Matrix Extensions of the RSA Algorithm

Chih-Chwen Chuang and James George Dunham
Department of Electrical Engineering
Southern Methodist University
Dallas, TX 75275

Abstract

A new matrix extension of the RSA algorithm is proposed which is based on the Cayley-Hamilton theorem and a one-way function. The security of this algorithm rests upon both that of the RSA algorithm and the one-way function. The computational efficiency of the new algorithm depends on the dimension of the matrix. The most efficient implementation is the 2X2 case in which both encryption and decryption use a single modulo arithmetic multiplication and single evaluation of the one-way function.

1. Introduction

The Rivest-Shamir-Adleman (RSA) [8] algorithm is the best known public-key cryptosystem. Although many papers have discussed efficient implementations of discrete exponentiation algorithms [5-7,9,11], the RSA runs substantially slower than many secret-key cryptosystems such as the Data Encryption Standard (DES) algorithm. V. Varadharajan and R. Odoni [10] proposed a matrix extension of the RSA algorithm, but did not carefully address security issues.

A new matrix extension is proposed that is based upon the Cayley-Hamilton theorem and a one-way function. Under a chosen plaintext attack on the key, the security of the new algorithm is equivalent to that of the RSA algorithm. Under a known plaintext attack on the message, the security of the system rests upon that of the one-way function.

The computational efficiency of the new algorithm depends upon the dimension of the matrix. The most efficient implementation is the 2X2 case in which, both encryption and decryption use a single modulo arithmetic multiplication and single evaluation of the one-way function. Thus these public-key cryptosystems have the potential of a fast implementation.

A.J. Menezes and S.A. Vanstone (Eds.): Advances in Cryptology - CRYPTO '90, LNCS 537, pp. 140-155, 1991.

2. Background

The main tool in computing the matrix RSA scheme will be the Cayley-Hamilton theorem. Let $\mathbf{A}$ be an $n \times n$ matrix. Define

$$P_{\mathbf{A}}(\lambda) \triangleq \det(\lambda \mathbf{I} - \mathbf{A}) = \lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_0$$

to be the characteristic polynomial of $\mathbf{A}$. Then the Cayley-Hamilton theorem states that

$$P_{\mathbf{A}} = \mathbf{A}^n + a_{n-1}\mathbf{A}^{n-1} + \cdots + a_0\mathbf{I} = 0.$$

One important use of the Cayley-Hamilton theorem is to evaluate powers of $\mathbf{A}$ as

$$\mathbf{A}^k = c_{n-1}\mathbf{A}^{n-1} + \cdots + c_0\mathbf{I} \quad . \tag{1}$$

The Cayley-Hamilton theorem holds for matrices whose entries are from any commutative ring such as arithmetic modulo R [3].

The eigenvalues are needed to calculate the coefficients c_i, $i = 0, 1, \cdots, n-1$, in Eq. (1). To simplify the calculation, a triangular matrix is chosen to construct the matrix RSA scheme because the diagonal entries are the eigenvalues of the matrix.

3. Construction of the Cryptosystem

With the above background, the matrix form of the RSA scheme is constructed as follow:

(1) Choose two large strong primes p and q as proposed by Gordon [2] and calculate $R = p \cdot q$;

(2) Choose a large integer e as a public key such that $0 < e < (p-1)(q-1)$ and it is relatively prime to $(p-1)(q-1)$;

(3) Calculate d as a private key from $d \cdot e \equiv 1(mod(p-1)(q-1))$;

(4) Construct an $n \times n$ triangular matrix $\mathbf{A}$ whose diagonal entries are all distinct and the differences between diagonal entries are relatively prime to R;

(5) The entries above (upper triangular matrix) or below (lower triangular matrix) the diagonal are reserved for messages (The details will be described later).

4. Encryption

The encryption algorithm is as follow:

(1) Choose diagonal entries at random subject to the constraints in step (4) stated above;

(2) The number r is a function $g(\cdot)$ computed from the diagonal entries and the function $r \equiv (a_{11}+a_{22}+ \cdots +a_{nn}) \, mod \; R$ is suggested;

(3) Set up the following equations, calculate the coefficients c_i^e, $i = 0, 1, \cdots, n-1$, and store them in a safe place;

$$
\begin{aligned}
a_{11}^e &\equiv c_{n-1}^e a_{11}^{n-1} + c_{n-2}^e a_{11}^{n-2} + \cdots + c_0^e \\
a_{22}^e &\equiv c_{n-1}^e a_{22}^{n-1} + c_{n-2}^e a_{22}^{n-2} + \cdots + c_0^e \\
&\vdots \\
a_{nn}^e &\equiv c_{n-1}^e a_{nn}^{n-1} + c_{n-2}^e a_{nn}^{n-2} + \cdots + c_0^e
\end{aligned}
\qquad (2)
$$

where a_{ii}, $i = 1, 2, \cdots, n$, denotes the i^{th} diagonal entry.

