Mobile Self-Encryption Based Private Storage System using AES Algorithm

Mr. Neeraj Singh[1], Mr. Nikhil Kumar[2], Ms. Kriti Sharma[3], Mr. Harsh Sharma[4], Mr M.S Chaudhari[5]
[1,2,3,4] *Sinhgad Institute of Technology, Pune, India*
[5]*Assistant Professor, Sinhgad Institute of Technology, Pune, India*

Abstract- **In today's world Smartphone users are increasing day by day. As Smartphone have always been connected within the internet, it plays very important role in user's daily life. Almost all the important data is stored in Smartphone compared to desktop or Pc's. Data Security is an important challenge that prevents wider acceptance of mobile devices especially within private and government fields. The need for encryption is strongly recommended to mitigate the risk of accidental disclosure of private data. Among various encryption techniques, asymmetric encryption is highly computationally intensive compared to symmetric encryption.**

Index Terms- **Security, Encryption, Stream cipher, Data security, Database protection, Privacy, Symmetric Cipher.**

I. INTRODUCTION

The pervasive use of wireless network and mobile devices have been changing out life style significantly, it has become an integral part of our everyday life. Such powerful lightweight computing devices and less cost mobile connectivity paved the way for data-driven applications in mobile environment. To access any data, anywhere, anytime is possible due to Mobile data-driven applications. These device also consist of user's sensitive business and private information which can be in form of images, videos, messages, phone numbers If the user device is stolen or lost then this private information can be used in various ways to harm a particular individual.

As a consequence, the content of such devices should be protected from unauthorized access. In addition, portability makes mobile devices prone to being stolen or lost. It is very challenging to protect the weakly encrypted information on a mobile device. However, it should be noted that this protection should not result in substantial computation overhead, which would include the reduction in both battery life and performance of mobile device.

Chen and Ku introduced the Self-Encryption (SE) Scheme for Data Security in Mobile Devices in 2009 in their paper they have investigated whether it is possible to implement a lightweight encryption algorithm, which provides data confidentiality by exploiting the availability of a secure connection with a central server.

A mobile computing device can connect to a mobile database over a mobile network. The client and server have wireless connections. A cache is maintained to hold frequent data and transactions so that they are not lost due to connection failure. A database is a structured way to organize information. This could be a list of items such as customer id, customer name, and phone number.

In case of loss of the device, the valuable information inside the mobile can be accessed via website and all the data can be recovered in an instant.

The next section will provide a brief review on how mobile vendors currently protect the data stored in mobile devices. Section IV will explain why Symmetric Encryption is more secure and in Section V will explain the advantages of AES over other Encryption Techniques and in Section V we will conclude.

II. RELATED WORK

Securing user sensitive and private information securely has always been an issue in day-to-day situations. Mobile Vendors like Apple and Samsung provide facilities to customer to store their data on cloud based system. Once its content has been securely erased, the device is useless for an adversary who is interested in obtaining confidential information.

Most devices provide encryption tools in order to protect the information contained in their memory. There are currently two approaches: encrypting the whole content of the device or providing a small-encrypted storage area for confidential information.

Many devices also have a small store space that is encrypted that stores the user private data. This space is called as "mobile wallet". However, this storage space does not support arbitrary file formats. In addition, both full-disk encryption and "mobile wallet" base their security on the strength of the password chosen for the encryption.

A. Self-Encryption Scheme

The aim of using this method is that it can be easily used to protect or secure the user sensitive data just by requiring the user to create a secure PIN of a relative small size. Self Encryption does not require any TPM device and can be easily implemented as a user-space application on most smartphone. Unfortunately, we discovered that the encryption algorithm used in Self Encryption does not guarantee data privacy.

It is very challenging to prevent an adversary from breaking the embedded cryptographic algorithm when the mobile devices are captured. It is also not desirable to implement a complex computing intensive encryption/decryption scheme in a mobile device. Therefore, the rationale of this project is to investigate a novel lightweight approach to protect the information effectively even if an adversary has good knowledge of the encryption algorithm and many more resources to break the cryptography.

$\text{Ciphertext} = \text{Plaintext} \oplus \text{Keystream}$ (1)

The keystream generator consists of two parts, a hash function H and a random number generator G. The hash function takes the user's PIN and a nonce as input and the output is an integer seed, which is used as the seed of the random number generator G. The output random number sequence $\{r0, r1, \ldots, rn\text{-}1\}$ indicates which bits are selected and abstracted from the message (plaintext) to form the keystream. Therefore, we have:

$\text{seed} = H(\text{PIN}, \text{nonce})$ (2)

$\{r0, r1, \ldots, rn\text{-}1\} = G(\text{seed})$ (3)

where $\{r0, r1, \ldots rn\text{-}1\}$ is a random number sequence generated by the random number generator G. Since the random numbers could beyond the length of the message, and the length of the message body decreases as bits are abstracted, the pointers to the keystream bits need to be normalized following the changing message size. Hence, among the n abstracted bits $\{r'0, r'1, \ldots r'n\text{-}1\}$, the position of the k-th bit is:

$r'k = rk \bmod (m\text{-}k)$ (4)

B. Self Encryption Framework

Cryptography is a technique that uses algorithm to encrypt and decrypt the user files and data. It allows users to transfer files securely over a insecure network so that it cannot be easily hacked by hackers. Cryptography consist of two parts:-

1. Encryption :

Plain text is converted to cipher code information. The operation of a cipher usually depends upon piece of auxiliary information, called a "key".

