An RSA with Customized Public Key

Hung-Min Sun*, Cheng-Ta Yang** and Vincent S. Tseng**

*Department of Computer Science
National Tsing Hua University
Email: hmsun@cs.nthu.edu.tw

**Department of Computer Science and Information Engineering
National Cheng Kung University
Email: zadayang@ismail.csie.ncku.edu.tw

***Abstract*— To speed up RSA operation, one tempt is to use short exponents. Recently, Sun and Yang designed a new variant RSA with balanced prime factors and trade-off exponents. Using Sun and Yang's method, it is successful in designing variants of RSA in which the public and private exponents can be chosen obviously smaller than those in typical RSA. However, their RSA variants don't provide the use of public key of special form 2^X+1, which can further reduce the encryption cost. In this paper, we design a new RSA variant which not only remains the same property as the previous RSA variants, but also provides that the public exponent can be the special form of $2^X + 1$.**

Keyword: RSA, Short Exponent Attack.

I. Introduction

RSA[15] was first announced in the August 1977 issue of Scientific American by Rivest, Shamir, and Adleman. It is most commonly used for providing secrecy and guaranteeing authenticity of data. Today RSA is the most widely deployed public key cryptosystem. RSA security solutions promote e-business by providing secure foundations. These solutions include identity management, access management, secure mobile and remote access, secure enterprise access, and secure transactions. One of the main drawbacks in using RSA public-key cryptosystem is that encryption/decryption and signature/verification are really slow. It is due to the modular exponentiation involved.

Over the past decades, a considerable number of studies have been made on accelerating RSA operation, including employing a short secret exponent. Since the time taken in modular exponentiation is linear to the length of the private exponent, a shorter private exponent can improve performance of decryption and signature. Many variants of RSA are designed to speed up RSA decryption and signature, like batch RSA[9], Multi-factor RSA[7][21], ... and so on. Unfortunately, there are many short exponent attacks on RSA in many literatures[4][8][11][22][23][24]. In 1990, Wiener[24] first demonstrated that the secret exponent $d < N^{0.25}$ (typically N is 1024 bits) can be revealed in polynomial time by using a continued fractions algorithm. This means RSA modulus N is factorable in polynomial time if the private exponent is too short. Verheul and Tilborg[22] showed an extension of Wiener's attack that breaks up RSA cryptosystem when d is a few bits longer than $\frac{1}{4}\log_2 N$, it is possible to expose d in less time than an exhaustive search; however, their algorithm requires exponential time when $d > N^{0.25}$. In 1999, Boneh and Durfee[4] pointed out that an attacker can recover the private RSA key given the public key in polynomial time, if $d < N^{0.292}$. At Asiacrypt '99, Sun, Yang and Laih proposed three variants of RSA using a short secret exponent. Their first RSA variant is an attempt to make the private exponent d shorter than Wiener's bound[24] and Boneh and Durfee's bound[4]. In their second RSA variant, they select slightly unbalanced prime factor p and q to generate the key pair in which both e and d are small. Their third RSA variant allows a trade-off between the sizes of e and d to rebalance the computation cost between encryption and decryption. At Asiacrypt'00[8], Durfee and Nguyen showed that unbalanced RSA in which RSA modulus $N = pq$ is the product of two primes of different size actually improves the attacks on short secret exponent. In other words, RSA variants with unbalanced p and q are more insecure than those with balanced p and q. Recently, Sun and Yang[17] proposed two new RSA variants to improve Sun-Yang-Laih's[18] RSA variants. Their first scheme consists of balanced prime factors and balanced exponents that is more secure than RSA variant with unbalanced prime factors and balanced exponents. Their second scheme is a common case of the first scheme, which allows balanced prime factors and trade-off exponent, $\log_2 e + \log_2 d \approx \log_2 N + l_k$, where $l_k = 112$ is a predetermined constant. Using Sun and Yang's method[17], it is possible to have the public and private exponents which can be chosen significantly smaller than in typical RSA. However, their RSA variants don't provide to use the public key of special form $2^X + 1$, which can further reduce the encryption cost. We would concentrate our attention on RSA public key of special form $2^X + 1$ in this paper.

