**Editor in Chief** Dr Wei WANG

# International Journal of Security (IJS)

Book: 2008 Volume 1, Issue 1
Publishing Date: 30-06-2007
Proceedings
ISSN (Online): 1985-2320



IJS Journal is a part of CSC Publishers
http://www.cscjournals.org


Published in Malaysia

Typesetting: Camera-ready by author, data conversation by CSC Publishing Services – CSC Journals, Malaysia

**CSC Publishers**

# Table of Contents

Volume 1, Issue 1, May/June 2007.

## Pages

# Performance Analysis of Mobile Security Protocols: Encryption and Authentication

**Anita Singhrova** email: nidhianita@gmail.com
Sr. Lecturer in CSE Department,
DBCR University of Science and Technology, Murthal,
Sonepat, Haryana, 130131 INDIA.

**Dr. Nupur Prakash** email: nupurprakash@rediffmail.com
Dean, University School of Information Technology,
Guru Gobind Singh Indraprastha University, Kashmere Gate,
Delhi 110006, INDIA.

**Abstract**

Due to extremely high demand of mobile phones among people, over the years there has been a great demand for the support of various applications and security services. 2G and 3G provide two levels of security through: encryption and authentication. This paper presents performance analysis and comparison between the algorithms in terms of time complexity. The parameters considered for comparison are processing power and input size. Security features may have adverse effect on quality of services offered to the end users and the system capacity. The computational cost overhead that the security protocols and algorithms impose on lightweight end users devices is analyzed. The results of analysis reveal the effect of authentication and encryption algorithms of 2G and 3G on system performance defined in terms of throughput which will further help in quantifying the overhead caused due to security.

**Keywords**: Encryption, Authentication, GSM (Global System for Mobile communication), UMTS (Universal Mobile Telecommunication System), Time complexity, Performance Analysis, Throughput.

## 1. INTRODUCTION

The fixed line telephones had revolutionized the concept of voice communication, but with passage of time, lack of mobility was felt seriously. Moreover, delay in new connection, last mile wired connectivity and security hazards were few other problems.

The first generation (1G) of mobile communication system was introduced in 1985 and was driven by analog signal processing technique. It had certain problems, like phone fraud through cloning phones and thus calling at someone else's expense, and the possibility of someone intercepting the phone call over the air and eavesdropping on the discussion.

The second generation (2G) of mobile communication systems popularly known as GSM (Global System for Mobile communication) was one of the first digital mobile phone systems to follow analog era and had started in 1992. The GSM system was supposed to overcome the phone fraud and call interception problems of an analog era by implementing strong authentication between the MS and the MSC, as well as implementing strong data encryption for over the air transmission channel between the MS and BTS [1][2]. The GSM system also suffered some of the shortcomings. The attack against A5, accessing the signaling network, retrieval of key and false base station attack etc.

2.5G is known as GPRS (General Packet Radio Services) came in 1995. This does not implement any new algorithms for authentication or confidentiality, but it uses same algorithm for authentication and encryption as 2G and multiple timeslots in parallel in order to achieve a greater transmission rate i.e. 171 kbps [3].

The third generation (3G) of mobile communication systems known as UMTS (Universal Mobile Telecommunication Systems) was introduced in 2002 and intends to establish a single integrated and secure network. The 3GPP (Third Generation Partnership Project), is a follow up project of GSM which implements UMTS (Universal Mobile Telecommunication Systems) [4][5]. It lays down standards to support broadband data services and mobile multimedia using a wideband radio interface international roaming for circuit switched and packet switched services. Mobile/wireless Internet is becoming available with 3G mobile communication

systems. The complete 3G security architecture consists of five major security classes: (i) network access security, (ii) network domain security, (iii) user domain security, (iv) application domain security and (v) visibility and configurability of security [6].

The fourth generation (4G) of mobile communication systems with its year of inception predicted as 2010-2012[7] is a futuristic approach and is envisioned as a convergence of different wireless access technologies [8].

Wireless networks are as such less secure and mobility further adds to security risk. Therefore, it is desirable that 2G and 3G are atleast as secure as fixed networks if not over secure. Security is achieved at the cost of performance degradation, therefore, it is critical and important to quantitatively measure overheads caused by various security services [9] [10].

The Section 2 presents the various security mechanisms of 2G and 3G in brief. Section 3 deals with the performance Analysis of Authentication and Encryption Algorithms of 2G and 3G both followed by discussion on the performance analysis in terms of throughput. Section 5 is devoted to review the future trends. Finally, summary and conclusion is given in section 6.

## 2. SECURITY FEATURES

### 2.1. 2G Security Overview

The security mechanism in 2G mainly consists of subscriber identity authentication and confidentiality i.e. encryption of user traffic.

#### 2.1.1 Authentication

In GSM the authentication algorithm used is $A_3$. Its function is to generate the 32-bit SRES (Signed Response) to the MSC's random challenge, RAND and the secret key $K_i$ from the SIM as input i.e. SRES= $A_3$ $K_i$ (RAND). The subscriber identity authentication is used to identify the MS to the PLMN (public land mobile network) operator [11]. Authentication is a one way process. The MS is authenticated but the visited PLMN is not. Therefore, GSM is open to false base station attack.

In GSM $A_8$ algorithm is used as key generation algorithm. It generates a 64 bit session key, $K_c$, from the 128 bit random challenge, RAND, received from the MSC and from 128 bit secret key $K_i$ i.e. $K_c$= $A_8$ $K_i$ (RAND). The BTS receives the same $K_c$ from the MSC. HLR is able to generate the $K_c$, because the HLR knows both the RAND (the HLR generated it) and the secret key $K_i$, which it holds for all the GSM subscribers of this network operator. One session key, $K_c$, is used until the MSC decides to authenticate the MS again.

The $COMP_{128}$ generates both the SRES response and the session key, $K_c$ on one run [11]. Therefore $COMP_{128}$ is used for both the $A_3$ and $A_8$ algorithms.

#### 2.1.2 Encryption

A5 algorithm is the stream cipher and is used to encrypt over-the-air transmissions to protect sensitive information against eavesdropping on the air interface [12].

$K_c = A_8K_i$ (RAND) and Ciphertext = $A_5K_c$ (Plaintext)

Each frame in over-the-air traffic is encrypted with a different key stream. The same $K_c$ is used throughout the call, but the 22-bit frame number changes during the call, thus generating a unique key stream for every frame [12]. The $A_5$ algorithm consists of three LSFRs of different lengths.

The data is encrypted only between the MS and BTS. After the BTS the traffic is transmitted in the plain text within the operator's network. Therefore, if attacker can access the operators signaling network, then attacker can listen to everything transmitted.

### 2.2 3G Security Overview

Third generation mobile systems such as UMTS revolutionized telecommunications technology by offering mobile users content rich services, wireless broadband access to internet, and worldwide roaming. However, this introduced serious security vulnerabilities [4]. Encryption and Authentication are the two main security mechanisms in 3G network access securities [5].

#### 2.2.1 Authentication

The UMTS authentication algorithm consists of seven functions $f_1$,$f_1$*,$f_2$,$f_3$,$f_4$,$f_5$ and $f_5$*. The standardized algorithm set for these seven functions is called MILENAGE. For MILENAGE, a specific kernel has to be chosen, and therefore Rijndael was selected [13] [14]. Rijndael is an iterated block cipher with a 128 bit block length and a 128 bit key length. It is composed of eleven rounds that transform the input into the output.

#### 2.2.2 Encryption

Within the security architecture of the 3GPP system there are two standardized algorithm: a confidentiality algorithm $f_8$, and an integrity algorithm $f_9$[4], which are based on the KASUMI algorithm [14] [15].

## 3. PERFORMANCE ANALYSIS

This section analyses the performance of algorithms used for 2G and 3G authentication and encryption. The time complexity computation has been carried out for this purpose.

### 3.1 Analytical Analysis of 2G

#### 3.1.1 Authentication

This involves the analysis of authentication and key generation algorithm using A3A8 algorithm for 2G [11]. Authentication initialization needs 48 operations.

| OPERATIONS | TIMES | TIME NEEDED | EQUIVALENT TOTAL |
|---|---|---|---|
| Load Rand # | 16 | 2 | 32 |
| Load Key | 16 | 1 | 16 |

Operations needed = 48

**TABLE 1:** Authentication initialization(load Rand # and key $K_i$)

| OPERATIONS | TIMES | TIME NEEDED | EQUIVALENT TOTAL |
|---|---|---|---|
| Load Rand # | 16 | 2 | 32 |
| Load $K_i$ | 8 | 16 | 128 |
| Substitution | 8 | 2080 | 16640 |
| Form bits | 8 | 768 | 6144 |
| Permutation | 8 | 1168 | 9344 |

Total number of operations = 32 288

**TABLE 2:** Total operations in 2G authentication

The next step of substitution involves 3 variable j, k, l. Where j can have any value between 0 to 4, k can have value $2^n$. For 1st iteration the value is 0 i.e $2^0$ for n = 0; for 2nd iteration there are two values 0 and 1 i.e $2^1$ for n=1, so on and so forth till n=4. And for l the value is $2^{4-n}$ . Therefore for n=0, j have values $2^4$ i.e j = 0..15 ,if n=1, j have value $2^3$ i.e j = 0..7 so on and so forth. Hence, total number of operation required for substitution is 5*16*26 where 26 are the number of operations carried out in 1 iteration. The next step of bit forming requires 32 * 4*6 operations and the permutation requires 72 for outer loop and 1168 for inner loop.

$T_{A3A8}$ is total number of operations for authentication = 32 288.
$T_{A3A8}$ = 32 288.
$s_d$ is the size of original message (in bytes).
N is the message size in bits. $N=8 * S_d$
n is the total number of blocks, n = Ceil (N ÷ 128) where Ceil(x) means the smallest integer ≥ operand
$U_{A3A8}$ is the total number of operations required for A3A8 authentication.
$U_{A3A8}$= ceil $((8 * S_d) \div 128) * T_{A3A8} = n * T_{A3A8}$
$C_p$ is MIPS performed by the processor.
$t_{A3A8}$ $(S_d, C_p)$ is the time required for encryption (decryption) for processor speed $C_p$ and message size $S_d$ in bytes. $t_{A3A8}(S_d, C_p) = U_{A3A8}(S_d) \div C_p$ or $t_{A3A8}(S_d, C_p) = (\text{ceil}((8 * S_d) \div 128) * T_{A3A8}) \div C_p$
or $t_{A3A8}(S_d, C_p) = (n * T_{A3A8}) \div C_p$

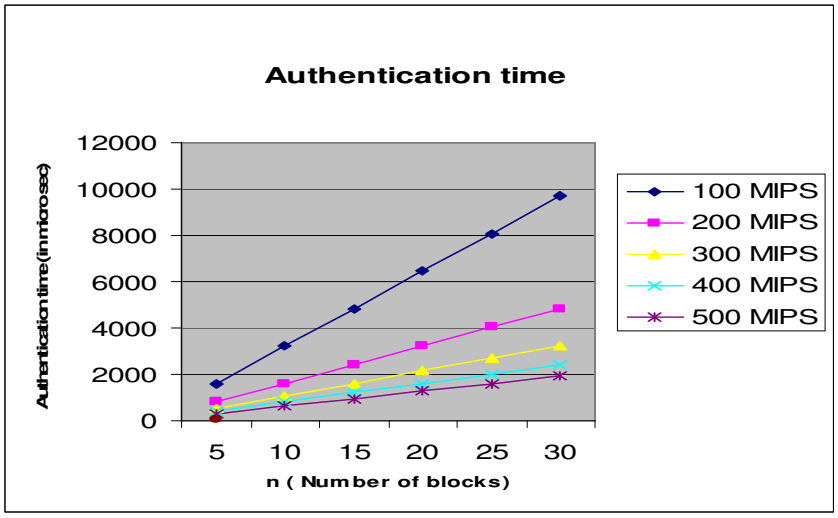



**FIGURE** 1: Authentication time (in μ sec) Vs n of i/p blocks and processing speed

**3.1.2 Encryption**

The LSFRs R1, R2, R3 are 19, 22 and 23 bits long respectively defined with the help of MASK 0x07FFFF (0..18 numbers) , 0x3FFFFF (0..21 numbers ) and 0x7FFFFF (0..22 numbers). For clocking the feedback registers feedback taps are used[12]. Middle bit of each of the three shift registers, are used for clock

control i.e. R1MID 0x000100, R2MID 0x000400, R3MID 0x000400. The highest bit of LSFRs are taken as output taps. 18th, 21st and 22nd bits respectively for R1, R2 and R3 respectively [12].

| OPERATIONS | TIMES | TIME NEEDED | EQUIVALENT TOTAL |
|---|---|---|---|
| Right shift by n | 5 | 1 | 5 |
| XOR | 5 | 1 | 5 |
| AND | 1 | 1 | 1 |

operations needed =11

**TABLE 3:** Operations in Parity function

| OPERATIONS | TIMES | TIME NEEDED | EQUIVALENT TOTAL |
|---|---|---|---|
| AND | 2 | 1 | 2 |
| Left shift by 1 | 1 | 1 | 1 |
| OR | 1 | 1 | 1 |
| Parity ( ) | 1 | 11 | 11 |

operations needed = 15

**TABLE 4**: Operations in Clockone function

| OPERATIONS | TIMES | TIME NEEDED | EQUIVALENT TOTAL |
|---|---|---|---|
| ADD | 2 | 1 | 2 |
| AND | 3 | 1 | 3 |
| Parity( ) | 3 | 11 | 33 |
| Comparison | 1 | 1 | 1 |

operations needed = 39

**TABLE 5:** operations in Majority function

| OPERATIONS | TIMES | TIME NEEDED | EQUIVALENT TOTAL |
|---|---|---|---|
| Clockone( ) | 3 | 15 | 45 |

operations needed = 45

**TABLE 6:** Operations in Clockallthree function

| OPERATIONS | TIMES | TIME NEEDED | EQUIVALENT TOTAL |
|---|---|---|---|
| Majority( ) | 1 | 39 | 39 |
| Comparison | 6 | 1 | 6 |
| Clockone( ) | 3 | 15 | 45 |

operations needed = 90

**TABLE 7:** Operations in Clock function

| OPERATIONS | TIMES | TIME NEEDED | EQUIVALENT TOTAL |
|---|---|---|---|
| AND | 3 | 1 | 3 |
| XOR | 2 | 1 | 2 |
| Parity ( ) | 3 | 11 | 33 |

operations needed = 38

**TABLE 8:** Operations in Getbit function

| OPERATIONS | TIMES | TIME NEEDED | EQUIVALENT TOTAL |
|---|---|---|---|
| Clockallthree( ) | 1 | 45 | 45 |
| Right shift | 1 | 1 | 1 |
| AND | 2 | 1 | 2 |
| Divide | 1 | 1 | 1 |
| XOR | 3 | 1 | 3 |

operations needed = 52

**TABLE 9:** Operations in Key setup function

| OPERATIONS | TIMES | TIME NEEDED | EQUIVALENT TOTAL |
|---|---|---|---|
| clockallthree | 1 | 45 | 45 |
| Right shift | 1 | 1 | 1 |
| XOR | 3 | 1 | 3 |
| AND | 1 | 1 | 1 |

operations needed = 50

**TABLE 10:** Operations in Frame# load function

| OPERATIONS | TIMES | TIME NEEDED | EQUIVALENT TOTAL |
|---|---|---|---|
| | | | |
| Clock ( ) | 1 | 90 | 90 |
| Getbit( ) | 1 | 38 | 38 |
| AND | 1 | 1 | 1 |
| Left shift | 1 | 1 | 1 |
| OR | 1 | 1 | 1 |
| Arithmetic operators | 2 | 1 | 2 |

operations needed =133

**TABLE 11**: Operations in Run function

| OPERATIONS | TIMES | TIME NEEDED | EQUIVALENT TOTAL |
|---|---|---|---|
| Keysetup | 64 | 52 | 3328 |
| Frame#Load( ) | 22 | 50 | 1100 |
| Clock | 100 | 90 | 9000 |
| Run ( ) | 228 | 133 | 30324 |

Total operations needed=43 752

**TABLE 12:** Total operations for $A_5/1$

$T_{A5/1}$ is total number of operations in block encryption = 43 752.
$T_{A5/1}$ = 43 752
$S_d$ is size of original message (in bytes)
N is the message size in bits. N=8 *$S_d$
n is the total number of blocks. n = Ceil (N ÷ 114)
where Ceil(x) means the smallest integer >= operand
$U_{A5/1}$ is the total number of operations required for encryption or decryption of message size Sd.
$U_{A5/1}$ = ceil ((8 * $S_d$) ÷114) * $T_{A5/1}$ = n * $T_{A5/1}$
$C_p$ is MIPS performed by the processor.
$t_{A5/1}$ ($S_d$, $C_p$) is the time required for encryption (decryption) for processor speed $C_p$ and message size $S_d$ in bytes.
$t_{A5/1}$ ($S_d$ , $C_p$ ) = $U_{A5/1}$ ($S_d$ ) ÷ $C_p$ or $t_{A5/1}$ ($S_d$ , $C_p$ ) = (ceil ((8 * $S_d$ ) ÷ 64) * $T_{A5/1}$) ÷ $C_p$
or $t_{A5/1}$ ($S_d$, $C_p$) = (n * $T_{A5/1}$) ÷$C_p$

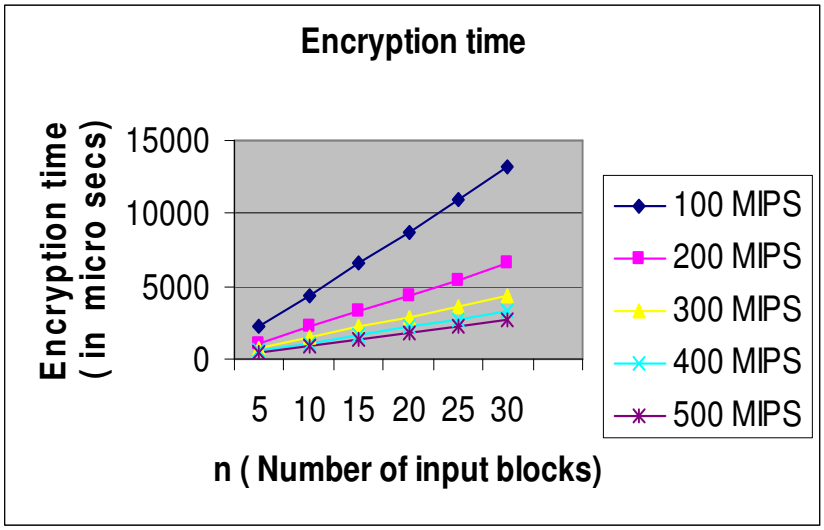



**FIGURE 2**: Encryption time (in µ sec) as a function of number of packets for different processing speeds (MIPS)

The total number of operations required by a processor to perform $A_3A_8$ and $A_5$/1 as a function of the packet size are presented in figure 3. $A_5$ requires more number of operations as compared to $A_3A_8$ .

