Fast RSA-Type Cryptosystem Modulo $p^k q$

Tsuyoshi Takagi

NTT Software Laboratories
3-9-11, Midori-cho Musashino-shi, Tokyo 180-0012, Japan
E-mail: ttakagi@slab.ntt.co.jp

Abstract. We propose a cryptosystem modulo $p^k q$ based on the RSA cryptosystem. We choose an appropriate modulus $p^k q$ which resists two of the fastest factoring algorithms, namely the number field sieve and the elliptic curve method. We also apply the fast decryption algorithm modulo p^k proposed in [22]. The decryption process of the proposed cryptosystems is faster than the RSA cryptosystem using Chinese remainder theorem, known as the Quisquater-Couvreur method [17]. For example, if we choose the 768-bit modulus $p^2 q$ for 256-bit primes p and q, then the decryption process of the proposed cryptosystem is about 3 times faster than that of RSA cryptosystem using Quisquater-Couvreur method.

Key words: RSA cryptosystem, Quisquater-Couvreur method, fast decryption, factoring algorithm

1 Introduction

The RSA cryptosystem is one of the most practical public key cryptosystems and is used throughout the world [19]. Let n be a public key, which is the product of two appropriate primes, e be an encryption key, and d be a decryption key. The algorithms of encryption and decryption consist of exponentiation to the e^{th} and d^{th} powers modulo n, respectively. We can make e small, but must consider low exponent attacks [3] [4] [6]. The encryption process takes less computation and is fast. On the other hand, the decryption key d must have more than one fourth the number of bits of the public key n to preclude Wiener's attack [24] and its extension [23]. Therefore, the cost of the decryption process is dominant for the RSA cryptosystem.

In this paper, we propose an RSA-type cryptosystem modulo $n = p^k q$. Even though the modulus is not of the form pq, we choose appropriate sizes for the secret primes p and q to preclude both the number field sieve and the elliptic curve method. Using this modulus $p^k q$, we construct a fast decryption public-key cryptosystem. In the key generation, we generate the public key e and secret key d using the relation $ed \equiv 1 \pmod{L}$, where $L = \mathrm{LCM}(p-1, q-1)$. Note that L is not the same as $\phi(n) = p^{k-1}(p-1)(q-1)$ or even $\lambda(n) = \mathrm{LCM}(p^{k-1}(p-1), q-1)$. Thus, the secret exponent d becomes much smaller than $n = p^k q$. Moreover, for decrypting $M_p \equiv M \pmod{p^k}$ we show that it is possible to apply the fast

decryption algorithm proposed in [22]. The running time for computing M_p is essentially equivalent to that for $C^d \pmod p$. Therefore, the decryption process is much faster than in the RSA cryptosystem using the Chinese remainder theorem [17].

The paper is organized as follows. In Section 2, we describe the algorithm of the proposed cryptosystem. We discuss the size of the secret primes which prevents the use of both the number field sieve and the elliptic curve method in Section 3. Then, we show the running time of the proposed cryptosystem in comparison with the RSA cryptosystem using the Quisquater-Couvreur method in Section 4. We explain the effectiveness of Wiener's attack in Section 5. We show some properties of our cryptosystem related to some attacks in Section 6.

Notation: $\mathbf{Z}$ is an integer ring. $\mathbf{Z}_n$ is a residue ring $\mathbf{Z}/n\mathbf{Z}$ and its complete residue class is $\{0, 1, 2, \ldots, n-1\}$. $\mathbf{Z}_n^\times$ is a reduced residue group modulo n. $\mathrm{LCM}(m_1, m_2)$ is the least common multiple of m_1 and m_2. $\mathrm{GCD}(m_1, m_2)$ is the greatest common divisor of m_1 and m_2.

2 Proposed public-key cryptosystem

In this section, we describe an RSA-type cryptosystem modulo $p^k q$, and discuss the size of its secret keys and the running time.

2.1 Algorithm

1. Generation of the keys: Generate two random primes p, q, and let $n = p^k q$. Compute $L = \mathrm{LCM}\,(p-1, q-1)$, and find e, d which satisfies $ed \equiv 1 \pmod L$ and $\mathrm{GCD}(e, p) = 1$. Then e, n are public keys, and d, p, q are the secret keys.
2. Encryption: Let $M \in \mathbf{Z}_n^\times$ be the plaintext. We encrypt the plaintext by the equation:

$$C \equiv M^e \pmod n. \tag{1}$$

3. Decryption: We decrypt $M_p \equiv M \pmod{p^k}$ and $M_q \equiv M \pmod q$ using the secret key d, p, q. The plaintext M can be recovered by the Chinese remainder theorem. Here, M_q is computed by $M_q \equiv C^d \pmod q$ and M_p is computed by the fast algorithm described in [22].