(4) (a) Let $i = 1$ and $j = 2$;

(b) Calculate $f(r)$ using the number r;

(c) Calculate $f(r) \oplus m_{ij}$ where m_{ij} is the message and place the result in the $(i, j)^{th}$ entry of the matrix where $\oplus$ denotes bit-by-bit exclusive OR;

(d) Update $r \equiv r + 1 \; mod \; R$ and $j = j + 1$. Repeat steps (b) and (c) until $j = n$;

(e) Let $i = i + 1$ and $j = i + 1$. Repeat steps (b), (c), and (d) until $i = n - 1$;

(5) Use the calculated coefficients in step (3) and Cayley-Hamilton theorem to encrypt the matrix. Send the ecrypted matrix to the recipients;

(6) Let $r \equiv r+1 \; (mod \; R)$ and repeat steps (4) and (5) until all encrypted messages are sent out.

5. Decryption

After the legitimate recipients receive the matrices, the message is decrypted as follow:

(1) Set up the following equations, calculate the coefficients c_i^d, $i = 0, 1, \cdots, n-1$, and store them in a safe place;

$$
\begin{aligned}
a_{11}^{ed} &\equiv c_{n-1}^{d} a_{11}^{e(n-1)} + c_{n-2}^{d} a_{11}^{e(n-2)} + \cdots + c_{0}^{d} \\
a_{22}^{ed} &\equiv c_{n-1}^{d} a_{22}^{e(n-1)} + c_{n-2}^{d} a_{22}^{e(n-2)} + \cdots + c_{0}^{d} \\
&\cdot \\
&\cdot \\
&\cdot \\
a_{nn}^{ed} &\equiv c_{n-1}^{d} a_{nn}^{e(n-1)} + c_{n-2}^{d} a_{nn}^{e(n-2)} + \cdots + c_{0}^{d}
\end{aligned}
\qquad (3)
$$

where a_{ii}^{e}, $i = 1, 2, \cdots, n$, denotes the i^{th} diagonal entry from the received matrix.

(2) Use the coefficients calculated in step (1) and apply the Cayley-Hamilton theorem to decrypt the matrix;

(3) Calculate $r \equiv (a_{11}+a_{22}+ \cdots +a_{nn})$ *mod* R and store it in a safe place;

(4) (a) Let $i = 1$ and $j = 2$;

(b) Calculate $f(r)$ using the number r;

(c) Calculate the message from

$$f(r) \oplus a_{ij} \equiv f(r) \oplus [f(r) \oplus m_{ij}] \equiv m_{ij} \ mod \ R$$

where a_{ij} is the $(i, j)^{th}$ superdiagonal entry of the received matrix;

(d) Update $r \equiv r + 1 \ mod \ R$ and $j = j + 1$. Repeat steps (b) and (c) until $j = n$;

(e) Let $i = i + 1$ and $j = i + 1$. Repeat steps (b), (c), and (d) until $i = n - 1$;

(5) Let $r \equiv r+1 \ (mod \ R)$ and repeat steps (2) and (4) until all messages are received.

Only e, R, and the function $f(\cdot)$ and $g(\cdot)$ are revealed to the public. Knowledge of d, r, a_{ii}, $i = 1, 2, \cdots, n$, and the primes p and q remain secret. The a_{ii}'s may be discarded.

Example 1:

Assume that e, d, and R are 47, 3983, and 7663, respectively and the message are 124 and 150.

A. Encryption

(1) Construct a 2X2 matrix by choosing $a_{11} = 53$ and $a_{22} = 59$.

(2) Set up the system of equations according to Eq. (2) as follow:

$$53^{47} \equiv 2824 \equiv c_0^e + 53\, c_1^e \ mod\ 7663$$

$$59^{47} \equiv 4194 \equiv c_0^e + 59\, c_1^e \ mod\ 7663$$

and solve this system of equations. The answer to c_0^e and c_1^e is 3494 and 5337, respectively.

(3) This step depends on $f(r)$, and we did not specifically define this function. So let us assume that the sequence generated by $f(r)$ is 47 and 1447.

(4) First calculate $f(r) \oplus m$ and fill the result in a_{12}. For example, $47 \oplus 124 = 83$. Then encrypt the first block of message by calculating

$$\mathbf{A}^e \equiv \mathbf{A}^{47} \equiv \begin{bmatrix} 3494 & 0 \\ 0 & 3494 \end{bmatrix} + 5337 \cdot \begin{bmatrix} 53 & 83 \\ 0 & 59 \end{bmatrix} \ mod\ 7663$$

$$\equiv \begin{bmatrix} 2824 & 6180 \\ 0 & 4194 \end{bmatrix} \ mod\ 7663\ .$$

(5) Encrypt the rest of the message in the same manner as step (4). The encrypted sequence is 6180 and 4194.

