International Journal of Latest Engineering Research and Applications (IJLERA) ISSN: 2455-7137

Volume – 02, Issue – 07, July – 2017, PP – 13-19

Analysis of Steganography as a Security Technique

Anirudh Sohil
Department of Information Technology
BVCOE, New Delhi

Abstract: Application developers faces many challenges while creating an application for use. One of the most important aspects that a developer must keep in mind is the security of his application. Various applications are potential targets for hackers, phishers etc. As these applications are developed, sometimes developers solely focus on visual and functionality of an application and takes only minimalist security measures for their applications due to some reasons. Even though security is one of the most important aspect, sometimes it is given least priorities. This research paper aims to provide reasonable approaches to make an application secure or make a security application for several applications.

I. INTRODUCTION

While researching information security of an application, we came across a lot of methods for security of an application. Several internal as well as external security features can be introduced like a security layer over an application, password protection and many more. But one of the methods that we found interesting was steganography. Developers sometimes think that an extra security layer over an application will provide them with the security for their application. In today's time, when even SHA-1 has been broken, there is a need to add some kind of extra encrypted layer over that security layer. Several cryptographic algorithms like RSA, AES, DES, Steganography etc. are there. This research paper aims to provide extra security to applications using steganography as its base and a key to be embedded using steganography (key can be generated using other encryption techniques).

Security has been one of the major issues for any kind of application that are made and there are many reasons for it. One of the major reasons is no defined software development process. These applications do tend to fail most of the time. Another problem is the understanding of application security. It is often taken as a term and not in SDLC. Regular audits to access potential threats are faced whenever there is a dynamic change is done in an application. Sometimes, security policies are not integrated in SDLC which not only makes it vulnerable to security attacks, but also generates some bugs in the application. Sometimes, application's requirements and security do not come hand-to-hand. Developers do have to tackle this issue and sometimes causing one of them to get almost neglected, usually security being the one to be neglected in the process.

This research aims at making an application which can be used for security purposes using current cryptographic techniques. To provide this illustration, we have divided this research paper into several sections. In first two sections, a brief introduction of steganography is given. As we are using a key generation method for it to be embedded in image generated, we will be using AES and will give its introduction as well. This will give a complete idea to the user how these cryptographic techniques work.Third section will analyze how key generation will be done using the application (which will be built on android). This section covers up the use steganography used with AES encryption technique which will be used for key generation. As steganography is not a new technique, use of modern technique with it improves its security quite significantly. In fourth section, we will analyzehow these two techniques work together to give the fruitful results that we desire from a security application. Also, we will be giving some analytics or an overview of that application using these techniques. If we can, we will provide with comparison of these techniques when used separately and when used together. And in final section, we will conclude the paper by providing the importance of our project, discussing its future scope and the importance of expanding this particular project and the work we wish to perform in future projects.

II. STEGANOGRAPHY

Steganography is a method of hiding or concealing a file of a form (which can be a text file, image or a video) into another file. The word steganography combines the Greek words steganos, meaning "covered, concealed, or protected". Steganography is an encryption technique that can be used along with cryptography as an extra-secure method in which to protect data. [1]

Steganography can be used in images, video file or an audio file. Usually, steganography is written in characters including hash marking. Due to this, in images, it is quite common to use steganography in it. The advantage of steganography over cryptography alone is that the intended secret message does not attract attention to itself as an object of scrutiny. Plainly visible encrypted messages—no matter how unbreakable—arouse interest, and may in themselves be incriminating in countries where encryption is illegal. Thus, whereas cryptography is the practice of protecting the contents of a message alone, steganography is concerned with concealing the fact that a secret message is being sent, as well as concealing the contents of the message.