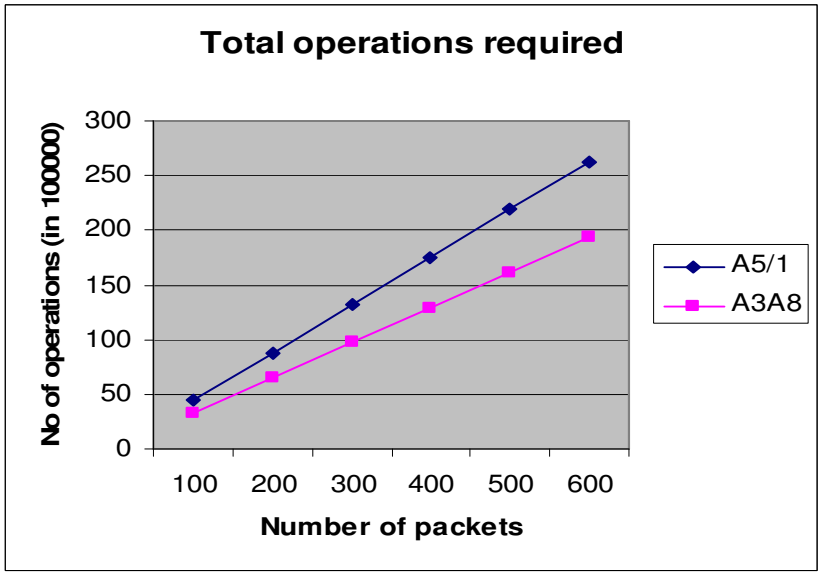



**FIGURE 3**: 2G, Total number of operations required as a function of number of packets for 2G algorithms

### 3.2 Analytical Analysis of 3G

#### 3.2.1 Authentication

3G authentication is implemented by Rijndael. Rijndael is an iterated block cipher with a variable block length and variable key length [13]. The block and key length are independently specified as 128 bits for 3GPP and is used in encryption mode. It consists of 9 rounds in addition to an initial and a final round to transform the input into the output. An intermediate result is called state. The state can be a 4 X 4 rectangular array of bytes (128 bits in total).

##### 3.2.1.1. The Byte Substitution Transformation

As described in paper [13] it is a non-linear byte substitution, operating on each of the State bytes independently. The substitution table is stored as S-box. In this transformation, we require 16 1-Dimensional lookup for 16 elements. Therefore, 16 operations are carried for 1 block of input of 128 bits.

##### 3.2.1.2 The Shift Row Transformation

In this transformation[13], the rows of the State are cyclically left shifted by different amounts. Row 0 is not shifted, four states in row 1 are shifted by 1 byte, four states in row 2 by 2 bytes and four states in row 3 by 3 bytes. Therefore, 0 + (4*1) + (4* 2) + (4* 3) = 24 operations.

##### 3.2.1.3. The Mix Column Transformation

The mix column transformation operates on each column of the State independently [13]. For each column there are (4 XOR + 1 multiplication + 1 optional XOR) *4. Since there are 4 columns there will be 80 or 100 operations.

##### 3.2.1.4. The Round Key addition

In this operation, a Round Key is applied to the State by a simple bitwise exclusive-or[13]. The Round Key is derived from the Cipher Key by means of the key schedule. The Round Key length is equal to the block length. There are total of 16 XOR. Therefore, 16 operations are required for this transformation.

##### 3.2.1.5. Key schedule

Rijndael has 11 Round Keys, numbered 0-10, that are each 4x4 rectangular arrays of bytes. Let $rk_{r,i,j}$ be the value of the $r^{th}$ Round Key at position (i, j) in the array and $k_{i,j}$ be the cipher key loaded into a 4x4 array.

| BASIC OPERATION | EQUIVALENT OPERATIONS | | |
|---|---|---|---|
| | Type | Time needed | Space needed |
| Round key Addition | 8-bit XOR | 1 | |
| Byte substitution Transformation | 1-D table lookup [B ] | 1 | b |
| Shift row Transformation | left shift by n bit | 1 | |
| Mix column Transformation | XOR | 1 | |
| | Multiply | 1 | |
| Key schedule | XOR | 1 | |
| | MULTIPLY | 1 | |
| Key schedule Initialization | 8-bit copy | 1 | |
| 2-D table look up (for i:j bit map) | Multiply | 1 | |
| | Add | 1 | |
| | 1-D table lookup | 1 | i i rows * j col |
| 2-D table ( for 4*4 bit map) into 1-D or vice versa | COPY | 1 | |
| | | | |
| | Add | 1 | |
| | Right shift by 2 bits | 1 | 4 rows * 4 col |

**TABLE 13:** Rijndael basic operations

| Operations | Times | Time needed | Equivalent Total |
|---|---|---|---|
| Round key addition | 16 | 1 | 16 |
| Byte substitution transformation | 16 | 1 | 16 |
| Shift row transf(1 for 1st row, 2 for 2nd row And 3 for 3rd row) | 4 | 1+1+1 =3 | 12 |
| Column transf.(4XOR +1 Multiply +1 XOR optional ) * 4 for 1 column. | 4 | (4+1)*4 =20 | 80 |
| Key initialization | 16 | 4 | 16 |
| Key schedule XOR+ 1 Multiply * 10 (for rounds ) + 12 XOR for 4 col.) | 1 | 60 + 12 = 72 | 72 |

**TABLE 14**: Rijndael operations

| step | operations | Times | Time needed | Equivalent total |
|---|---|---|---|---|
| 0 | Key initialization | 1 | 1 | 1 |
| | Key schedule | 1 | 72 | 72 |
| 0-10 | Key ( 2-D lookup) | 11 | 3 | 33 |
| 1 | 8-bit round key addition | 1 | 16 | 16 |
| 1-9 | Round | 9 | 16+12+80+16 =124 | 1116 |
| 10 | Final round | 1 | 16+12+16= 44 | 56 |
| 0 | 1-D rep. into 2D | 16 | 3 | 48 |
| 10 | 2-D rep. into 1-D | 16 | 3 | 48 |

Total = 1 441 operations

**TABLE 15**: Total Rijndael operations (1 block Auth.)

$T_{rijndael}$ is total number of operations in 1 block encryption.
$T_{rijndael}$ = 1 441
$S_d$ is the size of original message (in bytes).
N is the message size in bits. $N=8 * S_d$
n is the total number of blocks. n = Ceil (N ÷ 128)
where Ceil(x) means the smallest integer ≥ operand
$U_{rijndael}$ is the total number of operations required for Rijndael encryption or decryption of message size Sd.

$$U_{rijndael} = \text{ceil}((8 * S_d) \div 128) * \text{Trijndael} = n * T_{rijndael}$$

Cp is MIPS performed by the processor .
$t_{rijndael}$ ($S_d$, $C_p$) is the time required for encryption (decryption) for processor speed $C_p$ and message size $S_d$ in bytes.
$t_{rijndael}(Sd, Cp) = U_{rijndael}(Sd) \div Cp$ or $t_{rijndael}(Sd, Cp) = (\text{ceil}((8 * Sd) \div 128) * T_{rijndael}) \div Cp$
or $t_{rijndael}(Sd, C_p) = (n * T_{rijndael}) \div C_p$
The mobile devices are equipped with embedded processors, which can perform 100-500 Million of Instructions per Seconds (MIPS) [16].

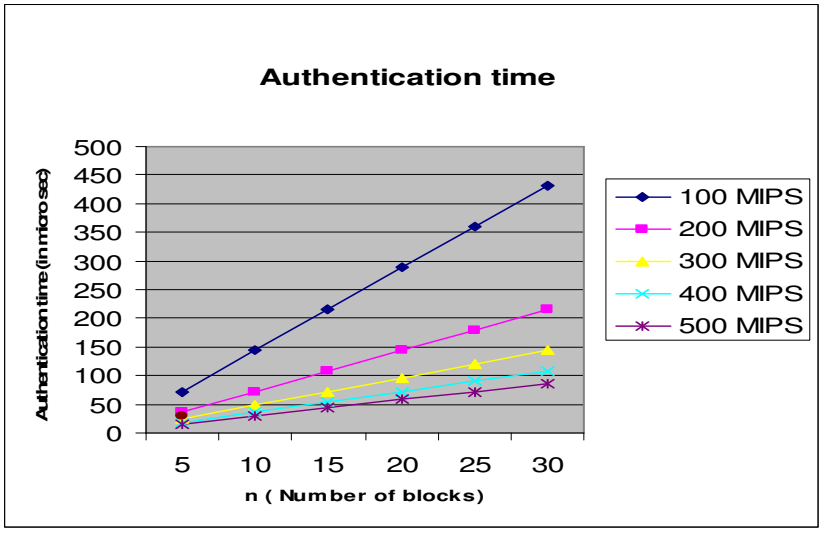



**FIGURE 4:** Authentication time (in μ sec) Vs n of i/p blocks and processing power

**3.2.2 Encryption**

3G encryption uses KASUMI algorithm. KASUMI uses a 128 bit key and block size of 64 bits. The algorithm has 8 distinct steps and 8 rounds [15]. Steps 1 to 8 are functionally identical and are dependent on different portions of input key.

**3.2.2.1. FL Function**

The function FL consists of two XOR (16-bit each), four 16-bit copy, one AND, one OR and two left shifts (cyclic) by one bit each [15]. The input to the function FL comprises a 32-bit data input I, a 32-bit sub key and a 32-bit output.

**3.2.2.2. FO Function**

The input to the function FO comprises a 32-bit data input I and two sets of sub keys, a 48-bit sub key KOi and 48-bit sub key KIi [15]. The 32-bit data input is split into two halves. The 48-bit sub keys are subdivided into three 16-bit sub keys and we return the 32-bit value (L3 || R3). This function consists of six 16-bit XOR, six 16-bit copy and three FI function call.

**3.2.2.3. FI Function**

The function FI [15] takes a 16-bit data input I and 16-bit sub key. The input I is split into two unequal components, a 9-bit left half $L_0$ and a 7-bit right half $R_0$ where $I = L_0 \| R_0$.Similarly the key KIi,j is split into a 7-bit component KIi,j,1 and a 9-bit component $KI_{i,j,2}$ where $KI_{i,j} = KI_{i,j,1} \| KI_{i,j,2}$. The function uses two S-boxes, $S_7$ and S9. The function returns the 16-bit value ($L_4 \| R_4$).This function consists of three 9-bit XOR, three 7-bit XOR and six 7-bit copy. Two times S9 and S7 mappings respectively, and invokes ZE( ) and TR( ) functions twice.

**3.2.2.4. S-boxes**

The two S-boxes [6] have been designed so that they may be easily implemented in combinational logic as well as by a look-up table.

**3.2.2.5. Key Schedule**

KASUMI has a 128-bit key K. Each round of KASUMI uses 128 bits of key that are derived from K [15]. The 128-bit key K is subdivided into eight 16-bit values and a second array of sub-keys, Kj' is derived from Kj. This function consists of eight 16-bit XOR, eight 1-bit cyclic left shift, eight 5-bit cyclic left shift, eight 8-bit cyclic left shift and eight 13-bit cyclic left shift. The extraction of round sub-keys is a 2-D table lookup.

| Basic operation | EQUIVALENT SIMPLE OPERATIONS | | |
|---|---|---|---|
| | TYPE | TIMES NEEDED | SPACE NEEDED |
| ZE | XOR | 1 | |
| TR | XOR | 1 | |
| 2D Table map ( i:j bit map) | MULTIPLY | 1 | |
| | ADD | 1 | |
| | 1-D TABLE LOOKUP | 1 | I ROWS * J COLUMN |
| Left Shift | LEFT SHIF T BY N BITS | 1 | |
| Copy** | N * 32-BITCOPY | N | |
| XOR** | N * 32-BIT XOR | N | |
| AND** | N * 32-BIT AND | N | |
| OR** | N* 32-BIT OR | N | |

** for 32-bit processor and up to 32-bits N=1.

**TABLE 16:** KASUMI basic operations

| Operations | Times | Time needed | Equivalent total |
|---|---|---|---|
| 16-bit XOR | 2 | 1 | 2 |
| 16-bit COPY | 4 | 1 | 4 |
| 16-bit AND | 1 | 1 | 1 |
| 16-bit OR | 1 | 1 | 1 |
| 16-bit left shift | 2 | 1 | 2 |
| 16-bit split | 1 | 3 | 3 |
| 16-bit combine | 1 | 3 | 3 |
| 2D key lookup | 2 | 3 | 6 |

Total=22 operations

**TABLE 17**: KASUMI - FL function

| Operations | Times | Time needed | Equivalent Total |
|---|---|---|---|
| Endian correct | 8 | 5 | 40 |
| K prime | 8 | 2 | 16 |
| construct subkeys | 8 | 34 | 272 |

Total = 328 operations

**TABLE 18:** KASUMI – Keys

| Operations | Times | Time needed | Equivalent total |
|---|---|---|---|
| Key XOR (KIi,j,1 and KIi,j,2) | 2 | 1 | 2 |
| 9-bit XOR | 2 | 1 | 2 |
| 7-bit XOR | 2 | 1 | 2 |
| 7-bit copy | 3 | 1 | 3 |
| 9-bit copy | 3 | 1 | 3 |
| S9 mapping (1-D table map) | 2 | 1 | 2 |
| S7 mapping (1-D table map) | 2 | 1 | 2 |
| ZE | 2 | 1 | 2 |
| TR | 2 | 1 | 2 |
| 7-9 bit split | 1 | 4 | 4 |
| 7-9 bit combine | 1 | 3 | 3 |
| Key split in seven | 1 | 2 | 2 |
| Key split in nine | 1 | 2 | 2 |

Total = 31 operations

**TABLE 19**: KASUMI - FI function

| Operations | Times | Time needed | Equivalent total |
|---|---|---|---|
| 16-bit XOR | 6 | 1 | 6 |
| 16-bit COPY | 6 | 1 | 6 |
| FI ( )* | 3 | 31 | 93 |
| 16 bit split | 1 | 3 | 3 |
| 16 bit combine | 1 | 3 | 3 |
| 2D key lookup | 6 | 3 | 18 |

Total = 129 operations

*in one FO( ), FI ( ) is called three time

**TABLE 20**: KASUMI -- FO function

| step | Operations | times | Time needed | Equivalent total |
|---|---|---|---|---|
| 1-8 | 32-bit COPY | 16 | 1 | 16 |
| 1-8 | 32-bit XOR | 8 | 1 | 8 |
| 1-8 | FL ( ) | 8 | 22 | 176 |
| 1-8 | FO ( ) | 8 | 129 | 1032 |
| | key setup | 1 | 328 | 328 |
| 1 | 32 bit split | 1 | 22 | 22 |
| 8 | 32 bit combine | 1 | 14 | 14 |

Total = 1596 operations

**TABLE 21:** KASUMI operations (1 block encry.)

$T_{kasumi}$ is total number of operations in block (encryption) = 1596.
$T_{kasumi}$ = 1596
$S_d$ is the size of original message (in bytes).
N is the message size in bits. N=8 *$S_d$
n is the total number of blocks. n = Ceil (N ÷ 64)
where Ceil(x) means the smallest integer >= operand
$U_{kasumi}$ is the total number of operations required for KASUMI encryption or decryption of message size Sd.

$$U_{kasumi} = \text{ceil}((8 * S_d) \div 64) * \text{Tkasumi} = n * T_{kasumi}$$

$C_p$ is MIPS performed by the processor.
$t_{kasumi}$ ($S_d$, $C_p$) is the time required for encryption (decryption) for processor speed Cp and message size $S_d$ in bytes.

$$t_{kasumi}(S_d, C_p) = U_{kasumi}(S_d) \div C_p \text{ or } t_{kasumi}(S_d, C_p) = (\text{ceil}((8 * S_d) \div 64) * T_{kasumi}) \div C_p$$

or $t_{kasumi}(S_d, C_p) = (n * T_{kasumi}) \div C_p$

The mobile devices are equipped with embedded processors, which can perform 100-500 Million of Instructions per Seconds (MIPS) [16].

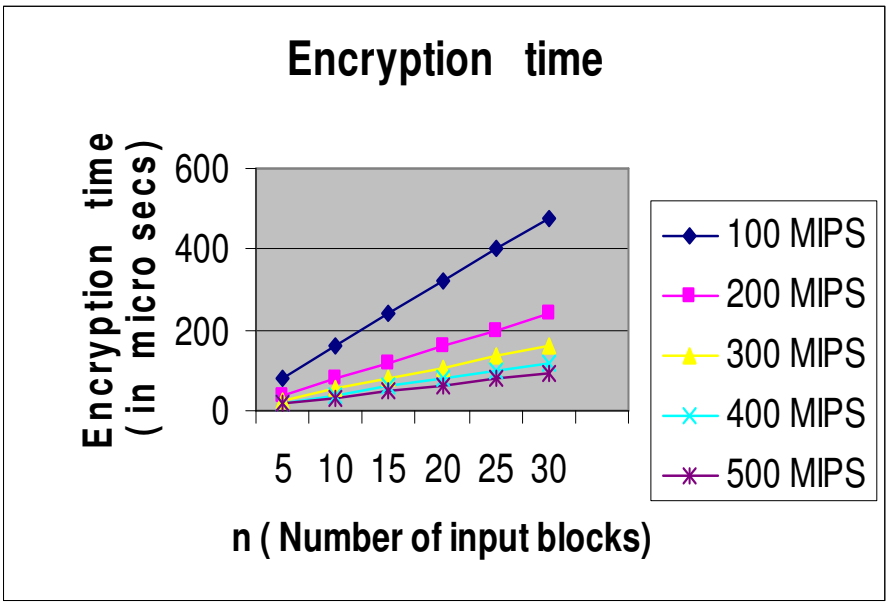



**Figure 5**: Encryption time (in μ sec) as a function of number of packets for different processing speeds (MIPS).

To draw comparison between the computational cost of encryption and authentication, a graph is drawn with number of operations and number of packets as inputs.

