INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS
ISSN 2320-7345

BLOWFISH ALGORITHM ON ITS OWN CLOUD COMPUTER PERFORMANCE AND IMPLEMENTATION

R.Prasanthini[1], M.Jothilakshmi[2]

[1]*Research scholar (Department of computer science) Vivekanandha College of arts and sciences for women, ammu29msc@gmail.com*
[2] *Assistant Professor (Department of computer science) Vivekanandha College of arts and sciences for women, mjothiadi@rediffmail.com*

Abstract

Cloud computing is the concept of implemented make out the daily computing problems. It is basically virtual pool of resources and cloud provides this resource to any users via internet. Cloud computing is the internet based development and used in computer technology, solves many problems of cycles, so the new technology has also created new challenges. Such as data security, data ownership &trans-code data storage. In this paper we will talk about the cloud computing security algorithms. Here various security algorithms, we build a model to analyze and compare the execution time and energy of the blowfish cryptographic algorithms.

Keywords: Cloud computing; Data security; Algorithms: DES, AES, RSA, BLOWFISH

1. Introduction

The concepts of cloud computing is linked into closely with those of Infrastructure as a service(IAAS),Platform as a service(PAAS),software as a service(SAAS).Which are a service oriented architecture, here cloud computing has pay as you go system. Because cloud based solutions rely on user communications across the internet, the solutions are at risks for man-in-the-middle attacks. A good defence against such attacks is to establish a secure (encrypted) connection with the remote server. Cipher text is an art of protecting information by encrypting it into an unreadable. This information can only be possessed by those who possess the secret key that can decrypt the message into original form. As each and every organization is moving its data to the cloud, which means its user the storage service provided by the cloud provide. So there is a need to protect that data against unauthorized access, modification or denial of services.

Cryptography in modern days is considered combination of three types of algorithm. The main aim of cryptography is to take care of data secure from invaders. In encryption key is a piece of information which

states the particular conversion of plaintext to cipher text. The strength of the encryption algorithm bank on the length of the key, secret of the key. There are two Encryption/Decryption key types: first key type of encryption technologies when two end points need to communicate with one another through Encryption. But it is mostly use the same algorithm and same key. Second type of encryption technologies mostly use different algorithm but related keys. Such as symmetric or asymmetric algorithms.

Security Algorithm Used In Cloud Computing

DES (Data Encryption Standard)

The DES is a cipher selected as official Federal information processing standard for the United States in 1976. The algorithm was initially controversial with elements. DES is a short key length, so now considered to be insecure for many applications. This is due to the 56-bit key size and 16 cycle of each 48 bit sub keys are formed being too small. Also some Analytical results which demonstrate theoretical weakness in the cipher fiestel network.

Triple DES

Triple DES was first standardized for use in financial application in ANSI standard X9.17 in 1985.3DES uses three keys and three executions of the DES algorithm. Such as encrypt-decrypt-encrypt (EDE) sequence.

1. $C = E(K_3, D(K_2, E(K_1, P)))$
2. $P = D(K_1, E(K_2, D(K_3, C)))$
3. $C = E(K_1, D(K_1, E(K_1, P))) = E[K, P]$

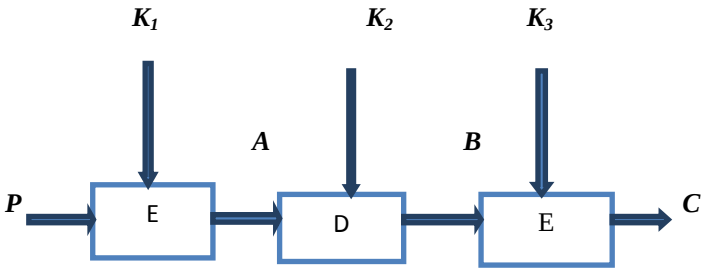

Figure1: 3DES Encryption

AES (Advanced Encryption Standard)

The principal drawbacks of 3DES are not a responsible candidate for long-term use. So AES is a new proposal algorithm for security. AES must be a symmetric block cipher with a block length of 128 bits and support for key length of 128,192 and 256 bits. The input to the encryption and decryption algorithm is a single 128bit block.