It is commonly known that the computation of RSA cryptosystem is based on modular exponentiation. The modular exponentiation algorithm[6] scans the bits of the exponent from left to right. A squaring is performed at each step, and depending on the scanned bit value, a subsequent multiplication is performed. So, the number of squaring is the bits of

the exponent and the number of multiplications is the same as the number of ones in the scanned bit. Generally, the use of low Hamming weight exponents have been regarded as a computational tool. Considering the computational efficiency, one often uses a special form of $2^X + 1$, which will help decrease the number of modular multiplications. A special number of $2^X + 1$ requires X squaring and 2 multiplications, whereas a random chosen number, which is represented to a $(X + 1)$-bit binary number with Y ones, requires X squaring and Y multiplications.

In this paper, we focus on designing RSA variants which not only keep the same property as the Sun and Yang proposed RSA variants[17], but also provide that the public exponent can be the special form of $2^X + 1$. We establish an RSA variant whose private exponent is of 512 bits (typically N is of 1024 bits) fitting Boneh and Durfee's suggestion[4] on the size of private exponent, and the public exponent is of the form $2^X + 1$, where X is an integer. As a result, we generate a special-made key pair on RSA in which the public key e is $2^{623} + 1$, the private key d is of 512 bits, and the factors p and q are about of 512 bits respectively. Compared with RSA with CRT decryption (RSA-CRT for short), our method can be applied to entity authentication of imbalanced network securely and efficiently, while RSA-CRT can not.

This paper is organized as follows. In Section 2, we design a new RSA variant to provide that the public exponent can be the special form of 2^X+1; moreover, we analyze the security of the scheme and estimated its computational load. Section 3 shows the experimental results of our implementations for our proposed algorithm. Finally, we conclude this paper in Section 4.

II. New RSA Variant with Special Public Exponent Form $2^X + 1$

A. The Proposed Algorithm

In this section, we would design a special variant RSA including a devisable key pair whose public exponent can be a special form $2^X + 1$ by extending the Sun and Yang's method[17]. Our scheme can choose a particular public key $e = 2^{623} + 1$, and generate a corresponding private key to keep in secret. Our scheme is also based on the Extended Euclidean algorithm[10]. Here, briefly recall that given two integers $a, b > 1$, if $\gcd(a, b) = 1$, we can find a unique pair (u_h, v_h) satisfying $au_h - bv_h = 1$, where $(h-1)b < u_h < hb$ and $(h-1)a < v_h < ha$, for any integer $h \geq 1$.

Our method is described as follows:

- **Scheme** : Here, we choose a public key with low Hamming weight exponent, $e = 2^{623}+1$ (e is of 624 bits), the private exponent d is of 512 bits, and the RSA modulus is $N = pq$, where the prime factors p and q are about of 512 bits.

Step 1. Fix e first (e.g. $e = 2^{623} + 1$).
Step 2. Randomly select a prime p of 512 bits.
Step 3. Solve $eu^{'} - (p-1)v^{'} = 1$ to obtain $u^{'}$ and $v^{'}$ where $0 < u^{'} < (p-1)$ and $0 < v^{'} < e$.

TABLE I
Largest δ (where $d < N^{\delta}$) for which Durfee-Nguyen's attack can be completed.

	$\log_N(e)$					
	1.0	0.9	0.8	0.7	0.6	0.55
0.5	0.284	0.323	0.363	0.406	0.451	0.475
$\log_N(p)$ 0.4	0.296	0.334	0.374	0.415	0.460	0.483
0.3	0.334	0.369	0.406	0.446	0.487	0.510

Step 4. Try to find $v^{'} = k^{''}q^{'}$, where $k^{''}$ is of 112 bits and $q^{'} + 1$ is a prime. If it fails, go to Step 2; else $d = u^{'}$, $q = q^{'} + 1$, and $N = pq$.
Step 5. The RSA parameters are p, q, e, d and N.

Following the above scheme, we provide that the RSA public exponent is of the special form of $2^X + 1$. Also the resulting scheme keeps the same properties as Sun and Yang's RSA variants. So, we can generate RSA instances in which e is a special form of $2^{623} + 1$ (about 624 bits), d is of 512 bits, and both p and q are approximately of $\frac{1}{2}\log_2 N$ bits (about 512 bits). Notice that e and d satisfy the equation $ed = k^{'}k^{''}(p-1)(q-1) + 1 = k\phi(N) + 1$, where $k = k^{'}k^{''}$.

