Cryptanalysis of REESSE1+ Public Key Cryptosystem

Shengli Liu[1], Fangguo Zhang[2]
[1]Dept. of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai 200030, P.R.China
email: liu-sl@cs.sjtu.edu.cn
[2]Dept.of Electronics and Communication Engineering,
Sun Yat-Sen University, Guangzhou 510275, P.R.China
email: isszhfg@mail.sysu.edu.cn

March 12, 2007

Abstract

A new public key cryptosystem, called REESSE1+, was proposed. REESSE1 consists of two primitive algorithms, a public key encryptio/decryption algorithm and a digital signature algorithm. We give some analysis to REESSE1+, and show that the system is totally unsecure. We show how to derive the private key from the public key. As the same time, we also show how to forge signatures for any messages, given two valid signatures.

Key words: REESSE1, digital signature, cryptanalysis

1 Introduction

A public key cryptosystem, named REESSE1+, was recently proposed in [2]. It is a revised version of REESSE1 presented in [1] in 2003. There are two primitive algorithms associated with REESSE1+: an encryption/decryption algorithm and a digital signature algorithm. Cryptoanalysis of REESSE1 was shown in [4, 3]. The aim of this paper is to give an analysis of the newly revised version.

The new version has even longer key size than the old REESSEE1. The analysis of the impractical length of the private/public key, and the complexity of the encryption/decrytpion algorithm of the REESSE1 was given in [4, 3].

In this paper, we show how that the REESSEE1+ encryption/decryption algorithm can be reduced to the REESSEE1. We present algorithms to derive the private key from the public key. We also show how to forge valid signatures for any message with the help of two valid signatures.

We will follow the original symbols used in [2].

2 REESSE1 encryption/decryption algorithm and its analysis

2.1 The Original Description of Encryption/Decryption Algorithm

Key Generation

- d, D, T, S are pairwise coprime integers.

- The pairwise coprime sequence $\{A_1, A_2, \cdots, A_n\}$;
- A prime number M satisfying $M > \prod_{i=1}^{n} A_i$;
- Choose δ such that $\gcd(\delta, M-1)$ and $ord(\delta) = dDT$;
- $W = (\prod_{i=1}^{n} A_i)^{-1} \cdot (\alpha\delta^{-1})^{1/S} \mod M$;
- Compute $l(1), l(2), \cdots, l(n) \in \{i\delta \mod M-1, i = 5, \cdots, n+4\}$;
- Compute $C_i = A_i W^{l(i)} \mod M$, $i = 1, 2, \cdots, n$.

The **public key** is $(\{C_1, C_2, ..., C_n\}, M)$.

The **private key** is $(\{A_1, A_2, ..., A_n\}, \{l(1), l(2), ..., l(n)\}, W, \delta)$.

Encryption Suppose that $F = \{b_1, b_2, ..., b_n\}$ is an n-bit plaintext to be encrypted. The corresponding ciphertext is $\hat{G} \equiv \prod_{i=1}^{n} C_i^{b_i} \mod M$.

Decryption Given a ciphertext $\hat{G} \equiv \prod_{i=1}^{n} C_i^{b_i}$. The decryption procedure works as follows.

step 1 Let $\hat{G} \leftarrow \hat{G}W^{-\delta} \mod M$;

step 2 Initialize the plaintext bits $b_k \leftarrow 0$ for $k = 1, 2, ..., n$.
Let $G \leftarrow \hat{G}$ and $i \leftarrow 1$.

step 3 If $A_i | G$, let $b_i = 1$ and $G \Leftarrow G/A_i$.

step 4 $i \Leftarrow i + 1$. If $i \leq n$ and $G \neq 1$, then goto **step 3**;

step 5 If $G \neq 1$, then goto **step 1**, otherwise end.

2.2 Simplified Description

We have some comments on the algorithms.

- The conditions imposed on the values of W and δ serve for digital signature algorithm. Hence, we can neglect these conditions in the encryption/decryption algorithms.

- To decrypt the ciphertext $\hat{G} \equiv \prod_{i=1}^{n} C_i^{b_i} \equiv W^{\sum_{i=1}^{n} b_i l(i)} \cdot \prod_{i=1}^{n} A_i^{b_i} \mod M$, the decryption algorithm will try to eliminate $W^{\sum_{i=1}^{n} b_i l(i)}$. However, the algorithm does not know the value of $W^{\sum_{i=1}^{n} b_i l(i)}$. It will try
$$\sum_{i=1}^{n} b_i l(i) \delta^{-1} \mod M-1$$
time to eliminate $W^{\delta \sum_{i=1}^{n} b_i l(i) \delta^{-1}}$ from the ciphertext $\hat{G}$, each time multiplying $\hat{G}$ with $W^{-\delta}$. Consequently, the decryption algorithm needs $\left(\delta^{-1} \sum_{i=1}^{n} b_i l(i) \mod M-1\right)$ modular multiplications and $\left(n\delta^{-1} \sum_{i=1}^{n} b_i l(i) \mod M-1\right)$ divisions.

