# ENHANCEMENT OF ADVANCED BROWSER SECURITY FOR ANDROID DEVICES USING ADVANCED SECURITY STANDARDS ALGORITHM

T. Senthil Kumar, S. Prabakaran, Koshtubh Mohata and Kartik Vaishnav
Department of Computer Science and Engineering, SRM University, Kattankulathur, Chennai, India
E-Mail: senthilkumar.t@ktr.srmuniv.ac.in

## ABSTRACT

Mobile browsers use client side efficiency measures such as larger cache storage and fewer plugins. However, the impact on data security of such measured is an understudied area. This paper reveals a method to scrutinize the security of lightweight browsers. Using this opposing model, we disclose previously unknown vulnerabilities in the following popular light browsers, namely: UC Browser, Dolphin, CM Browser, and Samsung Stock Browser, which allows an attacker to obtain unauthorized access to the user's private data. The latter include browser history, email content, and bank account details. The main issue with the storage remains the fact that all the cache is stored in the browser's folder in the external memory card of the android device. With the way permissions work in android, this cache folder can easily be accessed by the other apps with the same permissions. Therefore, it is advisable that all the sensitive browser cache be stored in the internal memory while large files such as video clips and images can be stored in the external memory if memory space is a constraint. It is usually advisable to wipe the cache stored by the applications, but in this case the utility of the browser will cease. Thus, to make the cache more secure we propose to encrypt the data using a Java crypto and implement the Advanced Encryption Standard or AES. This encryption standard is a symmetric block cipher and most hardware and software vendors use this standard for protecting the sensitive data.

**Keywords:** mobile browser, security, storage, encryption, decryption, AES, internal memory, java, android, wireless.

## 1. INTRODUCTION

Data stored in the device is worth more than the device. This raises a point that the data must be so secured that no other can access the data. There are certain hacking techniques like Key logger, Denial of Services, Waterhole attacks, Fake WAP, Phishing, etc. which can lead to loss of privacy of data. To overcome most of the hacking techniques and to make the data limited to an individual it is best to use certain Encryption techniques which can secure the data and to break certain encryption algorithms, it would take years for a hacker or hacker to decrypt the data and extract information especially for large files. Of the most encryption algorithm, the one which provides the best security is Advanced Encryption Standards Algorithm which encrypts the text of the file to 16-bit which would take decades for a hacker to decrypt the data.

Advanced Encryption Standards enables the user to be care free for the data hosted on the cloud storage. When the user downloads the file from cloud storage or when the user creates any file in the device then the file is being stored with the data encrypted with in it. Now looking from the attacker's point of view, the attacker can spoof the IP address of the target phone and will try to download all the important documents. At this point of time when the attacker tries to attempt to access the files, the attacker would see all the data as encrypted 16-bit long which he cannot decode and apparently destroying the purpose of hacking. There are other encrypting techniques as well like ASCII encryption, binary conversion of data, etc. but the Advance Encryption Standards is the perfect algorithm for mobile browser security scenario. The AES Algorithm converts the text of the file to its equivalent Hex-decimal form which almost makes impossible to hack the data. The AES algorithm is not in entirety a computer program rather a mathematical portrayal of a process of concealing data. Cloud technology users are on the rise and the need for encryption for protecting data is increasing by the minute. Although this is the case, the clients should make sure that the encryption solutions that they prefer are within the vendors' visibility and have been through NIST validation. Also, these solutions need to be based on proper standards and AES is one such method.

It is usually advisable to wipe the cache stored by the applications, but in this case the utility of the browser will cease. Thus, to make the cache more secure we propose to encrypt the data using a Java crypto and implement the Advanced Encryption Standard or AES. This encryption standard is a symmetric block cipher and most sought out for encryption of important data in the technological world.

## 2. METHODOLOGY

The Advanced Encryption Standard uses a strategical process which is the substitution-permutation network. This is a blend of substitution and permutation. The main benefit of this network is that it is swift in both software and hardware. The Data Encryption Standard was the antecedent of the Advanced Encryption Standard and it used the Feistel method while the latter does not use it, rather is based on the Rijndael method. This flow uses a fixed block size of 128 bits. The key size may differ and can be any of 128, 192 or 256 bits. The requisite for the Rijndael method is that the block and key size should be a multiple of 32 bits. Also, the minimum size should be 128 bits and maximum of 256 bits. AES functions on a particular 4x4 matrix termed as state which can also be larger based on the requirements. The system block diagram shown in Figure-1.

