Comparative Study and Hardware Implementation of Various Cryptographic Algorithms

Prashant Barthwal
Dept. Computer Science, THDC-Institute of Hydropower Engineering and Technology,
Tehri, Uttarakhand, India
pbarthwal9@gmail.com

Swati Sharma
Dept. Computer Science, THDC-Institute of Hydropower Engineering and Technology,
Tehri, Uttarakhand, India
swatu240496@gmail.com

Vivek Kumar
Dept. Computer Science, THDC-Institute of Hydropower Engineering and Technology,
Tehri, Uttarakhand, India
vivek9837@gmail.com

Abstract--As computing power is increasing by the day, there is a constant threat which is imposed on the security of the digital data. Over the years, either symmetric key, public key or hashing algorithms have been designed using the popular tools and technique for security purpose. Various algorithms have been studied and implemented. MD-5, SHA is used for hashing the message, RSA as signing algorithm and AES and DES for encryption of the overall message. The optimization of the algorithms will provide us with descent throughput. The comparative analysis of the algorithms has been done on various factors.

Keywords: Symmetric key, Asymmetric key, hashing, CPU time, Frequency.

1. Introduction

In today's era of fast computing information is transferred and processed along the network with great speed and precision. Security attacks on digital information is one of the primary concern today. Extensive efforts have been made by the security experts to avoid and prevent the consequence of security breach. To overcome this problem, encryption algorithms were proposed which are used to transform information into a non-recognizable, non-sensible data, which can only be decrypted by the intended user. For this purpose two categories of algorithms are used: Symmetric key, in which same key is used for encryption and decryption and Asymmetric key, in which a pair of private and public keys are used for encryption and decryption. Once information is encrypted, confidentiality is achieved but authentication is also important as in the absence of authentication, the message is vulnerable. For authentication also encryption is done but that is not performed over all of the message but over a small part. This part is hashed using hashing algorithm which converts any size message to constant byte output, a hash uniquely identifies a message.

II. Algorithm

2.1 Algorithms Studied and Implemented

2.1.1 RSA Algorithm

RSA is an asymmetric algorithm which was first proposed by Rivest, Shamir and Adleman, on whose name the algorithm is named. It is widely used in safeguarding the digital data specifically on the internet. RSA is a Public-key algorithm therefore uses two keys.

2.1.2 AES Algorithm

The Advanced Encryption Standard algorithm is a symmetric key algorithm that is implemented in software and hardware to encrypt the digital data. It was designed as a successor for DES, when it became vulnerable to attacks.

DOI- 10.18486/ijcsnt.2017.6.3.01
ISSN-2053-6283

2.1.3 DES Algorithm

Data Encryption Standard is a symmetric key algorithm and is now not considered much secure.

2.1.4 MD-5 Algorithm

The MD5 is a hashing algorithm which is used to produce a hash of 128-bit value. It is used to find checksum to check for data integrity of the corrupted data.

2.1.5 SHA-1 Algorithm

Secure hash algorithm-1 is a hashing algorithm which is used to produce a hash of 160-bit value. The hash value as output is known as the message digest.

2.2 Background Study

In today's world, there are several cryptographic algorithms that are highly efficient in encryption and decryption of message data such as DES, AES, SHA, MD-5 etc. But, the security of the digital information requires both the confidentiality and authentication. Therefore they both must be provided together for better security. So comparative analysis is done on the basis of study and implementation of the algorithms.

2.3 Future Work

The future work may deal with the design and implementation of the verifying corresponding hardware at the receiver side. The crypto-accelerator architecture, will verify the received data for authentication, and once authenticated, will accept the data that was intended to be communicated. It will basically comprise of decryption algorithms for RSA and AES while MD5 algorithm will remain the same. The LUT for AES can be communicated to the receiver who can now decrypt the message by just substituting the encrypted text with their corresponding plaintext. An addition checksum will also be included in the cryptographic algorithms to ensure extra data integrity and security mention in Fig 1-5 related data mention in Table 1.

```
Total REAL time to Xst completion: 926.00 secs
Total CPU time to Xst completion: 925.85 secs

-->
```

Figure1. RSA Algorithm Synthesis

```
Total REAL time to Xst completion: 34.00 secs
Total CPU time to Xst completion: 33.50 secs

-->
```

Figure **2.** AES Algorithm Synthesis

```
Total REAL time to Xst completion: 21.00 s
Total CPU time to Xst completion: 21.82 se

-->
```

Figure 3**.** MD-5 Algorithm Synthesis

```
Total REAL time to Xst completion: 13.00
Total CPU time to Xst completion: 12.93 s
```

Figure 4. SHA Algorithm Synthesis

```
Total REAL time to Xst completion: 27.00 secs
Total CPU time to Xst completion: 26.77 secs
```

Figure 5. DES Algorithm Synthesis

Table **1.** The Comparative Analysis Of Various Cryptographic Algorithms Which Includes Parameters As

Algorithm	CPU time (in	Frequency	Security	Type	Rounds	Algorithm

DOI- 10.18486/ijcsnt.2017.6.3.01
ISSN-2053-6283

	sec)					
RSA	925.85	0.00108	high	Asymmetric	1	RSA
AES	33.50	0.02985	Highly strong	Symmetric	10,12,14	AES
DES	27	0.0371	low	Symmetric	16	DES
MD5	21.82	0.04582	moderate	Hashing	4	MD5
SHA-1	13	0.0769	moderate	Hashing	80	SHA-1

III. Conclusions

All the cryptographic algorithms have an individual implementation and have upgrades to the existing algorithms. These provided one of the components, either confidentiality or authentication. The comparative analysis has been done for the five algorithms i.e. RSA, AES, DES, MD-5 and SHA-1. On the basis of the comparison table and timing summary, it was observed that MD5 had the highest frequency and the least CPU time. Since MD5 has the least CPU time, it will certainly give a better throughput, where throughput can be determined by the following formula:

$$\text{Throughput} = (\text{frequency}/\text{latency}) \text{ x input size}$$

VI. References

[1] D. Narh Amanor,C. Paar, J. Pelzl, V. Bunimov and M. Schimmler, “Efficient hardware architectures for modular multiplication on FPGAs”, Field Programmable Logic and Applications, pp. 539 - 542, 2005.

[2] Vivek Kumar, Ashish Joshi, Banit Negi, “FPGA-based Hardware Architecture of Elgamal Encryption using Carry Save Adder”, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 2, February 2014 ISSN: 2277 128X

[3] Samir Palnitkar, Verilog HDL: A Guide to Digital Design and Synthesis, Publisher: Prentice Hall PTR ISBN: 0-13-044911-3

[4] Rivest R., Shamir, A., and Adleman L, 1978. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. Communications of the ACM.

[5] Stallings W.2003, Cryptography and Network Security: Principles and Practices.

[6] Burnett S. and Paine S, 2001. RSA Security’s Official Guide to Cryptography, McGraw-Hill.