2. Decryption :

The cipher information is converted back into the plain text during decryption.

Types of cryptographic key:-

1. Public Key :

It is any cryptographic system that uses pair of key that may be disseminated widely. Every user must have public key and private key. These keys are used for encryption and decryption.

2. Symmetric Key :-

In Symmetric key system, every user has his own private key. Only one key is being used for encryption and decryption.

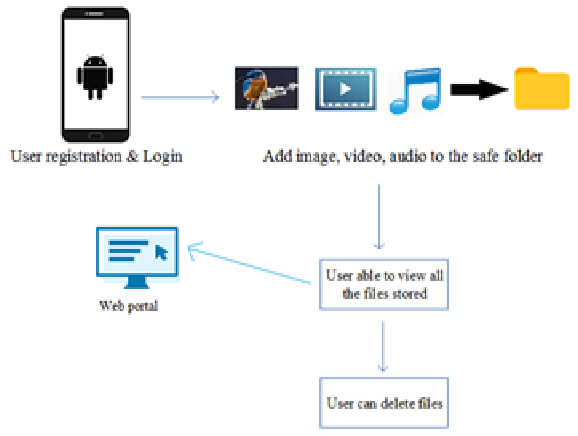
User registration & Login
Add image, video, audio to the safe folder
Web portal
User able to view all the files stored
User can delete files

Figure 1: Symmetric Encryption Scheme

III. HISTORY

AES, in full Advanced Encryption Standard is a data encryption standard endorsed by US National Institute of Standards and Technology (NIST) as a replacement for DES (Data Encryption Standard). AES offers greater security than DES for data communication and storage purpose.

During the 1990's DES algorithm was not sufficient to protect the data. Although various cipher algorithm techniques were introduced but none of them were sufficient. In 2001, Vincent Rijmen and Joan Daemon's submission, Rijndel(pronounced "Rhine-dahl) was selected as a replacement for DES algorithm.

It provided much more security and prevention from attacks previously effective against both DES and block cipher not succeeded in breaking Rijindel.

IV. WHY SYMMETRIC IS MORE SECURE

Symmetric encryption of data is always preferred over asymmetric encryption. There are many significant advantages of symmetric over asymmetric encryption methods.

Below mentioned characteristics explains why symmetric encryption method is more secured and preferred.

A. Size of cryptogram

Symmetric encryption does not increase the size of the cryptogram (asymptotically), but asymmetric encryption does. If we take the example of RSAES-OAEP in PKCS#1V2 with a 1024-bit key and 160-bit SHA-1 hash, a 1024-bit cryptogram can convey a maximum of 688 bit of useful information. Thus, data enciphered in this way would cost 49% more space to store, or more time to move over a given link.

B. Computational Cost

The public key algorithms known are relatively computationally costly compared with most symmetric key algorithms of apparently equivalent security.The difference factor is the use of typically quite large keys. This has important implications for their practical use. Most are used in hybrid cryptosystems for reasons of efficiency – in such a cryptosystem, a shared secret key ("session key") is generated by one party, and this much briefer session key is then encrypted by each recipient's public key. Each recipient then uses the corresponding private key to decrypt the session key. Once all parties have obtained the session key, they can use a much faster symmetric algorithm to encrypt and decrypt messages. In many of these schemes, the session key is unique to each message exchange, being pseudo-randomly chosen for each message.

C. Performance

On a modern CPU with hardware AES support, encryption or decryption speed is over 2000 megabyte/second (per core); while decryption of a 1024-bit cryptogram in the above scheme can perhaps run at 4000 per second (per thread of a comparable CPU), thus a throughput of 0.4 megabyte/second, 5000 times slower; that's also more less the ratio of power usage. That ratio tends to get even worse as security increases. While there are more efficient schemes, it is safe to say that a symmetric scheme is orders of magnitude faster and less power hungry than an asymmetric one, at least for decryption (some asymmetric schemes, including RSA with low public exponent, are considerably faster on the encryption side than they are on the decryption side, and can approach the throughput of some symmetric cryptography).

