Further Research on Registration System with Vandermonde Matrix

Ning Huang[1], Xian-tong Huang[2], Jing-li Ren[3], Xian-wen He[2], Yang Liu[2]
[1]Center of Modern Educational Technology, Gannan Normal University
Ganzhou, 341000, China
hngzjx@qq.com
[2]College of Mathematics and Computer Science, Gannan Normal University
Ganzhou, 341000, China
[3]College of Communication and Media, Gannan Normal University
Ganzhou, 341000, China

Abstract

We propose an improved software registration system from our previous research. Our improvements are mainly as follows. (1) Changing basic field to make the scheme suitable for all characters. (2) Changing encryption and decryption formulae to make the scheme more complex. (3) Using the technique of letter decomposition and composition to make the scheme more deceptive to a possible adversary. (4) Using mobile phone in the system to enhance the security. Experimental results and analysis show that the improvements are successful and the scheme is viable and secure.
Keywords: *software, registration, Vandermonde matrix*

1. Introduction

Copy protection for computer software started a long cat-and-mouse struggle between publishers and crackers. These were (and are) programmers who would defeat copy protection on software as a hobby, add their alias to the title screen, and then distribute the "cracked" product to the network BBSes or Internet sites that specialized in distributing unauthorized copies of software [1]. Research on the topic never ends. In [2], we proposed a scheme of registration with Vandermonde matrix in a Galois field $GF(p)$, which is an application of Hill cipher [8]. In this paper, we further discuss the improvements of the method. Our improvements are mainly as follows. (1) Using the Galois field $GF(2^m)$ instead of $GF(p)$ to make the scheme suitable for all characters. (2) Using matrix equation $Y = V^{-1}X+C$ instead of $Y = V^{-1}X$ to make the scheme more complex. (3) Using the technique of letter decomposition and composition to make the scheme more deceptive to a possible adversary. (4) Using mobile phone in the system to enhance the security. The rest of the paper is organized as follows. In Section 2, we briefly introduce our original research in $GF(p)$. In Section 3, we propose some novel ideas to improve our original scheme. In Section 4, we design the registration system. In Section 5, we give experimental results and analysis. We conclude the paper in Section 6.

2. Original Scheme

Agree on permission control character string such as $PROFESSIONALVERSION$ on both sides of the vendor and user. Then we take n different characters $\mu_1, \mu_2, , \cdots, \mu_n$, from hard id [7] to create a Vandermonde matrix in a Galois field $GF(p)$, where p is a prime. That is $V = V(\mu_1, \mu_2, \cdots, \mu_n)$

$$= \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \mu_1 & \mu_2 & \cdots & \mu_n \\ \vdots & \vdots & \ddots & \vdots \\ \mu_1^{n-1} & \mu_2^{n-1} & \cdots & \mu_n^{n-1} \end{pmatrix} mod\ p \quad (1)$$

Then we obtain a determinant formula

$$det(V) = \prod_{1 \le i < j \le n} (p + \mu_i - \mu_j)\ mod\ p \quad (2)$$

It follows from $\mu_i s$ are different from each other that V is invertible. The fast computation of the inverse of V is

$$A = V^{-1}(\mu_1, \mu_2, \cdots, \mu_n) = HL \ mod \ p \quad (3)$$

where H is an upper triangular matrix and L a lower triangular one. The elements of each can be obtained from recursive formulae. Let

$$X = (x_1, \quad x_2, \quad ... \quad , x_n)^T$$

be the plain text, define a linear mapping:
$X \mapsto Y = (y_1, \quad y_2, \quad ... \quad y_n)^T$

$$= VX \ mod \ p \quad (4)$$

as an encryption algorithm, which is equivalent to the following linear functions:

$$\begin{cases} y_1 = x_1 + x_2 + \cdots + x_n \\ y_2 = \mu_1 x_1 + \mu_2 x_2 + \cdots + \mu_n x_n \quad mod \ p \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ y_n = \mu_1^{n-1} x_1 + \mu_2^{n-1} x_2 + \cdots + \mu_n^{n-1} x_n \end{cases} \quad (5)$$

