 10.7763/IJET.2016.V8.880

significant sensitive personal and work data. This portability of mobile device increases the risk for data loss [9], [10].

B. Mobile Malware

Malwares are applications that perform malicious actions on the device. These applications run on the device without users' knowledge, which can steal sensitive data stored on the device, make charges to phone bill or give control of device to remote attacker [9], [10].

C. Spyware

Spywares are applications that steal information, which includes phone call history, text messages, user location, browser history, contact list, email, and private photos from the users device without their consent. These information are mainly used for performing identity theft or financial fraud [9].

D. Untrusted Applications

Untrusted applications can access information like sending Short Message Service (SMS), accessing address books, geo-location and record voice calls. Also, since the permissions are easily accessible, the untrusted applications can even wipe-out data without user's knowledge.

E. Storage of Sensitive Information

The business-related and even personal information stored on mobile devices is often sensitive. Encrypting this data is a must. If a device is lost and the Subscriber Identity Module (SIM) card stolen, the thief will not be able to access the data if proper encryption technology is utilized on the device.

F. Poor Authentication and Authorization

Poor authentication and authorization is definitely a mobile platform application challenge, which may lead to attacks like privilege escalation and unauthorized access. The authorization and authentication process should be more stringent in the mobile applications before the users edit/view/submit sensitive information over the Internet.

G. Unwanted Calls and SMS

Malicious use of premium rate Call/SMS services is spreading around the world making easy for cyber criminals to make money fast.

H. Phishing

A user cannot exactly identify what mobile application or website is being interacted as user interfaces for mobile devices are constrained by the devices' small screens and also mobile operating systems and browsers lack secure application identity indicators. This exposes users to the risk of mistaking a malicious application for a trusted one.

I. Rooting

Rooting introduces many serious security breaches like breaking most of the Android's security layers, application sandboxing and get control over the other applications data.

J. Privacy

Passwords, credit card numbers, and more personal information are all routinely discussed over mobile thinking that the phone frequency is not being monitored. While this type of passive monitoring is not cheap or always easy, it is possible.

K. Vulnerable Development

The legitimate applications can also pose a threat to the user if the developer has not taken security into account when developing the applications. Android applications that are over privileged can make the application vulnerable to threats [9].

Of all the threats discussed above, it is observed that, most of the threats are related to data compromise on the mobile device. This necessitates the requirement for securing the data stored on the mobile device.

III. Data Protection in Android

Android, a modern open mobile platform provides security by means of application sandboxing mechanism, in which no application by default has permission to perform any operation that would impact another application, the operating system, or the user [11].

The key components required for data protection are the data to be secured and the security key used for encrypting the data. The encrypted data can only be readable if the security key is known, for which there is a need to store the security key on the device or generate the security key on the fly with the password provided. Storing the keys on the device pose a threat of data compromise if the key is stolen and it is always recommended to generate the security key from the password as per the standard Password Based Encryption Standard [12]. With the enhancements in Android security framework, full file system encryption can be enabled, so that all user data can be encrypted in the kernel but it never applies for the external storage [11]. Android enables to store the security keys in the keystore from where the applications can access the keys for performing the cryptographic operations. Key chain concept, introduced in Android 4.0 enables user to store system-wide credentials that can be used by several applications with users consent. Android 4.3 supports an enhanced version of keychain where in the user can store credentials on per application basis [11], [13]. This is still enhanced with the hardware support [8] for storing credentials which many devices doesn't support currently.

Full file system encryption in Android [14], [15] uses the device password for encrypting the device, a lengthy process which makes the device unusable during that time. Once it is enabled, it cannot be reversed without a factory reset of the device. Encrypted Android device prompts for the device password at boot to enable the device for normal experience. Also, depending on the device capabilities, full file system encryption may degrade the device performance [16].

In order to address these issues, we have developed a Secure Storage application that is storage based following the password based encryption standard. This provides a option to the user for encrypting the files of interest rather than full file system encryption. The advantage of using this solution is the possibility of reversing the encrypted files, which is not the case with the in-built file system encryption mechanism of Android.

The implementation and the usage details of Secure Storage application are explained further in this paper.