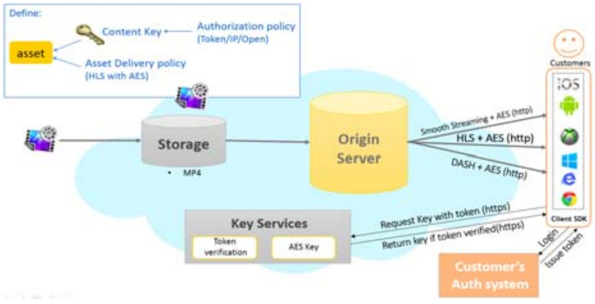

**Figure-1.** Architecture of AES.

### a) Light weight browsers

Light Weight Browser in android is a necessary parameter when considering the constraints like space and time complexity. These browsers consume very less space and the response time is also quick. These kinds of browsers are very much helpful in saving the storage space and help in maintaining the performance of the device. These browsers are also considered as the best testing platforms. In order to determine such browsers for our purpose we create a web view application and as soon as we hit any URL from this page a temporary file gets generated which gets stored in the internal storage of the phone.

### b) AES architecture

While the looping architecture uses feedback of data for each round and is implemented by Gaj, the fully unrolled pipelined architecture uses pipeline registers in the time unrolled stages of AES and is implemented by Sages *et al*. Deep sub pipelined fully unrolled architecture is done by further dividing each stage by pipeline registers and is implemented by I Hammed *et al*. Research has also been done on efficient utilization of area and speed(defined as frequency and calculated as inverse of delay) in hardware.

### c) Cache

Android is flexible with storage options and provides a plethora of storage solutions. The solution chosen depends on how the application may require the particular storage access. This logic is documented by the developer and may range to the internal storage or the external storage. The data may also be specified private to the application or may be available to the other applications for their usage. The solution may also depend on the amount of storage the application may need. Cache may be defined as a component of software or hardware which stores data for further use by the system which yields faster computation. The browser uses a cache to load a URL faster and may amount to large data. The cache is collected over a period and may be erased if the device is low on memory or if the data isn't required.

However, the system may not clean the cache on its own and may at times require manual clean up. When an application is uninstalled, the corresponding application data and the cache must be cleaned so that this storage can be used by other applications. It is also a good practice to limit the cache for an application so that only sufficiently important data is only stored.

### d) JAVA language as software

With the recent boom in the smartphone market, android application development is on the rise. Many methodologies have been presented to build applications but the most recommended way is to develop native applications using Java and the Android SDK. This method may differ a bit from developing Java applications but may also have some similarities in generic methods.

### e) Data encryption

```
public static String encrypt (String Data) throws Exception
{
      Key key = generateKey();
      Cipher c = Cipher.getInstance(ALGO);
      c.init(Cipher.ENCRYPT_MODE, key);
      byte[] encVal = c.doFinal(Data.getBytes());
      String encryptedValue = new
BASE64Encoder().encode(encVal);
      return encryptedValue;
   }
```

Data encryption is the important part of this project as the complete application revolves around this concept. We encrypt the data to 128-bit long string which is also possible that data can be encrypted to 156 bit long or 192-bit long key. At the end of time data is decrypted back to its original state so that the readers can get a proper view of the data without any inconsistent view of the data. Data encryption helps in maintaining the privacy of the data and the encrypted data is next to impossible to decode it back as the AES uses the technique of confusion and diffusion. The AES structure replaces the the bits in key value at regular intervals which cannot be predicated and the permutation logic also known as diffusion technique helps in maintaining the uniqueness of the Key which is provides the leading edge over other encryption algorithms.

### f) Storing files in cache memory

```
String filename = "samplefile";
String string = "Hello world!";
FileOutputStream outputStream;
try {
     outputStream = openFileOutput(filename,
     Context.MODE_PRIVATE);
     outputStream.write(string.getBytes());
     outputStream.close();
}
catch (Exception e) {
     e.printStackTrace();
}
```

The cache is an isolated area for the application and is accessible by the application operating on it. This area ceases to exist when the application is uninstalled by the client. A typical Android handheld has two sections for memory: the internal and external storage. These sections were in use when early smartphone pioneers used to split the storage into an internal storage section and a removable external storage. Various devices split the internal storage into internal and external storage where the internal is used only for the installation of applications. This results in an abstraction and the API acts similarly whether there is a removable storage or not.

### g) Application of AES algorithm

The Advanced Encryption Standard is based on the Rijndael method and uses a substitution-permutation method. In this progression, some of the input bits are replaced by definite outputs while the other bits are transposed, hence the name substitution-permutation network.