The decryption algorithm is

$$X = V^{-1}Y = AY \ mod \ p \quad (6)$$

which is equivalent to the following linear functions:

$$\begin{cases} x_1 = a_{11}y_1 + a_{12}y_2 + \cdots + a_{1n}y_n \\ x_2 = a_{21}y_1 + a_{22}y_2 + \cdots + a_{2n}y_n \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ x_n = a_{n1}y_1 + a_{n2}y_2 + \cdots + a_{nn}y_n \end{cases} mod \ p \quad (7)$$

In order to minimize the set of necessary formulae on the user's side, we use $A = V^{-1}$ on the vendor's side as the encryption key while V on the user's side as the decryption key. On the vendor's side, we take the permission control string as the plain text X. We compute $Y = V^{-1}X \ mod \ p$ as the cipher text. On the user's side, we use $X = VY \ mod \ p$ to verify the registration. Our original scheme takes $p = 37$ as the modulus, choose 10 numbers and 26 capital letters and the symbol '\$'. Lower case letters are converted to upper ones. Other symbols are ignored. The key space is $37 \cdot 36 \cdots (37 - n + 1)$ for a given n.

3. Improved Scheme

3.1 Basic field of the scheme

To make the scheme universal, we define our computation in the field $GF(2^m)$. In fact, $GF(2^m)$ is congruent to $Z_2[x]/f(x)$, where $Z_2[x]$ is the polynomial ring over Z_2 , $f(x)$ is a primitive polynomial of $Z_2[x]$. In $Z_2[x]/f(x)$, the addition of polynomials $\alpha(x)$ and $\beta(x)$ is defined by $\sigma(x) = \alpha(x) + \beta(x)$.With the character of Z_2 being 2, for arbitrary $\alpha(x) \in Z_2[x]$, we have $\alpha(x) + \alpha(x) = 0$.The multiplication of polynomials and is defined by $\pi(x) = \alpha(x) + \beta(x) \quad mod \ f(x)$.See more details in [4, 5, 6]. If we take $m = 8$,$GF(2^8)$ is just the letter set of extended $ASCII$. This means we can use every symbol in a plain text. However, that will cause the problem of unprintable letters in the registration string. We use the decomposition of letters to solve this problem by splitting a letter into two, each is in the range of $\{0, 1, 2, \cdots, 9, A, B, C, D, E, F\}$. Furthermore, we plus each value with $'A'$ in the generation of a registration string. For example, if a letter with the value 0 is used for a registration string, we really see it as $'A'$. Similarly, we see $'B'$ for value 1, $'C'$ for value 2 through $'P'$ for value F. This can easily be realized in C language. Suppose c is in the cipher text, the decomposition and conversion is expressed by

$$c_1 = c/0x10 +' A';$$
$$c_2 = c\%0x10 +' A';$$

This method also increases the complexity of the encryption. Bravo! It actually butters both sides of our bread.

3.2 Fast computation of $V^{-1} = HL$

We develop the method in [3] to use it in $GF(2^m)$. Let L be the matrix whose rows are associated with the coefficients of the polynomials

$$\begin{cases} \psi_1(s) = 1 \\ \\ \psi_i = (s + \mu_j)\psi_{i-1}(s) \end{cases} \quad (8)$$

L can be denoted by
$L(1, s, \cdots, s^{n-1})^T = (\psi_1(s), \psi_2(s), \cdots, \psi_n(s))^T$

$$L = \begin{pmatrix} l_{11} & l_{12} & \cdots & l_{1n} \\ l_{21} & l_{22} & \cdots & l_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{pmatrix} \quad (9)$$

$$\text{Let} \quad l_{ii} = 1, l_{ij} = 0 (i \neq j). \quad (10)$$

$$\begin{cases} l_{i+1,1} = (\mu_i) \cdot l_{i,j-1} \\ l_{i+1,j} = l_{i,j-1} \\ j = 2, 3, \cdots, i, i = 1, 2, \cdots, n \end{cases} \quad (11)$$