B. Decryption

(1) Set up the system of equations according to Eq. (3) as follows:

$$2824^{3983} \equiv 53 \equiv c_0^d + 2824\, c_1 mod\ 7663$$

$$4194^{3983} \equiv 59 \equiv c_0^d + 4194\, c_1^d \ mod\ 7663$$

and solve this set of equations. The answer to c_0^d and c_1^d is 1260 and 6645, respectively.

(2) Decrypt the first message by calculating the following equation

$$(\mathbf{A}^e)^{3983} \equiv \begin{bmatrix} 1260 & 0 \\ 0 & 1260 \end{bmatrix} + 6645 \cdot \begin{bmatrix} 2824 & 6180 \\ 0 & 4194 \end{bmatrix} \ mod\ 7663$$

$$\equiv \begin{bmatrix} 53 & 83 \\ 0 & 59 \end{bmatrix} \ mod\ 7663\ .$$

(3) Recover the first message by calculating $f(r) \oplus a_{12}$, while in this case is $m = 47 \oplus 83 = 124$.

(4) Repeat steps (2) and (3), except calculating the sequence of $f(r)$, to recover whole the message as 124 and 150. □

6. Security

The security of the matrix extension scheme is different from that of the original RSA scheme. We have to carefully analyze the structure of matrix **A** when **A** is raised to the e^{th} power. The structure of $\mathbf{A}^e$ can be formally stated in the following theorem.

Theorem 1: Let **A** be an $n \times n$ upper triangular matrix. Denote $a_{i,i+j}$ and $a_{i,i+j}^{(e)}$ as the $(i,i+j)^{th}$ entry and the $(i,i+j)^{th}$ entry after matrix **A** raised to the e^{th} power, respectively. Then $a_{i,i+j}^{(e)}$, $1 \le i \le n, 0 \le j \le n-i$, can be represented as

$$\begin{aligned}
a_{i,i+j}^{(e)} &= a_{i,i+j} \sum_{i_1=1}^{e} a_{ii}^{e-i_1} a_{i+j,i+j}^{i_1-1} \\
&+ \sum_{l_1=1}^{j-1} a_{i,i+l_1} a_{i+l_1,i+j} \sum_{i_1=1}^{e-1} \sum_{i_2=i_1}^{e-1} a_{ii}^{e-1-i_2} a_{i+l_1,i+l_1}^{i_2-i_1} a_{i+j,i+j}^{i_1-1} \\
&+ \sum_{l_1=1}^{j-2} \sum_{l_2=l_1+1}^{j-1} a_{i,i+l_1} a_{i+l_1,i+l_2} a_{i+l_2,i+j} \sum_{i_1=1}^{e-2} \sum_{i_2=i_1}^{e-2} \sum_{i_3=i_2}^{e-2} a_{ii}^{e-2-i_3} a_{i+l_1,i+l_1}^{i_3-i_2} a_{i+l_2,i+l_2}^{i_2-i_1} a_{i+j,i+j}^{i_1-1} \\
&\quad \vdots \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (4) \\
&+ a_{i,i+1} a_{i+1,i+2} \cdots a_{i+j-1,i+j} \sum_{i_1=1}^{e-j+1} \sum_{i_2=i_1}^{e-j+1} \cdots \sum_{i_j=i_{j-1}}^{e-j+1} a_{ii}^{e-j-i_j+1} a_{i+1,i+1}^{i_j-i_{j-1}} \cdots a_{i+j,i+j}^{i_1-1}
\end{aligned}$$

where each term

$$\sum_{i_1=1}^{e-k} \sum_{i_2=i_1}^{e-k} \cdots \sum_{i_k=i_{k-1}}^{e-k} \sum_{i_{k+1}=i_k}^{e-k} a_{ii}^{e-k-i_{k+1}} a_{i+l_1,i+l_1}^{i_{k+1}-i_k} \cdots a_{i+l_k,i+l_k}^{i_2-i_1} a_{i+j,i+j}^{i_1-1}$$

$$= \frac{\displaystyle\sum_{m=i,i+l_1,i+l_2,\ldots,i+l_k,i+j} (-1)^{\delta} a_{mm}^{e} \prod_{\substack{m_1,m_2=i,i+l_1,i+l_2,\ldots,i+l_k,i+j \\ m_1,m_2 \ne m \\ i \le m_1 < m_2 \le i+j}} (a_{m_2m_2} - a_{m_1m_1})}{\displaystyle\prod_{\substack{m_1,m_2=i,i+l_1,i+l_2,\ldots,i+l_k,i+j \\ i \le m_1 < m_2 \le i+j}} (a_{m_2m_2} - a_{m_1m_1})} . \qquad (5)$$

where $k = 1, 2, \cdots, j-1$, and

$$s = \begin{cases} k+1, & m = i \\ k+1+subscript(l_{k_1}), & m = i+l_{k_1}, k_1 = 1, 2, \cdots, k \\ 2k+2, & m = i+j \end{cases} .$$