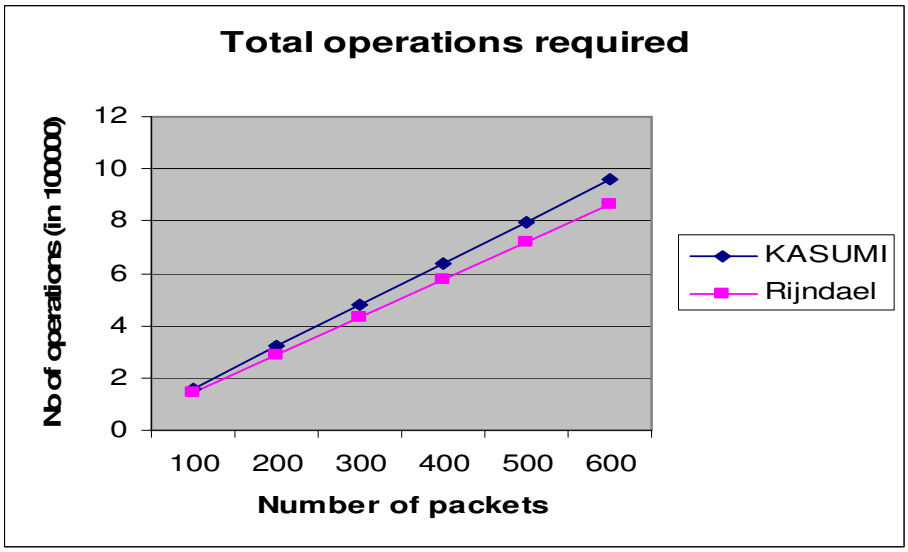



**Figure 6:** Number of operations required as a function of packet size

The above graph clearly shows that number of operations in case of encryption is more as compared to authentication.

## 4. THROUGHPUT

Throughput is defined as the number of bits in one time unit and is measured in Mbps [ 10].

**4.1 Throughput of 2G**

**4.1.1 ThroughputA3A8**

n blocks require tA3A8 (in μ sec) time period for authentication for a given Cp.

Therefore, in 1sec = (n*128 bits) ÷ tA3A8 (in μ sec)

**4.1.2 ThroughputA5/1**

n blocks require tA5/1 (in μ sec) time period for encryption for a given Cp.

Therefore, in 1sec = (n*114 bits) ÷ tA5/1 (in μ sec)

**4.2 Throughput of 3G**

**4.2.1 Throughput$_{kasumi}$**

n blocks require $t_{kasumi}$ (in μ sec) time period for encryption for a given Cp.
Therefore, in 1sec = (n*64 bits) ÷ $t_{kasumi}$ (in μ sec)

**4.2.2 Throughput$_{rijndael}$**

n blocks require $t_{rijndael}$ (in μ sec) time period for authentication for a given Cp.
Therefore, in 1sec = (n*128 bits) ÷ $t_{rijndael}$ (in μ sec)

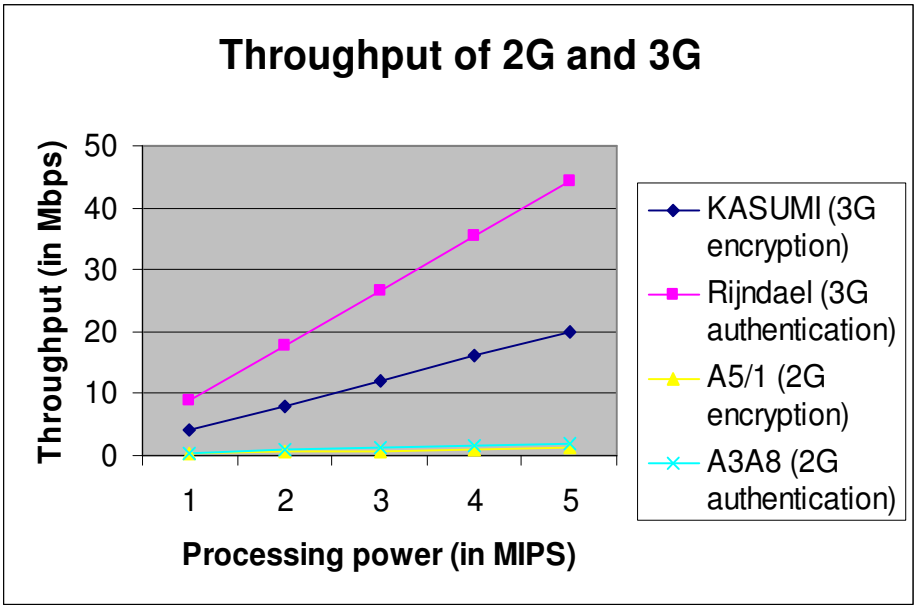



**Figure 7:** Throughput (in Mbps) of 2G and 3G (authentication and encryption algorithm) as a function of processing speed (in MIPS)

The number of operations for 2G authentication and encryption are 32,288 and 43,752 respectively. The encryption requires more number of operations as compared to authentication for same number of input blocks and same processing speed in MIPS. This is very obvious from the figure1, 2 and 3. Similarly in 3G, from figure 6, it is clear that the number of operations for authentication is less as compared to encryption. For one block of authentication 1441 operations are required and for one block of encryption 1596 operations are required. Therefore, time taken for authentication is less as compared to encryption for the same number of input blocks and same processing speed.

The throughput is the number of bits in one time unit. So, more are the number of operations, more is the processing time required, lesser is the bits in one time unit and hence lower throughput. Figure 7 shows authentication and encryption algorithms of 3G provides higher throughput as compared to 2G authentication and encryption algorithms. Even though 3G algorithms are more complex and provides certain enhanced features like two way authentication and integration key based more secure encryption, throughput of 3G is still high as the algorithm is more efficient as compared to 2G.

## 5. FUTURE TRENDS

This work can be further extended to 4G in near future. Though, 4G algorithms (as the predicted date for 4G [7] is given as 2010) will be available after 4G standardized documentation is made available. In 4G, heterogeneity will be the rule instead of exception and it would be of paramount importance to identify and explore the different issues and challenges related to mobility management in 4G. A seamless handoff should be supported between different interfaces like WMAN (using WiMax standard), WPAN (using Bluetooth), WLAN (using WiFi). A study of 802.11, 802.16 and 802.15 standards would be required for ensuring seamless mobility [19, 20].

While shifting from 2G to 3G, to acquire high speed transmission, improved voice quality, global roaming and service flexibility(which means both services – circuit and packet switching), first and the foremost challenge is, interoperation between 2G and 3G. Both the systems use different key lengths. After 3G authentication, the USIM and the SN/HE uses 128 bit cipher and integrity keys CK and IK whereas 2G uses 64 bit cipher key Kc. Therefore, certain conversion functions are needed, that convert the 3G keys to 2G length and vice versa [17].

Except for the transformation complexity and the processor capabilities, the real time required for a packet to be protected may depend on the overall system load as well. Security services not only have significant impact on the system throughput, but security services may further delay data transfer. The mean end to end delay values can be found out as a function of mean data rate for various security scenarios and MS processing capabilities. The mean packet delay is least for unprotected data flow and may vary differently for different algorithms.
Security services also affect mean buffer size. One of the main reason that causes system performance degradation is the packet congestion at the MS because of the computational complexity of the security tasks executed as well as its limited processing capabilities [18]. The mean buffer size at the MS may be calculated as the function of mean data rate for data protection algorithms.

## 6. SUMMARY AND CONCLUSION

The evolution of the security in mobile systems signifies a shift towards open and easily accessible network architecture, which raises major security concerns. The main thrust of research is to develop security which is secure and efficient (in terms of time overhead and space overhead ). The time required for security transformation increases proportionally with the required number of operations, but it also involves the processor capabilities. Since the numbers of operations are greater for encryption than authentication both in 2G and 3G, throughput for encryption is low compared to authentication as encryption consumes significantly more processing resources compared to authentication.
The time required for authentication is less as compared to encryption in both 2G and 3G respectively. However, throughput of 3G for both authentication and encryption is higher than that of 2G. 2G requires more number of processing resources as compared to 3G.The mobile companies are shifting from 2G to 3G for the following reasons:

i. Higher throughput of 3G as compared to 2G
ii. 3G is more secure than 2G. 3G offers two-way authentication i.e. not only network authenticates mobile equipment, mobile equipment also authenticates network, so as to overcome fraud base station attack.
iii. Higher data transfer bandwidth incase of 3G.

To reduce computational overheads encryption should be used in critical user information only and not for regular traffic flow. Encryption if needed should be combined with authentication. In this case if the message fails authentication, decryption process is saved (not performed).
Further the performance analysis determines the cost (in terms of time complexity and throughput). Quantifying the security overhead makes mobile users and mobile network operators aware of the price of added security features and further helps in making optimized security policy configurations.
Finally, except for the transformation complexity and the processor capabilities, the real time required for a packet to be protected depends on the overall system load and traffic conditions as well.

## 7. REFERENCES

1. Lauri Pesonen "*GSM Interception*", lecture notes, Helsinki University of technology, Lauri.Pesonen@iki.fi, 1999.

2. Paulo S. Pagliusi "*A contemporary foreword on GSM Security*", Lecture notes, Royal Holloway, University of London, UK, http://www.isg.rhul.ac.uk

3. Chengyuan Peng, "GSM and GPRS Security", Helsinki university of technology, IIUT TML 2000, TIK 110.501 Seminar on network security.

4. Stefan Pitz, Roland Schmitz, Tobias Martin,"*Security mechanisms in UMTS*", Datenschutz und Datensicherheit (DUD), vol. 25, pp 1-10. 2001.

5. Christos Xenakis, Lazaros Merakos, "*Security in 3G Mobile network*", Computer communications, vol. 27, pp 638-650, 2004.

6. 3GPP TS 33.102 (v3.12.0), " *3G Security: Security Architecture*", Release '99, June 2002.

7. Jun- Zhao Sun, Jaakko Sauvola, Douglas Howie " *Features in Future: 4G visions from a technical perspective* " , IEEE, 0-7803-7206-9/01, pp 3533-3537, 2001.

8. Suk Yu Hui and Kai Hau Yeung " *Challenges in the Migration to 4G Mobile Systems*", IEEE Communications Magazine, Dec 2003.

9. Christos Xenakis, Lazaros Merakos, "IPsec *based end_to_end VPN deployment over UMTS*", Computer Communications vol. 27, pp.1693-1708, May 2004.

10. O. Elkeelany, M. Matalgah, K. Sheikh, M. Thaker, G. Chaudhry, D. Medhi, J. Qaddouri, "*Performance analysis of IPSec protocol: encryption and authentication*", Proc. IEEE Int'l Conf. Communications, New York, NY, pp. 1164-1168, April-May 2002.

11. Marc Briceno, Ian Goldberg, David Wagner, "*An implementation of GSM $A_3$, $A_8$ algorithm*", http://www.scard.org/gsm/a3a8, 1999.

12. Ross Anderson, " *$A_5$ The GSM Encryption Algorithm*", sci.crypt, 1994.

13. 3GPP TS 35.205 (v.1.0), " *Specification of Milenage algorithm for the 3GPP authentication and key generation functions*", Nov 2000.

14. 3GPP TS 33.105 (v4.1.0), "3G *security: cryptographic algorithm requirements*", release 4, 2001.

15. 3GPP TS 35.201 (v.1.0), "*Specification of 3GPP confidentiality and integrity algorithm*", Document 1: $f_8$, $f_9$ specification, Dec 1999.

16. ARM microprocessor solutions from ARM Ltd, http://www.arm.com/products/CPUs.

17. 3GPP, [TR 31.900] . "*Interworking between 2G and 3G*" , V 5.1.0 , 2002.

18. Qingyang song and Abbas Jamalipour, " *A network selection mechanism for next generation networks*", IEEE-0-7803-8938-7/05, pp1418-1422, July 2005.

19. Pablo Vidales, Javier Baliosian, Joan Serrat, "*Autonomic system for mobility support in 4G networks*", IEEE Journal on selected areas in communications volume 23, No.12, IEEE 0733-8716, pp. 2288-2303, Dec 2005.

20. Chang Wei Lee, Li Ming Chen, Meng Chang Chen, Yeali Sunny Sun, "*A framework of handoffs in wireless overlay networks based on mobile IPV6*", IEEE journal on selected areas in communications, vol 23 No 11, pp 2118-2128, Nov 2005.

# Novel Methods of Generating Self-Invertible Matrix for Hill Cipher Algorithm

**Bibhudendra Acharya** bibhudendra@gmail.com
*Department of Electronics & Communication Engineering*
*National Institute of Technology Rourkela*
*Rourkela, 769 008, India*

**Girija Sankar Rath** gsrath@nitrkl.ac.in
*Professor, Department of Electronics & Communication Engineering*
*National Institute of Technology Rourkela*
*Rourkela, 769 008, India*

**Sarat Kumar Patra** skpatra@nitrkl.ac.in
*Professor, Department of Electronics & Communication Engineering*
*National Institute of Technology Rourkela*
*Rourkela, 769 008, India*

**Saroj Kumar Panigrahy** skp.nitrkl@gmail.com
*Department of Computer Science & Engineering*
*National Institute of Technology Rourkela*
*Rourkela, 769 008, India*

**Abstract**

In this paper, methods of generating self-invertible matrix for Hill Cipher algorithm have been proposed. The inverse of the matrix used for encrypting the plaintext does not always exist. So, if the matrix is not invertible, the encrypted text cannot be decrypted. In the self-invertible matrix generation method, the matrix used for the encryption is itself self-invertible. So, at the time of decryption, we need not to find inverse of the matrix. Moreover, this method eliminates the computational complexity involved in finding inverse of the matrix while decryption.

**Keywords:** Hill Cipher, Encryption, Decryption, Self-invertible matrix.

## 1. INTRODUCTION

Today, in the information age, the need to protect communications from prying eyes is greater than ever before. Cryptography, the science of encryption, plays a central role in mobile phone communications, pay-TV, e-commerce, sending private emails, transmitting financial information, security of ATM cards, computer passwords, electronic commerce and touches on many aspects of our daily lives [1]. Cryptography is the art or science encompassing the principles and methods of transforming an intelligible message (plaintext) into one that is unintelligible (ciphertext) and then retransforming that message back to its original form. In modern times, cryptography is considered to be a branch of both mathematics and computer science, and is affiliated closely with information theory, computer security, and engineering [2].

Conventional Encryption is referred to as symmetric encryption or single key encryption. It can be further divided into categories of classical techniques and modern techniques. The hallmark of conventional encryption is that the cipher or key to the algorithm is shared, i.e., known by the parties involved in the secured communication. Substitution cipher is one of the basic components of classical ciphers. A substitution cipher is a method of encryption by which units of plaintext are substituted with ciphertext according to a regular system; the units may be single letters (the most common), pairs of letters, triplets of letters, mixtures of the above, and so forth. The receiver deciphers the text by performing an inverse substitution [3]. The units of the plaintext are retained in the same sequence as in the ciphertext, but the units themselves are altered. There are a number of different types of substitution cipher. If the cipher operates on single letters, it is termed a simple substitution cipher; a cipher that operates on larger groups of letters is termed polygraphic. A monoalphabetic cipher uses fixed substitution over the entire message, whereas a polyalphabetic cipher uses a number of substitutions at different times in the message— such as with homophones, where a unit from the plaintext is mapped to one of several possibilities in the ciphertext. Hill cipher is a type of monoalphabetic polygraphic substitution cipher.

In this paper, we proposed novel methods of generating self-invertible matrix which can be used in Hill cipher algorithm. The objective of this paper is to overcome the drawback of using a random key matrix in Hill cipher algorithm for encryption, where we may not be able to decrypt the encrypted message, if the matrix is not invertible. Also the computational complexity can be reduced by avoiding the process of finding inverse of the matrix at the time of decryption, as we use self-invertible key matrix for encryption.

The organization of the paper is as follows. Following the introduction, the basic concept of Hill Cipher is outlined in section 2. Section 3 discusses about the modular arithmetic. In section 4, proposed methods for generating self-invertible matrices are presented. Finally, section 5 describes the concluding remarks.

## 2. HILL CIPHER

It is developed by the mathematician Lester Hill in 1929. The core of Hill cipher is matrix manipulations. For encryption, algorithm takes $m$ successive plaintext letters and instead of that substitutes $m$ cipher letters. In Hill cipher, each character is assigned a numerical value like $a = 0, b = 1, \ldots, z = 25$ [4]. The substitution of ciphertext letters in the place of plaintext letters leads to $m$ linear equation. For $m = 3$, the system can be described as follows:

$$\begin{aligned} C_1 &= (K_{11}P_1 + K_{12}P_2 + K_{13}P_3) \bmod 26 \\ C_2 &= (K_{21}P_1 + K_{22}P_2 + K_{23}P_3) \bmod 26 \\ C_1 &= (K_{31}P_1 + K_{32}P_2 + K_{33}P_3) \bmod 26 \end{aligned} \quad \ldots (1)$$

This case can be expressed in terms of column vectors and matrices:

$$\begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix} \begin{pmatrix} P_1 \\ P_2 \\ P_3 \end{pmatrix} \quad \ldots (2)$$

or simply we can write as $C = KP$, where $C$ and $P$ are column vectors of length 3, representing the plaintext and ciphertext respectively, and $K$ is a $3 \times 3$ matrix, which is the encryption key. All operations are performed $\bmod 26$ here. Decryption requires using the inverse of the matrix $K$. The inverse matrix $K^{-1}$ of a matrix $K$ is defined by the equation $KK^{-1} = K^{-1}K = I$, where $I$ is the Identity matrix. But the inverse of the matrix does not always exist, and when it does, it satisfies the preceding equation. $K^{-1}$ is applied to the ciphertext, and then the plaintext is recovered. In general term we can write as follows:

For encryption: $C = E_k(P) = K_p$ … (3)

For decryption: $P = D_k(C) = K^{-1}C = K^{-1}K_p = P$ … (4)

## 3. MODULAR ARITHMETIC

The arithmetic operation presented here are addition, subtraction, unary operation, multiplication and division [5]. Based on this, the self-invertible matrix for Hill cipher algorithm is generated. The congruence modulo operator has the following properties:

1. $a \equiv b \mod p$ if $n \mid (a-b)$
2. $(a \mod p) = (b \mod p) \Rightarrow a \equiv b \mod p$
3. $a \equiv b \mod p \Rightarrow b \equiv a \mod p$
4. $a \equiv b \mod p$ and $b \equiv a \mod p \Rightarrow a \equiv c \mod p$

Let $Z_p = [0,1,..., p-a]$ the set of residues modulo $p$. If modular arithmetic is performed within this set $Z_p$, the following equations present the arithmetic operations:

1. Addition : $(a+b) \bmod p = [(a \bmod p) + (b \bmod p)] \bmod p$
2. Negation : $-a \bmod p = p - (a \bmod p)$
3. Subtraction : $(a-b) \bmod p = [(a \bmod p) - (b \bmod p)] \bmod p$
4. Multiplication : $(a*b) \bmod p = [(a \bmod p) * (b \bmod p)] \bmod p$
5. Division : $(a/b) \bmod p = c$ when $a = (b*c) \bmod p$

The following Table exhibits the properties of modular arithmetic.

| Property | Expression |
|---|---|
| Commutative Law | $(\omega + x) \bmod p = (x + \omega) \bmod p$<br>$(\omega * x) \bmod p = (x * \omega) \bmod p$ |
| Associative law | $[(\omega + x) + y] \bmod p = [\omega + (x + y)] \bmod p$ |
| Distribution Law | $[\omega * (x + y)] \bmod p = [\{(\omega * x) \bmod p\} * \{(\omega * y) \bmod p\}] \bmod p$ |
| Identities | $(0 + a) \bmod p = a \bmod p$<br>$and\ (1 * a) \bmod p = a \bmod p$ |
| Inverses | For each $x \in Z_p$, $\exists\ y$ such that $(x + y) \bmod p = 0$ then $y = -x$<br>For each $x \in Z_p$ $\exists\ y$ such that $(x * y) \bmod p = 1$ |

**Table 1:** Properties of Modular Arithmetic

## 4. PROPOSED METHODS FOR GENERATING SELF-INVERTIBLE MATRIX

As Hill cipher decryption requires inverse of the matrix, so while decryption one problem arises that is, inverse of the matrix does not always exist [5]. If the matrix is not invertible, then encrypted text cannot be decrypted. In order to overcome this problem, we suggest the use of self-invertible matrix generation method while encryption in the Hill Cipher. In the self-invertible matrix generation method, the matrix used for the encryption is itself self-invertible. So, at the time of decryption, we need not to find inverse of the matrix. Moreover, this method eliminates the computational complexity involved in finding inverse of the matrix while decryption.

