Comparison of Encryption Algorithms implemented in FOTA packaging tool for Software Updation in ECUs

Srayan Sankar Chatterjee[1], Prof. Subrahmanya K.N[2]

[1]*Student, Electronics and Communication Engineering Department, R.V College of Engineering, Karnataka, India*
[2]*Assistant Professor, Electronics and Communication Engineering Department, R.V College of Engineering, Karnataka, India*

---***---

Abstract - *In an increasingly digital world, where electronic devices have formed an integral part of our lives one of the most important factors is user satisfaction or user experience. This can be enhanced in multiple ways. One of them is through Firmware Over The Air (FOTA) update. FOTA ensures that customers don't have to visit service centers on a regular basis to get the device software updated, instead it can be done remotely according to their comfort. Encrypting the update package is an important part of the packaging tool used in FOTA. This article compares 2 widely used encryption algorithms in the context of FOTA packaging tool.*

Key Words: *FOTA, packaging tool, encryption, AES, RSA, wireless updates*

1. INTRODUCTION

Devices of any kind must undergo maintenance and upgradation procedures at regular intervals for proper functioning. Some of these electronic devices are mobile in nature, for example cell phones, laptops, even the ECUs used in modern vehicles. Now, if we were to take such devices to some service centre, so that a technician can upgrade them, then it would be a time and money consuming process. Considering the importance of such devices in our day-to-day lives we can say that this would be an unnecessary ordeal. For solving this problem we have Firmware Over the Air(FOTA)

As the name suggests this is used for transmitting software updates wirelessly over the air to the respective electronic device. Thus the requirement of technicians is eliminated allowing hassle free updation of these devices.

There are multiple aspects of FOTA. One of them is the packaging tool used for developing the packets in which the software updates are to be sent. The packaging tool has to deal with creating the package, ensuring proper compression so that speedy transmission can take place and also securing the package to prevent third party attacks that can corrupt the package.

The main focus of this article is to compare encryption algorithms on different parameters that are used for securing these update packages. For this purpose a symmetric key technique-Advanced Encryption Standard(AES) and an asymmetric key technique- Rivest Shamir Adlemen (RSA) have been compared.

1.1 Motivation

There is a recurring need to fix the bugs present in the software present in devices manufactured. Operating system installed in the device when available for years on their policy, also have to be refreshed from time to time. Sometimes manufacturers have to give some new software features because of the market trends as well. That's why FOTA is needed. Therefore, to maintain the integrity of the update package being transmitted and to ensure that users receive the updates as early as possible it is essential to encrypt it using the fastest possible technique.

1.2 Review of Existing Literature

The paper [1] discusses about the basic idea ECU in detail. Along with that, designs of various components of ECU are also shown. An independent closed loop pressure control without expensive add-in sensors was proposed in this article. Therefore, to get an overall idea of the domain this article has been very useful.

Authors [2] proposed a very practical and effective method to convert any Attribute Based Encryption (ABE) scheme with non-verifiable outsourced decryption to an ABE scheme without much hassle. From the experimental results it was concluded that the proposed method introduces little to no overhead in exchange of verifiability. Thus confirming that the method is optimal in its usage.

In the paper [3] the authors gave a brief idea about FOTA, along with that the authors also agreed that the FOTA system can provide a safe and reliable network to perform remote firmware upgrade tasks without affecting the work efficiency of the ECUs. A few encryption algorithms like AES, ECC were discussed and the method for implementation of such algorithms was also shown.

2. ENCRYPTION ALGORITHMS

In Cryptography the original data which the sender wants to send is called 'Plain text' and Cipher text is the encrypted format of the plain text. The plain text is converted to the Cipher text using encryption algorithms and the reverse is done using decryption algorithms. These algorithms are mainly classified into two types symmetric key algorithm and asymmetric key algorithm. As mentioned before AES algorithm from symmetric key encryption will be compared with RSA algorithm from asymmetric key encryption.

2.2 Symmetric Key Encryption

In this method the sender and the receiver share the same key for encryption and decryption, which has to be kept secured by both the parties. There are different Symmetric Key Encryption techniques like Data Encryption Standard (DES), Triple Data Encryption Standard (3DES), and Advanced Encryption Standard (AES).

Advanced Encryption Standard (AES):
Blocks of three different sizes 128 bit, 192 bit and 256 bits are utilized in the execution of AES. 128 bit uses 10 rounds, 192 bit has 12 rounds and 256 bit consist of 14 rounds. Every round has to go through a series of steps namely, substitution byte, shift rows, mixed columns and add round key. AES encrypted files cannot be easily cracked by attackers by using brute force technique. However, a combination of boomerang and rectangle can break the round versions of AES but not complete AES. This happens mainly due to the S-BOX which is a somewhat vulnerable entity.