IV. Secure Storage: Overview

Sensitive/critical files, data, texts encrypted on the mobile device. Confidentiality of the data stored on device is maintained though the device gets compromised. This application provides improved performance over full file system or SD Card encryption by supporting selective encryption [17]. The key features that are supported by the application are listed below:

1) Password based encryption scheme
2) Confidentiality of personal & critical files/data (AES256 Symmetric)
3) Option to view encrypted critical files/data
4) Option to restore critical files/data back to file system
5) Support for different templates for storing critical information viz Bank Cards, Online Account details, etc.
6) Supports Store/Restore/View of files as background tasks for better user experience and capable of handling large files.

Fig. 1 depicts the overall architecture of the Secure storage application. The user is provided with a user interface for selecting/storing data on the mobile device. The Encrypt/Decrypt engine processes the selected data, by encrypting the data for storage and decrypting the data for viewing/restoring.

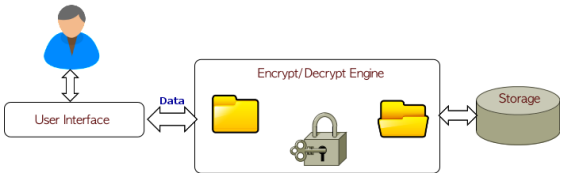


Fig. 1. Secure storage architecture.

This application enables the user to store files and texts securely. The texts can be stored in any of the predefined templates format. The key for storing the contents on mobile device is generated using the standard Password Based Encryption Standard (PBES) mechanism. This mechanism generates two security keys, a master key and a content protection key. The master key is generated from the user provided password and is used to encrypt the content protection key used for encrypting the sensitive data as depicted in Fig. 2. The process of key generation using PBES is explained further. Using a key derivation function, a key is derived from the password. The key derivation function is having five input parameters:

$$DK = KDF(PRF, password, salt, c, dkLen) \quad (1)$$

where:

- PRF is a pseudorandom function of two parameters with output length hLen
- Password is the master password from which a derived key is generated
- Salt is a cryptographic salt
- c is the number of iterations desired
- dkLen is the desired length of the derived key
- DK is the generated derived key

Each hLen-bit block *Ti* of derived key DK, is computed as follows:

$$DK = T_1 \backslash\backslash T_2 \backslash\backslash ... \backslash\backslash TdkLen/hlen \quad (2)$$

$$Ti = F(Password, Salt, Iterations, i) \quad (3)$$

The function *F* is the *x* or *c* iterations of chained PRFs. The first iteration of PRF uses password as the PRF key and salt concatenated with *i* encoded as a big-endian 32-bit integer (Note that *i* is a 1-based index). Subsequent iterations of PRF use password as the PRF key and the output of the previous PRF computation as the salt:

$$F\{password, salt, iterations, i) = \boldsymbol{U}_1 \text{ xor } U_2 \text{ xor...xor } Uc \quad (4)$$

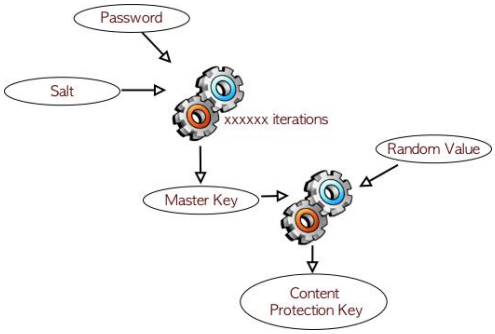


Fig. 2. Key generation using password based encryption standard.

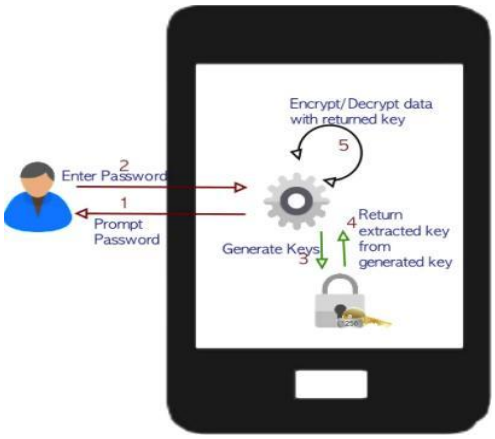
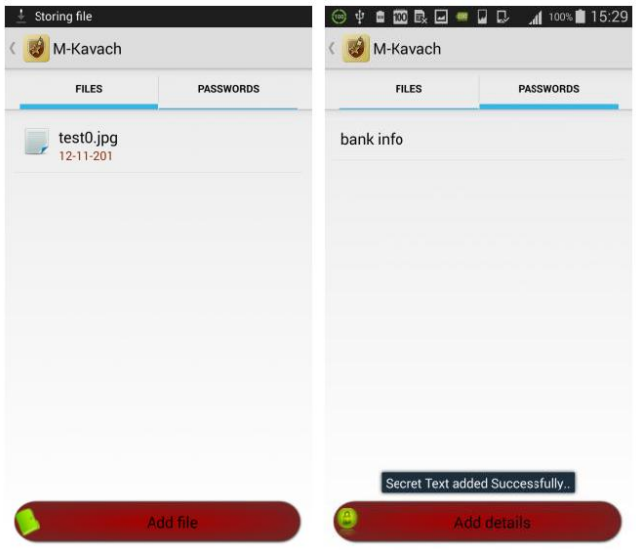


