Grid Signature: High Performance Digital Signature through Using Alchemi Grid Computing

Abdelfatah Hegazy
Computer Science Department
The Arab Academy for Science, Technology and Maritime Transport, Cairo, Egypt
E-mail: ahegazy@aast.edu

Bahaa Hasan
Arab Security Consultants, Cairo, Egypt
E-mail: bahaa.hasan@asc-egypt.org

Iyad Shaheen
Computer Science Department
The Arab Academy for Science Technology and Maritime Transport, Cairo, Egypt
E-mail: shaheen@teachers.org

Abstract

A lot of researchers did their best to accelerate the cryptographic algorithms and develop high performance cryptographic schemes by using approaches such as the use of high end Grid computing. Grid computing is one of the most powerful techniques that can achieve a high acceleration for cryptographic algorithms. This approach makes the digital signature attractive for adoption by businesses to secure their documents. In this paper we propose and develop an application for digital signature cryptography using enterprise grid middleware called Alchemi. The modifying of the digital signature schema through compute hash in Alchemi parallel execution at two phases sign and verify said. The analyses of its performance are presented in two sides' sender and receiver by GridSign and GridVerify phases.

Keywords: Grid computing, Digital signature, Hash, Cryptography

1. Introduction

Peer-to-peer (P2P) or enterprise grids are proven as one of the approaches for developing cost-Effective high-end computing systems. By utilizing them, one can improve the performance of digital signature cryptography through parallel execution (David P. Anderson, Eric Korpela, Rom Walton, 2005). A digital signature is an electronic analogue of a written signature; the digital signature can be used to provide assurance that the claimed signatory signed the information. In addition, a digital signature may be used to detect whether or not, the information was modified after it was signed (i.e., to detect the integrity of the signed data). These assurances may be obtained whether the data was received in a transmission or retrieved from storage (FIPS PUB 186-3. 2006). "Grid Signature" is a high performance digital signature scheme that we developed inside Peer-to-Peer Computational Grid Middleware, in this research we will use Grid Signature to accelerate the digital signature and compare the execution time of the accelerated that a using the " Alchemi " developed by Melbourne university in Australia .in this research paper, we separate the message, at blocks in same size of bits, and encrypted each block of the message by session key to get cipher blocks. Inside Alchemi we use this grid middleware to compute the hash for cipher blocks of type SHA1 by separate each blocks parallel, the hash value in independent Alchemi executers and add any result of computing hash 128 bit to other in Exclusive disjunction logic gate, and finally we have "Grid Hash". After that we use privet key to encrypt the Grid Hash in sender said. And use the public key to decrypt the Grid Hash in receiver said to proved authentication and integrity, which described in details in this paper.

2. Digital Signature Schema.

A digital signature is the transformation of a message using an asymmetric cryptosystem and a hash function such that a person having the initial message and the signer public key can accurately determine (1) whether the transformation was created using the private key that corresponds to the signer public key, and (2) whether the initial message has been altered since the transformation was made (Information Security Committee, 1996). In this paper we developed Grid Signature that is a digital signature schema through using Alchemi Grid Computing. In our digital signature schema as shown in figure 1, it divided in two parts, first grid sign phase in sender side, and grid verify phase in receiver side. In grid sign phase, we using the session key to encrypt the plaintext by using AES algorithm, to get cipher text, that compute for it the hash value of type “SHA1” through parallel execution (William Stallings, 2006).after that, encrypt the result of the hash value by using private key of sender to generate “Grid Sign” and concatenate with cipher text, which it send to receiver. In verify phase, we tokenize the cipher message from sign ,and we using the session key to decrypt the cipher text by using AES algorithm, to get plaintext, that compute for cipher text the hash value of type “SHA1” through parallel execution (William Stallings, 2006).and next step decrypt for grid sign that contain the hash value ,to compare between two hashes values (one from compute through parallel execution and other from decrypt of grid sign) if equal then the result of decryption by session key is verify (message not change of original) ,if not equal then not verify (message is change of original). In the information security system, the trusted units only is the owner node and any other one nodes out of the owner machine, manager and executers is not trusted in the grid. So all the data that is flow through grid nodes is cipher (data encrypted) to assurance that no attackers recover the plaintext from the grid digital signature schema system.

3. Grid Hash Function Schema

The hash is the deterministic procedure that takes an arbitrary block or number of blocks, that contain of data and returns a fixed-size bit string, which called “hash value”, such that an accidental or intentional for chick for change to the data will change the hash value. The data to be encoded is often called the "message", and the hash value is called the “message digest”. (Antoine Joux, 2004) In our search we modify for computing the hash value of type “SHA1” that returns 128 fixed-size. And compute the hash value for each block in parallel executers through Alchemi grid computing, and then collect all the hash values of results in Exclusive disjunction to use as a simple mixing function for detracted the change the hash value in each block and fixed-size bit because it is appear as adder but without carry. The result of this collection of hash values in this schema is called “Grid Hash”. The collection of hash values is both an associative and a commutative operation. Thus parentheses may be omitted in successive operations and the order of terms makes no difference to the result, if there is one of the executers in Alchemi finished first and the others are late of time, to obtain the same grid hash.

