Pollux Alarm Vault

[1]Chirag Dhanesha, [2]Krupesh Dobariya,[3]Kishorsinh Chudasama, [4]Darshan Karia, [5]Pratik Patel,
[1]B.Tech Student, [2]B.Tech Student, [3]B.Tech Student, [4]B.Tech Student, [5]Prof. [1-5]Dept.
of Computer Science Engineering [1-5]Parul Institute of Engineering & Technology
Vadodara, India

Abstract - Nowadays, Each and every Person have a mobile phone and access to the internet. They all are having their Personal and Professional information stored in that. But they are not able to keep track whether their data is being secured or have not being leaked by any other third party sources. Pollux Alarm Vault is an Advance Security Android Application which provides Additional Security to users. Also it will provide privacy and safety to secure their personal information and valid data using different encryption technique and AES algorithm which will be useful for them to access the digital world without restriction or any fear. Be it in any medium of image, video,pdf, document file we can gave them authentication assurance of protecting them to be in safe hands.

Keywords—AES Algorithm, Encryption, Android, Cyber Security, Cryptography, Authentication;

I. INTRODUCTION

Pollux Alarm Clock is an android application that will be used by users to secure their data or information. As we know So according to the famous phrase “From the great power comes great responsibility” But taking Responsibility word ,Privacy and security area is are dynamics that is concerned by the users also to protect their valuable data from unauthorized users or any strangers. Our main motto is to build an app which can protect the users data as well as psychologically no one can guess or get access to the private information of users without his/her permission and access. If we can achieve that we can build a secure society which can have transparent and pure atmosphere to live life without any tension or any problems. Pollux Alarm Vault is Android based Security Application which helps users to protect their valuable data through digital security medium. Smart phones and technology have become increasingly powerful in recent years. Dramatic breakthrough in processing power along with number of extra features include in these devices have opened the doors to a wide range of commercial possibilities, but security is the main concern about today’s area. Our project motivation is to secure your private data to hide from other users in secret and psychological smarter manner by our Pollux Alarm Vault. By using this app people can hide their private photos, videos and important documents with advanced encryption standard (AES) techniques which is widely used in IT industry to protect your privacy.

We gave Pollux name to this application because of two psychological reason,

1) Julius Pollux was a Greek writer who equivalent the hideand seek game to world in 2nd century, same concept followed by our app, you have to play hide and seek with data in our app.
2) Also Pollux is a closet giant and brightest star found to near our Sun. But you can’t see it with naked eyes. You need specific telescope to see it, same like our app, you can’t see data hidden with naked eyes,and you have to use secure functionality of our app.

The users will be able to perform many tasks like:

[A] Hide their private data under alarm clock

User can hide their private their data or some private files under alarm clock which can be accessible by just by passing security. So nobody can have access to their data without their permission.

[B] Encrypt files

User can encrypt their private file using AES algorithm, as it will be very secured for users to hide their information of financial data to be leaked or accessible to any unknown person without their permission.

[C] Hide image, video, document, audio files

Different types of file format are supported by application to give various facility for users to protect their privacy inform for image file, video file, document file or any audio file.

2. METHODOLOGY

To verify if an Android vault application was able to protect a user's personal files, the testing was divided into four phases: scenario creation, data acquisition, case analysis and tool development.

2.1. Scenario creation

The purpose of this phase was to simulate real user data on the Android device. As soon as the vault applications were Installed, we took photos (.jpg) and videos (.mp4) with clearly numbered identifiers (in each image / video) for each application and stored them in each `vault'. We had to also initially set up passwords / passcodes on the vault applications.

Alarm Clock Password: If password by user is 1000.Then user have to tap on 1,0,0,0 If password is correct than hidden files will be opened.
Optional Password: Fingerprint Authentication on toolbar users will get operation of opening hide files.
Screen lock password or pattern: setting correct pattern or password also helps users to find hidden files.

2.2. Data acquisition

To explore potential artifacts that may be utilized for breaking into the, two key items (listed in the following paragraphs) for each application were pulled from the Android device to the acquisition workstation. App-generated folders which are folders created by the application on the Android files system. We hypothesized that some les may contain hidden photos and videos, user passwords or other forensically relevant application artifacts that may help us break into the vaults. APK files which are Android Application Packages. These files contain installer les for each of the vault applications and are an important resource for reverse engineering the executable code. Precisely, for each vault application, three folders on the Android system were generally targeted during our data acquisition:

/data/app/package name which is a root user accessible folder that stores the APK file of each application. `Package name' refers to the package name of each application /data/data/package name is another folder accessible for root users that stores the private data / artifacts for each application. /sdcard which is the folder for the mounted Secure Digital (SD) card physically located in the Android device. In our tests, we found different paths that pointed to the same folder such as:
/storage/emulated/0 and /storage/emulated/legacy

It is important to note that in our work, acquisition was accomplished logically. We did not focus on deleted data as we were looking for low hanging fruit artifacts allowing us to break into each of the vault applications and reconstruct hidden les2. For each vault application, we employed the pull command in the adb tool to extract the following folders: /data/app/package_name, data/data/package_name and /sdcard. To replicate our work, one may use the commands presented in the following steps:

Step 1: The command chmod [OPTION] <MODE> <FILE> was used to modify the assigned folders to a normal user accessible under a root user's access.
Step 2: The command adb pull [-p] [-a] <remote> [<local>] was used to acquire the folders mentioned above.