2.2 Asymmetric Key Encryption

This method uses two keys, 'Private Key' and 'Public key'. The plain text is encrypted by the sender into cipher text with the 'public key' before transmission. This cipher text is then decrypted by the receiver with the help of 'private key'.

Rivest Shamir Adlemen (RSA):
Rivest, Shamir and Adlemen developed the algorithm in 1977. The sender utilizes the Public key to encrypt the plain text, on the receiver end private key is required to decrypt the message. This private key, is known only to the receiver. The RSA has been used to ensure message integrity, privacy, authentication and non-repudiation in various sectors, particularly e-commerce.

3. DESIGN METHODOLOGY

A FOTA packaging tool was designed using Tkinter library in Python 3.9. This packaging tool generated a Zip File as the update package that will be transmitted wirelessly. However, the zip file has to be encrypted before transmission. Therefore, to choose the faster encryption algorithm files of different sizes(1 MB, 5 MB, 10 MB) were first encrypted AES encryption and again with RSA encryption. The runtime for encryption was noted.

Then binary files of various sizes (1 MB, 5 MB, 10 MB) were encrypted using AES and RSA algorithms and their runtimes were noted again.

Finally, the same procedure was also repeated for image files of similar sizes. The simulation was conducted on a laptop with Windows 64bit, processor i3 and CPU 1.90GHz with 8GB of RAM.

The results have been tabulated in Table 1,2 and 3.

Table -1: Runtime for File Encryption

File Size	AES	RSA
1 MB	76 msec	420.2 msec
5 MB	365.2 msec	1698.43 msec
10 MB	674.92 msec	3011.87 msec

Table -2: Runtime for Binary File Encryption

File Size	AES	RSA
1 MB	71.3 msec	387.24 msec
5 MB	387.6 msec	1768.7 msec
10 MB	704.82 msec	3441.12 msec

Table -3: Runtime for Image File Encryption

File Size	AES	RSA
1 MB	98.6 msec	474.52 msec
5 MB	201.2 msec	1022.6 msec
10 MB	290.63 msec	1432 msec

4. INFERENCE FROM SIMULATION RESULTS

It can be inferred that AES is faster than RSA algorithm. In fact, it is quite visible that AES is almost five times faster than RSA. This is mainly due to the fact that AES uses symmetric key encryption technique, which means it uses the same key for encryption as well as decryption. On the other hand RSA is an asymmetric key encryption technique. Therefore, it takes longer time to encrypt these files. RSA algorithm does take longer time to encrypt and decrypt but in the process it provides strong security due to the usage of two different keys for encryption and decryption. This doesn't mean that AES encryption technique is weak and provides less security. In fact it can be estimated that average time taken for all PCs on earth, working together, to brute force crack AES-256 is: 13,689 trillion trillion trillion trillion years, which is more than the current estimated age of the universe!

5. CONCLUSIONS

It is concluded that AES is faster than RSA when it comes to encrypting different types of files as the runtime for encryption using RSA is almost five times than using AES.

Therefore, for faster generation of update packages using FOTA packaging tool, AES is recommended. Even though the paper has not compared the level of security provided by the two algorithms, it can be said that AES is one of the most secure encryption algorithms available for industrial purpose.

ACKNOWLEDGEMENT

The authors would like to thank Mr. Praneeth Kumar Kotagiri and Mr. Praveen Basappa from Robert Bosch Engineering and Business Solutions for providing the necessary infrastructure to conduct the study.

REFERENCES

[1]. V. Vasquez Lopez, J. M. Echeverry Mejia and D. E. Contreras Dominguez, "Design and Statistical Validation of Spark Ignition Engine Electronic Control Unit for Hardware-in-the-Loop Testing," in IEEE Latin America Transactions, vol. 15, no. 8, pp. 1376-83, 2017, doi: 10.1109/TLA.2017.7994782.

[2]. B. Qin, R. H. Deng, S. Liu and S. Ma, "Attribute-Based Encryption With Efficient Verifiable Outsourced Decryption," in IEEE Transactions on Information Forensics and Security, vol. 10, no. 7, pp. 1384-93, July 2015, doi: 10.1109/TIFS.2015.2410137.

[3]. A. Cheng, J. Yin, D. Ma and X. Dang, "Application and Research of Hybrid Encryption Algorithm in Vehicle FOTA System," 2020 Chinese Control and Decision Conference (CCDC), 2020, pp. 4988-4993, doi: 10.1109/CCDC49329.2020.9164481.