Let the initial vector $h_n = (c_1, c_2, \cdots, c_n)^T$ be determined from the partial fraction expansion

$$\frac{1}{(s+\mu_1)\cdots(s+\mu_n)} = \frac{c_1}{s+\mu_1} + \cdots + \frac{c_n}{s+\mu_n}.$$

Denote H by

$$H = (h_1, h_2, \cdots, h_n)^T \tag{12}$$

$$h_{ij} = \begin{cases} \frac{1}{\psi'_{j+1}(\mu_i)} & if\ i \le j \\ 0 \quad , & if\ i > j \end{cases} \tag{13}$$

Denote *d(s)* by

$$d(s) = (\mu_1 + s, \mu_2 + s, \cdots, \mu_n + s)^T \tag{14}$$

Let

$$h_{i-1} = h_i \otimes d(\mu_i), i = n, n-1, \cdots, 2 \tag{15}$$

ending at $h_1 = (1, 0, \cdots, 0)^T$. The right side of (14) is the inner product of h_i and $d(\mu_i)$, i.e., if $u = (u_1, u_2, \cdots, u_n)^T$,$v = (v_1, v_2, \cdots, v_n)^T$ then

$$h_i \otimes d(\mu_i) = (u_1 v_1, u_2 v_2, \cdots, u_n v_n)^T \tag{16}$$

It follows that the formulae to compute the elements in the upper triangular matrix H are given by

$$h_{ij} = \begin{cases} \frac{1}{\psi'_{j+1}(\mu_i)}, & if\ i \le j \\ 0 \quad , & if\ i > j \end{cases} \tag{17}$$

where

$$\psi'_{j+1}(\mu_i) = \prod_{k=1, k \ne i}^{j-1} (\mu_i + \mu_k)\ . \tag{18}$$

3.3 Improvements of the linear mappings

In the field $GF(2^8)$, Let

$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{pmatrix} \tag{19}$$

be the plain text, and

$$C = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1m} \\ c_{21} & c_{22} & \cdots & c_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nm} \end{pmatrix} \tag{20}$$

be the same size as X.Then

$$Y = V^{-1}X + C$$

is the matrix containing cipher text.

This is equivalent to m groups of linear mappings as follows.

$$\begin{cases} y_{11} = a_{11}x_{11} + \cdots + a_{1n}x_{n1} + c_{11} \\ y_{21} = a_{21}x_{11} + \cdots + a_{2n}x_{n1} + c_{21} \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ y_{n1} = a_{n1}x_{11} + \cdots + a_{2n}x_{n1} + c_{n1} \end{cases} \tag{21}$$

$$\begin{cases} y_{12} = a_{11}x_{12} + \cdots + a_{1n}x_{n2} + c_{12} \\ y_{22} = a_{21}x_{12} + \cdots + a_{2n}x_{n2} + c_{22} \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ y_{n2} = a_{n1}x_{12} + \cdots + a_{2n}x_{n2} + c_{n2} \end{cases} \tag{22}$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$\begin{cases} y_{1m} = a_{11}x_{1m} + \cdots + a_{1n}x_{nm} + c_{1m} \\ y_{2m} = a_{21}x_{1m} + \cdots + a_{2n}x_{nm} + c_{2m} \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ y_{nm} = a_{n1}x_{1m} + \cdots + a_{2n}x_{nm} + c_{nm} \end{cases} \tag{23}$$

In the field of character 2, the decryption is expressed as follows.

$$X = V(Y - C) = V(Y + C).$$

This is equivalent to m groups of linear mappings as follows.