- The values $l(1), l(2), \cdots, l(n) \in \{i\delta \mod M-1, i = 5, \cdots, n+4\}$.

  - Let $l(k) \equiv i\delta \mod M-1$. If $i\delta > M-1$, the time complexity $\left(n\delta^{-1} \sum_{i=1}^{n} b_i l(i) \mod M-1\right)$ for decryption may turn out to be of $O(M)$, which is an exponential time algorithm.

- More precisely, it should require that $\sum_{i=5}^{n+4} i\delta < M-1$, otherwise some plaintexts cannot be recovered by decryption alg. in poly-time.

Consequently, the values of $l(1), l(2), \cdots, l(n)$ can be considered as a random permutation of $\{5\delta, 6\delta, \cdots, (n+4)\delta\}$, to guarantee poly-time decryptions.

Now we give a simplified description of encryption/decryption algorithm of REESSEE+.

Key Generation

- Choose W, δ as before, let $V \equiv W^{\delta} \mod M$.
- Choose pairwise coprime sequence $\{A_1, A_2, \cdots, A_n\}$;
- Choose a prime number M satisfying $M > \prod_{i=1}^{n} A_i$;
- select $\{f(1), f(2), \cdots, f(n)\}$ as a random permutation of $\{5, 6, \cdots, n+4\}$.
- Compute $C_i = A_i V^{f(i)} \mod M$, $i = 1, 2, \cdots, n$.

The **public key** is $(\{C_1, C_2, ..., C_n\}, M)$.

The **private key** is $(\{A_1, A_2, ..., A_n\}, \{f(1), f(2), ..., f(n)\}, V)$.

Encryption the same as before.

Decryption Step 1 is replaced by "$\hat{G} \leftarrow \hat{G}V^{-1} \mod M$".

Remark. The simplified description is just REESSE1, the old version in [1].

3 Analysis of REESSE1+(REESSE1) Encrypion/Decryption Algorithm

3.1 Facts on Which the Attack Algorithms Based

Let $\texttt{Prime}[i]$ denote the i-th prime number. Then $\texttt{Prime}[1] = 2$, $\texttt{Prime}[2] = 3, \cdots$, etc.

Since $\{f(1), f(2), \ldots, f(n)\}$ is a random permutation of $\{5, 6, \ldots, n+4\}$, there must exist triples (i, j, k) such that $f(i) + f(j) = f(k)$. The number of such triples is

$$1 + 2 + \cdots + (n-5) = (n-4)(n-5)/2.$$

More precisely, if $f(k) = 10$, there is one triple; if $f(k) = 11$, there is 2 triples, etc.

When $f(i) + f(j) = f(k)$, we compute

$$Z \equiv C_i \cdot C_j \cdot C_k^{-1} \equiv A_i \cdot A_j \cdot A_k^{-1} \cdot V^{f(i)+f(j)-f(k)} \equiv A_i \cdot A_j \cdot A_k^{-1} \mod M.$$

Then, there must exists an integer l such that

$$\frac{Z}{M} = \frac{l}{A_k} + \frac{A_i \cdot A_j}{A_k \cdot M}. \quad (1)$$

Suppose the continued fraction of rational $\frac{Z}{M}$ is determined by integers $[a_0, a_1, \cdots, a_t]$ with

$$\frac{Z}{M} = a_0 + \cfrac{1}{a_1 + \cfrac{1}{\begin{matrix}\cdots & \cdots \\ \cdots & +\frac{1}{a_{t-1}+\frac{1}{a_t}}\end{matrix}}}.$$

Let $\frac{p_v}{q_v}$ be the rational determined by integers $[a_0, a_1, \cdots, a_v]$ with

$$\frac{p_v}{q_v} = a_0 + \cfrac{1}{a_1 + \cfrac{1}{\begin{matrix}\cdots & \cdots \\ \cdots & +\frac{1}{a_{v-1}+\frac{1}{a_v}}\end{matrix}}}. \tag{2}$$

Then $\{\frac{p_0}{q_0}, \frac{p_1}{q_1}, \cdots, \frac{p_t}{q_t}\}$ is the convergent sequence of continued fraction expansion of $\frac{Z}{M}$.

Theorem 1 *[5] Let α be a real number, and let r/s be a rational with $gcd(r, s) = 1$ and $|\alpha - r/s| < 1/2s^2$. Then r/s is a convergent of the continued fraction expansion of α.*

From Eq.(1), we see that $\frac{A_i \cdot A_j}{A_k \cdot M} < \frac{1}{2A_k^2}$. According to Theorem 1, $\frac{l}{A_k}$ must be a convergent of Z/M. Let $\frac{l}{A_k}$ is the u-th convergent, i.e., $q_u = A_k$ and $p_u = l$, i.e., $\frac{p_u}{q_u} = \frac{l}{A_k}$. Then we know that

$$|\frac{Z}{M} - \frac{p_{u+1}}{q_{u+1}}| < \frac{A_i \cdot A_j}{A_k \cdot M} = \frac{1}{2\left(\sqrt{\frac{A_k M}{2A_i A_j}}\right)^2}.$$