Fig. 3. Process flow.



(a) Secured files. (b) Secured texts.

Fig. 4. Secured files and texts.

where:

U_1 = PRF(password, salt-INT_32_BE(i))
U_2 = PRF(password, U_1)
Uc = PRF(Password, Uc-1)

V. Secure Storage : Process Flow

Fig. 3 depicts the process flow of secure storage.

The steps followed are illustrated below:

1) Application prompts the end user for the password/PIN
2) End user provides the application with the password/PIN for performing cryptographic operations.
3) Application requests the key generation engine for the encryption key.
4) Key generation engine.
- generates the master key with the user provided password/PIN.
- extracts the content protection key with the generated master key.
- send the extracted key to the application for performing encryption/decryption.
5) Application performs encryption/decryption of user data with the extracted key.

VI. Secure Storage : Usage

Secure Storage application provides tab views for files and texts respectively. Files tab contains the list of encrypted files names and Secret Text tab contains the description of the encrypted text as depicted in Fig. 4.

The process of securing files and texts are briefed below:

Securing files in Secure Storage includes the operations as mentioned below:

A. Add Files to Secure Storage

Browse through the file system and select a file to be stored, which will encrypt the file and update the filename in the files tab list

B. View Secured Files

Select the file to be viewed from the list, which will decrypt and open the same with an appropriate pre-installed application

C. Restore of Secured Files

Select the file to be restored from the list, which will decrypt the file in the user provided location.

Securing Texts in Secure Storage includes the operations as mentioned below

A. Add Text to Secure Storage

Select any of the template provided from the list and provide the description and remaining fields as per the selected template, which will encrypt the contents and update the description in the texts tab list

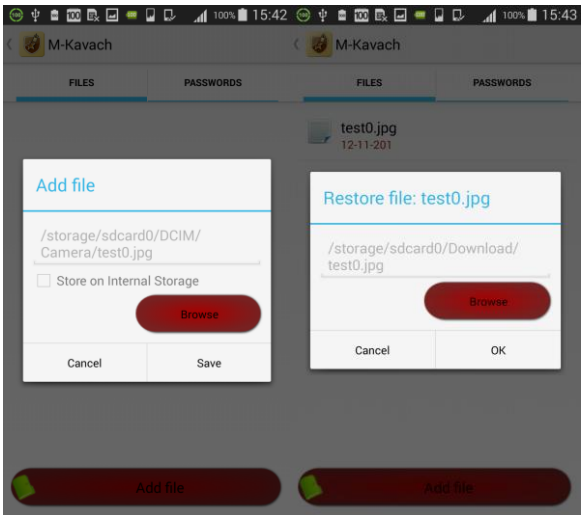
B. View Secured Texts

Select the secured texts to be viewed from the list to view or update the details

C. Delete Secured Texts

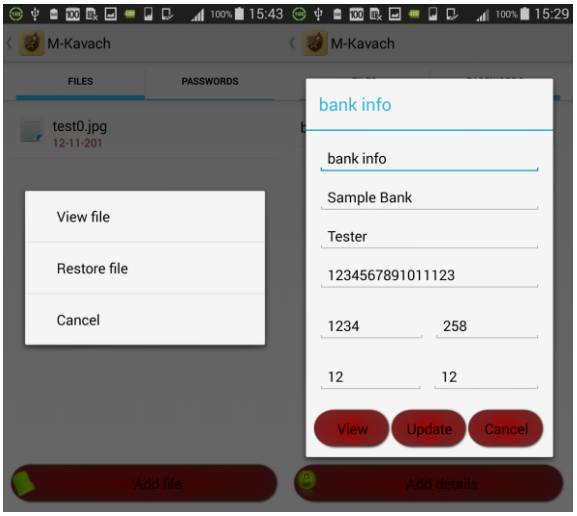
Select the secret texts to be deleted from the secure storage for removing the same.