D. Security

- Some encryption schemes can be proven secure based on the presumed difficulty of a mathematical problem, such as factoring the product of two large primes or computing discrete algorithm. Note that "secure" here has a precise mathematical meaning, and there are multiple different (meaningful) definitions of what it means for an encryption scheme to be "secure". The "right" definition depends on the context in which the scheme will be deployed.
- Another application in public key cryptography is the digital signature. Digital signature schemes can be used for sender authentication and non-repudiation. The sender computes a digital signature for the message to be sent, then sends the signature (together with the message) to the intended receiver. Digital signature schemes have the property that signatures can be

computed only with the knowledge of the correct private key. To verify that a message has been signed by a user and has not been modified, the receiver needs to know only the corresponding public key. In some cases (e.g., RSA), a single algorithm can be used to both encrypt and create digital signatures. In other cases (e.g., DSA), each algorithm can only be used for one specific purpose.

- To achieve both authentication and confidentiality, the sender should include the recipient's name in the message, sign it using his private key, and then encrypt both the message and the signature using the recipient's public key.

This security characteristic of symmetric encryption makes is more preferred than asymmetric encryption for large amount of data. Symmetric encryption technique can be developed to make it more secure for data encryption over secure server.

V. ADVANTAGES OF AES

AES data encryption is a more efficient and elegant cryptographic algorithm, but its main advantages rests in the option for various key lengths. AES allows you to choose different key bits such as 128-bit, 192-bit or 256-bit key, making it exponentially stronger than the 56-bit key of DES. In terms of structure, DES uses the Feistel network, which divides the block into two halves before going through the encryption steps. AES on the other hand, uses permutation-substitution, which involves a series of substitution and permutation steps to create the encrypted block.

AES has some drawbacks of slow processing and it takes the large time of data transferring, so to increase the speed of the process of the AES algorithm, we apply AES algorithm in parallel. There is a file of 5 megabit (5242880 bits) which requires to be sent from sender to receiver. Here using a 128 bit AES algorithm the number of steps expected will be 5242880/128=40960. This means 40960 data blocks will be made on which AES will be used individually. However, using the parallel approach of AES algorithm, the number of steps required will be 40960/64=640 where 64 is number of processors.

AES works in the following Steps.

Each round consists of the following four steps:

1) SubBytes Transformation: In this operation, the Substitution Box replaces each byte of the state matrix with another byte by substitution method. For the generation of the S-box the respective reciprocal of that byte in GF (2^8) is calculated and then affine transform is applied on it2
2) ShiftRows Transformation: In this operation, there is no change in the bytes in the first row of the state. There is a cyclic shift of the second, third, fourth and fifth rows to the left by one, two, three and four bytes respectively.
3) MixColumns Transformation: In this operation, the bytes in each column are mixed by the multiplication of the state using a fixed matrix of polynomial. It thus fully changes the setting of the cipher text even if all bytes look similar in appearance. There is no Inverse Polynomial Matrix to reverse the mix column operation of transformation.
4) AddRoundKey Transformation: In this transformation, there is an addition of a roundkey to the state by bitwise XOR (exclusive-OR) operation. This operation proceeds on column by column at a time. There is also an addition of a round key word with every state of the column matrix. The operation thus performed in this last segment of AES is addition of matrix.

VI. CONCLUSION

AES, which is a symmetrical encryption algorithm, uses a series of table look-ups to increase its efficiency of performance. Since these tables do not fully occupy into the cache, cache hits and misses are common during the encryption process, which causes various look up times, and encryption times that change according to the input text and the encryption key.

In this work, an improved AES algorithm will used to encrypt plaintext and ECC algorithm is then applied to encrypt the AES key thereby increasing overall security of the system by implementing software-based countermeasures to prevent possible vulnerabilities posed by the timing side channel attack. For further increasing the efficiency of the encryption of data, a higher order of AES will be implemented having the key size of 128-bit and along with the demonstration of 192 bit and 256-bit key sizes. The purpose is to provide high quality data encryption without any compromise using AES symmetric data encryption.

REFERENCES

[1] Roselin Selvarani and Dr. T. N. Ravi "A Review on Role of Encryption in Mobile Database Security" Volume 3, Issue 12, December 2014

[2] Medhavi S.Shriwas,Neetesh Gupta,Amit Sinhal Effeicient Method for Backup and Restore Data in Android, International Conference on Communication Systems and Network Technologies,2013.

[3] Vittorio ottaviani, alessandro lentini, antonio grillo, silvia di cesare and giuseppe f. Italiano, "shared backup & restore save, recover and share personal information into closed groups of smart phones" 2011 IEEE.

[4] Yu Chen and Paolo Gasti "Breaking and Fixing the Self Encryption Scheme for Data Security in Mobile Devices" 2010 18th Euromicro Conference on Parallel, Distributed and Network-based Processing

[5] Yu Chen and Wei-Shinn Ku "Self-Encryption Scheme for Data Security in Mobile Devices" Oct. 8, 2008 to CCNC'09, Las Vegas, NV, USA, Jan. 10 – 13, 2009.

[6] Hiroki Endo, Yoshihiro Kawahara, and Tohru Asami "A Self-Encryption Based Private Storage System over P2P Distributed File Sharing Infrastructure"