The AES Algorithm also helps in maintaining consistent data and the data can never be lost until it has been erased by user himself. Advantage of using this algorithm is, it can work with any Light Weight Browsers as well as Heavy Browsers. Encryption of the data doesn’t take much time as well as it never consumes more space of device. This algorithm can be implemented within the browser itself and can be called to encrypt the files present in the local storage. As soon as the user wants to get the view of the data again they can call the data by decrypting the data.

### h) Data decryption

```
public static String decrypt(String encryptedData) throws
Exception {
    Key key = generateKey();
    Cipher c = Cipher.getInstance(ALGO);
    c.init(Cipher.DECRYPT_MODE, key);
    byte[] decordedValue = new
BASE64Decoder().decodeBuffer(encryptedData);
    byte[] decValue = c.doFinal(decordedValue);
    String decryptedValue = new String(decValue);
    return decryptedValue;
  }
  private static Key generateKey() throws Exception {
    Key key = new SecretKeySpec(keyValue, ALGO);
    return key;
}
```

Data is decrypted back while the time of data retrieval and the user can again get the consistent view of data.

## 3. RESULTS AND DISCUSSIONS

The Advanced Encryption Standard, although the most secure encryption technique can be cracked but the practicality of that is pretty low. Researchers have tried to find vulnerabilities and have come close to only cracking some rounds of the algorithm. The number of steps required for the 128-bit key is an eight followed by thirty-seven zeroes. This is practically not possible at the moment and hence the result that AES is not in immediate danger. The AES algorithm follows the following flow for encrypting data and the number of rounds depend on the key size.

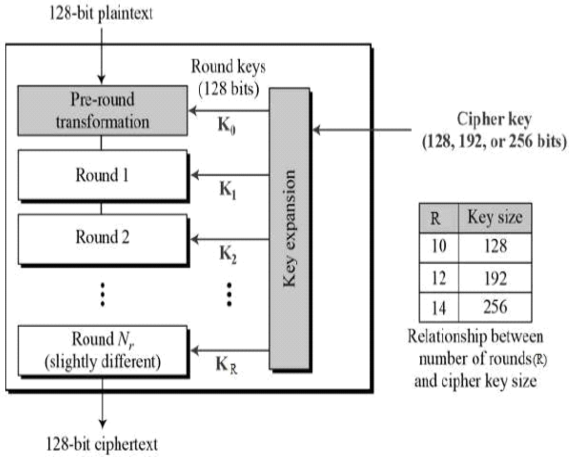

**Figure-2.** AES flow diagram for encryption.

## 4. CONCLUSIONS

Thus, to implement security to the cache files present in the storage of device which is coming from the cloud storage the AES Algorithm is the perfect fit which encrypts the data to 16 bytes long which is almost impossible to hack and this technique will find its way in future and can be implemented in the future to all the browsers coming for the device. This proposed technique will destroy the hazards related to hacking of data files from an individual's device.

## 5. FUTURE ENHANCEMENTS

Therefore, in future, to make the data more secure and preventing the harmful threats of hacks this logic finds its best way to be implemented for all browsers soon. Advanced Encryption Standards make the data so secured that even the best attackers would take decades to solve the Algorithm and retire the data. Thus, this algorithm resolves the security issues of the light weight browsers as well as the heavy browsers. The end to end data is never lost and is decoded back during the time of data retrieval by the trusted user. The main purpose of browser security is achieved in all standards without any compromise between the actual data and the size consumed by the data and browser of the storage (local or external).

## REFERENCES

[1] “Improving mobile device security with operating system-level virtualization”, Sascha Wessel, Manuel Huber, Frederic Stumpf, Claudia Eckert.

[2] “Mobile cloud security: An adversary model for lightweight browser security”, Sashi Pokharel, Kim-Kwang Raymond Choo, Jixue Liu.

[3] “A new approach to mitigating security risks of phone clone co-location over mobile clouds”, Seyed Yahya Vaezpour, Rui Zhang, Kui Wu, Jianping Wang, Gholamali C. Shoja.

[4] B. Thompson, S. Haber, W. G. Horne, T. Sander, and D. Yao, Privacy-preserving computation and verification of aggregate queries on out- sourced databases," in Proc. of the 9th Int. Symposium on Privacy Enhancing Technologies, ser. PETS '09, 2009, pp. 185{201.

[5] http://www.code2learn.com/2011/06/encryption-and-decryption-of-data-using.htm

[6] “http://www.code2learn.com/2011/06/encryption-and-decryption-of-data-using.html"l {AES Algorithm supporting code}

[7] https://www.sans.org/course/mobile-device-security-ethical-hacking

[8] https://www.wikipedia.org/AES-Algorithm

[9] https://heimdalsecurity.com/blog/ultimate-guide-secure-online-browsing/

[10] “Cryptography and Network Security: Principles and Practice” by William Stallings