The options available in Secure Storage are depicted in Fig. 5, Fig. 6.



(a) File to encrypt. (b) Restore encrypted file.

Fig. 5. Secure storage of files.



a) Secure file options. (b) Secure text options.

Fig. 6. Secure files and texts options.

IV. Performance Analysis of Secure Storage Application

The Secure Storage application was experimented on three different mobile devices of varied configurations for calculating the time taken by application. Files of different sizes were utilized and were subjected to 10-15 iterations. The average of 10-15 iterations was considered so as to minimize the error. During the experiment, all background running applications and services were also closed so as to get most accurate results.

Fig. 7- Fig. 9 depict the graphical representation of the results and it is observed that the application works independent of hardware and software versions. The performance varied in terms of hardware and software configuration. As depicted in the Fig. 7- Fig. 9 for 5MB file time taken by single core, 512MB RAM device was around 8.3sec. The same file on dual core mobile it took 5.78sec that is 25% less and on 1GB RAM and quad core device it took less than 2 sec, which shows more than 75% efficiency.

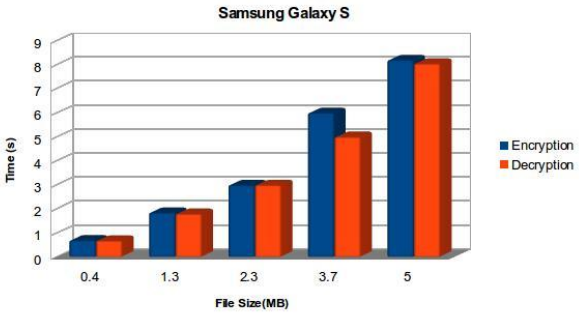


Fig. 7. Experimental results on Samsung galaxy s.

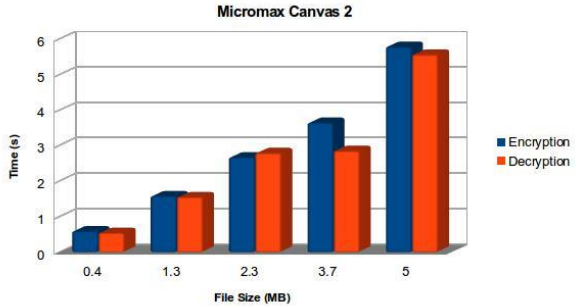


Fig. 8. Experimental results on micromax canvas 2.

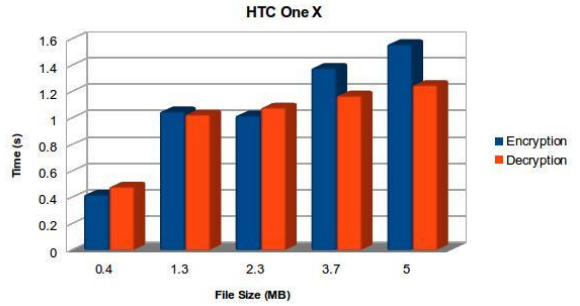


Fig. 9. Experimental results on HTC one X.

TABLE I: DEVICE CONFIGURATION

	Samsung Galaxy GT	**Micromax Canvas 2**	**HTC One X**
OS	Android 2.3.3	Android 4.0.1	Android 4.1.1
CPU	Single Core 1GHz Cortex-A8	Dual Core 1GHz Cortex-A9	Quad Core 1.5 GHz
RAM	512MB	512MB	1 GB

The configuration details of the devices used for the experimentation is depicted in Table I.

VII. CONCLUSION

In this paper we have tried to discuss the possible threats to mobile devices and importance of securing mobile device user's data. We introduced a flexible, easy to use and secure mechanism to secure the mobile device user's data. Also, we have evaluated and presented the details of performance analysis for the developed solution.

ACKNOWLEDGMENT

The authors acknowledge Department of Electronics and Information Technology (DeitY), Ministry of Communications & IT, Government of India for supporting this work.