B. Security Analysis

First, we consider the Durfee and Nguyen attack[8]. Table I shows the largest possible δ (where $d < N^{\delta}$) for which Durfee and Nguyen attack can succeed. In our instances, a devised public exponent $e = 2^{623} + 1$ is of 624 bits, p is a randomly selected 512-bit number, and the private exponent d is of 512 bits, so $\log_N(e) = \log_N(2^{623}+1) \approx 0.6$, $\log_N(p) \approx \log_N(2^{512}) = 0.5$, and $\log_N(d) \approx \log_N(2^{512}) = 0.5$. Following the Table I, Durfee and Nguyen's attack will be successful for $\delta = 0.451$ suggested in our instances. Fortunately, our parameter δ ($=\log_N(d) \approx \log_N(2^{512}) = 0.5$) is larger than 0.451. Therefore, Durfee and Nguyen's attack can not work in our instances.

Subsequently, we consider the exhaustive search attack. We can check a guess for k since $\phi(N) = N + 1 - (p + q) \equiv (-k)^{-1} \pmod{e}$ and so $(p + q) \equiv N + 1 + k^{-1} \pmod{e}$. Since $p + q < e$, this gives $p + q$ exactly and then we can test the guess by checking whether $a^{N+1-(p+q)} \equiv 1 \pmod{N}$ for a random value a. In our algorithm, k is large enough (112 bits), an exhaustive search method can not work effectively.

Note that others attacks[4][8][11][22][23][24] about short exponent on RSA do not apply to our algorithm.

C. Computational Load

If we randomly chose a 624-bit public exponent e. According Table **??** in Section 2.2 above, the average number of modular multiplications required by modular exponentiation algorithm for encryption/signature verification is found to be about 936 ($\frac{3}{2}\lfloor \log_2 e \rfloor$) in typical RSA. However, in our proposed scheme, we chose a special public exponent with low Hamming weight, $e = 2^{623} + 1$, which only takes 624 ($\lfloor \log_2 e \rfloor + 1$) modular multiplications. Notwithstanding all serious cryptosystems exponentiation is almost implemented by using the sliding window method, whose complexity is

TABLE II
COMPARISONS OF SUN-YANG'S 2ND RSA VARIANT AND OURS.

	Sun and Yang's 2nd scheme	Our proposed scheme
Which Key Is Selected First	Private Exponent	Public Exponent
Public Key Form	Random	$2^X + 1$
Time of key Generation	Fast	Slow
Encryption Time	Slow	Fast
# of Modular Multiplication in Encryption	$\frac{3}{2}\lfloor \log_2 e \rfloor$ (Average)	$\lfloor \log_2 e \rfloor + 1$

about of the order of $1.15 \sim 1.25$ times the length of the exponent ($1.15 \times \lfloor \log_2 e \rfloor \sim 1.25 \times \lfloor \log_2 e \rfloor$). Our proposed method still has the better performance in modular arithmetic.

D. Comparisons

Although remaining the same properties as the Sun-Yang RSA variants in our proposed scheme, there are some differences between them. Here, the comparisons are listed in Table II. The first item in the comparison is "Which key is selected first ?". In Sun and Yang's 2nd scheme, a prime p and the private exponent d are selected first, d is prime to $p-1$, a corresponding public exponent is generated finally. Our scheme fixes a particular public exponent ($2^X + 1$) first, and then builds a proportional private exponent. Note that we can not control and generate a public exponent with special form $2^X + 1$ directly using Sun and Yang's method. The item "Time of key generation" denotes how long it takes to generate a key pair? Because of factoring into an 112-bit factor, our scheme needs more time then Sun and Yang's 2nd scheme. Subsequently, the item "Encryption time" and "# of Modular Multiplication in Encryption" are identical to the encryption cost. The number of modular multiplication decreased using our proposed algorithm, so that the encryption is speeded up.

III. EXPERIMENTAL RESULTS

We implement our algorithm using C programming language under NTL with GMP (GNU Multi-Precision library) on MS-Windows system with Cygwin tools upon a personal computer (PC) with 1.7GHz CPU and 512MB DRAM. We factor 624 bit-length number by Pollard $p-1$ method[14] at Step 4 of the proposed algorithm. We notes that k'' is a 112 bit-length composite number and $q' + 1$ is a prime number.