4. Alchemi Grid Computing Framework

Alchemi is a .NET based grid computing framework developed at the University of Melbourne. It is an open source project which provides middleware for creating an enterprise grid computing environment by harnessing Windows machines. Alchemi supports multithreaded parallel operation in a manner similar to threading in Java or C#, but with their execution on distributed resources. The parallelism is realized at thread level and the programmer has to identify functions to be parallelized and implement them in the form of threads. Currently, inter-thread communication is not supported, so threads must be independent (Akshay Luther, Rajkumar Buyya, Rajiv Ranjan, and Srikumar Venugopal, 2004). A structure for Alchemi is shown in Figure 2. Its main components are manager and executor that support a master-worker parallel model. Alchemi has a number of features that ease the process of setting up of a grid environment in an enterprise. The executors can be setup in dedicated or no dedicated mode on employees’ desktop computers. In non-dedicated mode, Alchemi has no impact on the workstation as far as the user is concerned. The Alchemi manager also requires a Microsoft SQL Server instance, which is available in most companies. Before this research paper, we survey the techniques that use to Performance of Cryptography through Using Alchemi Grid Computing, and found the three techniques: GridCrypt, DotGrid and ULTRA GRIDSEC (Abdelfatah Hegazy, Bahaa Hasan and Iyad Shaheen, 2010). From this survey, generated to us an idea, how modify the digital signature through compute hash in Alchemi parallel execution at two phases sign and verify said.

5. Grid Sign Technique

In sender side; we separate the message, at blocks in same size of bits, and encrypted each block of the message by session key to get cipher blocks, as shown in figure 3. Inside Alchemi we use this grid middleware to compute the hash for cipher blocks of type SHA1 by separate each blocks parallel, the hash value in independent Alchemi executers and add any result of computing hash 128 bit to other in Exclusive disjunction logic gate, and finally we have “Grid Hash”. After that we use privet key to encrypt the Grid Hash in sender said. And concatenate grid hash with cipher text, which generates Grid Sign. As shown in figure4, the Normal Sign Time is equal the sum of an Encryption Message Time, Hash Time and Encrypt Hash Time. At otherwise as shown in figure 5, in ideal grid middleware the Total Grid Sign Time is equal the sum an Encryption Message Time, Hash last block Time and Encrypt Hash Time. That is means, the performance happen for lost the hash computing time from first block to before of the last hash block, which appears the different between normal and grid in sign time. So at running of the time, the Owner encrypts the block, at the same time; the executers in grid middleware compute the hash for previous block encrypted.

6. Grid Verify Technique

In receiver side; separate the message at blocks in same size of bits, and tokenize the cipher message from sign, and using the session key to decrypt the cipher text by using AES algorithm, to get plaintext. That compute for cipher text the hash value of type "SHA1" through parallel execution .And next step decrypt for grid sign that contain the hash value, to compare between two hashes values (one from compute through parallel execution and other from decrypt of grid sign by public key of sender) if equal then the result of decryption by session key is verify (message not change of original), if not equal then not verify (message is change of original) as shown in figure 6. As shown in figure7, the Total Normal Verify Time is equal the sum of Decryption Time, Hash Time, Decrypt Hash and Compare Time. At otherwise as shown in figure 8, in ideal grid middleware the Total Grid Verify Time is equal the sum of Decryption Time, Decrypt Hash and Compare Time. That is means, the performance happen for lost all the hash computing time that appears the different between normal and grid in verify time.

So at running of the time, the Owner decrypts the block, at the same time; the executers in grid middleware compute the hash for previous block encrypted.

7. Mathematical Grid sign and verify Complexity proof.

Assume a k number of Executers.

Assume an n message size.

So at 1st term $\rightarrow \frac{n}{k}$

at 2nd term $\rightarrow \frac{n}{2k}$

at t term $\rightarrow \frac{n}{2^t k}$

$T_{n,k} = \left(\frac{n}{k} + \frac{n}{2k} + \ldots\ldots\ldots + \frac{n}{2^t k} \right)$

$= \frac{n}{k} \left(1 + \frac{1}{2} + \frac{1}{4} + \ldots.. + \frac{1}{2^t}\right)$

Let sum $= 1 + \frac{1}{2} + \frac{1}{4} + \ldots.. + \frac{1}{2^t}$

$\frac{1}{2}$ sum $= \frac{1}{2} + \frac{1}{4} + \ldots.. + \frac{1}{2^t} + \frac{1}{2^{t+1}}$

sum $= 1 + \frac{1}{2}$ sum $- \frac{1}{2^{t+1}}$

So $\frac{1}{2}$ sum $= 1 - \frac{1}{2^{t+1}}$

Sum $= 2 - \frac{2}{2^{t+1}}$

$= 2 - \frac{1}{2^t}$

Observe $\frac{n}{2^t k} = 1$

$\log_2 n = \log_2 2^t + \log_2 k$

$= t + \log_2 k$

So $t = \log_2 n - \log_2 k$

$T_{n,k} = \frac{n}{k} \left(2 - \frac{1}{2^{(\log_2 n - \log_2 k)}} \right)$

Since n>> k then

$$\lim_{n\to\infty} \frac{1}{2^{(\log_2 n - \log_2 k)}} = \lim_{n\to\infty} \frac{1}{2^{\log_2 n}} = \lim_{n\to\infty} \frac{1}{n} = 0$$

$T_{n,k} = \frac{n}{k}(2-0)$

$T_{n,k} = \frac{2n}{k}$

Assume n message size is fixed-size.