The resultant data from the acquisition process for each vault application is referenced in the case study in Sec. Note that although data acquisition was achieved via adb, other methods to pull data from the aforementioned folders may be employed as well .It is of note that more research needs to be conducted to analyze how the media (images and videos) are stored by the vault applications in case data is copied and then deleted from the device, thus, there may be potential evidence residing on the device's storage when physical acquisition is employed.

2.3. Analysis

Once the artifacts were extracted from the Android device, we sequentially analyzed each vault application through artifact analysis and executable code reverse engineering. Executable code analysis was primarily employed when the security implementation was needed in order to gain access to encrypted material stored in the vault application. Artifact analysis was carried out manually by exploring the acquired data. During exploration, meaningful artifacts such as labels in Extensible Markup Language (XML) les (e.g. <string name="password">), column name in SQLite les (e.g. aeskey) or obvious folder / le names (e.g. /sdcard/.Enc ryptedFolder) were noted.

Some endings became

1) Direct artifacts e.g. unencrypted photos / videos or passwords found in clear text and

2) Indirect artifacts such as hashed passwords or les suspected to be the encrypted photos and videos. Executable code analysis was employed mainly if all artifacts retrieved through our analysis were deemed indirect. We focused on reverse engineering the APK files. Precisely, we disassembled APK les into small code.

By employing manual code analysis, we were able to retrieve

1) Artifacts missed by our primary artifact analysis, where access to information or hidden media files was stored.
2) The security implementations for authentication / authorization or data hiding. This allowed us to understand how to reconstruct hidden data and explore methods of decrypting encrypted data.

2.4. Tool development

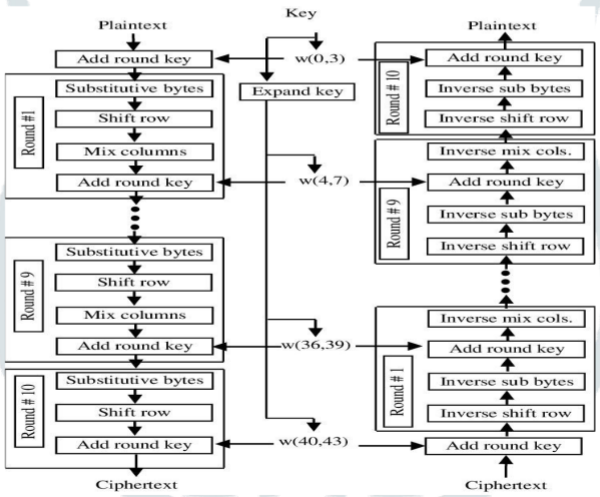
Based on our findings, we constructed software tools if breaking into the vault application required such implementations to aid in the reconstruction and decryption of vault-stored data. We posit that developing these tools is necessary to assist investigators in the recovery of potential evidence they might have missed. We note that these tools are not shared publicly and are provided on need basis, after the identity of the requesting party is vetted.We are using Android studio tool framework with AES algorithm and combination of java language and XML language to build the product.

3. CASE STUDY & FINDINGS

In this section, we present a comprehensive case study and ndings that entail vault applications. It is of note that we attempted two major methods to gain access to data stored by the vault applications. The first focused on finding ways of recovering media les from data acquired directly from the phone which is usually more relevant to forensic investigations. The second focused on gaining unauthorized access to the vault applications through various attack vectors.

Folder “\shared prefs" represents the folder that is located in /data/data/package_name for the vault applications.

Swap attack is a method in which an examiner can swap an assigned value in a certain le with a self-created one in order to reset the password, gesture for login or the status of an application (e.g. switch on/off the pattern lock). Password attacks which represents a variety of attacks against application passwords and gestures. In some instances, a rainbow table attack was used. In other instances, a brute force attack was used. Generally, these password attacks helped us in identifying the passwords used to access the application. In many of the applications, the password space was low (4-6 characters), and sometimes limited to digits, making a brute force attack a practical option. We used AES algorithm only after comparing it with different security algorithm available in market. We made a fair comparison between four most common symmetric key cryptography algorithms: AES, DES, CAST 128 and Blowfish. The comparison takes into consideration the behavior and the performance of the algorithm when different data load are used as the main concern here, is to study the performance of the algorithms under different settings. The comparison is made on the basis of these parameters: speed, block size, and key size. We also compared the Avalanche Effect and integrity checking using ECB and CBC mode of the different algorithms: Blowfish, Cast-128, DES and AES for one bit change in key and one bit changed in the cipher text. Crypto tool will be used for implementing the performance analysis for all algorithms mentioned above. After analysis has been conducted we found that AES gives the best security. The experiment shown that in both modes DES gives strong avalanche affect and AES and Cast 128 gives strong change in term of integrity checking compared with others algorithms using ECB and CBC mode.