According to Theorem 1 and convergence of sequence $\{\frac{p_0}{q_0}, \frac{p_1}{q_1}, \cdots, \frac{p_t}{q_t}\}$, we obtain that

$$q_{u+1} \geq \sqrt{\frac{A_k M}{2A_i A_j}} = A_k \cdot \sqrt{\frac{M}{2A_i A_j A_k}}. \tag{3}$$

Fact 1 If $f(i) + f(j) = f(k)$,

- **Fact 1.1** there exists a q_u such that $q_u = A_k$ in $\{\frac{p_0}{q_0}, \frac{p_1}{q_1}, \cdots, \frac{p_t}{q_t}\}$, the convergent sequence of continued fraction expansion of $\frac{Z}{M}$ with $Z \equiv C_i C_j C_k^{-1} \mod M$.
- **Fact 1.2** there is sharp increase from q_u to q_{u+1} since $q_{u+1} \geq \sqrt{\frac{A_k M}{2A_i A_j}}$.
- **Fact 1.3** Due to Fact 1.2, there is also a sharp increase from a_u to a_{u+1}, since $q_{v+1} = a_{v+1} q_v + q_{v-1}$ for $v = 1, 3, \cdots, t$. Here a_vs are items of Z/M determined by Eq. (2).

Fact 2 If the tuple (i, j, k, A_k) satisfies $f(i) + f(j) = f(k)$, we call it a **valid tuple**. Otherwise **invalid tuple**.

- **Fact 2.1** There are totally $(n - 5)(n - 4)/2$ valid tuples.
- **Fact 2.2** If we classified the output tuples (i, j, k, A_k) according to the value of $f(k)$, we have the following distribution.

$f(k)$	10	11	$\cdots$	$n+4$
number of tuples (i,j,k,A_k)	1 tuple	2 tuples	$\cdots$ $\cdots$	$n-5$ tuples

Table 1: distribution of tuples (i,j,k,A_k)

Fact 3 The maximal value of A sequence is up bounded by

$$\max\{A_1, A_2, \cdots, A_n\} < \frac{M}{\prod_{i=1}^{n-1} \texttt{Prime}[i]}$$

Fact 4 Let integer m satisfies

$$\prod_{i=1}^{m+1} \texttt{Prime}[i] \geq p,$$

but

$$\prod_{i=1}^{m} \texttt{Prime}[i] < p.$$

Then

$$\max_{i,j,k\in\{1,2,\cdots,n\}} A_i \cdot A_j \cdot A_k < \prod_{i=n-2}^{m} \texttt{Prime}[i].$$

According to Eq. (3), we have

$$q_{u+1} > q_u \cdot \sqrt{\frac{M}{2\prod_{i=n-2}^{m} \texttt{Prime}[i]}}.$$

3.2 Breaking RESSEE1+ Public Key Encryption/Decryption Cryptosystems

We break the public key encryption/decryption algorithm by deriving private key from the public key. There are two algorithms: **Alg1.** is to find valid tuples and **Alg2.** to derive the private key.

Alg.1: Finding Valid Tuples.

Input the public key $(\{C_1, C_2, ..., C_n\}, M)$;

Output tuples (i,j,k,A_k);

```
1. Let $\Delta = \sqrt{\frac{M}{2\prod_{i=n-2}^{m} \texttt{Prime}[i]}}$;
       Let maxA= $\frac{M}{\prod_{i=1}^{n-1} \texttt{Prime}[i]}$.
2. For $(i = 1, i <= n; i++)$
       For $(j = 1; j <= n; j++)$
       For $(k = 1; k <= n; k++)$
       { $Z \equiv C_i \cdot C_j \cdot C_k^{-1} \mod M$;
         Compute the convergent sequence of continued fraction of $Z/M$, and get
             $\left\{\frac{p_0}{q_0}, \frac{p_1}{q_1}, \ldots, \frac{p_t}{q_t}\right\}$;
```

```
    The denominators of convergent items constitute sequence $\{q_1, q_2, \ldots, q_t\}$;
    For($l = 1; l <= t; l++$)
    If (($q_l \cdot \Delta < q_{l+1}$))&&($q_l <$ maxA)
      then { Let $A_k = q_l$;
             Output $(i, j, k, A_k)$; }
  }
```

We cannot give precise estimations of q_{u+1}/q_u and the maximal value of A sequence. We use Δ as a lower bound of q_{u+1}/q_u and maxA as a up bound of the maximal value of A sequence. However, these two bounds are far from being tight. Consequently, **Alg. 1** output tuples more than $(n-4)(n-5)/2$, among which are valid tuple (i, j, k, A_k) with $f(i) + f(j) = f(k)$ and invalid tuple with $f(i) + f(j) \neq f(k)$. However, all $(n-4)(n-5)/2$ valid tuple must be among the output of the algorithm.