2.2 Details of the decryption algorithm

The order of the group $\mathbf{Z}_{p^k}^\times$ is $p^{k-1}(p-1)$. When $M_p \equiv M \pmod{p^k}$ is recovered using the standard algorithm of RSA, we have to compute $M_p \equiv C^d \pmod{p^k}$ for $d \equiv e^{-1} \pmod{\mathrm{LCM}(p^{k-1}(p-1), q-1)}$. Then the running time is slower than that of the method using the Chinese remainder theorem for $n = pq$ [17], so there are no significant advantages in using the modulus $p^k q$. Instead, we apply the method described in [22], where the author presents a fast algorithm

for computing RSA decryption modulo n^k using n-adic expansion. Then, the running time for computing M_p becomes essentially equivalent to computing $M_p \equiv C^d \pmod{p}$ for $d \equiv e^{-1} \pmod{\mathrm{LCM}(p-1, q-1)}$.

First, we modify the algorithm into a more efficient form. We denote the ciphertext reduced modulo p^k by C_p. Then the relationship between the ciphertext C_p and the plaintext is $C_p \equiv M_p^e \pmod{p^k}$. Note that M_p the plaintext modulo p^k, has the p-adic expansion such that

$$M_p \equiv K_0 + pK_1 + p^2K_2 + \ldots + p^{k-1}K_{k-1} \pmod{p^k}. \tag{2}$$

Here, we define the function $F_i(X_0, X_1, \ldots, X_i)$ as follows:

$$F_i(X_0, X_1, \ldots, X_i) = (X_0 + pX_1 + \ldots + p^iX_i)^e,$$

where $i = 0, 1, \ldots, k-1$. $F_{k-1}(X_0 + pX_1 + \ldots + p^{k-1}X_{k-1})^e$ is the same as the function that encrypts the plaintext M_p in equation (2). By reducing modulo p^{i+1}, we get the relationship

$$F_i(X_0, X_1, \ldots, X_i) \equiv F_{i-1} + p^iG_{i-1}X_i \pmod{p^{i+1}},$$

where $F_{i-1} = F_{i-1}(X_0 + pX_1 + \ldots + p^{i-1}X_{i-1})^e$ and $G_{i-1} = e(X_0 + pX_1 + \ldots + p^{i-1}X_{i-1})^{e-1}$ for $i = 0, 1, \ldots, k-1$. From this relationship, we can recursively calculate $K_1, \ldots, K_{k-1}$. For $i = 1$, K_1 is the solution of the following linear equation of X_1:

$$C \equiv F_0(K_0) + pG_0(K_0)X_1 \pmod{p^2}. \tag{3}$$

Assume we have already calculated $K_1, K_2, \ldots, K_{i-1}$. Using these values, we compute $F_{i-1}(K_0, K_1, \ldots, K_{i-1}), G_{i-1}(K_0, K_1, \ldots, K_{i-1})$ in $\mathbf{Z}$, and denote them by F_{i-1}, G_{i-1}, respectively. Then, K_i is the solution of the following linear equation of X_i:

$$C \equiv F_{i-1} + p^iG_{i-1}X_i \pmod{p^{i+1}}. \tag{4}$$

Note that $(G_{i-1}, p) = 1$, because $\mathrm{GCD}(K_0, p) = \mathrm{GCD}(e, p) = 1$, so we can uniquely decrypt K_i.

After computing $K_0, K_1, \ldots, K_{k-1}$, we can evaluate $M_p \pmod{p^k}$ from equation (2). Finally, the plaintext $M \pmod{p^kq}$ is also computed from the values $M_p \pmod{p^k}, M_q \pmod{q}$, and the Chinese remainder theorem.

Moreover, note that we do not have to use the secret exponent d for evaluating $K_1, K_2, \ldots, K_{k-1}$. Thus, when we compute the two values of $K_0 \equiv C^d \pmod{p}$ and $M_q \equiv C^d \pmod{q}$, the secret exponent d can be reduced modulo $p-1$ and $q-1$. Indeed, $C^d \equiv C^{d_p} \pmod{p}$ and $C^d \equiv C^{d_q} \pmod{q}$ hold, where $d_p \equiv d \pmod{p-1}$ and $d_q \equiv d \pmod{q-1}$.

In Appendix A, we describe the decryption program written in pseudo-code. For $x \in \mathbf{Z}$ and a positive integer N, $[x]_N$ denotes the remainder of x modulo N, which is in $\{0, 1, \ldots, N-1\}$.