Proof: This can be proved by induction by repeatedly multiplying the matrix by itself or with the aid of Cayley-Hamilton theorem. □

Note that in order to recover the original matrix after it being raised to the e^{th} power, the denominator of Eq. (5) has to be relatively prime to R. The constraint of a 2X2 matrix is that the difference between diagonal entries is relatively prime to R (a 2X2 matrix is computational the most efficient case).

The general form of the numerator of Eq. (5) can be decomposed and rewritten in a different form. This statement can be formally summarized in the following lemma.

Lemma 1: The general form of the numerator of Eq. (5) can be written as the nonlinear combination of the lower order equations with the same form, i.e.

$$\sum_{m=1}^{n} (-1)^S a_{mm}^{ed} \prod_{\substack{m_1,m_2 \\ m_1,m_2\neq m \\ 1\leq m_1<m_2\leq n}}^{n} (a_{m_2m_2}-a_{m_1m_1}) \ , s=n+m$$

$$= \sum_{m=1}^{n} (-1)^{S_1} a_{mm}^{n-2} [\sum_{\substack{m_1=1 \\ m_1\neq m}}^{n} (-1)^{s_2} a_{m_1m_1}^{ed} \prod_{\substack{m_2,m_3=1 \\ m_2,m_3\neq m_1 \\ 1\leq m_2<m_3\leq n}}^{n} (a_{m_3m_3}-a_{m_2m_2})] \qquad (6)$$

where $s_1 = n+m-1$ and

$$s_2 = \begin{cases} n+m_1, & m_1 < m \\ n+m_1-1, & m_1 > m \end{cases} .$$

Proof: The proof can be found in [1]. □

In general, cryptosystems are most vulnerable to a chosen-plaintext attack. There is no exception in our case because under a chosen-plaintext attack all the superdiagonal entries of the matrices can be carefully selected by the attacker. The diagonal entries, however, are fixed for a particular set of messages only and are chosen by the sender. So the diagonal entries are secret to everyone except the sender. In this case, the best way of breaking the extension scheme is finding the decryption key d. This can be formally stated in the following theorem.

Theorem 2: Consider a chosen-plaintext attack on the secret key d. If there is a polynomial time algorithm which can find the decryption key d and break the RSA algorithm, then one can find the value of the diagonal and superdiagonal entries and break the extension scheme. Conversely, if there is a polynomial time algorithm which can solve d from the equations which are set up according to Theorem 1 given $0 \equiv a_{ii}^{ed} - a_{ii} \; mod \; R$, one can break the RSA scheme using the same method.

Proof: The encrypted diagonal entries have exactly the same form as that of the RSA algorithm. So if there exists a polynomial time algorithm which can find the decryption key d and break the RSA algorithm, then this algorithm can also be employed to break the extension scheme. The proof of the first part of the theorem is done.

First define a function $f\,(i,\, i+l_1,\, i+l_2,\, \cdots,\, i+l_k,\, i+j)$ as

$$f\,(i,\, i+l_1,\, i+l_2,\, \cdots,\, i+l_k,\, i+j)$$

$$\triangleq \frac{\displaystyle\sum_{m=i,i+l_1,\ldots,i+l_k,i+j} (-1)^s \, a_{mm}^{ed} \prod_{\substack{m_1,m_2 \neq m \\ m_1,m_2=i,i+l_1,\cdots,i+l_k,i+j \\ m_1<m_2}} (a_{m_2m_2} - a_{m_1m_1})}{\displaystyle\prod_{\substack{m_1,m_2=i,i+l_1,\cdots,i+l_k,i+j \\ i\leq m_1<m_2\leq i+j}} (a_{m_2m_2} - a_{m_1m_1})}\,, \qquad (7)$$

where $k = 1, 2, \cdots, j-1$, and

$$s = \begin{cases} k+1, & m=i \\ k+1+subscript(l_k), & m=i+l_k,\; l_k = 1, 2, \cdots, j-1 \,. \\ 2k+2, & m=i+j \end{cases}$$