Nevertheless, we will use the following properties to pick up the $(n-4)(n-5)/2$ valid tuples and use the properties of the valid tuples to determine the private key $\{A_1, A_2, \cdots, A_n\}$, V and $\{f_1, f_2, \cdots, f_n\}$.

Property 1 If (i, j, k, A_k) is a valid tuple, then

$$A_i A_j \equiv C_i C_j A_k C_k^{-1} \mod M.$$

Property 2 If (i, j_1, k_1, A_{k_1}) and (i, j_2, k_2, A_{k_2}) are both valid,

$$\begin{aligned} A_i &= \gcd\left(C_i C_{j_1} A_{k_1} C_{k_1}^{-1} \mod M, \quad C_i C_{j_2} A_{k_2} C_{k_2}^{-1} \mod M\right) \\ A_{j_1} &\equiv C_i C_{j_1} A_{k_1} (C_{k_1} A_i)^{-1} \mod M \\ A_{j_2} &\equiv C_i C_{j_2} A_{k_2} (C_{k_2} A_i)^{-1} \mod M \end{aligned}$$

Property 3 Among all the invalid tuples output by **Alg. 1**, there are at most two tuples (i, j, k, A_k) and (j, i, k, A_k) associated with an invalid A_k, whose value is not correct due to the invalidity of the tuple (i, j, k, A_k).

Property 4 If all the valid tuples output by **Alg. 1** are classified by the different value of $f(k)$, the distribution of tuples is just like that in Table 1.

Property 5 If the two valid tuples (i_1, j_1, k_1, A_{k_1}) and (i_2, j_2, k_2, A_{k_2}) satisfy that $f(k_1) + 1 = f(k_2)$, then

$$V \equiv C_{k_2} \cdot A_{k_1} \cdot (C_{k_1} \cdot A_{k_2})^{-1} \mod M.$$

Alg.2. Picking up Valid Tuples to Derive Private Key.

Input tuples (i, j, k, A_k) output by **Alg.1**;

Output $(\{A_1, A_2, ..., A_n\}, \{f(1), f(2), ..., f(n)\}, V)$.

1. Classify all the tuples (i, j, k, A_k) according to the value of A_k.

Count the number of tuples associated with A_k, and denoted the number by N_k.

2. If there exists a unique A_k such that $N_k = l$, then set $f(k) = l + 9$.

Mark all the tuples associated with A_k valid.

Mark other tuples associated with $\widetilde{A_k}$, with $\widetilde{A_k} \neq A_k$, invalid.

3. Among all the tuples,

Repeat

(1) search two valid tuples (i_1, j_1, k_1, A_{k_1}) and (i_2, j_2, k_2, A_{k_2}) such that $i_1 = i_2$ or $j_1 = j_2$;
Without loss of generality, we assume that $i = i_1 = i_2$.

(2) compute

$$\begin{aligned} A_i &= \gcd\left(C_i C_{j_1} A_{k_1} C_{k_1}^{-1} \mod M, \quad C_i C_{j_2} A_{k_2} C_{k_2}^{-1} \mod M\right) \\ A_{j_1} &\equiv C_i C_{j_1} A_{k_1} (C_{k_1} A_i)^{-1} \mod M \\ A_{j_2} &\equiv C_i C_{j_2} A_{k_2} (C_{k_2} A_i)^{-1} \mod M \end{aligned}$$

(2) Mark all the tuples associated with A_i, A_{j_1}, A_{j_2} valid.
Mark other tuples associated with $\widetilde{A_i}$ with $\widetilde{A_i} \neq A_i$ invalid. Do the same to $\widetilde{A_{j_1}}, \widetilde{A_{j_2}}$ with $\widetilde{A_{j_1}} \neq A_{j_1}, \widetilde{A_{j_2}} \neq A_{j_2}$.

(3) Set $f(i) = N_i + 9$, $f(j_1) = N_{j_1} + 9$, $f(j_2) = N_{j_2} + 9$.

Until all valid tuples are searched.

3. If there are still tuples with undetermined validity,

Repeat

(1) search a valid tuple (i_1, j_1, k_1, A_{k_1}) and an undetermined tuple (i_2, j_2, k_2, A_{k_2}) with $i_1 = i_2$ or $j_1 = j_2$;
Without loss of generality, we assume that $i = i_1 = i_2$.

(2) If

$$\gcd\left(C_i C_{j_1} A_{k_1} C_{k_1}^{-1} \mod M, \quad C_i C_{j_2} A_{k_2} C_{k_2}^{-1} \mod M\right) = 1$$

then (i_2, j_2, k_2, A_{k_2}) is invalid, Mark all the tuples associated with A_{k_2} invalid.

Otherwise it is valid and set

$$A_i = \gcd\left(C_i C_{j_1} A_{k_1} C_{k_1}^{-1} \mod M, \quad C_i C_{j_2} A_{k_2} C_{k_2}^{-1} \mod M\right),$$

$$A_{j_2} = C_i C_{j_2} A_{k_2} (C_{k_2} A_i)^{-1} \mod M$$

Mark all the tuples associated with A_i valid.
Mark other tuples associated with $\widetilde{A_i}$ with $\widetilde{A_i} \neq A_i$ invalid.