$$\begin{cases} x_{11} = y^*_{11} + \cdots + y^*_{n1} \\ x_{21} = \mu_1 y^*_{11} + \cdots + \mu_n y^*_{n1} \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ x_{n1} = \mu_1^{n-1} y^*_{11} + \cdots + \mu_n^{n-1} y^*_{n1} \end{cases} \tag{24}$$

$$\begin{cases} x_{12} = y^*_{12} + \cdots + y^*_{n2} \\ x_{22} = \mu_1 y^*_{12} + \cdots + \mu_n y^*_{n2} \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ x_{n2} = \mu_1^{n-1} y^*_{12} + \cdots + \mu_n^{n-1} y^*_{n2} \end{cases} \tag{25}$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$\begin{cases} x_{1m} = y^*_{1m} + \cdots + y^*_{nm} \\ x_{2m} = \mu_1 y^*_{1m} + \cdots + \mu_n y^*_{nm} \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ x_{nm} = \mu_1^{n-1} y^*_{1m} + \cdots + \mu_n^{n-1} y^*_{nm} \end{cases} \tag{26}$$

where $y^*_{ij} = y_{ij} - c_{ij} = y_{ij} + c_{ij}$.

This cipher is obviously more complex compared with the original one. The security is enhanced.

3.4 Use of mobile phone as a receiver

In our original scheme, we assumed that the user got the registration string vie web service or email. Now we propose the registration string can also be received as a message via a mobile phone.

Step1 The user pays for the software and submits personal information, e.g. hard id, name, mobile phone number to the server via web;

Step2 The server checks the submission;

Step3 The sever creates a registration string;

Step4 The server sends the registration string to the user's mobile phone via wireless tunnel;

Step5 The user reads the message and register the software. The flow of the process is shown by Fig. Registration steps.

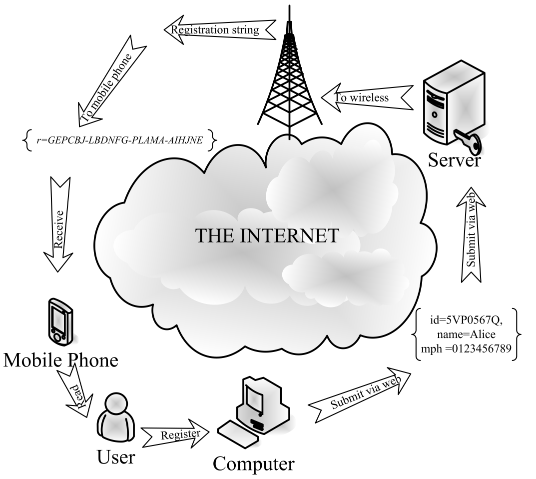

Fig. Registration steps

4. Registration scheme

Set preliminary conditions on both sides of the vendor and user:
A character string as a permission control string p_1 ; dimension n.

4.1 The Creation of registration string

The user submits the message as follows. Hard id(id for short), name, mobile phone number(mph for short).
On the vendor's side, the server computes as follows:
Input: $id, name, mph$;
Output: Registration string like
$r = xxxxxx - xxxxxx - \cdots - xxxxxx$

Algorithm:

Step1 Selects different elements from id, forms a new id_1;

Step2 Creates matrices of lower triangular L and upper triangular H from id_1 as shown in subsection 3.2;

Step3 Computes the inverse of Vandermonde matrix

$$V^{-1} = HL;$$

Step4 Appends necessary dots to the permission control string p_1 to fit the dimension, puts the result in p_2 ;

Step5 Puts p_2 into matrix X and creates a matrix C from $name$ which matches dimensions in step4, using the elements of $name$ circularly.