The difference between Eqs. (4) and (7) is that the encryption key e in Eq. (4) is substituted by ed. Now substitute Eq. (7) into Eq. (7) and rewrite Eq. (4) as

$$
\begin{aligned}
a_{i,i+j}^{(ed)} &\equiv a_{i,i+j} \\
&\equiv [a_{i,i+j} f_1(i,i+j) \\
&\quad + \sum_{l_1=1}^{j-1} a_{i,i+l_1} a_{i+l_1,i+j} \; f_2(i,i+l_1,i+j) \\
&\quad + \sum_{l_1=1}^{j-2} \sum_{l_2=l_1}^{j-1} a_{i,i+l_1} a_{i+l_1,i+l_2} a_{i+l_2,i+j} \times \\
&\qquad f_3(i,i+l_1,i+l_2,i+j) \\
&\qquad \cdot \\
&\qquad \cdot \\
&\qquad \cdot \\
&\quad + a_{i,i+1} a_{i+1,i+2} \cdots a_{i+j-1,i+j} \times \\
&\qquad f_j(i,i+1, \cdots ,i+j)] \bmod R \; .
\end{aligned}
\tag{8}
$$

It is clear that if Eq. (8) holds, then $f_1(i,\, i+j) \equiv 1 \ mod\ R$ and $f_k(i,\, i+l_1, \cdots ,i+l_k,\, i+j) \equiv 0 \ mod\ R$, $k = 1, 2, \cdots , j$. Under this condition, $f_1(i,\, i+j) \equiv 1 \ mod\ R$ can be explicitly written as:

$$0 \equiv (a_{i+j,i+j}^{ed} - a_{i+j,i+j}) - (a_{ii}^{ed} - a_{ii}) \ mod\ R \; . \tag{9}$$

The trivial solution to Eq. (9) is that d satisfies both

$$0 \equiv a_{ii}^{ed} - a_{ii} \ mod\ R \tag{10}$$

and

$$0 \equiv a_{i+j,i+j}^{ed} - a_{i+j,i+j} \ mod\ R \; . \tag{11}$$

Eqs. (10) and (11) have exactly the same form as that of the RSA algorithm. As discussed earlier, the solution to Eqs. (10) and (11) is a subset of the solution of Eq. (9). So if there exists a polynomial time algorithm which can find solutions to d given Eqs. (10) and (11) by solving Eq. (9), then this algorithm can also be employed to find the decryption key d of the RSA algorithm in the same manner.

If $f_k(i, i+l_1, \cdots, i+j) \equiv 0 \; mod \; R$ holds, Lemma 4.1 can be employed several times until the right hand side of Eq. (8) has the following form:

$$\sum_{m=0,l_1,\ldots,j} sgn_1 \, a_{mm}^{k-2} \cdots \sum_{\substack{m_{k-2}=0,l_1,\ldots,j \\ m_{k-2}\neq m, m_1, \ldots, m_{k-3}}} sgn_{k-1} \, a_{m_{k-2}m_{k-2}} \times \sum_{\substack{m_{k-1},m_k=1,l_1,\ldots,j \\ m_{k-1},m_k\neq m, m_1, \ldots, m_{k-2} \\ m_k > m_{k-1}}} \left(a_{m_k m_k}^{ed} - a_{m_{k-1}m_{k-1}}^{ed}\right) \cdot \tag{12}$$

where sgn_i, $i = 1, 2, \cdots, k$ denotes the sign with respect to each coefficient. The last summation term of Eq. (12) can be rewritten as:

$$\sum_{\substack{m_{k-1},m_k=1,l_1,\ldots,j \\ m_{k-1},m_k\neq m, m_1, \ldots, m_{k-2} \\ m_k > m_{k-1}}} \left(a_{m_k m_k}^{ed} - a_{m_{k-1}m_{k-1}}^{ed}\right)$$
$$= \sum_{\substack{m_{k-1},m_k=1,l_1,\ldots,j \\ m_{k-1},m_k\neq m, m_1, \ldots, m_{k-2} \\ m_k > m_{k-1}}} \left(a_{m_k m_k}^{ed} - a_{m_k m_k}\right) - \left(a_{m_{k-1}m_{k-1}}^{ed} - a_{m_{k-1}m_{k-1}}\right) \tag{13}$$

Eq. (13) is similar to Eq. (9), so the solution of d subject to $a_{m_k m_k}^{ed} \equiv a_{m_k m_k} \; mod \; R$ and $a_{m_{k-1}m_{k-1}}^{ed} \equiv a_{m_{k-1}m_{k-1}} \; mod \; R$ is a subset solution to Eq. (13). If there is a polynomial time algorithm which can solve Eq. (13) given Eqs. (10) and (11) to find solutions to d, then the same algorithm can be employed to break the RSA algorithm by finding its decryption key as in the proof of Theorem 2. This concludes the second part of the theorem. □

Clearly, from Theorem 2, finding the encryption key d from the extension scheme is as hard as finding it from the RSA scheme. Up to this point, it was assumed that all the information except the decryption key d is known to the attacker. Now assume that the attacker chooses his own messages and sends them to his partner. After receiving the message, his partner chooses the diagonal entries and extra messages and then sends the encrypted messages back to the attacker. Under this condition, the attacker has control only part of the messages and he tries to determine the remaining messages. The security of this scheme under a chosen-plaintext attack on the message can be summarized in the following theorem.