$A$ is called self-invertible matrix if $A = A^{-1}$. The analyses presented here for generation of self-invertible matrix are valid for matrix of +ve integers, that are the residues of modulo arithmetic on a prime number.

### 4.1 Generation of self-invertible $2\times 2$ matrix

Let $A=\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$, then, $A^{-1}=\dfrac{\text{adjoint}(A)}{\text{determinant}(A)}=\dfrac{(\text{cofactor}(A))^T}{\text{determinant}(A)}$

$\therefore A^{-1}=\dfrac{1}{\Delta a}\begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix}$, where, $\Delta a$ is the $\text{determinant}(A)$

$A$ is said to be self-invertible if $A=A^{-1}$

So, $a_{12}=-a_{12}/\Delta a$ & $a_{21}=-a_{21}/\Delta a$

$\therefore \Delta a=-1$ and $a_{11}=-a_{22} \Rightarrow a_{11}+a_{22}=0$ … (5)

**Example:** (For modulo 13)

$A=\begin{bmatrix} 2 & 3 \\ 12 & 11 \end{bmatrix}$

### 4.2 Generation of self-invertible $3\times 3$ matrix

Let $A=\left[\begin{array}{c:cc} a_{11} & a_{12} & a_{13} \\ \hdashline a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{array}\right]=\left[\begin{array}{c:c} A_{11} & A_{12} \\ \hdashline A_{21} & A_{22} \end{array}\right]$

where $A_{11}$ is a $1\times 1$ matrix = $[a_{11}]$, $A_{12}$ is a $1\times 2$ matrix = $[a_{12} \quad a_{13}]$,

$A_{21}$ is a $2\times 1$ matrix = $\begin{bmatrix} a_{21} \\ a_{31} \end{bmatrix}$ and $A_{22}$ is $2\times 2$ matrix = $\begin{bmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{bmatrix}$

If $A$ is self-invertible then,

$A_{11}{}^2+A_{12}A_{21}=I, \quad A_{11}A_{12}+A_{12}A_{22}=0,$

$A_{21}A_{11}+A_{22}A_{21}=0, \quad$ and $A_{21}A_{12}+A_{22}^2=I$ … (6)

Since $A_{11}$ is $1\times 1$ matrix = $[a_{11}]$ and $A_{21}(a_{11}I+A_{22})=0$

For non- trivial solution, it is necessary that $a_{11}I+A_{22}=0$

That is $a_{11}=-$ (one of the Eigen values of $A_{22}$)

$A_{21}A_{12}$ can also be written as

$A_{21}A_{12}=\begin{bmatrix} a_{21} & 0 \\ a_{31} & 0 \end{bmatrix}\begin{bmatrix} a_{12} & a_{13} \\ 0 & 0 \end{bmatrix}=\begin{bmatrix} a_{21}a_{12} & a_{21}a_{13} \\ a_{31}a_{12} & a_{31}a_{13} \end{bmatrix}$

So $A_{21}\ A_{12}$ is singular and

$A_{21}A_{12}=I-A_{22}^2$ … (7)

Hence $A_{22}$ must have an Eigen value $\pm 1$. It can be shown that $Trace[A_{21}A_{12}]=A_{12}A_{21}$.

Since it can be proved that if $A_{11}=a_{11}=-$ (one of the Eigen values of $A_{22}$),

then, any non-trivial solution of the equation (7) will also satisfy

$A_{12}A_{21}=1-a_{11}{}^2$ … (8)

**Example:** (For modulo 13)

Take $A_{22}=\begin{bmatrix} 2 & 5 \\ 1 & 6 \end{bmatrix}$ which has Eigen value $\lambda=1$ and 7

$a_{11}=-7=6$ or $-1=12$

If $a_{11}=6$,

then, $A_{21}A_{12}=I-A_{22}^2=I-\begin{bmatrix} 2 & 5 \\ 1 & 6 \end{bmatrix}\begin{bmatrix} 2 & 5 \\ 1 & 6 \end{bmatrix}=I-\begin{bmatrix} 9 & 1 \\ 8 & 2 \end{bmatrix}=\begin{bmatrix} 5 & 12 \\ 5 & 12 \end{bmatrix}$

$a_{21}a_{12}=5$. So, $a_{21}=5$ and $a_{12}=1$

$a_{21}a_{13}=12$. So, $a_{13}=\frac{12}{5}=5$ and $a_{31}=\frac{5}{1}=5$

So the matrix will be $A=\begin{bmatrix}6 & 1 & 5\\ 5 & 2 & 5\\ 5 & 1 & 6\end{bmatrix}$. Other matrix can also be obtained if we take $a_{11}=12$.

### 4.3 Generation of self-invertible $4\times4$ matrix

Let $A=\left[\begin{array}{cc:cc} a_{11} & a_{12} & a_{13} & a_{14}\\ a_{21} & a_{22} & a_{23} & a_{24}\\ \hdashline a_{31} & a_{32} & a_{33} & a_{34}\\ a_{41} & a_{42} & a_{43} & a_{44}\end{array}\right]$ be self-invertible matrix partitioned as $A=\left[\begin{array}{c:c} A_{11} & A_{12}\\ \hdashline A_{21} & A_{22}\end{array}\right]$,

where $A_{11}=\begin{bmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{bmatrix}$, $A_{12}=\begin{bmatrix}a_{13} & a_{14}\\ a_{23} & a_{24}\end{bmatrix}$, $A_{21}=\begin{bmatrix}a_{31} & a_{32}\\ a_{41} & a_{42}\end{bmatrix}$, $A_{22}=\begin{bmatrix}a_{33} & a_{34}\\ a_{43} & a_{42}\end{bmatrix}$

Then, $A_{12}A_{21}=I-A_{11}^{\,2}$, $A_{11}A_{12}+A_{12}A_{22}=0$,

$A_{21}A_{11}+A_{22}A_{21}=0$, and $A_{21}A_{12}=I-A_{22}^2$

In order to obtain solution for all the four matrix equations, $A_{12}A_{21}$ can be factorized as

$$A_{12}A_{21}=(I-A_{11})(I+A_{11}) \quad \ldots (9)$$

So, if $A_{12}=(I-A_{11})k$ or $(I+A_{11})k$

$A_{21}=(I+A_{11})\frac{1}{k}$ or $(I-A_{11})\frac{1}{k}$, where $k$ is a scalar constant.

Then, $A_{11}A_{12}+A_{12}A_{22}=A_{11}(I-A_{11})k+(I-A_{11})kA_{22}$ or $k(A_{11}+A_{22})(I-A_{11})$

So, $A_{11}+A_{22}=0 \quad or \quad A_{11}=I$ … (10)

Since $A_{11}=I$ is a trivial solution, then, $A_{11}+A_{22}=0$ is taken.

When we solve the 3rd and 4th matrix equations, same solution is obtained.

**Example:** (For Modulo 13)

Take $A_{22}=\begin{bmatrix}1 & 3\\ 8 & 4\end{bmatrix}$ then, $A_{11}=\begin{bmatrix}12 & 10\\ 5 & 9\end{bmatrix}$

Take $A_{12}=I-A_{11}$ with $k=1$. Then, $A_{12}=\begin{bmatrix}2 & 3\\ 8 & 5\end{bmatrix}$ and $A_{21}=\begin{bmatrix}0 & 10\\ 5 & 10\end{bmatrix}$

So $A=\begin{bmatrix}12 & 10 & 2 & 3\\ 5 & 9 & 8 & 5\\ 0 & 10 & 1 & 3\\ 5 & 10 & 8 & 4\end{bmatrix}$

### 4.4 A general method of generating an even self-invertible matrix

Let $A=\left[\begin{array}{ccc:cc} a_{11} & a_{12} & \ldots & \ldots & a_{1n}\\ a_{21} & a_{22} & \ldots & \ldots & a_{2n}\\ \ldots & \ldots & \ldots & \ldots & \ldots\\ \hdashline \ldots & \ldots & \ldots & \ldots & \ldots\\ a_{n1} & a_{n2} & \ldots & \ldots & a_{nn}\end{array}\right]$ be an $n\times n$ self-invertible matrix partitioned to $A=\left[\begin{array}{c:c} A_{11} & A_{12}\\ \hdashline A_{21} & A_{22}\end{array}\right]$,

where $n$ is even and $A_{11}$, $A_{12,}A_{21}$ & $A_{22}$ are matrices of order $\frac{n}{2}\times\frac{n}{2}$ each.

So, $A_{12}A_{21} = I - A_{11}^2 = (I - A_{11})(I + A_{11})$ … (11)

If $A_{12}$ is one of the factors of $I - A_{11}^2$ then $A_{21}$ is the other.
Solving the 2nd matrix equation results $A_{11} + A_{22} = 0$.
Then form the matrix.

**Algorithm:**

1. Select any arbitrary $\frac{n}{2} \times \frac{n}{2}$ matrix $A_{22}$.
2. Obtain $A_{11} = -A_{22}$
3. Take $A_{12} = k(I - A_{11})$ or $k\ (I + A_{11})$ for $k$ a scalar constant.
4. Then $A_{21} = \frac{1}{k}(I + A_{11})$ or $\frac{1}{k}(I - A_{11})$
5. Form the matrix completely.

**Example:** (For modulo 13)
Let $A_{22} = \begin{bmatrix} 10 & 2 \\ 3 & 4 \end{bmatrix}$, then, $A_{11} = \begin{bmatrix} 3 & 11 \\ 10 & 9 \end{bmatrix}$
If $k$ is selected as 2, $A_{12} = k(I - A_{11}) = \begin{bmatrix} 9 & 4 \\ 6 & 10 \end{bmatrix}$ and $A_{21} = \begin{bmatrix} 2 & 12 \\ 5 & 5 \end{bmatrix}$
So, $A = \begin{bmatrix} 3 & 11 & 9 & 4 \\ 10 & 9 & 6 & 10 \\ 2 & 12 & 10 & 2 \\ 5 & 5 & 3 & 4 \end{bmatrix}$

### 4.5 A general method of generating self-invertible matrix

Let $A = \left[\begin{array}{c:ccc} a_{11} & a_{12} & \dots & a_{1n} \\ \hdashline a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{array}\right]$ be an $n \times n$ self-invertible matrix partitioned to $A = \left[\begin{array}{c:c} A_{11} & A_{12} \\ \hdashline A_{21} & A_{22} \end{array}\right]$

$A_{11}$ is a $1 \times 1$ matrix = $[a_{11}]$, $A_{12}$ is a $1 \times (n-1)$ matrix = $[a_{12} \quad a_{13} \dots \quad a_{1n}]$

$A_{21}$ is a $(n-1) \times 1$ matrix = $\begin{bmatrix} a_{21} \\ a_{31} \\ \dots \\ a_{n1} \end{bmatrix}$, $A_{22}$ is a $(n-1) \times (n-1)$ matrix = $\begin{bmatrix} a_{22} & a_{23} & \dots & a_{2n} \\ a_{32} & a_{33} & \dots & a_{3n} \\ \dots & \dots & \dots & \dots \\ a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix}$

So, $A_{12}\ A_{21} = I - {A_{11}}^2 = 1 - {a_{11}}^2$ … (12)
and $A_{12}(a_{11}I + A_{22}) = 0$ … (13)
Also, $a_{11} =$ – (one of the Eigen values of $A_{22}$ other than 1)
Since $A_{21}A_{12}$ is a singular matrix having the rank 1
and $A_{21}A_{12} = I - A_{22}^2$ … (14)
So, ${A_{22}}^2$ must have rank of $(n-2)$ with Eigen values +1 of $(n-2)$ multiplicity.
Therefore, $A_{22}$ must have Eigen values $\pm 1$.
It can also be proved that the consistent solution obtained for elements $A_{21}$ & $A_{12}$ by solving the equation (14) term by term will also satisfy the equation (12).

**Algorithm:**

1. Select $A_{22}$, a non-singular $(n-1)\times(n-1)$ matrix which has $(n-2)$ number of Eigen values of either +1 or –1 or both.
2. Determine the other Eigen value $\lambda$ of $A_{22}$.
3. Set $a_{11} = -\lambda$.
4. Obtain the consistent solution of all elements of $A_{21}$ & $A_{12}$ by using the equation (14).
5. Formulate the matrix.

**Example:** (For modulo 13)

Let $A_{22} = \begin{bmatrix} 9 & 6 & 10 \\ 12 & 10 & 2 \\ 5 & 3 & 4 \end{bmatrix}$ which has Eigen values $\lambda = \pm 1,\ 10$

So, $A_{11} = [3]$, and one of the consistent solutions of $A_{12} = [11 \quad 9 \quad 4]$ and $A_{21} = \begin{bmatrix} 10 \\ 2 \\ 5 \end{bmatrix}$

So, $A = \begin{bmatrix} 3 & 11 & 9 & 4 \\ 10 & 9 & 6 & 10 \\ 2 & 12 & 10 & 2 \\ 5 & 5 & 3 & 4 \end{bmatrix}$

Another consistent solution of $A_{12} = [1 \quad 2 \quad 11]$ and $A_{21} = \begin{bmatrix} 6 \\ 9 \\ 3 \end{bmatrix}$

So, $A = \begin{bmatrix} 3 & 1 & 2 & 11 \\ 6 & 9 & 6 & 10 \\ 9 & 12 & 10 & 2 \\ 3 & 5 & 3 & 4 \end{bmatrix}$

### 4.6 Another method to generate self-invertible matrix

Let $A$ be any non-singular matrix and $E$ be its Eigen matrix. Then we know that $AE = E\lambda$, where $\lambda$ is diagonal matrix with the Eigen values as diagonal elements. $E$ the Eigen matrix is non-singular.

Then, $A = E\ \lambda E^{-1}$ … (15)

and $A^{-1} = \left(E\lambda E^{-1}\right)^{-1} = E^{-1}\ \lambda^{-1} E = E\lambda^{-1}E^{-1}$ … (16)

So, $A = A^{-1}$ only when $\lambda = \lambda^{-1}$

If $\lambda = \begin{bmatrix} \lambda_1 & 0 & 0 & 0 & . & . & 0 \\ 0 & \lambda_2 & 0 & 0 & . & . & 0 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & . & . & \lambda_n \end{bmatrix}$ then, $\lambda^{-1} = \begin{bmatrix} \frac{1}{\lambda_1} & 0 & 0 & . & . & . & 0 \\ 0 & \frac{1}{\lambda_2} & 0 & 0 & . & . & 0 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\lambda_n} \end{bmatrix}$

Thus $\lambda = \lambda^{-1}$ when $\lambda_i = \frac{1}{\lambda_i}$ or $\lambda_i = \pm 1$

**Algorithm:**

1. Select any nonsingular matrix $E$ .
2. Form a diagonal matrix $\lambda$ with $\lambda = \pm 1$ but all value of $\lambda$ must not be equal.
3. Then compute $E\lambda E^{-1} = A$ .

**Example:** (For modulo 13)

$$E = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 8 \end{bmatrix}, \quad E^{-1} = \begin{bmatrix} -\frac{8}{3} & \frac{8}{3} & -1 \\ \frac{10}{3} & \frac{13}{3} & 2 \\ -1 & 2 & -1 \end{bmatrix} = \begin{bmatrix} 6 & 7 & 12 \\ 12 & 0 & 2 \\ 12 & 2 & 12 \end{bmatrix}$$

Take $\lambda = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 12 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

$$A = E\ \lambda\ E^{-1} = \begin{bmatrix} 1 & 11 & 3 \\ 4 & 8 & 6 \\ 7 & 5 & 8 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 12 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 6 & 7 & 12 \\ 12 & 0 & 2 \\ 12 & 2 & 12 \end{bmatrix} = \begin{bmatrix} 5 & 0 & 5 \\ 10 & 1 & 6 \\ 3 & 0 & 8 \end{bmatrix}$$

## 5. CONCLUSION

This paper suggests efficient methods for generating self-invertible matrix for Hill Cipher algorithm. These methods encompass less computational complexity as inverse of the matrix is not required while decrypting in Hill Cipher. These proposed methods for generating self-invertible matrix can also be used in other algorithms where matrix inversion is required.

## 6. REFERENCES

1. Blakley G.R., "*Twenty years of cryptography in the open literature*", Security and Privacy 1999, Proceedings of the IEEE Symposium, 9-12 May 1999
2. Imai H., Hanaoka G., Shikata J., Otsuka A., Nascimento A.C., "*Cyptography with Information Theoretic Security*", Information Theory Workshop, 2002, Proceedings of the IEEE, 20-25 Oct 2002
3. A. J. Menezes, P.C. Van Oorschot, S.A. Van Stone, "*Handbook of Applied Cryptography*", CRC press, 1996
4. W. Stallings, "*Cryptography and Network Security*", 4th edition, Prentice Hall, 2005
5. Bruce Schneir, "*Applied Cryptography*", 2nd edition, John Wiley & Sons, 1996

# Multi-Dimensional Privacy Protection for Digital Collaborations

**Geoff Skinner** Geoff.Skinner@newcastle.edu.au
*Faculty of Science and IT*
*University of Newcastle*
*Callaghan, 2308, Australia*

**Abstract**

In order to sustain privacy in digital collaborative environments a comprehensive multidimensional privacy protecting framework is required. Such information privacy solutions for collaborations must incorporate environmental factors and influences in order to provide a holistic information privacy solution. Our Technical, Legal, and Community Privacy Protecting (TLC-PP) framework addresses the problems associated with the multi-facetted notion of privacy. The three key components of the TLC-PP framework are merged together to provide complete solutions for collaborative environment stakeholders and users alike. The application of the TLC-PP framework provides a significant contribution to the delivery of a Privacy Augmented Collaborative Environment (PACE).

**Keywords:** Information Privacy, Privacy Evaluator Module (PEM), Manual Privacy Management (MPM), Community Observed Privacy (COP), TLC-PP.