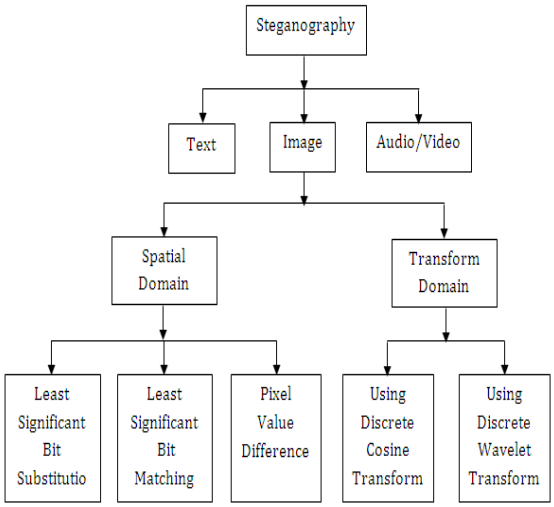

Fig 1. Steganography Fundamentals in different file formats.

Steganography is not limited to scope of only limited to digital file system. It was used one of the most significant cryptographic technique and was used since ancient times like in ancient tablets of Greece, using secret inks under other messages, in Morse Code and even in World War II when photos were having microdots as encrypted message in them. Now though, it is used in encryption of digital messages as well in all of the file formats. And also, some puzzles are actually made by using steganography. Steganography has occurred in the military for decades, even if not computer based. [2]

III. ADVANCED ENCRYPTION STANDARD (AES)

The National Institute of Standards and Technology (NIST) started development of AES in 1997 when it announced the need for a successor algorithm for the Data Encryption Standard (DES), which was starting to become vulnerable to brute-force attacks. [3]

This new, advanced encryption algorithm would be unclassified and had to be "capable of protecting sensitive government information well into the next century," according to the NIST announcement of the process for development of an advanced encryption standard algorithm. It was intended to be easy to implement in hardware and software, as well as in restricted environments (for example, in a smart card) and offer good defenses against various attack techniques.

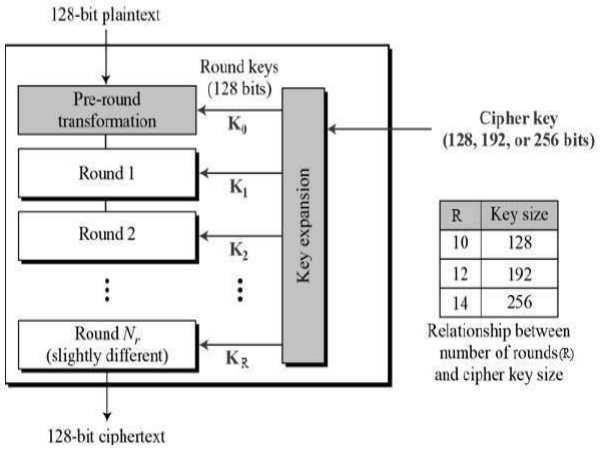

Fig 2. AES Encryption Process. [4]

AES is an iterative rather than Feistel cipher. It is based on 'substitution–permutation network'. It comprises of a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations). Interestingly, AES performs all its computations on bytes rather than bits. Hence, AES treats the 128 bits of a plaintext block as 16 bytes. These 16 bytes are arranged in four columns and four rows for processing as a matrix. [5]

Unlike DES, the number of rounds in AES is variable and depends on the length of the key. AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Each of these rounds uses a different 128-bit round key, which is calculated from the original AES key.

In present day cryptography, AES is widely adopted and supported in both hardware and software. Till date, no practical cryptanalytic attacks against AES has been discovered. Additionally, AES has built-in flexibility of key length, which allows a degree of 'future-proofing' against progress in the ability to perform exhaustive key searches.

IV. IMPROVEMENT AREAS IN STEGANOGRAPHY

While steganography is an encryption method of concealing a message into another file, but in today's time, steganography alone is not enough to conceal a message within another file. Although it's a good way to hide any kind of useful information, steganography can be countered using steganalysis. There are some measures that can be taken while performing steganography.