Until all the $(n-5)(n-4)/2$ valid tuples are marked.

3. Search a valid tuple (i, j, k, A_k) satisfies $f(k) = 10$ and $i = j$, then set $f(i) = 5$.

Search valid tuple (i, j, k, A_k) satisfies $f(k) = 10 + t$ (t is a positive integer) and $f(i) = 5$, then set $f(j) = 5 + t$.

4 Output all A_ks, $f(k)$s, and V.

3.3 Example

Let $n = 10$, $V = 709863737651593824387533$; $M = 1640976313637848358971801$;
$f[1] = 10, f[2] = 13, f[3] = 9, f[4] = 14, f[5] = 6,$
$f[6] = 8, f[7] = 7, f[8] = 12, f[9] = 5, f[10] = 11;$
$A[1] = 9, A[2] = 253, A[3] = 323, A[4] = 205, A[5] = 1369,$
$A[6] = 3481, A[7] = 4, A[8] = 2809, A[9] = 2263, A[10] = 49;$

$C[1] = 656980308978034175699516, C[2] = 529118527878261775263063,$
$C[3] = 1117492693060345271717610, C[4] = 1009005619984027518080917,$
$C[5] = 407140262259854747498280, C[6] = 919158732131835174270358,$
$C[7] = 197336528727655645732846, C[8] = 480167833213793003341972,$
$C[9] = 635798888164869683821836, C[10] = 651849566821592027079423;$

The **Alg.1** output 30 tuples.

We classify the 30 tuples according to the values of A_ks in Table 2.

Now we will use **Alg.1** to pick up the 15 valid ones and derive the private key.

A_k	Tuples (i, j, k)
$A_4 = 205$	**(3,9,4) (5, 6, 4)(9, 3, 4)(6, 5, 4)(7, 7, 4)**
$A_2 = 253$	**(5, 7, 2) (6, 9, 2) (7, 5, 2) (9, 6, 2)**
$A_{10} = 1894$	(6, 9, 10) (9, 6, 10)
$A_8 = 2809$	**(7, 9, 8) (9, 7, 8) (5,5,8)**
$A_{10} = 6957$	(9, 7, 10) (7, 9, 10)
$A_4 = 3$	(8, 3, 4) (3, 8, 4)
$A_{10} = 49$	**(9, 5, 10) (5, 9, 10)**
$A_{10} = 53022327$	(3, 4, 6) (4, 3, 6)
$A_6 = 4471789987666990$	(3, 5, 6) (5, 3, 6)
$A_4 = 152391460756$	(7, 8, 4) (8, 7, 4)
$A_4 = 16127$	(7, 10, 3) (10, 7, 3)
$A_1 = 9$	**(9, 9, 1)**
$A_6 = 1572955621791218$	(5, 5, 6)

Table 2: Distribution of tuples (i, j, k, A_k) by the algorithm

- $A_4 = 205$ must be correct, since only it has 5 tuples. Hence we know that $f(4) = 14$. The validity of $A_4 = 205$ invalid the rows for $A_4 = 3$ and $A_4 = 152391460756$.

- A_2 =253 must be correct, since only it has 4 tuples. Hence $f(2) = 13$.

- $A_8 = 2809$ must be correct, since only it has 3 tuples. Hence $f(8) = 12$.

- Recover $V \equiv C_4 \cdot A_2 \cdot (C_2 \cdot A_4)^{-1} \equiv 709863737651593824387533 \mod M$.

- From valid tuples $(3, 9, 4, A_4)$ and $(6, 9, 2, A_2)$, we have

$$A_3 \cdot A_9 \equiv C_3 \cdot C_9 \cdot A_4 \cdot C_4^{-1} \equiv 730949 \mod M,$$

$$A_6 \cdot A_9 \equiv C_6 \cdot C_9 \cdot A_2 \cdot C_2^{-1} \equiv 7877503 \mod M.$$

  Then $A_9 = \gcd(730949, 7877503) = 2263$. Consequently $A_3 = 730949/2263 = 323$ and $A_6 = 7877503/2263 = 3481$. This invalidates the rows for $A_6 = 4471789987666990$ and $A_6 = 1572955621791218$ in the table.

- From the valid tuple $(5, 6, 4, A_4)$, we have

$$A_5 \cdot A_6 \equiv C_5 \cdot C_6 \cdot A_4 \cdot C_4^{-1} \equiv 4765489 \mod M.$$

  $A_5 = 4765489/A_6 = 1369$.

- From the valid tuple $(5, 7, 2, A_2)$, we have

$$A_5 \cdot A_7 \equiv C_5 \cdot C_7 \cdot A_2 \cdot C_2^{-1} \equiv 5475 \mod M.$$

  $A_7 = 5476/A_5 = 4$.