$T k = \frac{1}{k}$

So $O (T) = \frac{1}{k}$

That means in grid sign or verify, the complexity of execution depends on number of executers that compute the hash value.

8. Performance Evaluation

We have done a runtime comparison for the GridSign application using 4 executor nodes each with the same specification of Pentium IV 3000 MHz processor, 512 MB of memory and Running Windows XP Professional operating system. All these nodes were interconnected over a shared student laboratory LAN network of 100 Mbps. The Alchemi manager was installed on a separate computer together with SQL Server 2000. A separate user machine was used to initiate the execution of the GridSign or GridVerify applications. We monitored the CPU usage and the threads execution details using the Alchemi console (see Figure 8).

The grid sign and verify experiments were conducted on files of size 45 MB and 82 MB with 1Mb block size, which lead to the creation of 5 work units respectively. For each file the sign or verify was carried on 1, 2, 3, 4 and 5 executor nodes. The performance results of these experiments are shown in Figure 9 and 10, and that it is appear the GridSign and Grid Verify for 45 MB file size with 1MB block size. The performance results of 82MB file size experiments are shown in Figure 11 and 12, which it is appearing the Time of GridSign and Grid Verify for 82 MB file size with 1MB block size. In all experiments, that it is appear the time of running standalone on a single machine without any grid (zero executer) overhead for sign or verify 45 MB and 82 MB file size, it take more than running time if using Alchemi Grid computing for at least one executer. Because in Alchemi Grid, compute the hash in parallel executers at the same time the owner encrypts in sign phase or decrypts in verify phase, but in running standalone on a single machine without any grid (zero executer) overhead, encrypts in sign phase or decrypts in verify phase and compute the hash in serial, that means in the Alchemi Grid is save the running time for compute the hash blocks (more performance) compared with a single machine without any grid. Also in all experiments, there was a reasonable performance improvement when 1, 2, 3 and 4 executors were used roughly respectively (Daniel M. Pressel.). This gain is not linear; a similar phenomenon is also observed in. Although there was a reasonable performance improvement when less to 4 executors were used, there was drop in performance gain when number of executors was increased as show in Figure 9,10,11,12. The performance improvement is limited by the encryption time of the grid signature, I/O and communication overhead. This is due to various overhead factors including (a) involvement of large datasets with low computation to communication ratio, (b) existence of serial processing component (file splitting and collating results), (c) the use of slow and shared network, and (d) the overhead of the distributed execution environment (e.g., distribution of executable, initiation of execution on a remote node, and management of threads) (Akshay Luther, Rajkumar Buyya, Rajiv Ranjan, and Srikumar Venugopal, 2004).

9. Conclusion

Enterprise grids are proven as one of the approaches for developing cost-Effective high-end computing systems. We can improve the performance of digital signature cryptography through parallel execution, in Alchemi grid computing. This implemented using Alchemi shows an increase over the single processor version of the digital signature cryptography, the performance improvement is limited by the encryption time of the grid signature, the I/O and communication overhead.

References

Abdelfatah Hegazy, Bahaa Hasan and Iyad Shaheen. (2010). Survey of Three Techniques to Performance of Cryptography Using Alchemi Grid Computing. *International journal: Research, Analysis and Evaluation,* India, June 2010.

Akshay Luther, Rajkumar Buyya, Rajiv Ranjan, and Srikumar Venugopal. (2004). "Alchemi: A .NET-based Desktop Grid Computing Framework, High Performance Computing: Paradigm and Infrastructure", Laurence Yang and Minyi Guo, Wiley Press, New Jersey, USA. Fall 2004.

Akshay Luther, Rajkumar Buyya, Rajiv Ranjan, and Srikumar Venugopal. (2004). "GridCrypt: High Performance Symmetric Key Cryptography Using Enterprise Grids ", Laurence Yang and Minyi Guo (editors), Wiley Press, New Jersey, USA. Fall 2004.

Antoine Joux. (2004). Multicollisions in Iterated Hash Functions. Application to Cascaded Constructions. LNCS 3152/2004, pages 306-316.

Daniel M. Pressel. Scalability vs. Performance. U.S. Army Research Laboratory Aberdeen Proving Ground, MD

21005-5066. http://www.hpcmo.hpc.mil/Htdocs/UGC/UGC00/paper/daniel_pressel_scale_paper.pdf

David P. Anderson, Eric Korpela, Rom Walton. (2005). *"High Performance Task Distribution for Volunteer Computing"*. First IEEE International Conference on e-Science and Grid Technologies, Dec. 2005.

FIPS PUB 186-3. Digital Signature Standard. National Institute of Standards and Technology, 13 March 2006. Information Security Committee. (1996). *"Digital Signature Guidelines"*. Electronic Commerce and Information Technology, Division Section of Science and Technology, American Bar Association, 1996.

William Stallings. (2006). *"Cryptography and Network Security Principles and practices",* Fourth Edition, PEARSON, USA, 2006.