A. Data Compression

Steganography is a very wide technique that is used to send the information from sender to reciever. Using cryptanalysis, one can tell that the file has encrypted text in it by checking the difference in bits sent on sender and reciever end. To avoid that, data compression is one of the best techniques that can be used to compress the total bits to a suitable amount from sender's end. This avoids intrusion from any outside interferer (hackers, pishers etc.). Any universal compression algorithm can be used to compress that data.

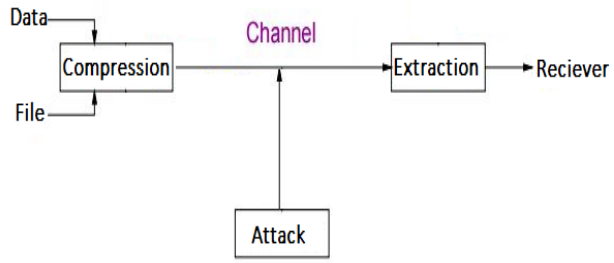

Fig 3. Stegosystem with data compression

B. Public-Key Steganography

In previous system, it is quite plausible to even detect that a file in the form of message has been sent on the channel. To counteract that problem, a public-key stegosystem, much like a public-key encryption system is used. A public-key steganography or a public-key stegosystem, much like any other public key encryption system allows the users to send message on any given channel (public channel). Due to this feature, it becomes theoretically impossible to trace any kind of message transfer on public network when a message is transferred from sender to receiver. Data compression of an stegosystem can be further improved by this steganography.

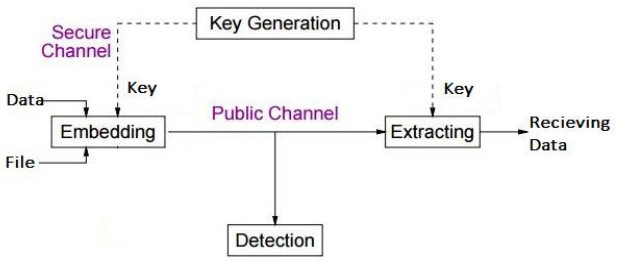

Fig 4. A Public-Key Stegosystem

C. Digital Watermarking

Due to many android devices, it becomes a hassle to make an application which can not only work on different android versions, but can also work on different android versions having their own specific set of hardware. As hardware on a smartphone is impossible to change, it becomes an important aspect for an application developer not only to work on software, but also on limited hardware.

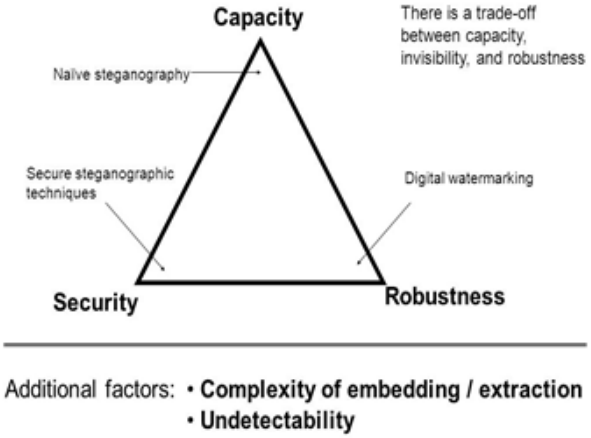

Fig 5.Relationship triangle between Data Capacity, Digital Watermarking and Security. [6]

D. Distributed Steganography

In normal stegosystem, only one carrier or channel is used at the time of transfer of data. This makes it easy for hackers to find out the channel for interference. Distributed steganography aims at sending data through different covert channels by distributing data into multiple carrier files. This, in turn, makes the data detection more difficult. Distributed Steganography, compared to other conventional steganographic techniques, can improve security and information hiding capacity because it leaves reduced signatures of hidden information in host data. [7,8]