Vol.2 Issue.1, Pg.: 132-138
January 2014

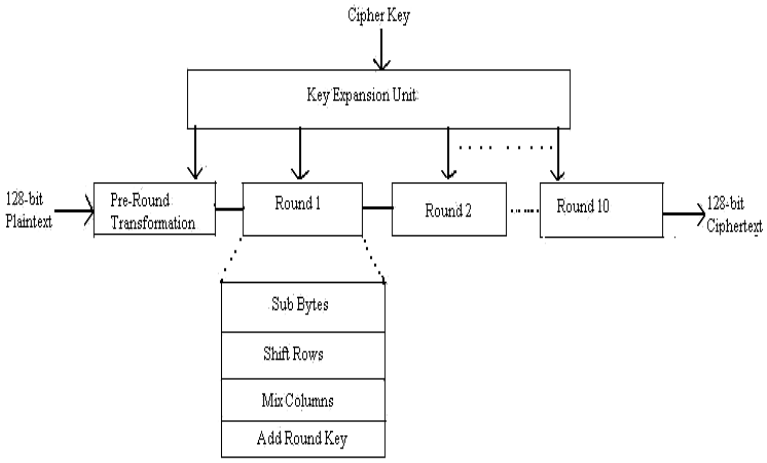

Figure 2: AES

RSA

RSA cryptosystem realize the properties of the multiplicative Homomorphism encryption [9]. Ronald Rivest, Adi Shamir and Leonard Adleman have invented the RSA algorithm and named after its inventors. RSA uses modular exponential for encryption and decryption. RSA uses two exponents, a and b, where a is public and b is private. Let the plaintext is P and C is cipher text, then at encryption

C = Pa mod n
And at decryption side
P = Cb mod n.
N is a very large number, created during key generation process.

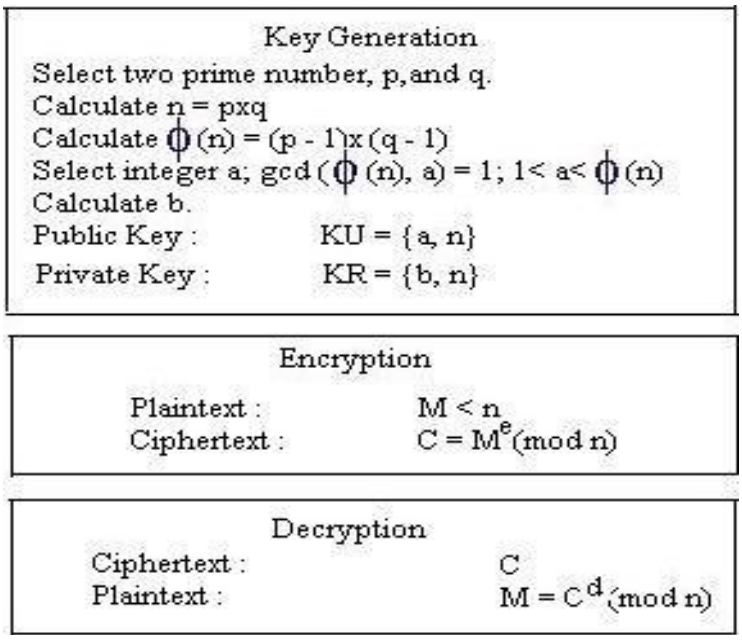

Figure 3: RSA

Vol.2 Issue.1, Pg.: 132-138
January 2014

Blowfish Algorithm

Blowfish was designed in 1993 by *Bruce Schnier* as a fast free alternative to existing encryption algorithm. Blowfish is a symmetric block cipher that can be effectively used for encryption and secure of data. It takes variable length key from 32 bits to 448 bits making it ideal for securing data.

