A Comparative Evaluation of Cryptographic Algorithms

Er. Kumar Saurabh  
Assistant Professor CS &IT department  
Ramgarhia Institute of Engg & Tech  
Phagwara, Punjab, India. Email: Saurabh.er@gmail.com

Er. Ravinder Singh Mann M.tech in CSE  
Ramgarhia Institute of Engg & Tech  
Phagwara, Punjab, India. Email: Mr.rsmann@gmail.com

Abstract:
Security of information has become an important issue in data communication. Encryption has come up as a solution, and plays a vital role in information security system. This security mechanism uses some algorithms to scramble data into unreadable text which can only be decoded or decrypted by party those possesses the associated key. These algorithms consume a significant amount of computing resources such as CPU time, memory and battery power and computation time. This paper performs comparative analysis of three algorithm; ECC, AES and RSA considering certain parameters such as computation time and complexity of the algorithms. A cryptographic tool is used for conducting experiments. Experiments results are given to analyses the effectiveness of each algorithm

Keywords:  
Encryption, secret key encryption, public key encryption, ECC, RSA, AES, encryption.

I. Introduction:
For secure communication over public network data can be protected by the method of encryption. Encryption converts that data by any encryption algorithm using the ‘key’ in scrambled form. Only user having access to the key can decrypt the encrypted data [4]. Encryption is a fundamental tool for the protection of sensitive information. The purpose to use encryption is privacy (preventing disclosure of information or confidentiality) in communications. Encryption is a way of talking to someone while other people are listening, but such the other people cannot understand what you are saying [6]. Encryption algorithms play a vital role in providing data security against malicious attacks. In mobile devices security is very important and different types of algorithms are used to prevent malicious attack on the transmitted data. Encryption algorithm can be categorized into symmetric key (private) and asymmetric (Public) key [1].

In Symmetric keys encryption or secret key encryption, only one key is used to encrypt and decrypt data. In Asymmetric Keys, two keys are used; private and public keys. Public key is used for encryption and private key is used for decryption (e.g. RSA). Public key encryption is based on mathematical function, computationally intensive and is not very efficient for small mobile devices [10, 5].

The present scenario uses encryption which includes mobile phones, passwords, smart cards and DVDs. It has permeated everyday life and is heavily used by much web application.
A. RSA Algorithm:
RSA stands for Rivest, Shamir and Adleman. RSA is a commonly adopted public key cryptography algorithm [12]. The first, and still most commonly used asymmetric algorithm RSA is named for the three mathematicians who developed it, Ron Rivest, Adi Shamir, and Leonard Adleman. RSA today is used in hundreds of software products and can be used for key exchange, digital signatures, or encryption of small blocks of data. RSA uses a variable size encryption block and a variable size key. The key-pair is derived from a very large number, \( n \), that is the product of two prime numbers chosen according to special rules. Since it was introduced in 1977, RSA has been widely used for establishing secure communication channels and for authentication the identity of service provider over insecure communication medium. In the authentication scheme, the server implements public key authentication with client by signing a unique message from the client with its private key, thus creating what is called a digital signature. The signature is then returned to the client, which verifies it using the server’s known public key [9].

Encryption algorithms consume a significant amount of computing resources such as CPU time, memory, and battery power[2]. This paper examines a method for evaluation performance of various algorithms based upon time and complexity. A performance characteristic typically depends on both the encryption key and the input data. A comparative analysis is performed for those encryption algorithms at different sizes of data blocks, finally encryption/decryption speed.

The paper is organized as follows: Section 1 covers the introduction part. Section 2 covers literature reviews. In section 3 experimental set up design of experiments is covered. In section 4 result analysis is performed. We conclude briefly in section 5.

B. ECC Algorithm:
In 1985, Neal Koblitz and V. S. Miller independently proposed using them for public-key cryptosystems. They did not invent a new cryptographic algorithm with elliptic curves over finite fields, but they implemented existing public-key algorithms, like Diffie-Hellman, using elliptic curves.

Some of the key concepts in ECC are:

1. ECC offers considerably greater security for a given key size.

2. The smaller key size also makes possible much more compact implementations for a given level of security.

ECC stands for Elliptic Curve Cryptography. It contains certain advantages. ECC devices require less storage, less power, less memory, and less bandwidth than other systems. This allows you to implement cryptography in platforms that are constrained, such as wireless devices, handheld computers, smart cards, and thin-clients. It also provides a big win in situations where efficiency is important. For example, the current key size recommendation for legacy public schemes is 2048 bits. A vastly smaller 224-bit ECC key offers the same level of security. This advantage only increases with security level— for example, a 3072 bit legacy key and a 256 bit ECC key are equivalent—something that will be important as stronger security systems become mandated and devices get smaller. Elliptical curve cryptography is a public key encryption technique based on elliptic curve theory that can be used to create faster, smaller, and more efficient cryptographic keys. ECC generates keys through the properties of the elliptic curve equation instead of the traditional method of generation as the product of very of large prime numbers like in RSA. Because ECC helps to establish equivalent security
with lower computing power and battery resource usage, it is becoming widely used for mobile applications. Unlike other popular algorithms such as RSA, ECC is based on discrete logarithms that are much more difficult to challenge at equivalent key lengths.