$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{pmatrix}$$

$$C = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1m} \\ c_{21} & c_{22} & \cdots & c_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nm} \end{pmatrix}$$

Step6 Computes

$$Y = V^{-1}X + C;$$

Step7 Decomposes Y and adds $'A'$ to each element respectively to update Y;

Step8 Takes elements from Y to create a registration string like

$$r = xxxxxx - xxxxxx - \cdots - xxxxxx.$$

Then the server sends the registration string above to the user's mobile phone via wireless tunnel.

4.2 The use of registration string

Reading the registration string from mobile phone, the user keys in to register the software. **Input:** Registration string like

$$r = xxxxxx - xxxxxx - \cdots - xxxxxx.$$

Output:

$$TRUE/FALSE$$

Algorithm:

Step1 Takes elements from r to create a matrix Y

Step2 Minuses $'A'$ from each element of Y to update Y;

Step3 In the same way as step6 in last subsection, creates matrix C ;

Step4 In the field of character 2, computes $Y - C = Y + C$;

Step5 Computes Vandermonde matrix V from local hardware id , id_1;

Step6 Computes $X = V(Y - C) = V(Y + C)$ to get a possible plain text;

Step7 Obtains p_2 from X;

Step8 Removes redundant dots if there are any at the end to get p_1 ;

Step9 Compares p_1 with the preliminary one;

Step10 The result is $TRUE$ or $FALSE$. If the verification is successful, the registration will be approved, otherwise it will be defied. A registered software picks up the registration string automatically and verifies it according to the above routine to decide which functions the software can use.

5. Experimental results and analysis

Embeds in the software on both sides of the vendor and user:

$p_1 = VIPversion$

as a permission control string; Agree on dimension $n = 6$.

5.1 The Creation of registration string

The user submits the messages as follows.$id = 5VP0567Q, name = Alice, mph = 0123456789$.
On the vendor's side, the server computes as follows.

Input:
$id = 5VP0567Q, name = Alice$;

Output:
$r = GEPCBJ - LBDNFG - PIMAMA - AIHJNE$;

Algorithm:

Step1 Selects different elements from id , forms a new id_1 :
$id_1 = 5VP067Q$;

Step2 Creates lower triangular and upper triangular matrices:

$$L = \begin{pmatrix} 01 & 00 & 00 & 00 & 00 & 00 \\ 35 & 01 & 00 & 00 & 00 & 00 \\ 27 & 63 & 01 & 00 & 00 & 00 \\ BE & 22 & 33 & 01 & 00 & 00 \\ 19 & 3A & 2B & 03 & 01 & 00 \\ B0 & CC & DF & 27 & 35 & 01 \end{pmatrix}$$

and

$$H = \begin{pmatrix} 01 & 18 & 73 & 99 & 92 & 37 \\ 00 & 01 & 19 & BE & A2 & 8B \\ 00 & 00 & 18 & 95 & B3 & 39 \\ 00 & 00 & 00 & 73 & 88 & 17 \\ 00 & 00 & 00 & 00 & 99 & A5 \\ 00 & 00 & 00 & 00 & 00 & 92 \end{pmatrix}$$

Step3 Computes$V^{-1} = HL$

$$= \begin{pmatrix} 07 & A3 & C0 & A2 & 83 & 85 \\ A5 & 10 & BA & F5 & D3 & 06 \\ B1 & AA & F0 & D7 & FE & 73 \\ CD & F5 & 86 & 71 & 2D & 64 \\ 92 & 53 & 50 & 8A & F8 & 39 \\ 4D & BF & 5C & 7B & 7B & AD \end{pmatrix}$$

Step4 Appends 2 dots to the permission control string to fit the dimension,
$p_2 = VIPversion..$;