We demonstrate the two instances of RSA in which e is assigned to be $2^{623} + 1$ (624 bit-length), d is of 512 bits, p and q are about of 512 or 513 bits. The instances are as follows:

Example 1

$e =$ $2^{623} + 1$;
$d =$ EEAB5CDC C5BB8412 6E469CB0 304C460F
4CC9BD4E 2CE34185 72D8342F 85DDB3C1
72DA499F 68BD0BD9 79F23726 3C064375
08997B92 B6D6D68B 959A41C6 A190A211;
$p =$ 9608E778 138F94DE 00E55DB1 350F7FBA
047297FB E529C05B D741B969 1624F151
D913FC86 56A46E45 12E82FEF 86DA82F0
A0F7F3A6 1EC89E64 76BA27E2 B1547C23;
$q =$ 00000001 79DF0EEF 751EDFF8 767F72D4
F6BFABA7 E73CDEAE B7EE60B7 DAE5532D
DDABF832 96772EBE 7512F7FB ACFEEC12
C9B7998E 3C0C33A1 CE305450 D5A8C38B
1E9CFCC9;

Example 2

$e =$ $2^{623} + 1$;
$d =$ F08369D9 3CF2728D 6DF9B230 A624616B
C487AE3D DCF3272E 81C7D8BA D2CF82B2
5D4F90D8 C0284124 A7659F41 FBB1B0AA
58424373 FCF17102 1C9A81F8 FCE2A661;
$p =$ D866B988 94C695BD E7D33C05 4D4B26A3
7B4E3B2D BF358ADE 7E9F4DD2 C676A4B4
61406094 B382D6E4 D919BE8F 2E02BAFA
3305596E 653E97FC 2401105C 91D8EA59;
$q =$ BA8C67C5 E75F9C8E F869D924 D03B6FF6
84E137ED 42FCD601 51CFC426 904075D9
246E0EC6 DE6CE58A B67ABD25 61409B2B
3AC3FD73 A908A88B 2F886D86 DFF71195;

Consider the results quoted above. It takes about 929.87 seconds and $2,977$ loops running from Step 2 to Step 4 in our first example, and it takes about $9,235.07$ seconds and $29,500$ loops in our second example. Note that these experimental data are just for demonstrated that our algorithm can be executed correctly. The key generation process may be a heavily computational load for end users. However, we can adopt the better factoring method, like NSF[5] or ECM[13], to reform the experimental results' performance. Besides, the key generation process can be executed on some parallel techniques, and/or high-performance computers, such that SDC (Trust Shadow Distribution Center). For example, a trusted CA can generate the RSA key pairs beforehand and issue the IC cards with the applicant's private informations (public key, private key and certificate... and so on).

IV. CONCLUSIONS

Recently, Sun and Yang[17] proposed a new RSA variant with balanced prime factors p, q and trade-off exponents e, d for reducing and rebalancing the computational load between encryption and decryption. Considering the RSA operation is a modular exponentiation, the special form $2^X + 1$ will help decrease the number of modular multiplications. In this paper, we designed a special variant of RSA with devisable key pair whose public exponent is formed $2^X + 1$ to further reduce the encryption cost and private exponent is of 512-bit to meet the secure requirement. As an example, we construct the two instances of RSA where e is assigned to be $2^{623} + 1$ (624 bit-length), d is of 512 bits, p and q are about of 512 or 513 bits. Our proposed scheme continues the characteristic of Sun-Yang scheme and strengthens the performance , but our scheme is distinct from Sun-Yang scheme[17] in essence. We can set a particular public key first (representative or commemoratory number) to make known to the public, and

build a corresponding private key to keep in secret, that is to say, we can obtain a special-made key pair using our proposed algorithm. Our innovation is taken advantage of the fact that the computational complexity is reduced using low Hamming weight exponents and the encryption/decryption processes needn't know p and q. Owing to factorization, the key generation of our scheme is very slow. Fortunately, we can generate key pair previously by means of the computation-powerful server. One drawback of our scheme is that RSA operation is done using Chinese Remainder Theorem, our schemes may be not to provide astonishing performance. However, not every RSA cryptosystem applications[17] suit to RSA-CRT technique[12][19][20]. This topic is still an interesting discussion on typical RSA.

REFERENCES

[1] A. Aziz and W. Diffie, "A secure communications protocol to prevent unauthorized access - privacy and authentication for wireless local area networks", *IEEE Pers. Commun.*, First Quarter, 1994.

[2] M. Bellare and P. Rogaway, "Entity authentication and key distribution", Proceedings of Crypto'93 , LNCS 773, pp. 232-249, 1994.