Theorem 3: Under a chosen-plaintext attack on the message, the extension scheme is secure when $f(\cdot)$ is a one-way function and satisfies the following property:

(P1) Let $r \equiv \sum_{i=1}^{n} a_{ii}$ *mod* R. It is infeasible to compute $f(r + I \ mod \ R)$ given $f(r + i \ mod \ R), 0 \leq i < I$.

Proof: In order to analyze this problem, let us first set up two equations:

$$\begin{aligned} a_{11}^{e} &\equiv c_0 + c_1 a_{11} + \cdots + c_{n-1} a_{11}^{n-1} \ mod \ R \\ a_{22}^{e} &\equiv c_0 + c_1 a_{22} + \cdots + c_{n-1} a_{22}^{n-1} \ mod \ R \\ &\cdot \\ &\cdot \\ &\cdot \\ a_{nn}^{e} &\equiv c_0 + c_1 a_{nn} + \cdots + c_{n-1} a_{nn}^{n-1} \ mod \ R \end{aligned} \tag{14}$$

and

$$\begin{aligned} a_{i,i+j}^{(e)} \equiv\ & a_{i,i+j} \cdot g(i,\ i+j) + \sum_{l=1}^{j-1} a_{i,i+l}\, a_{i+l,i+j} \cdot g(i,\ i+l,\ i+j) \\ & + \sum_{l_1=1}^{j-2} \sum_{l_2=l_2+1}^{j-1} a_{i,i+l_1} a_{i+l_1,i+l_2} a_{i+l_2,i+j} \cdot g(i,\ i+l_1,\ i+l_2,\ i+j) \\ & \quad \cdot \\ & \quad \cdot \\ & \quad \cdot \\ & + \sum_{l_1=1}^{1} \sum_{l_2=l_1+1}^{2} \cdots \sum_{l_{j-1}=l_{j-2}+1}^{j-1} a_{i,i+l_1} a_{i+l_1,i+l_2} \cdots a_{i+l_{j-1},i+j} \times \\ & g(i,\ i+l_1,\ \cdots,\ i+l_{j-1},\ i+j) \quad ,\ j=1,2,\ldots,n-i\ . \end{aligned} \tag{15}$$

According to Theorem 1, $g(\cdot)$ is a function of diagonal entries which are constant for a particular set of messages and $a_{i+k_1,i+k_2}$ can be represented as:

$$a_{i+k_1,i+k_2} \equiv f\left(\sum_{i=1}^{n} a_{ii} + L \ mod \ R\right) \oplus m_{i+k_1,i+k_2}\,, \tag{16}$$

where

$$L = i+k_2 - 1 + \frac{2n(i+k_1) - 2n - (i+k_1)^2 + (i+k_1)}{2} + \frac{n(n-1)}{2} \cdot (k-1)$$

for the k^{th} matrix.

Suppose there exists a polynomial time algorithm that can solve Eq. (15) given $m_{i+k_1,i+k_2}$ and uniquely determine the value of $g(\cdot)$. Consequently, the value of $f(\cdot)$ can be calculated. Since $f(\cdot)$ is a one-way function, it is infeasible to compute $\sum_{i=1}^{n} a_{ii} \ mod \ R$ from $f(\cdot)$. Without knowing $\sum_{i=1}^{n} a_{ii} \ mod \ R$, it is impossible to calculate the value of subsequent $f(\cdot)$'s. Thus, the enemy still cannot compute the remaining messages. In general, there is more than one block of known message available. In order to prevent this scheme from being broken under this condition, $f(\cdot)$ has to satisfy a tighter restriction, i.e., Property (P1).