- Now test whether $(9, 9, 1, 9)$ is valid or not. If it is valid, then

$$A_9^2 \equiv C_9^2 \cdot A_1 \cdot C_1^{-1} \equiv 5121169 \mod M.$$

  $A_9 = 2263$ implies $A_9^2 = 5121169$, hence it is valid and $A_1 = 9$.

- Now test whether $(9, 5, 10, 49)$ is valid or not. If it is valid, then

$$A_9 \cdot A_5 \equiv C_9 \cdot C_5 \cdot A_{10} \cdot C_{10}^{-1} \equiv 3098047 \mod M.$$

  $A_9 = 2263$ and $A_5 = 1369$ implie $A_9 \cdot A_5 = 3098047$. Hence it is valid and $A_{10} = 49$. This invalidates the rows for $A_{10} = 4471789987666990$, $A_{10} = 6957$ and $A_{10} = 1894$ in the table.

- The number of valid tuples in the valid rows in Table 1 shows that $f(4) = 14$, $f(2) = 13$, $f(8) = 12$, $f(7) = 11$, $f(1) = 10$.

- $f(1) = 10$ and valid tuple $(9, 9, 1, A_1)$ shows that $f(9) = 5$.
  From valid tuple $(9, 5, 10, A_{10})(5, 9, 10, A_{10})$, we know that $f(5) = 6$;
  From valid tuple $(7, 9, 8, A_8)$, we know that $f(7) = 7$.
  From valid tuple $(6, 9, 2, A_2)$, we know that $f(6) = 8$.
  From valid tuple $(3, 9, 4, A_4)$, we know that $f(3) = 9$.

Now we totally recover the private key $(\{A_1, A_2, ..., A_n\}, \{f(1), f(2), ..., f(n)\}, V)$ from the public key $(\{C_1, C_2, ..., C_n\}, M)$.

4 REESSE1+ Digital Signature Algorithm and the Forging Algorithm

Let us review the parameters in the signature algorithm.

- d, D, T, S are pairwise coprime integers.
- The pairwise coprime sequence $\{A_1, A_2, \cdots, A_n\}$;
- A prime number M satisfying $M > \prod_{i=1}^{n} A_i$, $dDT|(M-1)$ and $i|(M-1)$ for $i = 1, 2, \cdots, n+4$;
- Choose δ such that $\gcd(\delta, M-1)$ and $ord(\delta) = dDT$;
- $W = (\prod_{i=1}^{n} A_i)^{-1} \cdot (\alpha\delta^{-1})^{1/S} \mod M$,
  $\alpha = \delta^{\delta^n} \mod M$, $\beta = \delta^{(\delta+1)WS} \mod M$,$\gamma = \delta^{W^n} \mod M$;
- Compute $l(1), l(2), \cdots, l(n) \in \{i\delta \mod M-1, i = 5, \cdots, n+4\}$;
- Compute $C_i = A_i W^{l(i)} \mod M$, $i = 1, 2, \cdots, n$.

Signing key: $\{A_1, A_2, \cdots, A_n\}, \{l_1, l_2, \cdots, l_n\}, W, \delta, D, d$;

Verification key: $\{C_1, C_2, \cdots, C_n\}, \alpha, \beta, \gamma$;

4.1 Signing

Suppose that F is the message to be signed. Let $hash(\cdot)$ be a proper one-way hash function.

The signer will use his signing key $\{A_i\}, \{l_i\}, W, \delta, D, d$ and public parameters M to sign message $F = (b_1, b_2, \cdots, b_n)$ in the following way.

Signing process(according to [1])

1. Compute $H = hash(F)$.
2. Let $k_1 = \sum_{i=1}^{n} b_i l(i)$, $G_0 = \prod_{i=1}^{n} A_i^{\overline{b_i}}$, where $\overline{b_i} = 1 - b_i$.
3. Pick Q such that

$$D|(\delta Q - W) \tag{4}$$

$$d \nmid ((SQ)^n - W^n) \mod M-1 \tag{5}$$

   Compute R such that $Q \equiv (RG_0)^S H\delta \mod M$

4. $U = \left(RW^{k_1-1}\delta^{\delta(\delta+1)}\right)^{QT} \mod M.$
   If

$$d \nmid \left((\delta+1)SU + \sum_{i=0}^{n-1}(\delta Q)^{n-1+i}W^i\right) \mod M-1, \tag{6}$$

   go to 3.

Then the signature for F is Q, U.

Since $R = (Q/H)^{\frac{1}{S}} G_0^{-1} \delta^{-\frac{1}{S}}$, we re-describe the signing algorithm as follows.

1. $H = hash(F)$.

2. Choose Q satisfying

$$D | (\delta Q - W) \tag{7}$$

$$d \nmid ((SQ)^n - W^n) \mod M - 1 \tag{8}$$

3. $U = \left((Q/H)^{\frac{1}{S}} G_0^{-1} \delta^{-\frac{1}{S}} W^{k_1 - 1} \delta^{\delta(\delta+1)} \right)^{QT} \mod M$. If U satisfies

$$d | \left((\delta + 1)SU + \sum_{i=0}^{n-1} (\delta Q)^{n-1+i} W^i \right) \mod M - 1, \tag{9}$$

   output (F, Q, U), otherwise goto 2.