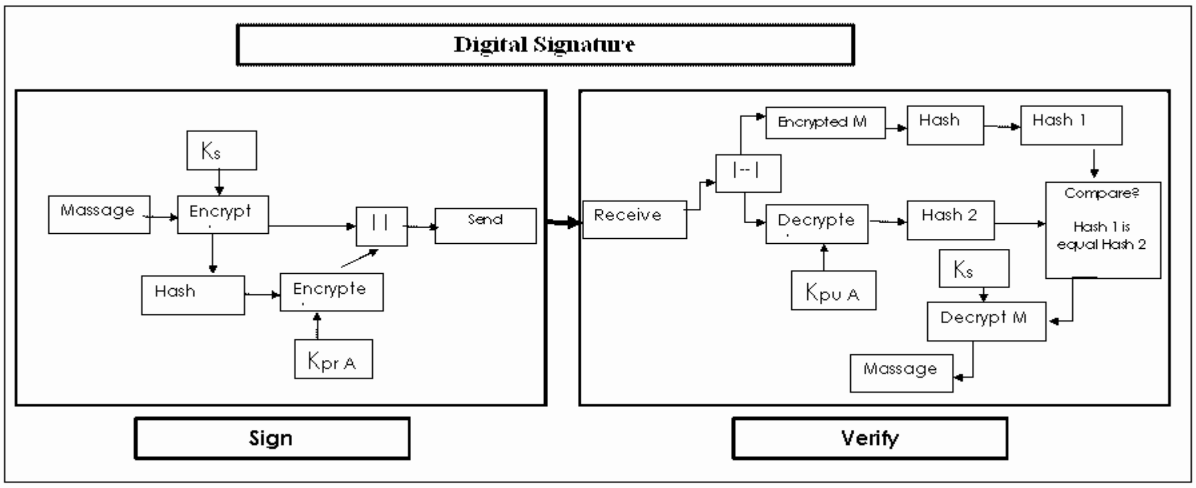


Figure 1. Digital Signature Schema

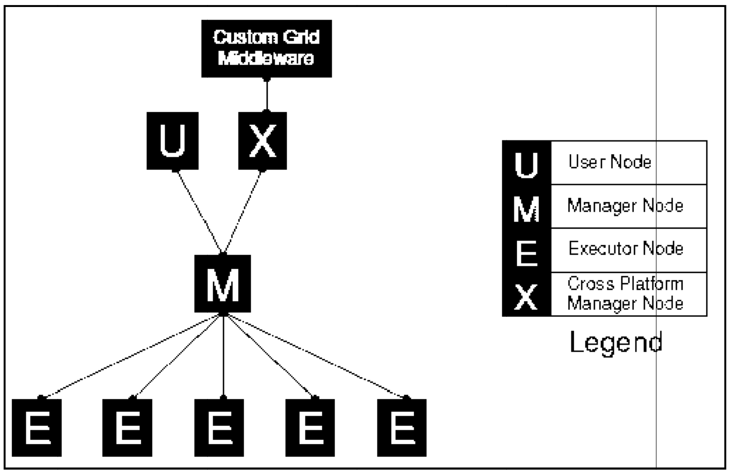


Figure 2. Alchemi Grid Computing Structure

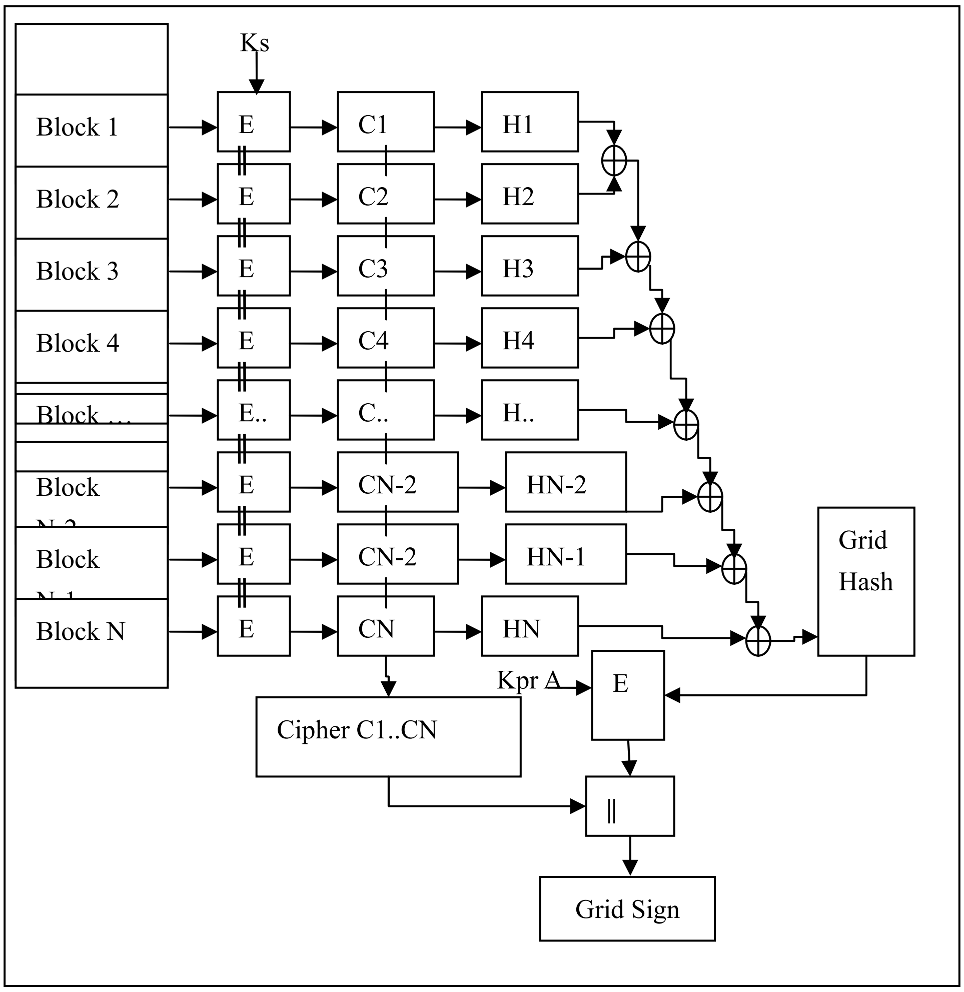


Figure 3. Grid Sign

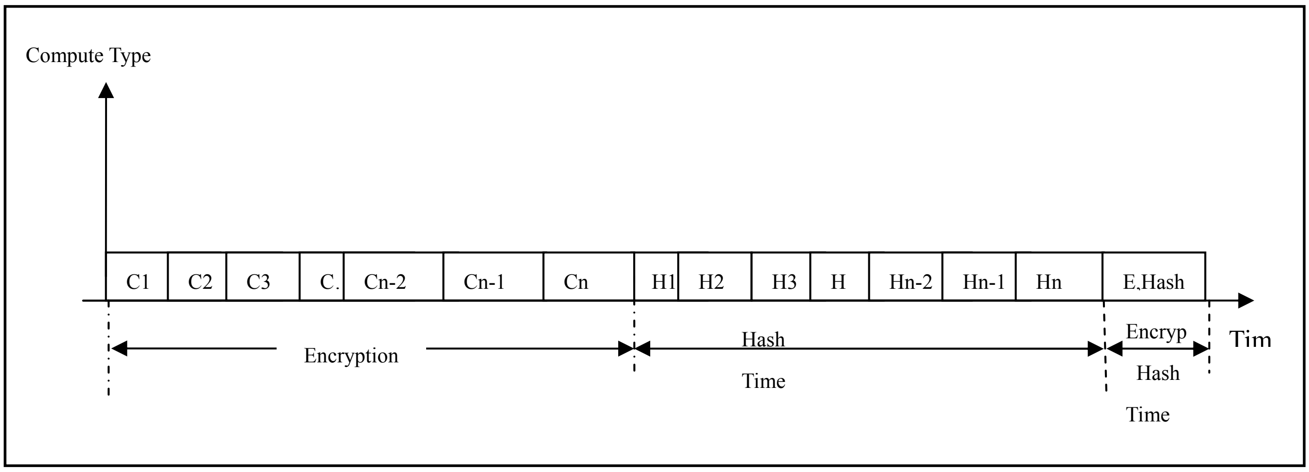


Figure 4. Total Normal Sign Time