System of equations of Eq. (14) can be added up and represented as

$$\sum_{i=1}^{n} a_{ii} \equiv c_1^{-1} \sum_{i=1}^{n} a_{ii}^{e} - (n \cdot c_0 + c_2 \sum_{i=1}^{n} a_{ii}^{2} + \cdots + c_{n-1} \sum_{i=1}^{n} a_{ii}^{n-1}) c_1^{-1} mod \ R \tag{17}$$

One can break the scheme under this condition if $\sum_{i=1}^{n} a_{ii} \ mod \ R$ is known. So $\sum_{i=1}^{n} a_{ii} \ mod \ R$ can be considered as one variable and Eq. (17) can be rewritten as

$$r \equiv c_1^{-1} \sum_{i=1}^{n} a_{ii}^{e} - c_1^{-1} \cdot k \ mod \ R \tag{18}$$

There are three unknowns r, k, and c_1 in Eqs. (16) and (18). It is practically impossible to enumerate the value of r for testing given $\sum_{i=1}^{n} a_{ii}^{e}$ is fixed and known because there are too many possible combinations. So it is impossible to uniquely determine r from both Eqs. (16) and (18) simultaneously. Thus the scheme is still secure even if there exists a polynomial time algorithm that can uniquely solve for $g(\cdot)$ and $a_{i+k_1,i+k_2}$ from Eq. (16). This concludes the proof of this theorem. □

7. Computational Complexity

The main issue of this section is to show that this scheme has much faster encryption and decryption algorithms than that of the RSA scheme. The encryption and decryption algorithms were discussed in sections 3, 4, and 5. Now the speed with which one can raise the matrix **A** to the e^{th} (for encryption) or to the d^{th} (for decryption) power is considered. The encryption and decryption algorithms include two parts. The first part is to set up the Eq. (2) or Eq. (3) and calculate the coefficients c_i^e and c_i^d. The second part of the algorithm is to calculate the matrix raised to the high power.

A. Precalculation of Coefficients

The decryption algorithm has exactly the same computational complexity as that of the encryption algorithm, so here we only discuss the computational complexity of encryption algorithm. Raising a_{ii} to the e^{th} power requires at most $2\left\lfloor \log_2 e \right\rfloor$ multiplications [4]. Thus, to set up Eq. (2) requires at most

$$2n\left\lfloor \log_2 e \right\rfloor + n(n-2) \tag{19}$$

multiplications. There are two ways to solve Eq. (2). One is using the Cramer's rule and the other is using the Vandermonde algorithm. If Cramer's rule is used to calculate the coefficient c_{ii}^{e}, then this stage needs at most

$$2n\left\lfloor \log_2 e \right\rfloor + n(n-2) + (n+1)\sum_{i=1}^{n-1}\Big(\prod_{j=n-i+1}^{n} j\Big) + 15\left\lfloor \log_{10} R \right\rfloor \tag{20}$$

multiplications and

$$(n+1)\sum_{i=1}^{n-1}\Big[\Big(\prod_{j=n-i}^{n} j\Big)/(n-i+1)\Big] + 10\left\lfloor \log_{10} R \right\rfloor \tag{21}$$

additions. If one observes Eq. (2) carefully, one notices that it is a Vandermonde system, hence the Vandermonde algorithm can be used to calculate the coefficients. This algorithm has a distinct computational advantage when the size of the matrix is large (e.g. $n \geq 7$). This stage needs at most

$$2n\left\lfloor \log_2 e \right\rfloor + n(n-2) + \frac{n(n-1)}{2}\Big(15\left\lfloor \log_{10} R \right\rfloor + 3\Big) \tag{22}$$

multiplications and

$$\frac{n(n-1)}{2}\Big(10\left\lfloor \log_{10} R \right\rfloor + 3\Big) \tag{23}$$

additions.

B. Encryption and Decryption

Eq. (1) shows that one only has to calculate $\mathbf{A}^2$, $\mathbf{A}^3$, $\cdots$, $\mathbf{A}^{n-1}$ and multiply them with the corresponding coefficients and add them together. This stage takes

$$\frac{n(n+1)(n+2)(n-2)}{6} + \frac{n(n+1)(n-1)}{2} \tag{24}$$

multiplications and

$$\frac{n(n+1)(n-1)(n-2)}{6} + \frac{n(n+1)(n-2)}{2} \tag{25}$$

additions. In fact the diagonal entries are fixed for a particular set of messages and the encrypted diagonal entries are not functions of the superdiagonal entries. So only the superdiagonal entries from the second matrix on need to be calculated. Thus this stage can speed up the encryption and decryption even more. In this case, we can reduce the computational complexity to

$$\frac{n(n+1)(n+2)(n-2)}{6} + \frac{n(n-1)^2}{2} \tag{26}$$

multiplications and

$$\frac{n(n+1)(n-1)(n-2)}{6} + \frac{n(n-1)(n-2)}{2} \tag{27}$$

additions.

We now give an example to illustrate the computational advantage of this scheme.