REFERENCES

[1] International Data Corporation (IDC). Smartphone OS Market Share. (2014). [Online], Available: http://www.idc.com/prodserv/smartphone-os-market-share.jsp
[2] M. Hypponen. F-Secure Threat Report. (2013). [Online], Available: http://www.f-secure.com/documents/996508/1030743/.
[3] 05 Reasons 97% of all New Mobile Malware is Targeting Android. [Online]. Available: http://safeandsavvy.f-secure.com/2014/03/13/5-reasons-97-of-all-new-mobile-malware-is-targeting-android/
[4] B. Donohue. (October 2013). No Shortage of Mobile Secure Storage Apps. [Online]. Available: http://blog.kaspersky.com/no-shortage-of-mobile-secure-storage-apps/.
[5] E. William, G. Peter, C. B. Gon, C. P. Landon, J. Jaeyeon, M. D. Patrick, N. S. Anmol, and *T. Droid*, "An Information-flow tracking system for realtime privacy monitoring on smartphones," in *Proc. the 9th USENIX Conference on Operating Systems Design and Implementation*, 2010, pp. 1-15.
[6] M. Ongtang, S. Mclaughlin, W. Enck, and P. Mcdaniel, "Semantically rich application-centric security in android," in *Proc. the ACSAC 09: Annual Computer Security Applications Conference*, 2009, pp. 1-27.
[7] B. Faysal and L. Jorn, and D. D. Bart, and N. Vincent, "Secure storage on android with context-aware access control," *Communications and Multimedia Security*, vol. 2, 2014.
[8] Samsung, KNOX: Mobile Enterprise Security. [Online]. Available: https://www.samsungknox.com/
[9] Lookout, What Is a Mobile Threat? [Online]. Available: https://www.lookout.com/resources/know-your-mobile/what-is-a-mobile-threat.
[10] Tech Target, Mobile Device Protection. [Online]. Available: http://www.searchmobilecomputing.techtarget.com/guides/Mobile-device-protection-and-security-threat-measures.
[11] Android Security Overview. [Online]. Available: https://www.source.android.com/devices/tech/ security.
[12] RSA Laboratories. PKCS #5: Password-Based Cryptography Standard. [Online]. Available: http://www.emc.com/emc-plus/rsa-labs/standards-initiatives/pkcs-5-password-based-cryptography-standard.htm.
[13] N. Elenkov. Storing Application Secrets in Android's Credential Storage. [Online]. Available: http://www.nelenkov.blogspot.in/2012/05/storing-application-secrets-in-androids.html.
[14] G. Rutenberg, Some Thoughts About Androids Full Disk Encryption. [Online]. Available: http://www.guyrutenberg.com/2012/06/29/some-thoughts-about-androids-full-disk-encryption/.
[15] J. Gotzfried and T. Muller, *Analysing Androids Full Disk Encryption Feature*, 2014.
[16] How To: Encrypt your Android Device. [Online]. Available: http://www.pocketmeta.com/encrypt-android-device-2390/.
[17] *Guide to Storage Encryption Technologies for End User Devices*, NIST Special Publication 800-11.

P Poonguzhali is currently working as a senior technical officer at C-DAC, Hyderabad. She received her B.E (ECE) degree from Anna University and She received her MS in electronics and communication from JNT University, Hyderabad in 2005 and 2011 respectively. She holds certifications in CEH, ECSA and GSSP-Java.

She holds a significant expertise in the area of Android application development, enterprise java application development, network simulators, wireless sensor networks and ASIC & FPGA Design. Her research interests include mobile security, network security, reconfigurable computing systems, FPGA & ASIC designs and ubiquitous computing.

Prajyot Dhanokar was born in Akola (Maharashtra) India in 1990. He received his B.E (IT) degree from Pune University in 2008 and he got his post graduate diploma in system software development from C-DAC, Hyderabad in 2013. Currently he is working as a project engineer at C-DAC, Hyderabad.

M. K. Chaithanya is currently working as a technical officer at C-DAC, Hyderabad. He received his B.Tech (ECE) degree and M.Tech (CS) from JNT University, Hyderabad in 2005 and 2010 respectively. He holds certifications in CEH and GSSP-Java.

He holds a significant expertise in the area of android application development, enterprise java application development, windows system programming, apache module development. His research interests include network security, mobile security, ubiquitous computing, network protocol design and Tizen OS.

Mahesh U. Patil was born in Pune District, in 1979. He has done his post graduation in the field of electronics and communication. Presently he is working as a principal technical officer at C-DAC, Hyderabad. His research interests include mobile security and embedded systems.