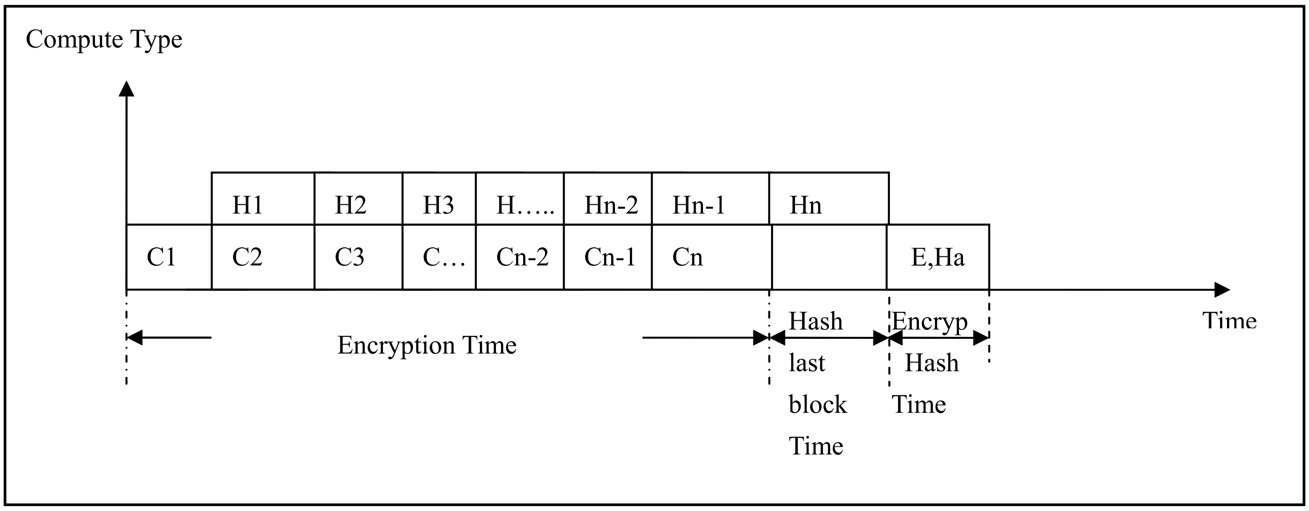


Figure 5. Total Grid Sign Time

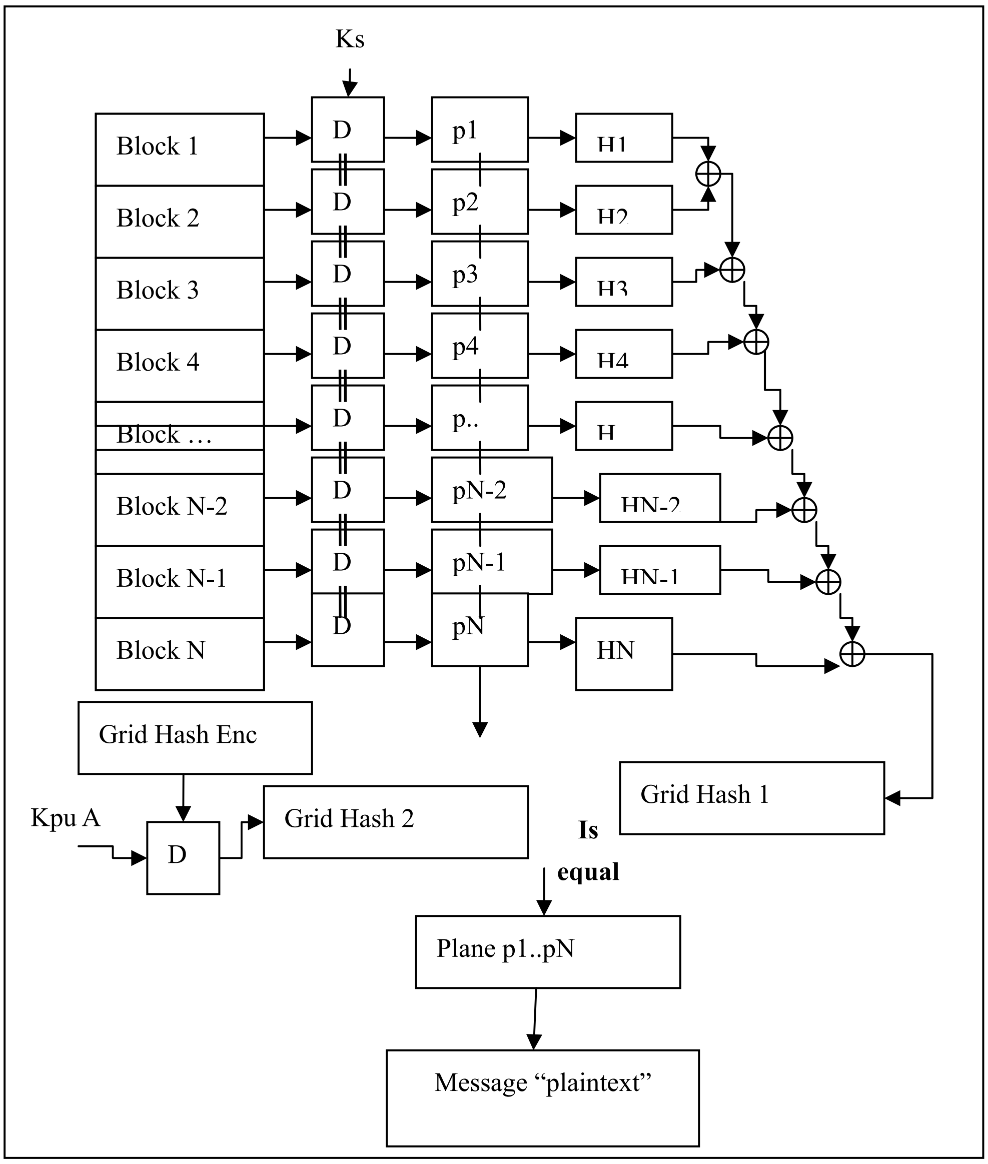


Figure 6. Grid Verify

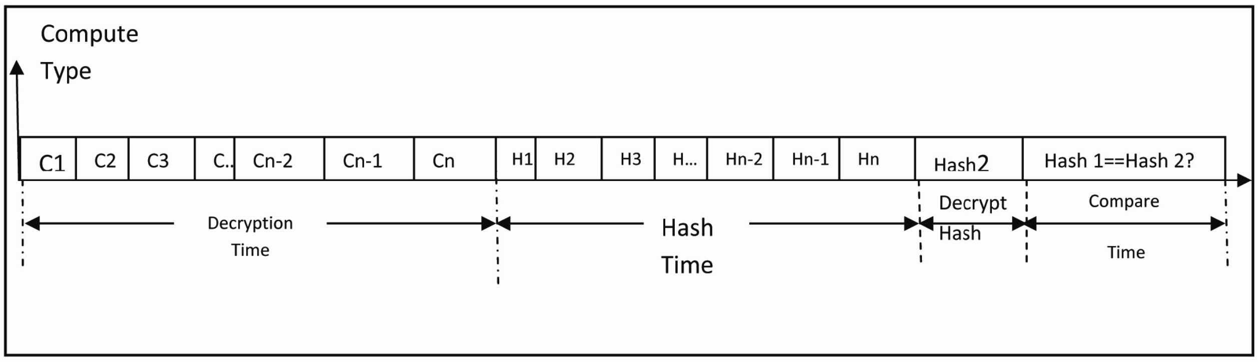


Figure 7. Total Normal Verify Time

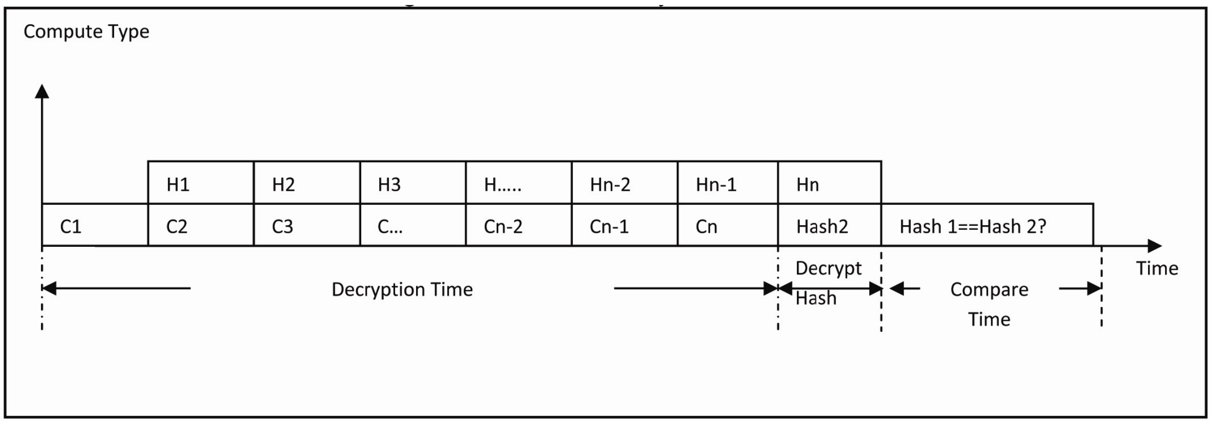
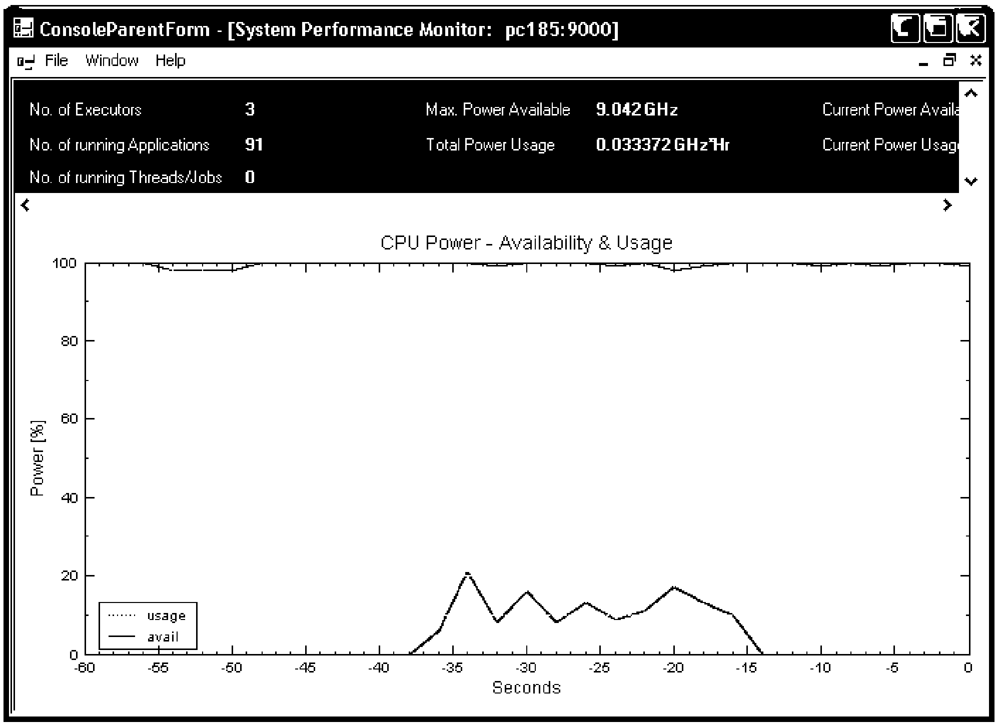