## 1. INTRODUCTION

Collaborative environments fulfill a very important role in a knowledge society, providing a digital 'place' for the exchange of ideas and knowledge, seen as one of the most important activities of man [1]. The storing of data in a commonly accessible structure has both a great potential for the knowledge society as well as a high risk for the user's privacy. Here in lies one of the greatest challenges for collaborative environments. That is, a continual balance must be sought between the interests of open easily accessible information with the protection of personal data and entity privacy. Therefore, information privacy and collaborative environments are two information system related concepts that are identified as priority research fields [2] and [3], vital to the continued and successful growth of many Information Communications and Technology (ICT) dependant industries.

A number of areas including e-Business, e-Learning, knowledge management, and intelligent analysis are direct beneficiaries of advances in information privacy protection in collaborative environments. Significantly improving information privacy protection and personal data management in collaborative environments provides many advantages to information requestors and information providers alike. Strong privacy controls are a major contributor to increased trust between member entities [4] which in turn can facilitate increased participation and contribution to a collaborative environment. As the collaboration grows so to does the need to ensure privacy is preserved along with clearly defined bounds of information flow for effective personal data management.

Ongoing research into the field of Collaborative Environments (CEs) has produced a number of potentially beneficial results for knowledge sharing and increasing productivity for small to

medium enterprises. CE's by their very nature promote cooperation and the development of open and adaptive technologies [5]. Such environments present many interesting issues and challenges for information privacy and data security. As with classical computer system evolution the relatively new field of e-collaborative environments is already at risk of following a similar path of overlooking information privacy concerns. Clarke [6] defines information privacy as being a combination of communications and data privacy. Formally defined as '… the interest an individual has in controlling, or at least significantly influencing, the handling of data about themselves' [6].

The focus of this paper is to provide a foundational perspective of our work investigating Information Privacy issues in the realm of collaborative environments. Information Privacy conformance needs to be integrated from system inception, but an effective privacy solution must be a symbiotic molding of technical, legal, and social elements. Due to the complex systems involved and their self-organizing nature no single model of privacy protection is adequate for collaborative environments. Rather, all models need to be incorporated into the environments and continually monitored and updated to ensure they maintain privacy while also facilitating the functionality of the collaboration.

The rest of the paper follows a common structure outline as follows. Section 2 provides relevant background material on Information Privacy and research in this area. Additionally, a review of our previous work and publications in the field are discussed. Current collaborative environment approaches to Information Privacy and Data Security is included in Section 3. Section 4 provides our proposals of the TLC Framework for Collaborative Environments and the importance of the TLC-PP framework for a Privacy Augmented Collaborative Environment (PACE). A brief conclusion and future work is provided in Section 5.

## 2. BACKGROUND AND RELATED WORK

Modern privacy solutions are often derived from the application, both in combination and isolation, of the four main models of privacy protection [7]. The models listed in [7] are Comprehensive Laws, Sectoral Laws, Self Regulation, and Technologies of Privacy. Of interest to our own work is the impact of collaborative environments on information privacy and what modifications are required for privacy protections to operate effectively in collaborations. The reason being is that many of the technology of privacy solutions, that are proving to be the most popular form of protection, rely on varying levels of computationally secure methods, such as encryption, to provide security and privacy of personal data [8]. With progression to more open collaborations and increased data sharing, application and regulation of personal data protection methods become more complex. As the collaborations become more distributed and composed of an increasing number of information systems it becomes harder to ensure consistency and enforcement for all types of privacy protection. Further, maintenance of privacy controls becomes more complex across diverse distributed systems that may differ in operating environments and requirements. This often results in devaluing or overlooking information privacy.

Our focus is on Information Privacy rather than Information Security, and specifically the development of a comprehensive collaboration wide approach to information privacy. From a technological perspective this involves the development and integration of Privacy Enhancing Technologies [8] with legislative, regulatory and social components. The uniqueness of privacy in terms of its subjective nature and openness to individual interpretation and representation has allowed it to evolve with similar advances in technology, society, culture and values [9]. In the field of IS research privacy solutions are not always based on technological approaches. The use and enforcement of legal regulations, laws (sectoral and comprehensive), and even self regulation attempts will still be applicable and perhaps even more significant to information privacy in distributed collaborative environments. However, we readily acknowledge that protection against intentional malicious attacks is still heavily reliant on technological solutions.

Therefore, a number of PETs make extensive use of encryption in some manner to help protect privacy. These include the Identity Protector [10], Shield Privacy [11], and Privacy Protector [12].

From a social privacy protection perspective what is important is the fact that information privacy benefits from any type of exposure. Raising user and system owner's awareness is an important phase in the over all process of protection of personal data and entity privacy. Collaborative environments assist in empowering small to medium enterprises to form transitory structures through collaboration. They not only facilitate knowledge transfer but also resource and expertise sharing. An ideal situation is to ensure that privacy best practices can be formulated and spread through out the collaboration by the sharing of resources. For example, one member of the collaborative community is recognized as providing good privacy protection to which other members are able to benchmark against. The synergy of sharing community resources should not be limited to only business related objectives. Rather it should also encompass the knowledge of providing effective information privacy and security. Our work serves a number privacy protecting purposes. One of the main objectives is to highlight potential threats to information privacy and any advantages that may be gained from the nature of collaborative environments. Another is the proposal of a framework to address the threats to privacy in collaborative environments. We show that many of these solutions will require a unique molding of technical, legal and community (social) elements to ensure information privacy.

## 3. INFORMATION PRIVACY ISSUES IN COLLABORATIVE ENVIRONMENTS

Advances in technology are providing valuable ways for entities to share information of any nature with others [13]. With increased sharing of information in addition to escalating methods of data collection it is imperative that adequate privacy practices are in place to protect and effectively manage entity personal data. In addition to these information privacy challenges collaborative environments inherit from the information systems making up their structure they also face others that are a result of their distributed, knowledge sharing functionality. Privacy is a major concern for all members and stakeholders of collaborative environments, particularly when personal data transactions are involved. Issues relating to uncertainty and establishing trust with 'unknown' entities produces additional risks when interacting with collaborative environments. Further, the inability to clearly determine the boarders of information flows within a collaborative environment contributes to user privacy concerns and complicates personal data management [14].

Privacy protection problems escalate in collaborative environments operating across multiple countries and regions. Due to the diverse and inconsistent legislative and regulatory global privacy landscape, enforcement and protection of privacy can be difficult in multi-national collaborations. For example a fictitious collaborative environment is represented with information system infrastructure located in six different countries all subject to very different privacy laws and regulations. That is, very different models of privacy protection are followed in the European Union (EU), which favor overarching privacy legislations, as compared to the United States, which favor a self-regulation approach. So while collaborations are adept at overcoming space and time obstacles for rapid knowledge sharing they are currently very limited in managing and protecting privacy of personal information that may constitute part or all of the knowledge being shared. As stated in [15] organizations need to "… develop privacy policies and procedures that allow local privacy laws to be respected without restricting the global flow of information.".

Collaborative environments not only need to protect privacy but they must also effectively manage personal data transmitted in to, within, and out of the collaboration. The usual approach to simply restricting access to personal data is counter-productive and not suitable for collaborative environments. The primary function of collaborations is to share information not restrict it. Therefore, privacy protection in collaborative environments should be more concerned with how the data is used and ensuring an entity retains complete or significant control over their personal data. Hence, assistance in the form of tools, notifications and accessible information

should be provided to members of the collaboration to enable better management of their privacy. Allowances should also be made for the individualistic and multi-dimensional nature of privacy by providing controls that can be configured by each entity depending on the situation. This will help accommodate the diversity and often dynamic conditions that are encountered within collaborative environments and likely to influence a member's privacy perception.

## 4. A TECHNICAL, LEGAL, AND COMMUNITY – PRIVACY PROTECTING FRAMEWORK FOR THE PRIVACY AUGMENTED COLLABORATIVE ENVIRONMENT (PACE)

Research to date strongly indicates that no single model of privacy protection is sufficient to provide a complete information privacy solution [7]. Therefore, we propose that a solution to this issue is to develop systems and operating environments that integrate a symbiotic molding of all four models of privacy protection. In addition, privacy by design and information system Hippocratic principles [16, 17] should be adhered to throughout the systems life cycle. To compliment the for-mentioned factors and provide robust information privacy protection architectures, the operating contexts [18, 19] as well as social and cultural environmental conditions need to be accounted for within the framework during development, deployment and operation. Any sustainable privacy solution must make every effort to take into consideration all current and foreseeable future factors that pose a threat to information privacy. Therefore, we propose a framework entitled Technical, Legal, and Community Privacy Protection (TLC-PP). It is an approach that combines all four models of privacy protection [7], as well as consideration for the influence of social and cultural ideals and perceptions from the collaborative environment community.

The TLC-PP objective is to address the issue of information privacy that is at risk from the increasing computational capacities, distributed nature, and information sharing objectives of collaborations. The remainder of this section details each of the Technical, Legal, and Community privacy protecting components and our solutions within each component of the TLC-PP framework for collaborative environments. Due to space limitations a general outline and overview of solutions within each of the three components is provided. Readers are encouraged to read our additional related publications for more comprehensive discussion of our information privacy protecting solutions for collaborative environments.

### 4.1 Technical Privacy Protection

Technical privacy protections are frequently referred to as Privacy Enhancing Technologies (PETs). Common PETs include proxies and firewalls, anonymizers, Platform for Privacy Preferences Project (P3P), encryption tools, spam filters, cookie cutters, and automated privacy audits [20]. Since the initial demand for PETs their application and variety has increased significantly. They have come to represent more than technological support for personal data protection and now provide informational self-defense [21]. PETs now provide methods of protection for entities against many privacy invasive behaviors including unwanted surveillance and disruption. PETs in the context of our research have a broad scope, due to PETs not having a widely accepted definition, but their primary function is to minimize the exposure of private date for entities using electronic services within a collaborative environment. More generally the purpose of PETs is to protect the privacy of entities, while still enabling them to interact with other entities within a collaborative environment through digital mediums [22].

We recognize the importance of technologies of privacy and have made it one of the three critical framework components for comprehensive privacy protection. Our ongoing research has developed a number of technical solutions for enhancing entity privacy protection and personal data management. Each element is an integral part of the technical component of our TLC-PP framework. They are:

- Shield Privacy: In order to meet space requirements interested readers are directed to [11] and [23] for the complete details of shield privacy. The technical methodology consists of four privacy by design and implementation rules. The rules guide the design and implementation of information systems and collaborative environments to ensure information privacy and personal data management requirements are accommodated. The four rules are the following:
    - PDM-ADM Design and Implementation Rule: Our approach to Personal Data Minimization (PDM) and Anonymous Data Maximization (ADM). PDM is used for determining and ensuring the minimum amount of personal information required by the collaboration or information system to function. ADM is used for determining and ensuring the maximum amount of personal information can be made anonymous for use throughout the collaboration or information system.
    - SDD Design and Implementation Rule: Our approach to the Separation of Duty and Data (SDD) within the information system. SDD involves the segregation of system roles and data based on sensitivity, context of use, and entity assigned personal data access permissions for information requestors.
    - HPP Design and Implementation Rule: Hippocratic Privacy Policies (HPP) is built upon the work proposed on Hippocratic Databases [18]. Hippocratic implies taking responsibility to ensure confidentiality and integrity of personal data. When applied to information systems and collaborative environments it infers that the information systems and collaborations take responsibility for the information privacy of entities using them and the protection of personal data they manage.
    - Data Security Design and Implementation Rule: the latest data security technologies should be reviewed and continually integrated into the collaborative environment to ensure the protection of personal data at rest and in transit.
- Privacy Using Graphs (PUG): PUG is a PET for managing privacy and personal data requests. The application uses directed weighted graphs to visually represent privacy, security, trust, and contextual relationships between entities in a collaborative environment. The two primary nodes of the dynamically generated graphs represent the starting node of the Information Provider (IP) and the final node of the Information Requestor (IR). When an IP receives a personal data request from an IR the IP can use the PUG application to generate a directed weighted graph mapping the 'social' or 'association' network from them to the IR. PUG requires an initial configuration by each member entity to appoint up to three 'trusted' member entities. Using the idea of 'six degrees of separation' a social or trust network of entities can be established for the collaboration. IP's can use this network to assist in visualizing personal data requests in order to determine whether they should be granted or denied. Again due to space limitations readers are directed to [24] for full details.
- Fair Privacy Principles and Preferences (F3P): F3P is our unique contribution to privacy preference technologies. After identifying the absence of situational and compensation elements in current privacy preference technologies we addressed the problem by extending privacy preferences to include two new elements. We labeled the new elements SITUATION and REWARD. As privacy is widely accepted as being an individualistic notion meaning many different things to many different people then privacy preferences should reflect this. For an entity their perception of privacy and its worth changes with situation and possible compensation. Therefore, by allowing configuration of privacy preferences based on different situations and expected rewards they are more adept at catering for more unique individuals. Complete details of F3P are discussed in [18] and [25].

### 4.2 Legal Privacy Protection

We use the term Legal to encompass all types of legislative and regulatory privacy protection models. Multinational collaborative environments can be composed a host of different information systems governed by different privacy legislations and regulations. Ideally privacy policies and practices for a collaborative environment should be consistent for all member entities. Therefore our legal privacy protections focus on the development and production of uniform privacy laws, regulations, and policies based on best practice adoption or benchmarking. Each element is an integral part of the legal component of our TLC-PP framework. They are:

- Privacy Evaluator Module (PEM): PEM is an XML based privacy legislation, regulation, and policy comparison tool. As collaborative environments can span multiple countries they are subject to a diverse set of privacy laws and regulations. We have developed an application that is able to compare the various privacy policies, based on a standard collaboration wide XML template, to identify differences. Information system stakeholders than are members of the collaborative environment are provided with the XML template to complete and submit to PEM. The XML privacy policy template is used to represent the information privacy legislations and regulations applicable to the information system in question. The templates are also structured in such a way that 'most complete' or 'most comprehensive' privacy policy can be identified and set as the benchmark privacy policy and practices for the collaborative environment. For specific details of its operation readers are directed to our relevant publications [26] and [27]. A diagrammatic representation of the PEM functionality is shown in Figure 1.

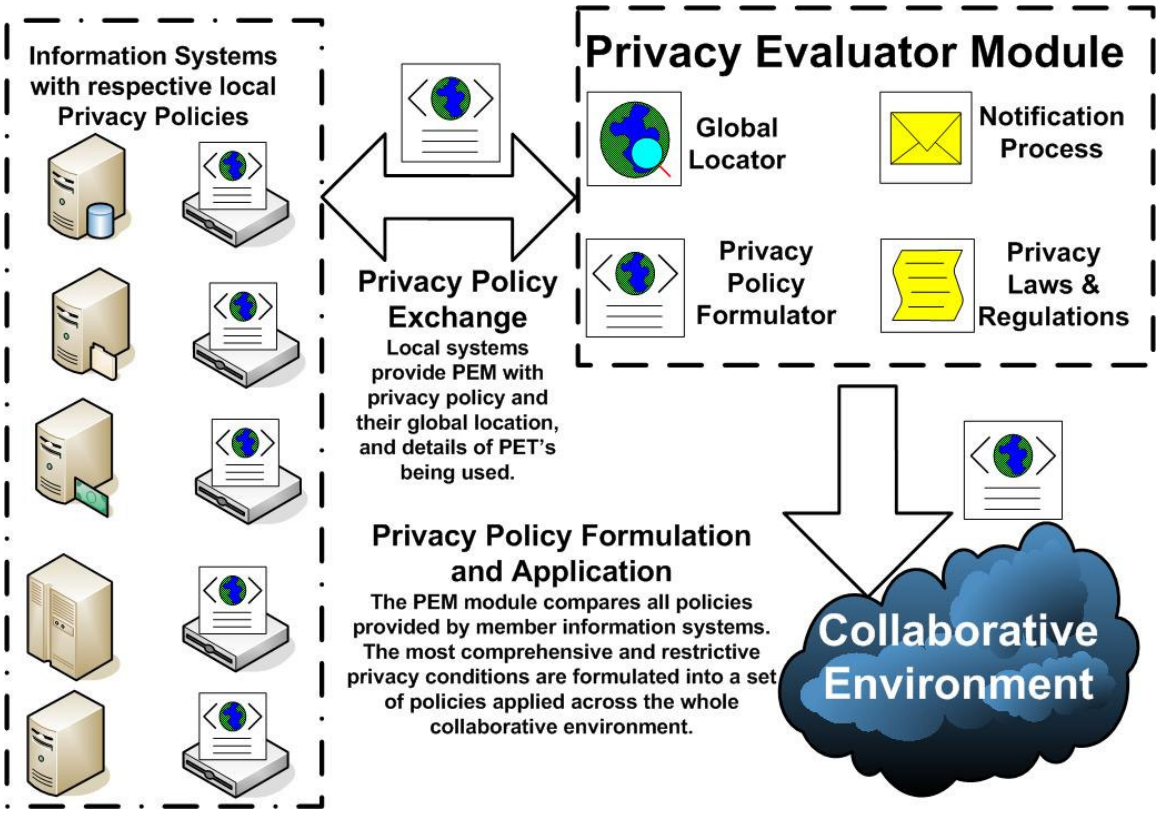



**FIGURE 1:** The Privacy Evaluator Module (PEM).

- Manual Privacy Management (MPM): Through our own experiences and those documented in the literature we have acknowledged that the legal component of information privacy protection and personal data can not be completely automated with current technologies and operating environments. Therefore, in the absence of a globally enforceable uniform set of privacy principles and practices manual enforcement and monitoring is required. As part of our MPM solution we endorse the appointment of a Privacy Officer (PO) that is tasked with legal privacy protection management. The MPM also includes a detailed list of privacy objectives and guidelines for the PO to follow in the administration of privacy across the collaboration.
- Privacy Benchmarked Policy (PBP): Through the application of PEM and practice of MPM a collaborative environment can produce a Privacy Benchmarked Policy (PBP) for

use across the collaboration. The PBP is not necessarily the representation of a single member information systems privacy policy. The PBP should encompass all of the relevant privacy legislations and regulations applicable to all entities within the collaborative environment.

### 4.3 Community Privacy Protection

The element of Community Privacy Protection is perhaps the most important model in terms of the overall success of entity privacy acknowledgment and understanding. However, it is also the element faced with the most difficult challenges and the hardest tasks to successfully implement, as it is heavily reliant of many of the same sociological influences of privacy. Due to the very nature of the Community model it is very hard to develop tangible solutions that an entity can readily implement and integrate into a collaborative environment. The general premise is that the community of member entities that constitute a digital collaborative environment must acknowledge, understand, support, and encourage good information privacy and personal data management practices and protection. We address these issues through the provision of three solutions. Each element is an integral part of the community component of our TLC-PP framework. They are:

- Privacy Awareness and Notification (PAN): PAN is a set of techniques, tools, and procedures for providing comprehensive privacy awareness and notification. Through the use of 'tools-tips', 'roll-overs', multi-layered contextual privacy policies, and privacy statements member entities of the collaboration are constantly presented with an abundance of privacy and personal data information. Additionally the PAN solution is implemented using readily available free web technologies present in most collaboration's.
- Privacy Protecting - System Development Life Cycle (PP-SDLC): The PP-SDLC is an extension to the common system development life cycle that integrates detailed privacy protection guidelines and strategies throughout each phase of the methodology. The privacy protecting and personal data management guidelines are expressed in a straightforward and easy to comprehend manner to ensure all information system stakeholders are capable of completing the necessary privacy objectives and tasks detailed in PP-SDLC.3) Community Observed Privacy (COP): COP represents policing by a collaborations stakeholders and users to instill and maintain a privacy protecting culture. Support is provided for anonymous logging of privacy violations or unsatisfactory privacy services to the Privacy Officer for follow up and action. It is a key solution in fostering an information privacy culture.