Step5 Puts $p_2, name$ into matrices:

$$X = \begin{pmatrix} V & s \\ I & i \\ P & o \\ v & n \\ e & . \\ r & . \end{pmatrix} \quad or \quad \begin{pmatrix} 56 & 73 \\ 49 & 69 \\ 50 & 6F \\ 76 & 6E \\ 65 & 2E \\ 72 & 2E \end{pmatrix}$$

then

$$C = \begin{pmatrix} A & l \\ l & i \\ i & c \\ c & e \\ e & A \\ A & l \end{pmatrix} \quad or \quad \begin{pmatrix} 41 & 6C \\ 6C & 69 \\ 69 & 63 \\ 63 & 65 \\ 65 & 41 \\ 41 & 6C \end{pmatrix}$$

which matche dimension in step4;

Step6 Computes

$$Y = \begin{pmatrix} 64 & F8 \\ F2 & C0 \\ 19 & C0 \\ B1 & 08 \\ 3D & 79 \\ 56 & D4 \end{pmatrix};$$

Step7 Decomposes Y and adds $'A'$ to each element respectively to get

$$Y = \begin{pmatrix} G & L & P & A \\ E & B & I & I \\ P & D & M & H \\ C & N & A & J \\ B & F & M & N \\ J & G & A & E \end{pmatrix};$$

Step8 Takes elements from Y to create a registration string: $r = GEPCBJ - LBDNFG - PIMAMA - AIHJNE$; Then the server sends the registration string to the user's mobile phone via wireless tunnel.

5.2 The use of registration string

Reading the registration string from mobile phone, the user keys in to register the software.

Input:
$r = GEPCBJ - LBDNFG - PIMAMA - AIHJNE$

Output:
$TRUE/FALSE$

Algorithm:

Step1 Takes elements from
$r = GEPCBJ - LBDNFG - PIMAMA - AIHJNE$
to get a matrix

$$Y = \begin{pmatrix} G & L & P & A \\ E & B & I & I \\ P & D & M & H \\ C & N & A & J \\ B & F & M & N \\ J & G & A & E \end{pmatrix};$$

Step2 Minuses $'A'$ from each element of Y to get a new matrix

$$Y = \begin{pmatrix} 64 & F8 \\ F2 & C0 \\ 19 & C0 \\ B1 & 08 \\ 3D & 79 \\ 56 & D4 \end{pmatrix};$$

Step3 In the same way as step5 in last subsection, creates

$$C = \begin{pmatrix} A & l \\ l & i \\ i & c \\ c & e \\ e & A \\ A & l \end{pmatrix} \quad or \quad \begin{pmatrix} 41 & 6C \\ 6C & 69 \\ 69 & 63 \\ 63 & 65 \\ 65 & 41 \\ 41 & 6C \end{pmatrix};$$

Step4 In the field of character 2, computes

$$Y - C = Y + C = \begin{pmatrix} 25 & 94 \\ 9E & A9 \\ 70 & A3 \\ D2 & 6D \\ 58 & 38 \\ 17 & B8 \end{pmatrix}$$

Step5 Computes Vandermonde matrix

$$V = \begin{pmatrix} 01 & 01 & 01 & 01 & 01 & 01 \\ 35 & 56 & 50 & 30 & 36 & 37 \\ 96 & D9 & CD & 87 & 93 & 92 \\ D0 & 5F & 33 & 05 & AF & 0B \\ DA & 69 & 52 & F0 & CB & CA \\ 33 & B4 & 6D & CD & 5D & A1 \end{pmatrix}$$

from local hardware
$id = 5VP0567Q, id1 = 5VP067Q;$

Step6 Computes a possible plain text
$X = V(Y - C) = V(Y + C)$

$$= \begin{pmatrix} 56 & 73 \\ 49 & 69 \\ 50 & 6F \\ 76 & 6E \\ 65 & 2E \\ 72 & 2E \end{pmatrix} \quad or \quad \begin{pmatrix} V & s \\ I & i \\ P & o \\ v & n \\ e & . \\ r & . \end{pmatrix};$$

Step7 Obtains
$p_2 = VIPversion..;$
from X;

Step8 Removes 2 redundant dots at the end to get $p_1 = VIPversion$

Step9 Compares p_1 with the preliminary one;

Step10 The result is $TRUE$.