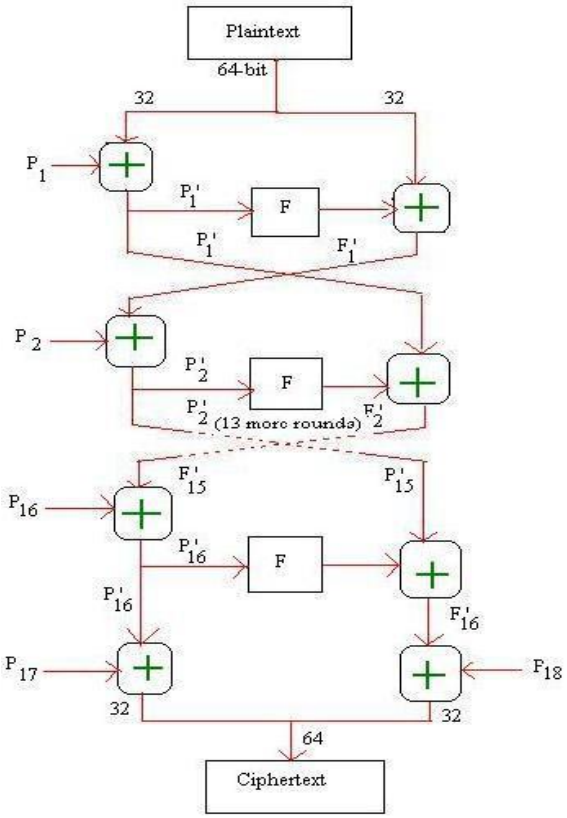

Figure 4: Blowfish Algorithm

Encryption

Blowfish has 16 rounds.
The input is a 64-bit data element, x.
Divide x into two 32-bit halves: xL, xR.
Then, for i = 1 to 16:
xL = xL XOR Pi
xR = F (xL) XOR xR
Swap xL and xR
After the sixteenth round, swap xL and xR again to undo the last swap.
Then, xR = xR XOR P17 and xL = xL XOR P18.

Finally, recombine xL and xR to get the cipher text.
Pocket Brief 5 of 7.Decryption is exactly the same as encryption, except that P1, P2..., P18 are used in the reverse order. Implementations of *Blowfish* that require the fastest speeds should unroll the loop and Ensure that all sub keys are stored in cache.

The Blowfish algorithm consists of two parts: a key-expansion part and a data-encryption part. Key expansion converts a variable-length key of at most 448 bits into several sub key arrays, totalling 4168 bytes. Blowfish algorithm is a fiestel network, iterating a simple encryption function 16 times. The block size is 64 its, and the key can be any length up to 448 bits.

- P-box: Permutation box that performs bit shuffling;
- S-box: Substitution box for non-linear functions;
- XOR: Logic function to achieve linear mixing.

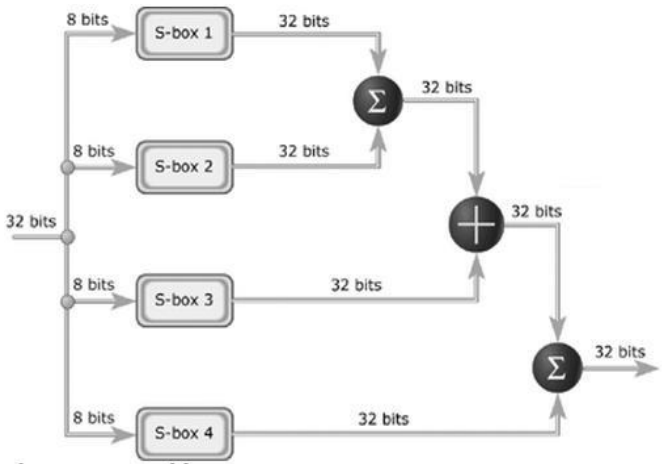

Figure 5: Logic Function

It shows a graphical representation of the *F* function, which has been shown as the most accessed function of the Blowfish algorithm [1]. It requires a 32-bit input data to be decomposed into four 8-bit blocks. Each block references an S-Box and each entry of the S-Box outputs a 32-bit data. First, the output of S-Box 1 and S-Box 2 are added. Then the result of the addition is XORed with S-Box 3. Finally, S-Box 4 is then added to the output of the XORed operation and provides a 32-bit output. Level and every 4 tiles can form a voltage domain that can operate on a different voltage level. For every voltage level, there is a range of supported frequencies. This enables us to minimize the amount of wasted energy on the SCC platform by selecting an appropriate voltage and frequency level for each of the computing cores and the idle cores