According to the schematics AES is more secure than DES, Blowfish, CAST-128 algorithms. Hence, after analyzing the most popular symmetric algorithms AES was found the more reliable, faster and better among the entire existing algorithm with no serious weaknesses, there are some flaws in symmetric algorithms such as weak keys, insecure transmission of secret key. During this analysis it was observed that AES was the best among all in terms of Security, Flexibility, and Encryption performance.

Factors	Cast 128	DES	Blowfish	AES
Key length	128-bits	56 bits	448 bits	128 bits
Cypher Type	Symmetric Block Cipher	Symmetric Block Cipher	Symmetric Block Cipher	Symmetric Block Cipher
Block Size	64 bits	64 bits	64 bits	128 bits
Developped	1996	IBM in 1975	1993	2000
Speed	Fast	Slow	Fast	Fast
Security	Less Secure	Not Secure Enough	Secure Enough	Excellent Security
number of rounds	12	16	16	10
No. of S-boxes.	4	8	4	1
Structure	Feistel Network	Feistel Network	Feistel Network	substitution-permutation Network

AES method has been adopted to protect unwanted interception and viewing of any file like image file while in transmission over the networks. Provide Highly Secure Separable Data Hiding in Image using AES Algorithm. AES will be more Non-Vulnerable than any other algorithm which is providing security till now. Maximum amount of Accuracy and security will be provided by AES.

4. CONCLUSION & FUTURE WORK

As shown from our results,Pollux Alarm Vault application developers indeed code protect their applications and protect users private data using AES algorithm, While this may not be privacy preserving, the discovered security implementations aid digital forensic examiners in reconstructing media les that may be relevant to a case.

Future work should examine vault applications on iOS and also explore network traffic analysis.Work should also test the viability of reconstructing media less from media that is physically acquired. Future research should replicate our methods and focus on other privacy enhancing technologies such as mobile password vault applications. Lastly, work should explore the feasibility of designing a tool to automate the discovery of the security implementations of privacy preserving mobile applications.

5. REFERENCES:

1. A Sophos Article 04.12v1.dNA, eight trends changing network security by James Lyne.

2. International Journal of Scientific & Engineering Research, Volume 4, Issue 9, September-2013 Page nos.68 – 71 ISSN 2229-5518, “Study of Cloud Computing in HealthCare Industry “ by G.Nikhita Reddy, G.J.Ugander Reddy.

3. Ham, K., Chien, Y. R., Kiesler, K., ―An Extended Cryptographic Key Generation Scheme for Multilevel Data Security‖, Fifth Annual Computer Security Applications Conference, Tucson, AZ, USA 1989.

4. Mr. Gurjeevan Singh, Mr. AshwaniSingla and Mr. K S Sandha, "Cryptography Algorithm Comparison for Security Enhancement in Wireless Intrusion Detection System", International Journal of Multidisciplinary Research, Vol.1 Issue 4, pp. 143-151, August 2011.

5. Steven J. Murdoch and Stephen Lewis, “Embedding covert channels into TCP/IP”, Information Hiding, Lecture Notes in Computer Science, vol. 3727, Springer Berlin / Heidelberg, 2005, pp. 247-261.

6. H.J. Shiu, K.L. Ng, J.F. Fang, R.C.T. Lee, C.H.Huang, “Data hiding methods based upon DNA sequences”, Information Sciences, Elsevier, Vol.180, Issue 11, 1 June 2010, pp. 2196-2208.

7. M. Zeghid, M. Machhout, L. Khriji, A. Baganne, and R. Tourki (2007): A Modified AES Based Algorithm for Image Encryption, International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol: 1, No: 3, pp: 745-750.

8. Sushmita Singh and Musheer Ahmad, Dhruv Malik (2016): Breaking an Image Encryption Scheme Based on Chaotic Synchronization Phenomenon, IEEE, pp: 1-4.

9. Roshni Padate and Aamna Patel (2015): Image Encryption and Decryption Using AES Algorithm, International Journal of Electronics and Communication Engineering & Technology, pp: 23-29.

10. Federal Information, Processing Standards Publication 197, November 26, 2001 announcing the “Advanced Encryption Standard (AES)”.

11. Alanazi Hamdan O., Zaidan B.B., Zaidan A.A., JalabHamid.A., Shabbir M and Al Nabhani Y.: New Comparative Study Between DES, 3DES and AES withinNine Factors, Journal of Computing,vol. 2, issue 3, March 2010, ISSN 2151-9617, pp.152-157.

12. W. Liu, W. Zeng, L. Dong, and Q. Yao.: Efficient compression of encrypted grayscale images, IEEE Trans. Image Process., vol. 19, no. 4, Apr. 2010,pp. 1097–1102.

13. Christopher Paar, Jan Pelzl.: The Advanced Encryption Standard, Textbook for Students and Practitioners.