5.3 Analysis of the results

In our novel scheme, the computations are performed in $GF(2^8)$ to make the scheme suitable for the whole extended $ASCII$ table. Meanwhile, the key space expands from $37 \cdot 36 \cdots (37 - n + 1)$ for a given $256 \cdot 255 \cdots (256 - n + 1)$. The uses of mobile phone, user's name, the decomposition and composition of letters also enhance the cipher. Experimental results show the success of the scheme.

6. Conclusions

It follows from Experimental results and analysis that the novel scheme is enhanced. We get better results in our further research.

Acknowledgements

1)The authors would like to express their thanks to the hard work of the editors and reviewers of this paper.
2)This work was supported by the fund from Natural Science of Jiangxi Province of China under Grant No.20114BAB201033. The authors would like to express their thanks to the Committee of the fund.

References

[1] Copy protection for computer software , http://en.wikipedia.org/wiki/Copy _ protection;2012.

[2] N. Huang, "Permission control of software based on registration system with Vandermonde matrix in a Galois field ",In Instrumentation & Measurement, Sensor Network and Automation (IMSNA), 2012 International Symposium on, 2012, Vol. 2, pp.487-490.

[3] S.H. Hou, and E. Hou, "Triangular Factors of the Inverse of Vandermonde Matrices," In Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 Vol 2, IMECS 2008, pp.19-21.

[4] Darrel Hankerson, "Alfred Menezes and Scott Vanstone.Guide to Elliptic Curve Cryptography", Berlin:Springer,2003.

[5] Roberto M. Avanzi, Henri Cohen, Christophe Doche,Gerhard Frey, Tanja Lange, Kim Nguyen, and Frederik Vercauteren.,"Handbook of elliptic and hyperelliptic curve cryptography", London:Taylor & Francis Group,2006.

[6] William J.Gilbert and W.Keith Nicholson, "Modern Alegbra with Applications",Second Edition.New Jersy:John Wiley &Sons,Inc,2003.

[7] S.D.S. Monteiro and R.F. Erbacher, "Exemplifying Attack Identification and Analysis in a Novel Forensically Viable Syslog Model",In Washington:IEEE Computer Society,Proceedings of the Third International Workshop on Systematic Approaches to Digital Forensic Engineering,2008,pp.57-68.

[8] L. S. Hill, "Cryptography in an Algebraic Alphabet," The American Mathematical Monthly, Vol. 36, No. 6. (Jun. - Jul., 1929), pp. 306-312.

Ning Huang, born in 1958, received Master's degree in applied mathematics and computer science from Jiangxi University, China in 1991, awarded senior engineer of the Industrial and Commercial Bank of China in 2001. He is now with Center of Modern Educational Technology, Gannan Normal University, Ganzhou, China,as an associate professor. His research interests include information security and digital campus.

Xian-tong Huang, born in 1966, received Doctor's degree in computational mathematics from Hunan University, China in 2006. He is now with College of Mathematics and Computer Science, Gannan Normal University, Ganzhou, China as a professor. His research interests include computational mathematics and information security.

Jing-li Ren, born in 1980, received Master's degree in computer science from Wuhan University of Technology, China in 2006. She is now with College of Communication and Media, Gannan Normal University, Ganzhou, China, as a lecturer. Her research interests include Intelligent computation and simulation.

Xian-wen He, born in 1974, received Master's degree in computer science from Nanchang University, China in 2007. He is now with College of Mathematics and Computer Science, Gannan Normal University, Ganzhou,China, as an associate professor. His research interests include network security and image recognition.

Yang Liu, born in 1972, received Master's degree in computer science from Nanchang University, China in 2007. He is now with College of Mathematics and Computer Science, Gannan Normal University, Ganzhou,China, as a lecturer. His research interests include algorithm optimization and image recognition.