### 4.4 Privacy Augmented Collaborative Environment

The Technical, Legal, and Community Privacy Protecting (TLC-PP) framework provides collaborative environment stakeholders with a comprehensive set of privacy protecting and personal data management solutions. Integration of implementation of all of the TLC-PP components contributes to the production of a Privacy Augmented Collaborative Environment. Due to space limitations of paper length full details of our ongoing research into the development and delivery of a PACE is limited. However, as part of our future work we plan to publish more complete details of our continuing work in this area. This includes our PIVOTAL methodology with compliments the TLC-PP framework. Privacy by Integration, Visualization, Optimization, Technology, Awareness, and Legislation (PIVOTAL) provides a unique set of privacy protecting and personal data solutions to work in combination with those provided by TLC-PP.

Our TLC-PP framework, in combination with our PIVOTAL methodology, focuses on a collaborative wide effort towards privacy protection. The use of community controls is unlike traditional solutions for managing security and information privacy, where they are usually centrally managed. With our proposals entities are provided with greater control over their data security and information privacy. The individual entity has more influence over the management of their personal data. This includes whom that data can be shared with and how it can be used. Management of personal data in this manner is in conformance with world leading privacy policies such as those stipulated for use by member states of European Union. However, more

importantly and as stressed a number of times throughout our work, individual management of personal data are especially important in multi-national collaborations. This is due to the fact that the collaboration can be subject to a diverse and inconsistent set of privacy laws between countries. A summary of the three key elements of the TLC-PP framework, along with their respective three components is shown in Figure 2.

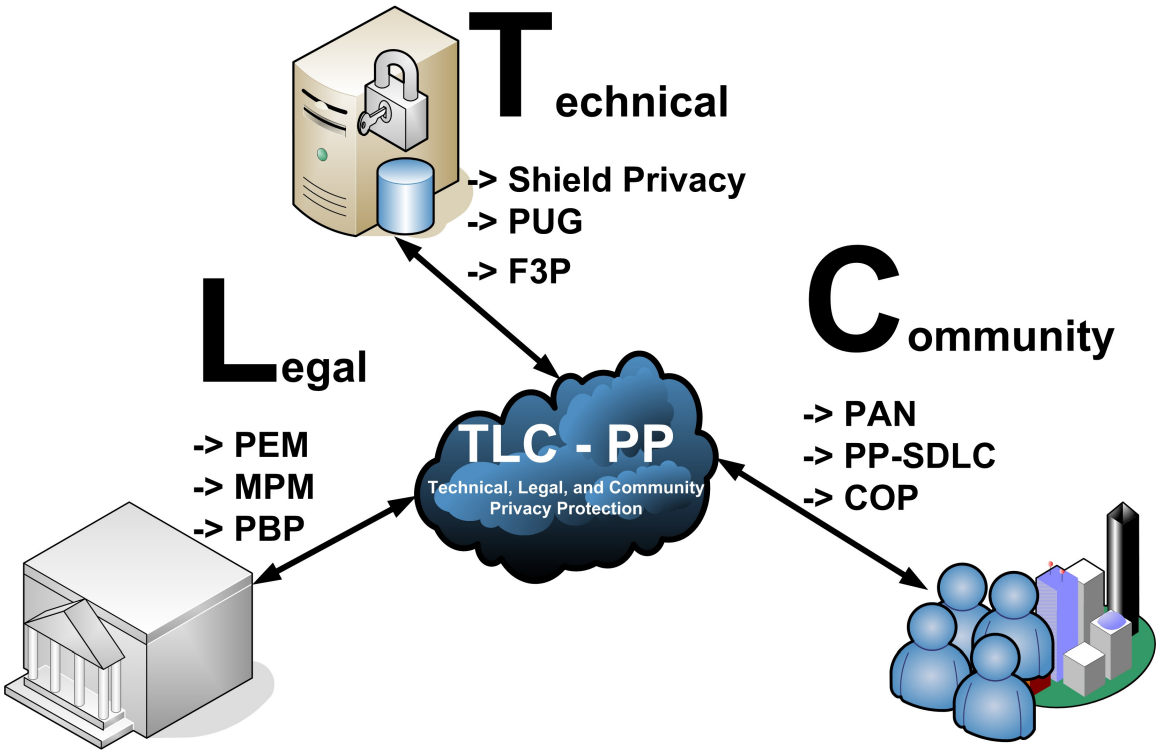



**FIGURE 2:** Representation of the Technical, Legal, and Community – Privacy Protection (TLC-PP) framework.

## 5. CONCLUSION

The Technical, Legal, and Community Privacy Protecting framework proposed in this paper provides a sustainable information privacy solution for collaborative environments. The three key components being Technical, Legal and Community models of protection each provide three unique privacy protecting and personal data management utilities for member entity use. The integration of application of the TLC-PP framework is a significant contribution towards the delivery of a Privacy Augmented Collaborative Environment (PACE). Our contribution is setting the PACE for sustaining privacy in autonomous collaborative environments.

## 6. REFERENCES

1. K. Borcea-Pfitzmann, K. Liesebach, and A. Pfitzmann, “Establishing a Privacy-Aware Collaborative eLearning Environment,” in Proceedings of the EADTU Annual Conference 2005: Towards Lisbon 2010: Collaboration for Innovative Content in Lifelong Open and Flexible Learning, Rome, November 2005.
2. J. Feigenbaum and D.J Weitzner, “Report on the 2006 TAMI/PORTIA Workshop on Privacy and Accountability,” Workshop on Privacy and Accountability, Massachusetts Institute of Technology, MA USA, June 2006.

3. I.L. Ballesteros, “New Collaborative Working Environments 2020,” Report on industry-led FP7 consultations and 3rd Report of the Experts Group on Collaboration@Work, European Commission, February 2006.
4. R. Clarke, “Privacy as a Means of Engendering Trust in Cyberspace,” June 2001, http://www.anu.edu.au/people/Roger.Clarke/DV/eTrust.html.
5. European Commission, “Technologies for Digital Ecosystems”, http://www.digital-ecosystems.org
6. R. Clarke, “Introduction to Dataveillance and Information Privacy, and Definitions of Terms”, http://www.anu.edu.au/people/Roger.Clarke/DV/Intro.html, September, 1999.
7. EPIC, “Privacy and Human Rights 2003”, Electronic Privacy Information Centre, http://www.epic.org.
8. I. Goldberg, “Privacy-enhancing technologies for the Internet II: Five years later”, PET 2002, San Francisco, 2002.
9. R.M. Davison, R. Clarke, J. Smith, D. Langford, and B. Kuo, “Information Privacy in a Globally Networked Society: Implications for IS Research”, Communications of the Association for Information Systems, Volume 12, 2003, 341-365.
10. G.W. van Blarkom, J.J. Borking, and J.G.E. Olk, “Handbook of Privacy and Privacy-Enhancing Technologies”, Privacy Incorporated Software Agent (PISA) Consortium, The Hague, 2003.
11. G. Skinner and E. Chang, “A Conceptual Framework for Information Privacy and Security in Collaborative Environments”, International Journal of Computer Science and Network Security, Vol. 6 No. 2B, Feburary 28, 2006.
12. D.A. Gritzalis, “Embedding privacy in IT applications development” Information Management and Computer Security, Vol. 12 No. 1, 2004.
13. JJ. Cadiz and A. Gupta, “Privacy Interfaces for Collaboration,” Technical Report MSR-TR-2001-82, Microsoft Corporation, 2001.
14. M/Cyclopedia of New Media, “Virtual Communities – Privacy Issues,” Creative Industries Faculty, QUT, http://wiki.media-culture,org.au/.
15. J.B. Spira, “Privacy in the collaborative business environment,” KM World, November 2004, http://www.kmworld.com/ReadArticle.aspx?ArticleID=9595.
16. G. Skinner and E. Chang, “PP-SDLC The Privacy Protecting Systems Development Life Cycle”, IPSI-2005 FRANCE, April 23 till April 26, 2005.
17. R. Agrawal, J. Kiernan, R. Srikant, Y. Xu, “Hippocratic Databases”, 28th International Conference on Very Large Databases (VLDB), Hong Kong, 2002.
18. G. Skinner and E. Chang, “Fair Privacy Principles and Preferences (F3P) – Evaluating Context Based Privacy Preferences”, The 10th WSEAS International Conference on Computers, ICCOMP-06, Vouliagmeni, Athens, Greece, July 13-15, 2006.
19. M. Ackerman, T. Darrell and D.J. Weitzner, “Privacy in Context”, Massachusetts Institute of Technology Discussion Paper, http://www.eecs.umich.edu/~ackerm/pub/01a12/context-privacy.final.pdf.
20. L.F. Cranor, “The Role of Privacy Enhancing Technologies,” Centre for Democracy and Technologies, March 2007, http://www.cdt.org/privacy/ccp/roleoftechnology1.shtml.
21. G. Danezis, “An Introduction to Privacy Enhancing Technologies,” presented at Internet Society Geneva's Monthly Conferences Cycle, Geneva, Switzerland, July 2004.
22. Meta Group Report, “Privacy Enhancing Technologies,” Ministry of Science and Technology, Denmark, March 2005.
23. G. Skinner, S. Han, S. and E. Chang, E., “Shield Privacy: A conceptual framework for Information Privacy and Data Access Controls”, WSEAS Transactions on Computers, Issue 6, vol. 5, June 2006, pp. 1375-1381.
24. G. Skinner and M. Miller, “Managing Privacy, Trust, Security, and Context Relationships Using Weighted Graph Representations”, WSEAS International Journal of Information Science and Applications, Issue 2, vol. 3, February 2005, pp. 283-290.
25. G. Skinner, S. Han, and E. Chang, “Integration of Situational and Reward Elements for Fair Privacy Principles and Preferences (F3P)”, in proceedings of IEEE International Conference on Industrial Technology (ICIT2006), Mumbai, India, December, 2006.

26. G. Skinner and E. Chang, “An Environmentally Adaptive Conceptual Framework for Addressing Information Privacy Issues in Digital Ecosystems”, in proceedings of 2007 IEEE International Conference on Digital Ecosystems and Technologies, Cairns, Australia, February, 2007.
27. G. Skinner and E. Chang, “Information Privacy Concerns in Quantum Computational Distributed Environments,” International Journal of Information Security and Privacy, 1(3), July-September 2007, pp. 1-12.

# MODIFIED APPROACH FOR SECURING REAL TIME APPLICATION ON CLUSTERS

**Abhishek Songra** sw0512@mnnit.ac.in
Computer Science & Engineering Department
Motilal Nehru National Institute of technology, Allahabad, UP, India
**Rama Shankar Yadav** rsy@mnnit.ac.in
Computer Science & Engineering Department
Motilal Nehru National Institute of technology, Allahabad, UP, India
**Sarsij Tripathi** cs0620@mnnit.ac.in
Computer Science & Engineering Department
Motilal Nehru National Institute of technology, Allahabad, UP, India

**Abstract**

In today arena security critical real time applications running over clusters are growing very rapidly. As an application running on clusters demand both timeliness and security thus, an efficient scheduling algorithm is needed that have better performance in terms of both number of task accepted and security value received. This paper modifies the security aware scheduling approach [5] by utilizing the concept of task criticality and adaptive threshold value. Also, this paper discus the system architecture used, mathematical model, lemmas and modified scheduling approach. Further, simulation studies have been carried out in MATLAB (module for Real-time) to measure the performance of modified approach. The modified approach is applicable over wide range of application differing in there requirement and have better performance.

**Keywords:** Real time System, Scheduling, Security Services, Clusters

## 1. Introduction

A Real-time system is a system in which computations must satisfy stringent timing constraints besides providing logically correct results i.e. a correct computation of the result must finish before its specified deadline is met. Failure to meet the specified deadline in such system leads to catastrophic loss in case of hard real time systems whereas degraded performance is observed in soft real time application.

Many real time applications are using clusters for satisfying the need of high computing power where nodes are inter connected through high speed network. A real time applications using clusters faces security threats for example in stock quote update and trading system, incoming requests coming from different business partner while outgoing response from an enterprise back-end machine these application composed of clusters that has to satisfy both timeliness of response and security requirements [13]. As cluster executes vast number of unverified application submitted by vast number of different type of users both applications and users can be source of security threats to cluster [20]. These applications are vulnerable to attacks such as: attack by malicious user, malicious application running on clusters itself. The malicious users intercept applications running and launch denial of service whereas blocking of resources is observed in the case of malicious applications. The security threats to these applications are primarily related to the authentication, integrity, and confidentiality of application. An attacker may breach the above security service by spoofing, snooping and alteration kind of attack. These attacks are briefly defined below.

**Spoofing attack** is a situation in which one person or program successfully masquerades as another by falsifying data and thereby gaining an illegitimate advantage.

**Snooping attack** is not necessarily limited to gaining access to data during its transmission. Hacker may gain access to data while it is in transmission but can also gain access while the data is in not in transmission.

**Alteration** is a kind of attack in which a malicious user, which may be inside the cluster or outside the cluster, after gaining access to data performs unauthorized changes to it.

Application having real time constraints running over clusters requires secure computation. These applications have to satisfy both timeliness and security issues. Also, applications require preference of one security service to another one and different security services require overhead. Thus, an efficient scheduling algorithm is needed that achieves high performance in term of completing more number of computations while maintaining higher security level.

Rest of the paper is organized as follows. Section 2 deals with related work whereas system model along with modified scheduling approach are discussed in section 3. Section 4 includes simulation and result while paper is being concluded in section 5.

## 2. Related Work

Here, first we discuss related work done in the area of real time scheduling, followed by cluster based security issues and then proposed solution for problem. Extensive work has been done in the field of real time task scheduling whereas few work is reported on scheduling of real time tasks with security constraints. Based on the time of when scheduling decision, is taken scheduling algorithms are categorized as Offline (static) and Online (dynamic). In offline scheduling is performed well before system starts functioning however, scheduling decision are taken at run time in case of online. Authors in [8] have proposed an algorithm which schedules the task on uniprocessor systems whereas scheduling algorithm for multiprocessor system is given in [9] [11].

In [10] a non preemptive static scheduling algorithm is used whereas dynamic scheduling algorithm for multi-processor system is given in [11]. These algorithms did well for the real time systems but they fails to satisfy security constraints required for real time cluster based system.

T. Sterling and D. Savarese [14] used static scheduling on the clusters whereas dynamic scheduling approach is employed in [15].These works are focused for scheduling non real time tasks with security constraints on the multiprocessor systems and fail to satisfy the real time task requirement. Thus, Scheduling Real time task with security on clusters has become open area of research and few studies has been made in this area. Manhee Lee et. al. has discussed the security issues related with clusters [17] whereas grid computing discussed in [18].

Xie et. al. [5] has used a security aware scheduling strategy for real time applications on clusters to satisfy minimum security requirement. Scheduling decisions are taken based on earliest deadline first (EDF) [5]. Scheduling decisions are taken at two phase: first that satisfy the minimum security requirement while improvement in security is received in second phase. Authors [5] uses improvement in second phase on the basis of the arrival time, i.e., a job arrived later have lesser chance for improvement as compared to arrived earlier. The improvement on the basis of arrival time may lead to a situation that already feasible task in phase 1may rejected. This could be understood by an example given below.

Consider a task having attribute ($a_i, e_i, f_i, d_i, \mathcal{E}_i, S_i, \mathcal{L}_i$) where $a_i, e_i$, $f_i$, $d_i$, $\mathcal{E}_i$, $S_i$, $\mathcal{L}_i$ are the arrival time, execution time, finish time, deadline, amount of data to be secured, security level requirement and the criticality of a task respectively. Also, a task $\mathcal{T}_i$ requires q security services which are represented by set of security level ranges, e.g., $S_i = (S_i^1, S_i^2, \ldots\ldots, S_i^q)$ where $S_i^j$ is security level range for $j^{th}$ security service. The security criticality of a task is the cumulative security requirement of a task. A task is said to be more security critical if its security requirement is more than threshold value. Detailed security criticality will be explained in section 3.1.5. Consider set of two tasks ($\mathcal{T}_1, \mathcal{T}_2$) having attributes value as below.

Tasks requiring three security services (authentication, confidentiality and integrity) along with security level range, given in square brackets. For task $\mathcal{T}_1$ minimum authentication security level required is 0.2 and this is compared with security level and corresponding overhead given in Table 5. Incase security requirement does not directly match with table value next higher security level is being considered. For this authentication requirement (0.2) is not matched with the value given in the table so, next higher value (0.55) is considered and corresponding overhead is computed as authentication overhead as 90. Similarly minimum confidentiality (0.3) is selected from Table 3 as 0.36 with overhead is 5.33ms (200/37.5). Table 4 is used to determine integrity overhead. Similarly we can determine overheads of the three services for task $\mathcal{T}_1$. Finish time of a task is the sum of security overhead, execution time of $\mathcal{T}_1$ and waiting time due to higher priority task. The values are summarized in table 1(a) below.

**Table 1(a): Feasibility of task set after phase one**

| Task | Authentication overhead (Min) | Confidentiality overhead (Min) | Integrity overhead (Min) | Finish time ($e_i$+Overhead+ $w_i$ ) | Deadline |
|---|---|---|---|---|---|

| | | | | | |
|---|---|---|---|---|---|
| $\mathcal{T}_1$ | 90 | 5.33 | 8.368 | 107.701 | 150 |
| $\mathcal{T}_2$ | 90 | 4 | 12.5 | 216.201 | 222 |

It is clear from the table that both task are feasible with minimum securities after phase 1.In second phase author[5] consider task $\mathcal{T}_1$ for improvement as its arrival time is earlier than T2 .Finish time of T1 after improvement in services (authentication, confidentiality and integrity are 0.5,0.5 and 0.4 respectively) is 124.033 ms However ,finish time of $\mathcal{T}_2$ become 232.533 ms which is more than its deadline leading to rejection of $\mathcal{T}_2$. That is either both tasks are forced to run with minimum security or $\mathcal{T}_2$ will be rejected shown in table 1(b).