[3] S. M. Bellovin and M. Merritt, "Encrypted key exchange: password-based protocols secure against dictionary attacks.", Proc. IEEE Computer Society Conference on Research in Security and Privacy, pp. 72-84, 1992.

[4] D. Boneh and G. Durfee, "Cryptanalysis of RSA with private key d less than $N^{0.292}$", Proceedings of Eurocrypt '99, LNCS 1592 , pp. 1–11, 1999.

[5] S. Cavallar, B. Dodson, A. K. Lenstra, W. Lioen, P. L. Montgomery, B. Murphy, H. te Riele, K. Aardal, J. Gilchrist, G. Guillerm, P. Leyland, J. Marchand, F. Morain, A. Muffett, C. Putnam, C. Putnam and P. Zimmermann, "Factorization of 512-bit RSA key using the number field sieve", Proceedings of Eurocrypt'00, LNCS 1807, pp. 1-18, 2000.

[6] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithm,second edition, McGraw-hill Book Company, 2001.

[7] T. Collins, D. Hopkins, S. Langford, and M. Sabin. Public Key Cryptographic Apparatus and Method. US Patent #5,848,159. Jan. 1997.

[8] G. Durfee and P. Nguyen, "Cryptanalysis of the RSA Schemes with Short Secret Exponent from Asiacrypt'99", Proceedings of Asiacrypt'00, LNCS 1976, pp. 14–29, 2000.

[9] A. Fiat. "Batch RSA." In G.Brassard, ed., Proceedings of Crypto '89, LNCS 435, pp. 175–185, 1989.

[10] I. N. Herstein, Topics in Algebra, Xerox Corporation, 1975.

[11] H. S. Hong, H. K. Lee, H. S. Lee and H. J. Lee, "The better bound of private key in RSA with unbalanced primes", Applied Mathematics and Computation, Vol. 139, pp. 351-362, 2003.

[12] S. D. Galbraith, C. Heneghan AND J. F. McKee, "Tunable balancing of RSA.Information Security and Privacy", 10th Australasian Conference - ACISP 2005, LNCS Vol. 3574, pp. 280-292, 2005.

[13] H.W. Lenstra, Jr., "Factoring integers with elliptic curve", Annuals of Mathematics, Vol. 126, pp. 649–673, 1987.

[14] J. Pollard, "Theorems of factorization and primality testing", Proc. Cambridge Philos. Soc., pp. 76:521–528, 1974.

[15] R. L. Rivest, A. Shamir and L. M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", Comm. ACM, Vol. 21,pp. 120-126, 1987.

[16] R. Rivest and R. D. Silverman, "Are strong primes needed for RSA?", The 1997 RSA Laboratories Seminar series, Seminar Proceedings, 1997.

[17] H. M. Sun and C. T. Yang, "RSA with Balanced Short Exponents and Its Application to Entity Authentication", Proceedings of PKC 2005, LNCS 3386, pp. 199–215, 2005.

[18] H. M. Sun, W. C. Yang and C. S. Laih, "On the design of RSA with short secret exponent", Proceedings of Asiacrypt'99, LNCS 1716, pp. 150–164, 1999.

[19] H. M. Sun and M. E. Wu, "Design of Rebalanced RSA-CRT for Fast Encryption", Information Security Conference 2005, pp. 16-27, 2005.

[20] H. M. Sun, M. J. Hinek and M. E. Wu, "On the Design of Rebalanced RSA-CRT", Technical Report CACR 2005-35, 2005.

[21] T. Takagi. "Fast RSA-type Cryptosystem Modulo $p^k q$." In H. Krawczyk, ed., Proceedings of Crypto '98, LNCS 1462, pp. 318–326, 1998.

[22] E. Verheul and H. van Tilborg, "Cryptanalysis of less short RSA secret exponents", Applicable Algebra in Engineering, Communication and Computing, Vol. 8, pp. 425-435, 1997.

[23] B. de Weger, "Cryptanalysis of RSA with small prime difference", Applicable Algebra in Engineering, Communication and Computing, Vol. 13, pp. 17-28, 2002.

[24] M. Wiener, "Cryptanalysis of short RSA secret exponents", IEEE Transactions on Information Theory, Vol. 36, no. 3, pp. 553–558, 1990.

[25] S.B. Wilson and A. Menezes, "Authenticated Diffie-hellman key agreement protocols", 5th Annual International Workshop, SAC'98, pp. 339-361, 1998.