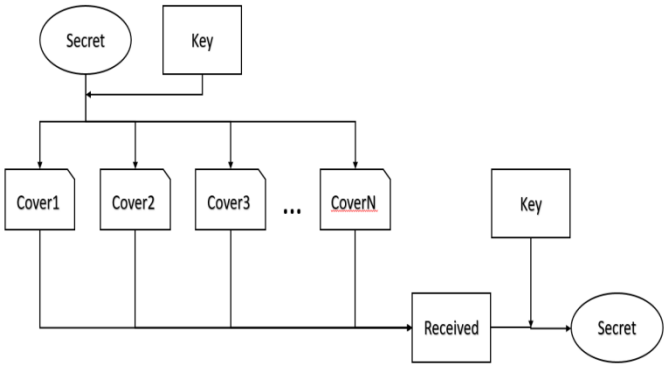

Fig 4. Distributed Stegosystem flowchart. [9]

E. Encoding data before steganography

Steganography involves embedding the data intended for receiver into a file. Recent studies have made it easy to decode data from the file. But an extra security can be added by encoding the data before it can be embedded into a file. Many techniques of encryption like public-key encryption can be used to encrypt the data. In this way, it makes it more secure even when some external attack is used to decrypt the data.

These and many other improvement areas are there to improve steganography or a stegosystem whenever a data is being encrypted using this.

V. ANALYSIS

The android smartphone used for the performance and battery analysis is Redmi 3S Prime. The hardware and software specifications are as follows:

- *OS:*Android v6.0.1.
- *Chipset:*Qualcomm MSM8937 Snapdragon 430.
- *CPU:* Quad Core 1.5 GHz Cortex-A53 and quad-core Cortex-A53.
- *GPU:*Adreno 505.
- *Battery.* Non-removable Li-Po 4100 mAh
- *Internal Memory.* 3GB RAM.

As we have developed an app lock application using steganography, performance will vary from phone-to-phone having different OS and UI. Below are some of the results that we got from the above device:

CASE I

Question embedded with steganography: -

I/Timeline: Timeline: Activity_launch_request time:60865486
W/art: Suspending all threads took: 6.578ms
D/MainActivity: File Size:202800
D/MainActivity: Encoded Message: android.graphics.Bitmap@d2f2993
D/MainActivity: Encoded File Size:202800
D/MainActivity: Decoded Message: bruno

Question code first encrypted then embedded: -

I/Timeline: Timeline: Activity_launch_request time:62938243
W/art: Suspending all threads took: 10.570ms
I/Timeline: Timeline: Activity_launch_request time:62951795
D/MainActivity: File Size:202800
D/MainActivity: Encoded text:W53mmvgEIRrZFBwnzcYPPg==
D/MainActivity: Encoded Message: android.graphics.Bitmap@836835c
D/MainActivity: Encoded File Size:202800
D/MainActivity: Decoded text:W53mmvgEIRrZFBwnzcYPPg==
D/MainActivity: Decoded Message: bruno

CASE II

Question embedded with steganography: -

I/Timeline: Timeline: Activity_launch_request time:60445497
W/art: Suspending all threads took: 6.422ms
D/MainActivity: File Size:202800
D/MainActivity: Encoded Message: android.graphics.Bitmap@a45f991
D/MainActivity: Encoded File Size:202800
D/MainActivity: Decoded Message: 003

Question code first encrypted then embedded: -

I/Timeline: Timeline: Activity_launch_request time:63828547
W/art: Suspending all threads took: 10.016ms
I/Timeline: Timeline: Activity_launch_request time:63828547
D/MainActivity: File Size:202800
D/MainActivity: Encoded text:4+jbkh814tZd9MmttUQmxQ==
D/MainActivity: Encoded Message: android.graphics.Bitmap@f36475c
D/MainActivity: Encoded File Size:202800
D/MainActivity: Decoded text:4+jbkh814tZd9MmttUQmxQ==
D/MainActivity: Decoded Message: 003