**Table 1(b): Feasibility of task set after phase two with existing approach**

| Task | Authentication overhead (security value) | Confidentiality overhead (security value) | Integrity overhead (security value) | Finish time ($e_i$+Overhead+ $w_i$ ) | Deadline |
|---|---|---|---|---|---|
| $\mathcal{T}_1$ | 90 (0.5) | 9.483 (0.5) | 20.55 (0.4) | 124.033 | 150 |
| $\mathcal{T}_2$ | 90(0.3) | 4(0.2) | 12.5(0.3) | 232.533 | 222 |

In this paper we modify criteria for selecting candidate task for the security improvement phase by using the concept of task criticality other than its arrival time. For purpose of adaptation between improvement in security and reduction in rejection of task, a threshold is considered. The value of threshold is determined dynamically, i.e., incase rejection is more the higher threshold value is taken; improvement in security is less consequently rejection ratio may be reduced. The next section deals with system model followed by modified approach.

## 3. System Model

This paper uses on line scheduling approach which is targeted for real time applications having security requirements on clusters. Cluster is a group of N nodes {$N_1$, $N2$, $N3$ ….$N_n$) connected through a high speed network where real time application having high computational and security requirements are submitted. These applications due to their high computational demands are incapable of executing on a single node; hence they are partitioned into sub application or tasks. For simplicity we presume that the tasks incorporated in an application are independent of each other. Real time application is accepted if and only if the cluster can schedule the task so that they complete within their respective deadline and ensures for at least minimum security requirement (related to application) in phase 1. Improvement over minimum security guarantee may be achieved through utilization of available slack in schedule. We consider a task set having n tasks, T = { $\mathcal{T}_1, \mathcal{T}_2$ …… $\mathcal{T}_n$}. Each task $\mathcal{T}_i$ is described with the attribute ($a_i, e_i, f_i, d_i, \mathcal{E}_i, S_i, \mathcal{L}_i$) where $a_i$ is the arrival time, $e_i$ is the execution time, $f_i$ is the finish time $d_i$ is the deadline, $\mathcal{E}_i$ is the amount of data to be secured, $S_i$ is security level requirement, $\mathcal{L}_i$is the criticality of a task. Suppose a task $\mathcal{T}_i$ requires q security services which are represented by the security level ranges e.g. $S_i = (S_i^1, S_i^2, \ldots\ldots, S_i^q)$. The parameter and assumptions are same as used in [5].

Before we proceed for modified scheduling algorithm in detail, we first discuss the various terms used in this paper. These terms are summarized in Table 2.

**Table 2: Terms and Description**

| Term | Description |
|---|---|
| *m* | Number of nodes in the cluster. The nodes may be or may not be identical. |
| *R* | Number of users submitting tasks to the cluster. A user can submit any task at any point of time. |
| $e_i$ | Execution time of a task $\mathcal{T}_i$ . |
| $a_i$ | Arrival time of the task $\mathcal{T}_i$. |
| $d_i$ | Deadline of task $\mathcal{T}_i$. It is the time beyond which the utility of the result of the task degrades. |
| $f_i$ | $f_i$ is the allowable finish time of the task $\mathcal{T}_i$ by which the utility of the result is within acceptable quality of service. |
| $\mathcal{L}_i$ | Criticality of the task $\mathcal{T}_i$. |

| | |
|---|---|
| $sl_i$ | Security level value assigned to a security algorithm based on its performance. |
| $S_i$ | Security level of task $T_i$ |
| $£_i$ | Amount of data that is to be secured. |
| $Th$ | Criticality Threshold of the cluster. |
| *rej_ratio* | Rejection ratio. |
| *Min(rej_ratio)* | Minimum rejection ratio. It is a measure of quality of service of the cluster that must be maintained. |
| *Max(rej_ratio)* | Gives the extreme limit of tasks rejection in percentage. |

As snooping, alteration and spoofing are three common attacks on cluster that can be handled by security services such as Authentication, Integrity and Confidentiality. These services incurred computational overhead, which depends upon amount of data secured used for securing these attacks. The following sub section describes detail about these services along with mathematical model for computation of overhead as used in [5].

### 3.1 Security Overhead Model

This paper focused on deploying security services (authentication, integrity and confidentiality) to secure cluster based real- time application against the basic attacks (spoofing, snooping and alteration). Snooping, an unauthorized interception of information can be tackled by confidentiality service whereas authentication service is deployed for spoofing. The alteration is unauthorized modification to information; this can be taken care by integrity services. Different applications require different type of integration of these security services for example; one may weight these services of equal importance whereas other may weight one service over another one. Thus, different combination of these services leads to complex integration of these services. The security aware scheduler running over complex integration has to adapt security overhead experience by a task in order to achieve desired quality of services (QoS) may be measured as number of tasks accepted, cumulative security level etc. Similar type of consideration is used in [5]. The security services are independent of one another. . The user can select different security services from the available services to form a complex integrated security solution. The following paragraph discusses detailed mathemetical model for confidentiality followed by integrity and then authentication.

#### 3.1.1 Confidentiality Overhead:

Confidentiality is achieved by encrypting & decrypting both real time application as well as data to receive safegaurd from malicious user. We consider eight standard encryption algorithms to calculate confidentiality overhead which is shown in Table 3 where each security algorithm is assigned a security level in the range of 0.08 to 1 on the basis of its security performance. Beside these security algorithms (given in table) security of other algorithm security overhead is calculated with the use of equation 1.

$$sl_i^c = \frac{13.5}{v_i^c}, 1 \leq i \leq 8 \qquad (1)$$

where $v_i^c$ is performance of the $i^{th}$ ($1 \leq i \leq 8$) standard encryption algorithm and $sl_i^c$ is the confidentiality security level of task $T_i$.

The security level of a algorithm is inversely proportional to algorithm's performance.

$$sl_i^c \propto 1/v_i^c$$

In case required confidential security level of of task $T_i$ is $S_i^c$, the overhead for this service can be computed by the use of equation 2 where $£_i$ is the amount of data (in terms of Bytes/KB/MB) which is to be secured & $\sigma^c(S_i^c)$ is a function used for mapping a security level to its corresponding encryption algorithm's performance.

$$c_i^c(S_i^c) = \frac{£_i}{\sigma^c(s_i^c)}, 1 \leq i \leq 8 \quad (2)$$

**Table 3: Cryptographic Algorithms for Confidentiality Service**

| *Cryptographic Algorithms* | *$SL_i^c$: SL Security level* | *$\nu_i^c$:KB/ms* |
|---|---|---|
| **Seal** | 0.08 | 168.75 |
| **RC4** | 0.14 | 96.43 |
| **Blowfish** | 0.36 | 37.5 |
| **Knufu/Khafre** | 0.40 | 33.75 |
| **RC5** | 0.46 | 29.35 |
| **Rijndael** | 0.64 | 21.09 |
| **DES** | 0.90 | 15 |
| **IDEA** | 1.00 | 13.5 |

### 3.1.2 Integrity Overhead:

Integrity security service is used to guard data against unauthorized modification or tampering while task is executing. We consider that seven integrity algorithms are deployed for providing integrity service and these consideration are same as considered in [5]. Integrity is achieved by implementing hash function [24] where each function is assigned a security level in accordance with its performance. The hash functions are shown in Table 4 along with their respective performance & security level.The security level for other hash function except shown in table, can be computed from equation 3.

$$sl_i^g = \frac{4.36}{\nu_i^g}, 1 \leq i \leq 7 \quad (3)$$

Where $\nu_i^g$ is the performance of the $i^{th}$ ( $1 \leq i \leq 7$) hash function.

**Table 4: Hash Function for Integrity Service**

| Hash Function | $SL_i^g$: Security level | $\nu_i^c$:KB/ms |
|---|---|---|
| **MD4** | 0.18 | 23.90 |
| **MD5** | 0.26 | 17.09 |
| **RIPEMD** | 0.36 | 12.00 |
| **RIPEMD-128** | 0.45 | 9.73 |
| **SHA-1** | 0.63 | 6.88 |
| **RIPEMD-160** | 0.77 | 5.69 |
| **Tiger** | 1.00 | 4.36 |

Let $S_i^g$ is the security level of integrity service for task $\mathcal{T}_i$, the overhead due to integrity service can be computed using equation 4.

$$c_i^g(S_i^g) = \frac{£_i}{\sigma^g(s_i^g)}, 1 \leq i \leq 7 \quad (4)$$

where $£_i$ is the amount of data whose integrity is to be assured and $\sigma^g(S_i^g)$ is a function used for mapping a security level to its corresponding hash function's performance.

### 3.1.3 Authentication Overhead:

Authentication is used to tackel spoofing attack. The authentication service insured that all task must be submitted by authorized users. Three authentication methods are used in paper which is shown in Table 5 where each authentication method is assigned a security level value. Security level of a required authentication method (other than given in table 4) can be calculated using equation 5.

$$sl_i^a = \frac{\nu_i^a}{163}, 1 \leq i \leq 3 \quad (5)$$

where $v_i^a$ is the performance of i$^{th}$ $(1 \leq i \leq 3)$ authentication method .
Authentication overhead $c_i^a(S_i^a)$ of task $T_i$ is a function of $T_i$s security level $S_i^a$.

**Table 5: Authentication Methods for authentication service**

| Authentication Methods | $SL_i^a$: Security Level | $v_i^c$ Computation Time(ms) |
|---|---|---|
| **HMAC-MD5** | 0.55 | 90 |
| **HMAC-SHA-1** | 0.91 | 148 |
| **CBC-MAC-AES** | 1 | 163 |

### 3.1.4 Security Overhead Model:

The overall security overhead for task $T_i$ which is the sum of overhead incurred by each of the three security services imployed in forming the integrated security solution , can be computed using equation 6. Consider a task $T_i$ requires w security services in sequential order and $s_i^k$ and $c_i^k$ be the security level & security overhead of the k$^{th}$ security service applied on the task respectively. The overall security overhead of task can be calculated using equation 6.

$$c_i = \sum_{j=1}^{t} c_i^j(s_i^j), where\ s_i^j \in S_i^j \quad (6)$$

### 3.1.5 Security Criticality

The term security criticality is extracted from security services ranges for a given task and it is cumulative security requirement of task for different security services. For example already considered in section 2 the security criticality of task $T_i$ is the average of lowest limit of the range for three security services ,i.e., security criticality of task $T_1$ ($\mathcal{L}_1$) is (0.2+0.3+0.1)/3 = 0.2 and $\mathcal{L}_2$ for $T_2$ is 0.2667 . Thus $T_2$ is more security critical than $T_1$.

### 3.2 System Architecture Used

System architecture used in this paper consist of 'm' identical nodes connected through a high speed network, where real time task submitted by the 'r' number of users is shown in Figure 1. The schedule queue maintained by admission controller is a buffer used to hold newly arrived task without any consideration. The task submitted by the user is dispatched to the accepted queue if it pass acceptance test. A task is said to be pass acceptance test if task is able to complete in its deadline with minimum security requirement. This acceptance test is the responsibility of admission controller. A task fail to pass the acceptance test is said to be rejected and such task are places to the rejected queue. In contrast to acceptance test performed by admission controller (where acceptance test of individual task is taken into account) real time scheduler performed feasibility analysis of newly accepted task along with other task waiting for service or partially executed. A task passes feasibility analysis join dispatch queue where security enhancement is achieved (phase 2). A task fail to satisfy feasibility test join rejected queue and accepted task is dispatched to local queue of nodes in cluster. Similar type of system architecture is used in [5].

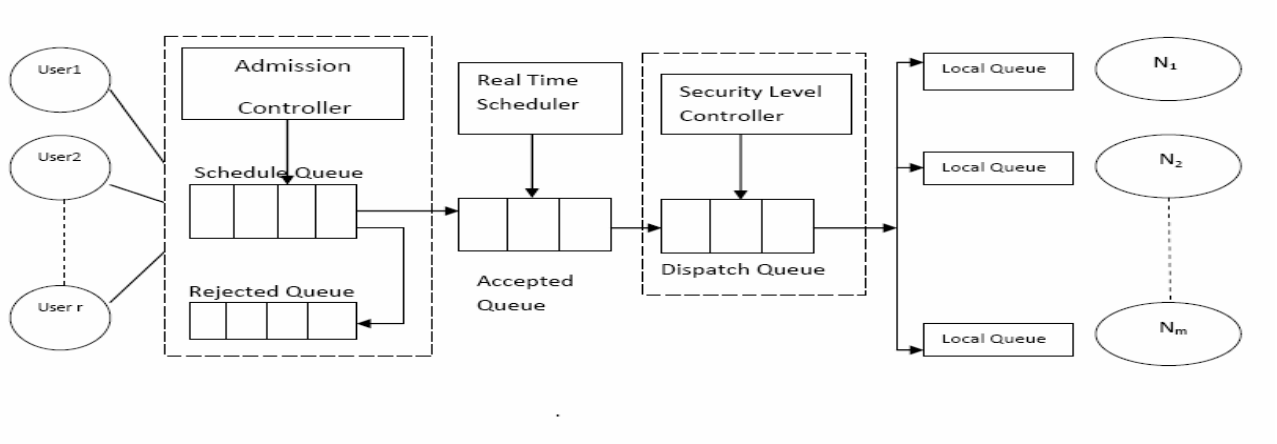



**Fig.1 System Architecture Used**

An application submitted to the cluster has the following property.

**Prpoerty1** This paper considers hard real time application submitted to the cluster .The application is composed of 'n' independent tasks requesting different level of security. An application is said to be accepted if and only if all tasks are feasible. Each node estimates the wait time $w_i^j$ of $\mathcal{T}_i$ on node $N_j$, will be the sum of remaining time of the executing task interrupted and execution time of all the tasks of higher priority, thus, $w_i^j = \text{remaining\_time}_h + \sum_{h \in H} e_h + \text{cost}(\min\_S_i)_i^j$ where H refers to the set of higher priority tasks (having deadline earlier to that of task $\mathcal{T}_i$).

After estimation of waiting time on a node, cost of the minimum security level feasibility analysis have been performed to obtain a valid schedule. A valid schedule can be stated by the following lemma used in [5].

**Lemma 1** A valid schedule is the one in which the incoming task can be scheduled on at least one node on the cluster such that it can be granted minimum security guarantee without missing its own deadline nor forcing any previously accepted task to miss its respective deadline. Mathematically it is written as,

$$\exists N_j \in N \text{ such that } w_i^j + e_i + \text{cost}(\min\_S_i)_i^j \leq d_i \quad \text{(i)}$$

$$\forall \mathcal{T}_k \in \text{local queue to } N_j \text{ and having lower priority than arriving task } \mathcal{T}_i: w_k^j + e_i + \text{cost}(\min\_S_i)_i^j \leq f_k \quad \text{(ii)}$$

Where $e_i$, $w_i^j$ are the worst case execution time, estimated wait time of the task $\mathcal{T}_i$ on node $N_j$ respectively. The $f_k$ is the allowable finish time of the task $\mathcal{T}_k$ by such that utility of the result is within acceptable quality of service.

**Proof:** If a task misses its own deadline then the utility of the result is lost. If it forces any previously accepted task to miss its deadline then an entire application will fail as refer property 1. In case, a task is accepted its security guarantee is improved in the best effort way if and only if the criticality of the task is more than the threshold of the cluster. This threshold is dynamically adjusted to maintain a desired QoS (rejection ratio not more than the value allowed for it) on the cluster i.e. to provide lower rejection ratio by allowing more tasks to be accepted by increasing the threshold. This can be stated as the following lemma.

***Property 2:*** The estimated waiting time of a task $\mathcal{T}_i$ is given as $w_i^j = w_i^j + e_h + \text{cost}(S_h)_h^j$ where $e_h$ and $\text{cost}(S_h)_h^j$ are execution time and overhead of security on node j respectively of task $\mathcal{T}_h$ (such that $d_h < d_i$) and its arrival time is $a_h$ ,i.e., the task $\mathcal{T}_i$ may have to wait more than its estimated time because of the arrival, of a higher priority task before it can be scheduled. The estimated wait time of $w_i^j$ can be given as $\sum_{h \in H} \text{actual\_finish\_time}_h^j \leq w_i^j \leq \sum_{h \in H} \text{estimated\_finish\_time}_h^j$ where H refers to set of higher priority tasks (having deadline earlier than the task $\mathcal{T}_i$ ), $\text{actual\_finish\_time}_h^j$ refers to the exact time by which the higher priority

task actually completes (by taking execution time between best and worst case), the $estimated\_finish\_time_{ij}$ refers to the execution time expected to be taken by the task in worst case.

**Lemma 2:** Threshold of a cluster is proportional to the rejection ratio on the cluster.

**Proof:** The value of the threshold of the cluster can lie between 0 and 1. If the value of threshold is equal to zero it indicates that all tasks will be improved in the best effort way at the time of acceptance, hence each task will demand maximum security overheads and will take at higher computation time. In such case, less number of tasks can be accepted. If the value of threshold is one then all tasks will be accepted with minimum security overhead and will be improved later at the time of their execution (if slack for improvement exists), hence more number of tasks can be accepted hence, lowering the rejection ratio. Thus threshold is directly proportional to rejection ratio.

### 3.3 Modified Security Aware Scheduling Approach (MSASA)

In [5] authors have used improvement in the security of a task on the basis of first come first service and reject a tasks whose minimum security requirement is not satisfied. As a result the scheme faces higher rejection ratio and lesser improvement in security too. In this paper beside given preference on the first come first service basis we schedule task with earliest deadline first to satisfy minimum security requirement. However, in improvement phase preference is given to more critical task (measured in terms of security requirement)
The security benefit received by a task is measured using security level function is given by equation 7.

$$SL(s_i) = \sum_{j=1}^{q} w_i^j s_i^j, 0 \leq w_i^j \leq 1, \sum_{j=1}^{q} w_i^j = 1 \quad (7)$$

where $X_i$ denotes all possible schedules for task $\mathcal{T}_i$ and $x_i \in X_i$ is a scheduling decision for $\mathcal{T}_i$ . For a given a real time task $\mathcal{T}_i$, the security benefit is maximized by security level controller using the following security benefits (SB), security value (SV) constraints as given below:

$$SB(X_i) = \max_{x_i \in X_i} \left\{ \sum_{j=1}^{q} w_i^j s_i^j(x_i) \right\} \quad (8)$$

The security level of task is increased up to a level at which task completes with in its deadline and does not make any previously accepted tasks to miss their deadline. The following security value function needs to be maximized under certain timing and security constraints:

$$SV(X) = \max_{x_i \in X_i} \{\sum_{i=1}^{p} y_i SB(x_i)\} \quad (9)$$

Where, p is the number of submitted tasks, $y_i$ is set 1 if the task is accepted and is set to 0 otherwise. Our aim is to schedule tasks, while maintaining the guarantee ratio, in a way to maximize equation 10.

$$SV(X) = \max_{x \in X} \{\sum_{i=1}^{p} (y_i \max_{x_i \in X_i} \{\sum_{j=1}^{q} w_i^j s_i^j\})\} \quad (10)$$

After the possible improvement in the task's security level it is dispatched to node accepting it and promising the best security level or minimum wait time (if less critical).The modified security aware scheduling algorithm is given below.