As was pointed by [2], the step 2 and 3 will repeat d time on average.

4.2 Verification

With the public key $\{C_i\}, \alpha, \beta, \gamma$ and the public parameters S, T, M the verifier can verify whether (F, Q, U) is valid or not.

Verification process(according to [2])

1. Compute $H = hash(F)$, and let $H = (b_1, b_2, \cdots, b_n)$ be a binary string of length n.

2. Compute $\hat{G} \equiv \prod_{i=1}^{n} C_i^{b_i} \mod M$.

3. Compute $X \equiv \left(\alpha H Q^{-1}\right)^{QUT} \alpha^{Q^n T} \mod M$,
   $Y \equiv \left(\hat{G}^{QT} U^{-1}\right)^{US} \beta^{UT} \gamma^{T} \mod M$.

4. if $X = Y$, accept (F, Q, U) as a valid signature; otherwise reject.

5 Forging Valid Signatures without the Signing Key

We show some basic facts about the signature scheme.

Fact 1 Any triple (F, Q, U) is a valid signature triple, as long as Eq.(7) Eq.(8) and Eq.(9) are satisfied.

- For a random Q, Eq.(8) is satisfied with probability $(d-1)/d$.

Fact 2 For any valid signature triple (F, Q, U), the signing part Q is not related to the message F and it satisfies Eq.(7) and Eq.(8).

Fact 3 For any valid signature triple (F, Q, U), the signing part U is uniquely determined by the Q, F and the secret $(\frac{\delta^{\delta(\delta+1)-1/S}}{GW})^{QT}$.
And U satisfies Eq.(9) with probability $1/d$.

5.1 About Fact 3

A valid signature triple (F, Q, U) implies

$$U \equiv \left((Q/H)^{\frac{1}{S}} G_0^{-1} \delta^{-\frac{1}{S}} W^{k_1-1} \delta^{\delta(\delta+1)}\right)^{QT} \mod M,$$

where $F = (b_1, b_2, \cdots, b_n)$, $H = hash(F)$ and $G_0 = \prod_{i=1}^{n} A_i^{\overline{b_i}}$ with $\overline{b_i} = 1 - b_i$.

- $G_0 = \prod_{i=1}^{n} A_i^{\overline{b_i}}$. Let $G_1 = \prod_{i=1}^{n} A_i^{b_i}$;
- Let $G = \prod_{i=1}^{n} A_i$ and $\widehat{G} \equiv \prod_{i=1}^{n} C_i^{b_i} \mod M$;
- We have $G = G_0 G_1$ and $\widehat{G} \equiv G_1 W^{k_1} \mod M$.

Since $R \equiv (Q/H)^{1/S} \cdot G_0^{-1} \cdot \delta^{-\frac{1}{S}} \mod M$,

$$\begin{aligned}
U &\equiv \left(RW^{k_1-1}\delta^{\delta(\delta+1)}\right)^{QT} \equiv \left((Q/H)^{\frac{1}{S}} G_0^{-1} \delta^{-\frac{1}{S}} W^{k_1-1} \delta^{\delta(\delta+1)}\right)^{QT} \mod M \\
&\equiv \left((Q/H)^{\frac{1}{S}} \frac{G_1}{G_0 G_1} W^{k_1-1} \delta^{\delta(\delta+1)-1/S}\right)^{QT} \equiv \left((Q/H)^{\frac{1}{S}} \frac{G_1 W^{k_1}}{GW} \delta^{\delta(\delta+1)-1/S}\right)^{QT} \mod M \\
&\equiv \left((Q/H)^{\frac{1}{S}} \frac{\hat{G}}{GW} \delta^{\delta(\delta+1)-1/S}\right)^{QT} \equiv \left((Q/H)^{\frac{1}{S}} \hat{G}\right)^{QT} \left(\frac{\delta^{\delta(\delta+1)-1/S}}{GW}\right)^{QT} \mod M
\end{aligned}$$

Hence we have $\left(\frac{\delta^{\delta(\delta+1)-1/S}}{GW}\right)^{QT} \equiv \frac{U}{((Q/H)^{S^{-1}} \hat{G})^{QT}} \mod M$

5.2 The Forging Algorithm

From the above facts, we know that as long as we can find

- a Q' satisfying Eq.(7),
- and the secret information $\left(\frac{\delta^{\delta(\delta+1)-1/S}}{GW}\right)^{Q'T}$,

we can uniquely determine a U' with F', Q' and the secret information, such that (F', Q', U') is a valid signature with probability $(d-1)/d^2$.

Now we show how to forge signatures for any message F' without the signer's private key, but with help of two valid signature triple (F_1, Q_1, U_1) and (F_2, Q_2, U_2).