3 Size of secret parameters

Here, we discuss the size of the secret parameters p and q. The RSA cryptosystem uses a composite number of the symmetry type pq, where p and q are the same bit size. The cryptosystem proposed in this paper depends on the security of factoring the modulus $p^k q$. We have to carefully choose the size of p and q.

There are two types of fast factoring algorithm to consider: the number field sieve [11] and the elliptic curve method [10]. Other factoring algorithms have the same or slower running times, so the size of the RSA-modulus can be estimated by these two factoring algorithms [7] [13] [20]. Let $L_N[s,c] = exp((c+o(1))\log^s(N)\log\log^{1-s}(N))$. The number field sieve is the fastest factoring algorithm, and the running time is estimated from the total bit size of the integer n to be factored, which is expected as $L_n[1/3,(64/9)^{1/3}]$. If we choose n to be larger than 768 bits, the number field sieve becomes infeasible. In our case, we have to make the modulus $n = p^k q$ larger than 768 bits. The elliptic curve method is effective for finding primes which are divisors of the integer n to be factored. The running time is estimated in terms of the bit size of the prime divisor p. Its expected value is $L_p[1/2,2^{1/2}]$. Note that the running time of the elliptic curve method is different from that of the number field sieve, and the order is much different. If we choose p to be larger than 256 bits, the elliptic curve method becomes infeasible. In our case, we have to make the primes p and q of the modulus larger than 256 bits.

The factoring algorithm strongly depends on the implementation. In my knowledge, the fastest implementation record for the number field sieves factored 130-digit RSA modulus [5] and that for the elliptic curve method found 48-digit prime factor [8]. Here, we again emphasize that there is a big difference in the cost between the number field sieve and the elliptic curve method. Therefore, if we choose the 768-bit modulus p^2q with 256-bit primes p and q, neither of the factoring algorithms is feasible, so the scheme is secure for cryptographic purposes. But the size of secret primes must be thoroughly discussed for the practical usage of our proposed cryptosystem, and this is work in progress.

Here, we wonder if there exists factoring algorithms against the modulus with a square factor p^2q. This factoring problem appeared in the list of the open problems in number theoretic complexity by Adleman and McCurley [1], and it is unknown whether there exists $L_p[1/3]$-type sub-exponential algorithm which finds the primes of the composite number p^2q. Recently, Peralta and Okamoto proposed a factoring algorithm against numbers of the form p^2q based on the elliptic curve method [16]. They focused on the fact the Jacobi symbol is equal to one for a square integer, and the running time becomes a little bit faster than that of the original elliptic curve method.

Remark 1. A digital signature scheme [14] and two public key cryptosystems [9] [15] which rely on the difficulty of factoring numbers of the type p^2q have been proposed. These cryptosystems are fast and practical. For secure usage of these cryptosystems and our proposed cryptosystem, the research of factoring algorithms against a composite number with a square factor is desirable.

4 Running time

In this section, we estimate the running time of the proposed cryptosystem. We assume that the public modulus $n = p^2q$ is 768 bits for 256-bit primes p and q in the following. We also assume the running time for computing $Z^a \pmod b$ is $O(\log_2^2(b)\log_2(a))$. Below, we estimate the worst-case running time.

In the decryption process of the proposed cryptosystem, the algorithm does not depend on the secret exponent d except when we compute

$$C^d \pmod p, \quad C^d \pmod q. \tag{5}$$

After calculating $C^d \pmod p$, we compute only a few multiplications for obtaining $M_p \equiv M \pmod{p^k}$. This costs the same as the encryption process. If we choose a very small e, this algorithm is very efficient. For example, if the modulus be p^2q, then we only compute at most $\lfloor \log_2 e \rfloor$ multiplications modulo p^2 and one division of p, two multiplications modulo p, and one inversion modulo p. Moreover, when we compute the two values of equation (5), the secret exponent d can be reduced modulo $p-1$ and $q-1$. In other words, $C^d \equiv C^{d_p} \pmod p$ and $C^d \equiv C^{d_q} \pmod q$ hold, where $d_p \equiv d \pmod{p-1}$ and $d_q \equiv d \pmod{q-1}$. Thus, the size of the secret exponent can be reduced.

Denote by T the running time for computing the decryption algorithm of the original RSA cryptosystem, i.e., $C^{d'} \pmod n$, where d' is as large as n. Then, the running time of the proposed cryptosystem for a 768-bit modulus is about $(2(1/3)^3 + \alpha_e)T = (0.074 + \alpha_e)T$, where α_e depends only on the encryption exponent e. When we make the encryption exponent e very small, α_e becomes negligible.