CASE III

Question embedded with steganography: -

I/Timeline: Timeline: Activity_launch_request time:61387439
W/art: Suspending all threads took: 6.578ms
D/MainActivity: Encoded File Size:202800
D/MainActivity: Encoded Message: android.graphics.Bitmap@b456033

D/MainActivity: Encoded File Size:202800
D/MainActivity: Decoded Message: ronaldo

Question code first encrypted then embedded: -
I/Timeline: Timeline: Activity_launch_request time:64083241
W/art: Suspending all threads took: 10.570ms
I/Timeline: Timeline: Activity_launch_request time:64083241
D/MainActivity: File Size:202800
D/MainActivity: Encoded text:HpygE+I5y0bP1o92hMGlwg+=
D/MainActivity: Encoded Message: android.graphics.Bitmap@1f297eb
D/MainActivity: Encoded File Size:202800
D/MainActivity: Decoded text:HpygE+I5y0bP1o92hMGlwg+=
D/MainActivity: Decoded Message: Ronaldo

As we can observe from above data that the file size of image is constant before and after embedding for any given string sizes. Also, we can observe that the time to suspend all threads while encrypting the answer for security question is increased than normal steganography.

A. Reason for same size

It is so because these strings take up any unused space generated as the image code we generate for app lock is in the bitmap form. In bitmap form, any sort of message can be compressed in .bmp file format to retain its original size. That is why even large strings of appropriate sizes can be embedded in password image generated. Being a lossless compression standard, BMP is quite ideal for steganosystems.

B. Reason for increased thread time

1). Total threads increased due to inroduction of AES encryption and decryption.
2). AES along with steganography will take more time altogether to encrypt the app lock password as well as decrypting to unlock.

VI. CONCLUSION

From examining the usage of application in both phases where steganography is used and steganography along with AES is used, the app tends to work faster when only steganography is used as it takes less time to finish less amount of threads. But in case of image formation by stegosystem, the file size remains same due to the BMP file format. This concludes to the fact that for more protective application, we must use stegosystem along with AES encryption and if performance is the main priority, steganography is enough without

VII. FUTURE WORK

Future work within this area could involve digital watermarking in stegosystem which improves stegosystem's security. Right now, application is made more secure. Due to which, there is a bit more time to perform. So, some performance improvements must be done in order to improve time for running every thread when AES or other type of encryption is used. Also, some other encryption can be used along with steganography like DES, RSA, SHA and many others.

VIII. GLOSSARY

SDLC: Software Development Life Cycle
AES: Advanced Encryption Standard
CPU: Central Processing Unit
RAM: Random Access Memory
Li-Po: Lithium Polymer

IX. REFERENCES

[1] Steganography, Wikipedia Main Page, Wikipedia.org,https://en.wikipedia.org/wiki/Steganography
[2] The history of steganography, Tim Green, Network World, http://www.networkworld.com/article/2870165/lan-wan/the-history-of-steganography.html
[3] Advanced Encryption Standard, Wikipedia Main Page, Wikipedia.org, https://en.wikipedia.org/wiki/Advanced_Encryption_Standard

[4] 128bit Ciphers - AES Reference implementation and derived code http://embeddedsw.net/Cipher_Reference_Home.html

[5] Advanced Encryption Standard, TutorialsPoint.com, https://www.tutorialspoint.com/cryptography/advanced_encryption_standard.htm

[6] Digital Watermarking & Data Hiding research papers at forensics.nl, http://www.forensics.nl/digital-watermarking

[7] A Novel Method for Distributed Image Steganography, Volume 315 of the book series Lecture Notes in Electrical Engineering (LNEE), https://link.springer.com/chapter/10.1007/978-3-319-07674-4_58 The Android Story, https://www.android.com/history/

[8] Distributed Steganography, IEEE Explore Digital Library, http://ieeexplore.ieee.org/document/6079557/

[9] Distributed Key-Based stegosystem implementation and analysis, https://www.cise.ufl.edu/~nemo/anonymity/lectures/Mehta_Overview.pptx