Forging a signature Q', U' for message F'. Let $F' = (b_1', b_2', \cdots, b_n')$.

Input Two valid signatures (F_1, Q_1, U_1) and (F_2, Q_2, U_2) with $Q_1 \neq Q_2$.

Output A valid signature (F', Q', U').

(1) Compute Q': $Q' = Q_1 + v(Q_1 - Q_2) = (v+1)Q_1 - vQ_2$, where v is an integer.

(2) Evaluate $\left(\frac{\delta^{\delta(\delta+1)-1/S}}{GW}\right)^{Q'T}$: From (F_1, U_1, V_1), we determine the secret $\left(\frac{\delta^{\delta(\delta+1)-1/S}}{GW}\right)^{Q_1T}$ with

$$\left(\frac{\delta^{\delta(\delta+1)-1/S}}{GW}\right)^{Q_1T} \equiv \frac{U_1}{((Q_1/H_1)^{S^{-1}}\hat{G}_1)^{Q_1T}} \mod M.$$

From (F_2, U_2, V_2), we determine the secret $\left(\frac{\delta^{\delta(\delta+1)-1/S}}{GW}\right)^{Q_2T}$ with

$$\left(\frac{\delta^{\delta(\delta+1)-1/S}}{GW}\right)^{Q_2T} \equiv \frac{U_2}{((Q_2/H_2)^{S^{-1}}\hat{G}_2)^{Q_2T}} \mod M.$$

Then we have

$$\left(\frac{\delta^{\delta(\delta+1)-1/S}}{GW}\right)^{Q'T} = \left(\frac{\delta^{\delta(\delta+1)-1/S}}{GW}\right)^{(v+1)Q_1T} \left(\frac{\delta^{\delta(\delta+1)-1/S}}{GW}\right)^{-vQ_2T}$$

(3) Compute U': Let $\hat{G}' \equiv \prod_{i=1}^{n} C_i^{b'_i} \mod M$.

(i) $U' \equiv \left((Q'/H')^{\frac{1}{S}}\hat{G}'\right)^{Q'T} \left(\frac{\delta^{\delta(\delta+1)-1/S}}{GW}\right)^{Q'T} \mod M$.

(ii) Compute $X' \equiv \left(\alpha H' Q'^{-1}\right)^{Q'U'T} \alpha^{Q'^{n}T} \mod M$,

$Y \equiv \left(\hat{G}'^{Q'T} U'^{-1}\right)^{U'S} \beta^{U'T}\gamma^{T} \mod M$.

if $X' = Y'$, output (F', Q', U'); otherwise goto (2).

Here we give a brief explanation for the validity of the forged signature (F', Q', U').

- $Q' = Q_1 + v(Q_1 - Q_2) = (v+1)Q_1 - vQ_2$ satisfies Eq.(7);
  From the validity of (F_1, Q_1, U_1) and (F_2, Q_2, U_2), it follows that
  $D|(\delta Q_1 - W), D|(\delta Q_2 - W) \Rightarrow D|\delta(Q_1 - Q_2) \Rightarrow D|v\delta(Q_1 - Q_2) \Rightarrow D|\delta v(Q_1 - Q_2) + \delta Q_1 - W$.
  Since $V' = v(Q_1 - Q_2) + Q_1$, it follows that

  $$D|(\delta Q' - W).$$

  On the other hand, Q' satisfies Eq.(8) with probability $1 - 1/d$.

- U' is uniquely determined by Q', F', and it satisfies Eq.(9) with probability $1/d$.

- Then (F', Q', U') is valid signature with probability $(d-1)/d^2$. Invalid triples (F', Q', U') are excluded by testing whether $X = Y$ holds. Consequently, on average the forging algorithm outputs a valid signature (F', Q', U') by repeating step (2) and (3) about $d^2/(d-1)$ times. The computation complexity of forging a valid signature corresponds the signing procedure of RESSEE1+.

6 Conclusion

This paper gives some analysis of REESSE1+ public key algorithm. We point out that REESSE1+ is not secure at all. The encryption scheme can be reduced to the old version REESSE1. Regarding to REESSE1, we show that the private key can be derived from the public key. On the other hand, the digital signature algorithm of REESSE1+ is not secure as well. Every one can make use of two known valid signature to forge new signatures for any messages.

References

[1] S. Su, *The REESSE 1 Public Key Cryptosystm*. Computer Engineering & Science, pp.13-16, Vol. 25, No. 5, 2003.

[2] S. Su, and S. Lü, *The REESSE1+ Public-key Cryptosystem*, http://eprint.iacr.org/2006/420

[3] Liu Shengli, Zhang Fangguo, Chen Kefei, *Crypatanalysis of REESSE1 Public Encryption Cryptosystem*, China Information Security, No. 7, 2005.

[4] Liu Shengli, Zhang Fangguo, Chen Kefei, *Crypatanalysis of REESSE1 Digital Signature Algorithm*, CCICS 2005, Xi'an, China.

[5] Kenneth H. Rosen. Elementary Number Theory and its application. 2004, p. 460.