Example 2: Suppose one chooses two large primes p and q such that $R=p\cdot q$ is a 200-digit number, i.e. $\lfloor \log_{10}R \rfloor=200$. We assume that the block lengths of each message are the same in both the RSA scheme and the extension scheme. The RSA scheme needs at most $2\lfloor \log_2 e \rfloor$ multiplications for encryption. The RSA algorithm and the 2X2 matrix can encrypt or decrypt one block of message, and the 3X3 matrix can encrypt 3 blocks of messages each time, respectively. In this example, we compare how efficiently one can encrypt 1 block of message using the RSA algorithm, 2X2 matrix, and 3X3 matrix. The results of the comparison are listed in the following table.

The results show that the extension scheme has more computational advantage with a smaller size and large encryption key over the RSA scheme. □

Table 1. The computational complexity comparison of RSA, 2×2 matrix, and 3×3 matrix.

	RSA	Pre-computation				Post-computation			
		2×2 matrix		3×3 matrix		2×2 matrix		3×3 matrix	
	×	×	+	×	+	×	+	×	+
$\lfloor \log_2 e \rfloor = 50$	100	3205	2003	3390	2020	1	0	$\frac{16}{3}$	$\frac{7}{3}$
$\lfloor \log_2 e \rfloor = 100$	200	3405	2003	3690	2020	1	0	$\frac{16}{3}$	$\frac{7}{3}$
$\lfloor \log_2 e \rfloor = 150$	300	3605	2003	3990	2020	1	0	$\frac{16}{3}$	$\frac{7}{3}$
$\lfloor \log_2 e \rfloor = 200$	400	3805	2003	4210	2020	1	0	$\frac{16}{3}$	$\frac{7}{3}$

8. Conclusion

A new way of extending the RSA algorithm using a triangular matrix and a one-way function was proposed. The security of this scheme has been shown to be equivalent to that of the RSA algorithm under a chosen plaintext attack on the key and a ciphertext only attack. Under a known plaintext attack on the message, the security of this scheme rests on the security of the RSA algorithm as well as the one-way function $f(\cdot)$. The fast encryption and decryption algorithms of this scheme are based on the Cayley-Hamilton theorem. The speed of this algorithm depends on both the dimension of the matrix and the capability of evaluating the one-way function. The most efficient implementation is the 2×2 case in which both encryption and decryption use a single modulo arithmetic multiplication and single evaluation of the one-way function.

In practice, the first block of the message can be transmitted using the RSA scheme and then the remaining message can be encrypted by calculating $f(m_i) \oplus m_{i+1}$. The extension scheme of the RSA algorithm was developed independently from the above system. However, the matrix version of the RSA algorithm turned out to have a similar form as that of the above system.

The criteria for choosing the one-way function is the efficiency of evaluating this function. A question remains as to which one-way function should be chosen.

Does this function has to be a one-way function in order to keep this scheme secure. This topic requires further investigation.

Acknowledgements

We would like to express our sincere gratitude and appreciation to Dr. Don Coppersmith at IBM Thomas J. Watson Laboratory and Dr. Andrew Odlyzko at AT&T Bell Laboratory for their invaluable suggestions during our research.

References

[1] C.C. Chuang, *Matrix Extension of the RSA Algorithm, Ph.D. Dissertation,* SMU, Dallas, Texas, 1990.

[2] J. Gordon, "Strong primes are easy to find," *Proceedings of Crypto'87*, pp. 216-223.

[3] R. A. Horn, and C. A. Johnson, *Matrix Analysis,* Cambridge University Press, NY, 1985.

[4] D. E. Knuth, *The Art of Computer Programming, Vol. 2: Seminumerical Algorithms,* Addison-Wesley, Reading, 2nd ed., 1981.

[5] R.F. Rieden, J.B. Snyder, R.J. Widman, and W.J. Barnard, "A two-chip implementation of the RSA public-key encryption algorithm," *GOMAC* (Government Microcircuit Applications Conference), (Orlando, FL), pp. 24-27 Nov. 1982.

[6] R.L. Rivest, "A description of a single-chip implementation of the RSA cipher," *Lambda,* vol.1, no. 3, pp. 14-18, Fall 1980.

[7] R.L. Rivest, "RSA chips (past/present/future)," *Eurocrypt'84*, pp. 159-165.

[8] R.L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Comm. ACM,* vol. 21, pp. 120-126, 1978.

[9] G.J Simmons, "High speed arithmetic utilizing redundant number systems," *National Telecommunications Conf.* (Houston,TX), pp. 49.3.1-2, Nov.30-Dec. 4, 1980.

[10] V. Varadharajan, and R. Odoni, "Extension of RSA cryptosystems to matrix rings," *Cryptologia,* vol. 9, no. 2, pp. 140-153, April, 1985.

[11] K. Yiu and K. Peterson, "A single-chip VLSI implementation of the discrete exponential public key distribution system," *GOMAC* (Government Microcircuit Applications Conference), (Orlando, FL), pp. 18-23, Nov. 1982.