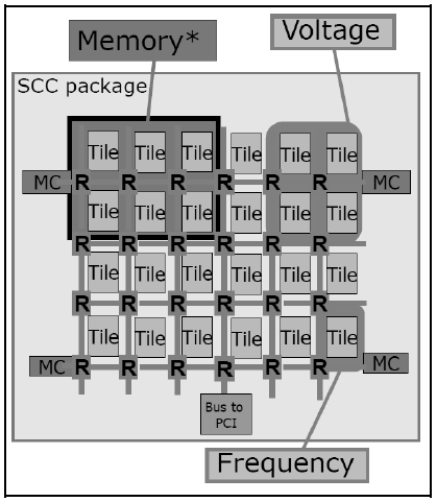

Figure 6: Frequency

Please note after Core 25, we send the data to Core 36, 37, 38, 39, and then come back to Core 26, 27, 28, and 29. So in total we still utilize 36 cores, corresponding to the 34 rounds for encipher and decipher with 2 cores for input and output. The reason for this topology is to reduce the overall communication overhead. As explained before, there is no extra hop between two cores on the same tile, Therefore, in order to minimize the execution time and delay associated with the communication overhead between the cores, all the communicating cores are located maximum one hop away from each other.

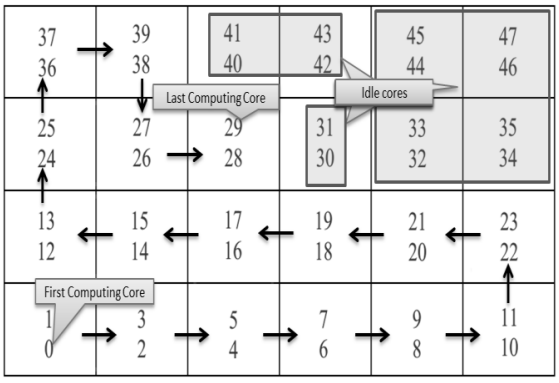

Figure 7: Massage passing core

However, the available bandwidth for the message passing is limited; thus, in the case of using large data chunks, communication overhead can cause bandwidth saturation
And higher latency. This can eventually lead to a longer execution time of the pipelined model, comparing to the sequential model where the program runs on only one core. In our design, the input file is split into different number of chunks and each chunk of data undergoes a series of computations based on the Blowfish algorithm. Each core is responsible for executing only one round of computations associated with the Blowfish algorithm and the data is then sent to the next core. In order to characterize and analyze the latency of the mesh network between the cores, we split the 1MB plaintext file into different chunk sizes.

Figure 8: Execution Time

It represents the execution time comparison between the sequential and the pipelined approach. As we can see, the pipelined approach runs 27.14 times faster than the sequential approach. While it takes for the sequential approach to run the program in 1.904956 seconds, the proposed pipelined model achieves to finish the program in only 0.0702 seconds. Only one core. In our design, the input file is split into different number of chunks and each chunk of data undergoes a series of computations based on the Blowfish algorithm. Each core is responsible for executing only one round of computations associated with the Blowfish algorithm and the data is then sent to the next core. In order to characterize and analyze the latency of the mesh network between the cores, we split the 1MB plaintext file into different chunk sizes.

Conclusion

To meet the ever increasing need of data security, cryptographic algorithms are getting more mathematically complicated day after day. On the other hand, the need for a reliable, energy-efficient, and fast computation is another necessity of our time. There is a significant improvement in performance gained by the use of multi-core systems. With multi-core and many-core systems becoming main-stream, we can make use of such systems for a faster and more energy-efficient computation of complicated cryptographic algorithms. However, possible performance gains are limited by the fraction of the software that can be run in parallel, since the rest of the program still needs to run sequentially. Also this gain is only possible if the computation overhead of the program is greater than the on-chip communication overhead.

REFERENCES

1. Kris Jams', Jones & Bartlett Student Edition **Cloud Computing Saas, Paas, Iaas,Virtualization,Business Models,Mobile,Security** First Indian Edition 2014.
2. William Stallings, **Network Security Essentials: Applications and Standards, 3rd** Edition, Pearson Education.
3. B.Schneier, **Applied Cryptography, John Wiley & Sons, New York, 1994.**
4. Bill Gatliff, **Encrypting Data With the Blowfish Algorithm Embedded Systems Programming.**
5. Uma Somani,Kanika Lakhami,Manish Mundra **Implementing Digital Signature With RSA Encryption Algorithm to Enhance the data Security of Cloud Computing[2010]**
6. Keahey, Fortes, Freeman **Science Clouds: Early experience in Cloud Computing For Scientific applications.**
7. http://en.wikipedia.org/wiki/Cloudcomputing
8. https://Clousecurity.org/
9. http://aws.typepad.com/aws/2008/07/