**Improve_security ()**

Arrange security services according to their weights
  For each security services do
    Calculate overhead for $S_i^k$ for kth security service $C_i^k$
    EFTij= wij + $e_i$ + $C_i^k$
    If ( EFTij > Di)
      Decrease $S_i^k$ break

Increase $S_i^*$

Continue till security level of all security services is not maximized

**MSASA Algorithm ()**

//Input: Task to be scheduled with their security requirements

//Output: Tasks are scheduled on nodes.

For every task $T_i$ arriving into schedule queue.

For every node $N_j$ do

Calculate wait time of $T_i$ on $N_j$ is $w_i^j$

Calculate cost of $T_i$ on $N_j$ is $c_i^j$ (min (SL))

Estimated finish time $EFT_i^j = w_i^j + e_i + c_i^j$ (min (SL))

If ( $EFT_i^j < d_i$ )

$Accept_i^j$=1 on the node $N_j$

Else

$Accept_i^j$=0 on node $N_j$

If ($accept_i^j$ ==1 && criticality of $T_i$ > threshold Th)

Call improve_security ()

If task is accepted on any node then

Increase accepted task

Select the best node for scheduling $T_i$ ( let it be $N_k$)

If (my_id==k)

Insert the task $T_i$ in local queue based on EDF

Delete the task $T_i$ from the arrive queue

Else

Increase rejected_task

Rejection_ratio = rejected_task / (rejected_task+accepted_task)

if ( rejection_ratio > MAX(rejection_ratio))

Increase threshold Th

Else

Decrease threshold Th

Continue with next task if any

Let us consider task $T_1$ and $T_2$ used in the section 2.Now we will examine the effect of the modified approach on these two tasks. As we know task $T_2$ is more security critical than $T_1$ and in section 2 from table 1(a) it is clear that both task are schedulable with minimum security requirements whereas from table 1(b) it is evident that improving security of task $T_1$ causes task $T_2$ to miss its deadline. By our modified approach the task $T_1$ is accepted at minimum security requirement and security improvement is done in task $T_2$. These results are shown in the table 5.

**Table 5: Feasibility analysis after improvement phase with modified approach**

| Task | Authentication overhead(security value) | Confidentiality overhead(security value) | Integrity overhead(security value) | Finish time ($e_i$+Overhead+ $w_i$ ) | Deadline |
|---|---|---|---|---|---|
| $T_1$ | 90(0.2) | 5.33(0.3) | 8.368(0.1) | 107.701 | 150 |
| $T_2$ | 90(0.55) | 5.11(0.46) | 15.416(0.45) | 220.227 | 222 |

**4. Performance measurement and discussion:** The performance of modified security aware scheduling approach (MSASA) is measured through simulation in MATLAB environment using scheduling tool. The simulation

parameters used in this paper is same as used in [5] and are summarized in table 6. The performance of MSASA is compared with that of security aware scheduling approach (SASA) [5]. The key parameters are guarantee ratio (ratio of number of tasks accepted over total number of tasks arrived in the system) and security value received (sum of achieved security for the entire accepted task).

**Table 6: Simulation Parameters**

| Parameter | Value (Fixed)-(Varied) |
|---|---|
| β (Deadline base, or Tbase) | (0ms) - (10,50,100……800)ms |
| Execution time $e_i$ | Uniform random number [5, 20]. |
| Required Security Service | (Mixed)- (confidentiality only, Integrity Only, Authentication Only) |
| Weight of Authentication | (0.2)- (0.1,0.3) |
| Weight of Confidentiality | (0.5)- (0.1,0.2……0.8) |
| Weight of Integrity | (0.3)- (0.1,0.2……0.8) |
| Threshold | (0.5)- (0.1,0.2…….1) |

Generation of task set:
Task has Poisson distribution arrival pattern with execution time uniformly generated. The range of security services are chosen by selective uniform random number between 0.1 to 1.0.

We used the following equation to generate $\mathcal{T}_i$ *s* deadline $d_i$.

$$d_i = a_i + e_i + cost_i^{max} + \beta \quad \textbf{(11)}$$

Where, $a_i$= arrival time of task, $e_i$= execution time of task and $cost_i^{max}$ is maximal security overhead which is computed as follows:

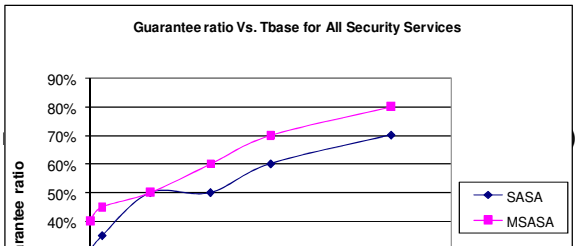
$$cost_i^{max} = \sum_{j \in \{a,c,g\}} cost_i^j \left(\max\{S_i^j\}\right) \quad \textbf{(12)}$$

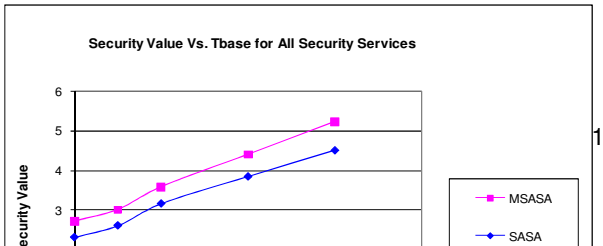
Where, $cost_i^j \left(\max\{S_i^j\}\right)$ represents the overhead of the $j^{th}$ security service for $\mathcal{T}_i$ when the corresponding maximal requirement is satisfied.

### 4.1 Results and discussion:

Simulations results are obtained in variety of applications requesting different type of services with different security levels. In following section we first discuss effect of Tbase for application where all there security requirements are needed followed by application requesting only special kind of security.

**Effect of Tbase for all security requirements:** Figure 2 (a) and 2 (b) shows performance of modified security aware scheduling approach for the case where authentication, integrity and confidentiality services are required. Figure 2 (a) measured the performance in term of guarantee ratio whereas security value is measured in Figure 2(b) .It is observed that with increment in Tbase both guarantee ratio and security value increases but this increment in performance is more in MSASA as compared to SASA. This is because increment in Tbase deadline of a task relaxed giving better performance in both cases. However , in improvement in rejection ratio by the use of threshold we decrease the number of task whose security is improved and accept more number of tasks with minimum security this gives better performance in both terms as compared to that received incase of existing one.



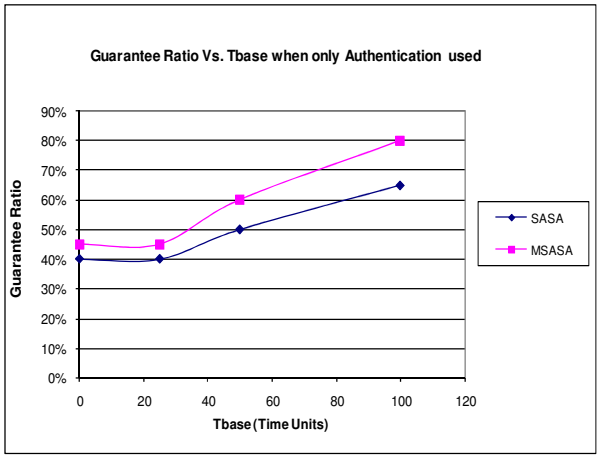



.

**Fig 2(a): Effect of Tbase.**

**Fig 2(b): Effect of Tbase**

**Effect of Tbase for authentication service only:** Performance of MSASA is shown in Figure 3(a) and 3(b) for the case where application request for authentication services. It is observed that guarantee ratio of modified approach increases with increment in Tbase value. However, security value received is almost same both of the approach.



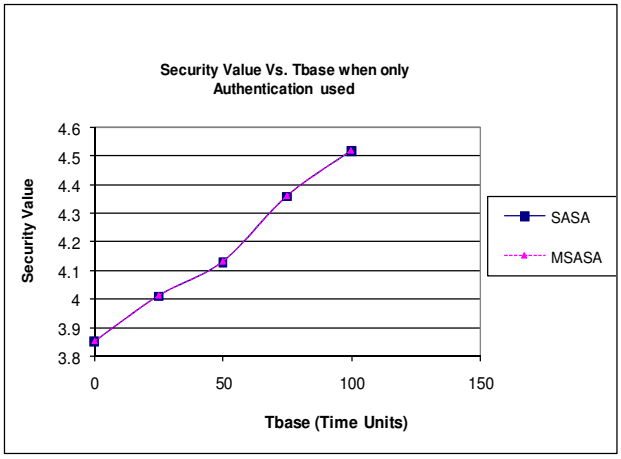
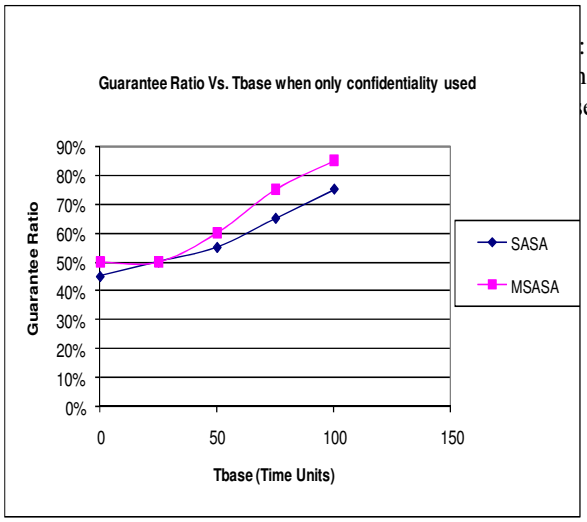




**Fig3 (a): Impact of the Authentication service**

**Fig 3(b): Impact of the Authentication Service**

: Figu
hat mod
erved r



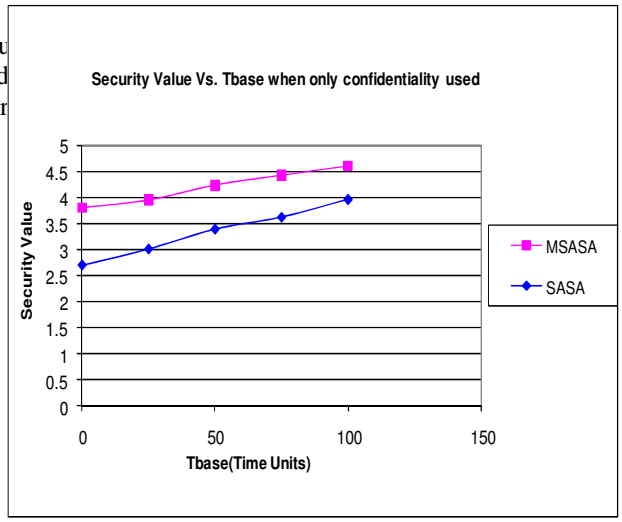




**Fig 4(a): Impact of the Confidentiality Service**

**Fig4 (b): Impact of the confidentiality Service.**

**Effect of Tbase for integrity services only:** The impact of integrity service is shown in figure 5(a) and 5(b). Similar type of trained is obtained as observed in the case confidentiality service only.

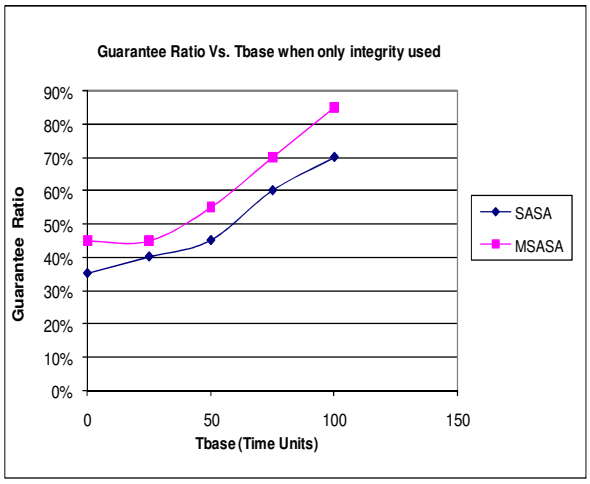



**Fig 5(a): Impact of the integrity Security Service**

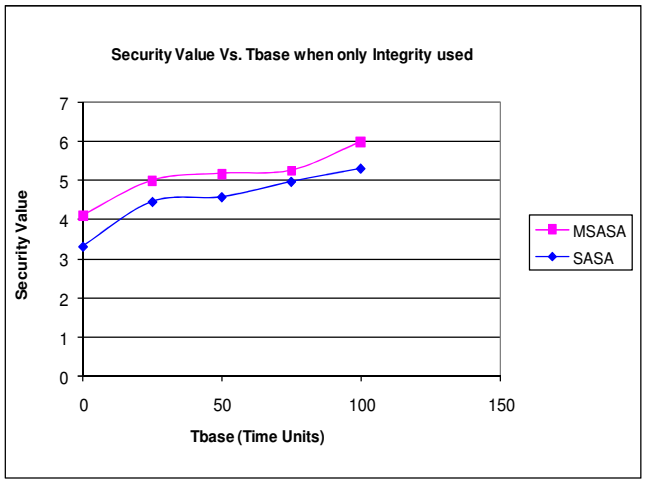



**Fig 5(b): Impact of the integrity Security Service**

## 5. Conclusion:

Security and timeliness both are equally important parameter for real time applications running over clusters. In this paper we propose a modified security aware scheduling approach that utilizes the concept of criticality and threshold based improvement in security of task over its minimum security requirement. This paper discusses system architecture, mathematical modeling and modified approach. The performance of modified approach is observed to simulation studies and example used. It is observed that modified approach have improvement about 15 % in terms of both guarantee ratio and security value received. The modified approach is applicable over wide range of application requesting different kind of security services and trimming constraints.

## References:

1. Makan Pourzandi, Ibrahim Haddad, Charles Levert, MiroslawZakrzewski: A New Architecture for Secure Carrier-Class Clusters. IEEE International Conference on Cluster Computing, 23-26 Sept. 2002, Page(s):494 – 497.
2. Dessouly, Alaa Amin and Reda Ammar and Ayman El: Scheduling Real Time Parallel Structures on Cluster Computing with Possible Processor Failure. IEEE 9th International Symposium on Computers and Communications, Volume 1, 28 June-1 July 2004, Page(s):62 – 67.
3. Parnas, J.Xu and D.L.: Scheduling Processes with Release Times, Deadlines, Precedence and Exclusion Relations. Transactions on Software Engineering, IEEE Volume 16, Issue 3, March 1990, Page(s):360 - 369
4. SHILOH, O. t. Amnon BARAK: Scalable cluster computing with MOSIX for LINUX. In Proceedings of 5th Annual Linux Expo, pages 95--100, May 1999.

5. Qin, Tao Xie and Xiao: Scheduling Security Critical Real Time Applications on Clusters. IEEE transactions on computers, Vol. 55, no 7, pp. 864-879 July 2006.
6. Gagne, T.Shepard and M: A Pre-Run-Time Scheduling Algorithm for Hard Real Time Systems. Transactions on Software Engineering, IEEE Volume 17, Issue 7, July 1991, Page(s):669 - 677.
7. X. Zhang, Y. Qu, and L. Xiao.: Improving Distributed Workload Performance by Sharing both CPU and Memory Resources. 20th International Conference on Distributed Computing Systems, IEEE. 10-13 April 2000, Page(s):233 – 241.
8. Kavi, Wenming Li and Krishna: A Non Preemptive Scheduling Algorithm for Soft Real Time Systems. Computers & Electrical Engineering, Volume 33, Issue 1, January 2007, Pages 12-29.
9. O.Elkeelany, M.matalgah, K.Sheikh: Performance Analysis of IPSEC Protocol: Encryption & Authentication. International conference on Communication IEEE 2002. Volume 2, page(s):1164-1168.
10. Martel, K. Jeffay and C. U: On Non-Preemptive Scheduling of periodic and Sporadic Tasks: Proceedings of the 12th IEEE Real-Time Systems Symposium, San Antonio, Texas, December 1991, IEEE Computer Society Press, pp. 129-139.
11. M. L. Dertouzos and A. K Mok : Multi-Processor Online Scheduling of Hard Real- Time Tasks: IEEE Transactions on Software Engineering, Vol. 15, No. 12, December 1989 , pp. 1497-1506.
12. J.Deepkumara, H.M. Heys and R.venkatesan: Performance Comparison of Message Authentication Code for Internet protocol Security. www.engr.mun.ca/~howard/PAPERS/necec_2003b.pdf 2003.
13. Genesis, M. H. A.M. Goscinski and J. Silock: The operating system managing parallelism and providing single system image on cluster. LNCS volume 2790/2004, publisher Springer Berlin / Heidelberg.
14. Savarese, T. Sterling and D : A parallel workstation for scientific computation. Proceedings of the 24th International Conference on Parallel Processing, August 14-18, 1995, Urbana-Champain, Illinois, USA. Volume I: Architecture.
15. A J.Hong, X. Tan and D. Towsley: performance analysis of minimum laxity and earliest deadline scheduling in a real time system. IEEE Transactions on Computers, Volume 38, Issue 12, Dec. 1989, Page(s):1736 - 1744.
16. Foster, Ian, Nicholas Karonis: Managing Security in High Performance Distributed Computations. Journal of Cluster Computing Volume 1, Issue 1 pages 95-107, publisher Springer Netherlands 1998.
17. Manhee Lee, Eun Jung Kim, Ki Hwan Yum: An overview of security issues in cluster interconnects. Sixth IEEE International Symposium on Cluster Computing and the Grid Workshops, 2006. Volume 2, 16-19 May, Page(s):9 pp.
18. Ian Foster, Carl Kesselman, Gene Tsudik, Steven Tuecke : A security architecture for computational grids: Proceedings of the 5th ACM conference on Computer and communications security CCS 1998.
19. Ferrari, Adam et al. A flexible security system for Metacomputing Environments www.cs.virginia.edu/papers/hpcn99.pdf 1999.
20. R. David, S. Son and R. Mukkamala: Supporting Security Requirements in Multilevel Real Time Database. IEEE Symposium on Security and Privacy, 8-10 May1995, Page(s):199 – 210,.
21. R. Mukkamala and S. Son: A Secure Concurrency Control Protocol for Real-Time Database: IFIP Workshop on Database Security. 1995.
22. S.H. Son, C. Chaney, C. and N. Thomlinson: Partial Security policy to Support Timeliness in Secure Real Time Databases. IEEE Symposium on Security and Privacy, 3-6 May 1998, Page(s):136 – 147.
23. Bosselares, R.Govaerts : Fast Hashing on the Pentium. Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology. Springer-Verlag, 1996 Pages: 298 - 312.

COMPUTER SCIENCE JOURNALS SDN BHD
M-3-19, PLAZA DAMAS
SRI HARTAMAS
50480, KUALA LUMPUR
MALAYSIA