Moreover, due to the apparent hardness of the underlying elliptic curve discrete logarithm problem (ECDLP), ECC systems are also suited for applications that need long-term security requirements. This requires much less processing while at the same time being much harder to crack. For instance, a 256-bit ECC key is as secure as a 3,072-bit RSA key. An elliptic curve E over a field F is defined by the Weierstrass equation:

\[ y^2 = x^3 + ax + b \]

C. AES Algorithm:
To provide security AES uses types of transformation. Substitution permutation, mixing and key adding each round of AES except the last uses the four transformations [11]. AES is a specification for the encryption of electronic data. It has been adopted by the U.S. government and is now used worldwide. It supersedes DES. The algorithm described by AES is a symmetric-key algorithm, meaning the same key is used for both encrypting and decrypting the data. AES was announced by National Institute of Standards and Technology (NIST). Originally called Rijndael, the cipher was developed by two Belgian cryptographers, Joan Daemen and Vincent Rijmen, and submitted by them to the AES selection process. The name Rijndael is a play on the names of the two inventors. Strictly speaking, AES is the name of the standard, and the algorithm described is a (restricted) variant of Rijndael. However, in practice the algorithm is also referred to as ”AES” (a case of totum pro parte). AES is based on a design principle known as a substitution-permutation network. It is fast in both software and hardware. Unlike its predecessor, DES, AES does not use a Feistel network.

AES uses 10, 12, or 14 rounds. The key size that can be 128,192 or 256 bits depends on the number of rounds. AES uses several rounds in which each round is made of several stages. AES has a fixed block size of 128 bits and a key size of 128, 192, or 256 bits, whereas Rijndael can be specified with block and key sizes in any multiple of 32 bits, with a minimum of 128 bits. The block size has a maximum of 256 bits, but the key size has no theoretical maximum. AES operates on a 4×4 column-major order matrix of bytes, termed the state (versions of Rijndael with a larger block size have additional columns in the state). Most AES calculations are done in a special finite field.

The AES cipher is specified as a number of repetitions of transformation rounds that convert the input plaintext into the final output of cipher text. Each round consists of several processing steps, including one that depends on the encryption key. A set of reverse rounds are applied to transform cipher text back into the original plaintext using the same encryption key.

II. Literature Review:
It was shown in [7] that RSA consumes more time than AES and DES.
It is seen that ECC consumes longest time than AES and RSA. RSA consumes less time than all of these.
It is found that ECC has more complexity among ECC, AES and RSA algorithms. AES have more complexity than RSA.

III. Experimental Design:
AES, RSA and ECC algorithms are run in C++. The files created are in form of algo.aes.cpp, algo.rsa.cpp, algo.ecc.cpp. These files are converted to object files of NS2. These files are extracted in Fidora using NS2. Here the comparison of three algorithms is performed.
The object files that are in a form algo.aes.o, algo.rsa.o and algo.ecc.o are run in NS2 and are converted to algo.rsa.tcl, algo.aes.tcl and algo.ecc.tcl form.

Evaluation Parameters:
Performance of encryption algorithms AES, RSA, ECC is evaluated considering the following parameters
   a. Time consumption
   b. Complexity
The time consumption is the time that an encryption algorithm takes to produce a cipher text from a plain text.

IV Experimental Results and Analysis:
Experimental result for encryption algorithm AES, RSA and ECC shows that RSA is most efficient algorithm among the three algorithms.
By analysing Figure 1, Which shows the computation time of three algorithms. It is noticed that RSA consumes least time and ECC consumes longest time for encryption?

Table 1: Value of complexity

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Value of complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSA</td>
<td>3.325231969999999</td>
</tr>
<tr>
<td>AES</td>
<td>3.565231969997</td>
</tr>
<tr>
<td>ECC</td>
<td>3.820740650000003</td>
</tr>
</tbody>
</table>

V Conclusion:
Algorithms play a vital role in Cryptography. Where these are concerned with the security of data, time consumption and complexity are the major issues of concern. The algorithms named AES, RSA and ECC are compared for their performance taking in account the time and complexity factors.
Based on experimental result it was concluded that ECC has more complexity as compared to AES and RSA.
ECC spent largest amount of time in encryption as compared to RSA and ECC.
ferences:
[3] Erik Ol son, “encryption for mobile computing”