Figure 8. Total Grid Verify Time


ConsoleParentForm - [System Performance Monitor: pc185:9000]
File
Window
Help
CPU Power - Availability & Usage
Power [%]
Seconds
usage
avail

No. of Executors 3
No. of running Applications 91
No. of running Threads/Jobs 0
Max. Power Available 9.042 GHz
Total Power Usage 0.033372 GHz*Hr
Current Power Availa
Current Power Usag

Figure 9. Alchemi performance while running 3 executors

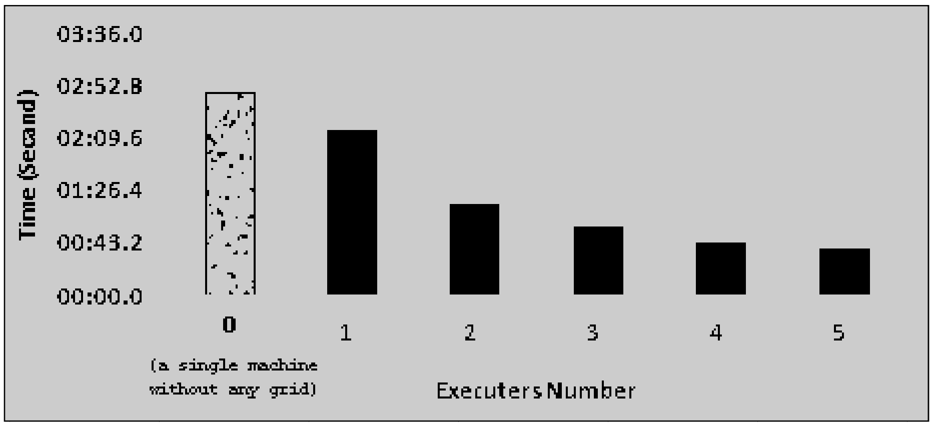


Figure 10. Time of a single machine without any grid and GridSign for 45 MB file size with 1MB block size

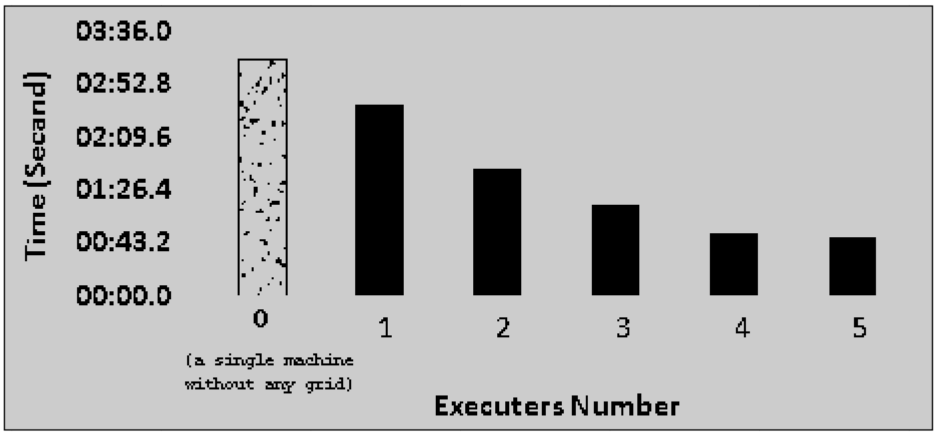


Figure 11. Time of a single machine without any grid and Grid Verify for 45 MB file size with 1MB block size

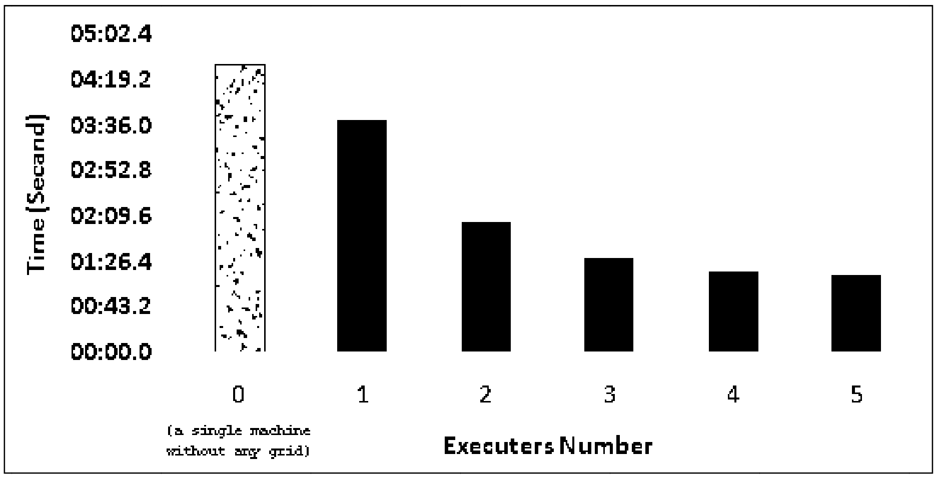


Figure 12. Time of a single machine without any grid and GridSign for 82 MB file size with 1MB block size

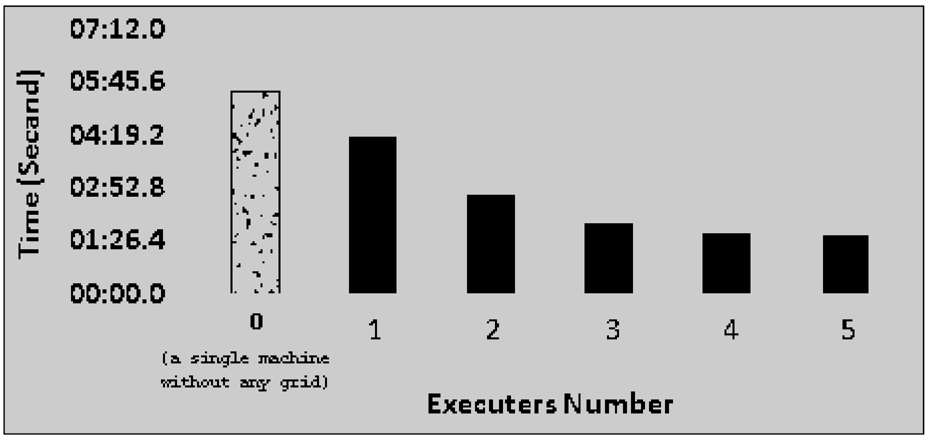


Figure 13. Time of a single machine without any grid and Grid Verify for 82 MB file size with 1MB block size