A similar decryption algorithm for the RSA cryptosystem using Chinese remainder theorem, the Quisquater-Couvreur method, mainly computes $C^d \pmod p$ and $C^d \pmod q$, where $n = pq$ is the RSA modulus, both p and q are as large as $(\log_2 n)/2$ bits, and we assume d is as large as p and q. So, the running time of Quisquater-Couvreur method is about 4 times faster than the original RSA cryptosystem.

Here, we compare the running time of our proposed cryptosystem with that of Quisquater-Couvreur method. The comparison is carried out based on the common bit length of the modulus. The proposed cryptosystem with the small encryption exponent e is about 3 times faster than the RSA cryptosystem applying the Quisquater-Couvreur method for the 768-bit modulus.

In addition, consider the RSA cryptosystem with the square-free modulus $n = p_1p_2\cdots p_l$, where we assume that p_i are as large as $(\log_2 n)/l$ bits for $i = 1, 2, \ldots, l$. As we discussed in Section 3, we can use a 768-bit modulus $n = p_1p_2p_3$ with 256-bit primes $p_i (i = 1, 2, 3)$ for the cryptographic purpose. This version of RSA will be faster when we use the decryption technique using the Chinese remainder theorem. Indeed, the decryption time with this modulus is dominant for computing $C^{d_i} \pmod{p_i}$, where we assume d_i are as large as p_i for $i = 1, 2, 3$. So, the running time of this RSA variant is about 9 times faster than the original RSA cryptosystem. Here, we compare this RSA variant with our

proposed cryptosystem. Our proposed cryptosystem is about 1.5 times faster for a 768-bit modulus.

5 Short secret exponent d

A short secret exponent is desirable for the fast decryption algorithm. However, Wiener reported an attack based on the continued fraction algorithm which detects a short secret exponent d [24]. This attack is effective for $d < n^{1/4}$.

The secret key d and the public key e of the proposed cryptosystem have the relation $ed \equiv 1 \pmod{\mathrm{LCM}(p-1, q-1)}$, and the primes p and q are much smaller than n. So, we wonder if Wiener's attack is applicable to larger secret exponents d. Moreover, if the attacker can compute d' such that

$$ed' \equiv 1 \pmod{\mathrm{LCM}(p^{k-1}(p-1), q-1)}, \tag{6}$$

then proposed cryptosystem will also be broken.

Here, we discuss Wiener's attack for relation (6). From $\mathrm{LCM}(p^{k-1}(p-1), q-1) = p^{k-1}(p-1)(q-1)/\mathrm{GCD}(p^{k-1}(p-1), q-1)$, we have $ed' = 1+mp^{k-1}(p-1)(q-1)/\mathrm{GCD}(p^{k-1}(p-1), q-1)$ for some integer m. Generally, $\mathrm{GCD}(p^{k-1}(p-1), q-1)$ is very small compared with p and q. Let $m/\mathrm{GCD}(p^{k-1}(p-1), q-1) = h/g$, where $\mathrm{GCD}(h, g) = 1$. Then, we get the relation

$$\left| \frac{e}{p^k q} - \frac{h}{gd'} \right| = \delta', \tag{7}$$

where $\delta' = \frac{h}{gd'} \frac{p^k + p^{k-1}q - p^{k-1} - g/h}{p^k q}$. From $h/d'g \leq 1$, the upper bound of δ' is of the size $n^{-1/(k+1)}$. It is known that for a rational number x such that $|x - P/Q| < 1/2Q^2$, P/Q is a convergent in the continued fraction of x, where P and Q are relatively prime integers. Therefore, if $n^{-1/(k+1)} < 1/2(gd')^2$ holds, then Wiener's attack is applicable by computing the continued fraction of $e/p^k q$. Therefore, Wiener's attack is effective for $d' < n^{\frac{1}{2(k+1)}}$. During key generation one must ensure that $d' \equiv e^{-1} \pmod{\mathrm{LCM}(p^{k-1}(p-1), (q-1))}$ is sufficiently large.

In the same manner, we can discuss the Wiener's attack for the relation $ed \equiv 1 \pmod{\mathrm{LCM}(p-1, q-1)}$. In this case, we get the relation

$$\left| \frac{e}{p^k q} - \frac{h}{gdp^{k-1}} \right| = \delta, \tag{8}$$

where $\delta = \frac{h}{gd} \frac{p+q-1-g/h}{p^k q}$. The lower bound on δ is of the size $1/gdn^{k/(k+1)}$, and $1/gdn^{k/(k+1)}$ is larger than the upper bound $1/2(gdp^{k-1})^2 \sim 1/2(gdn^{(k-1)/(k+1)})^2$ which the continued fraction can detect. So, Wiener's attack seems infeasible for the relation $ed \equiv 1 \pmod{\mathrm{LCM}(p-1, q-1)}$. Further work on this is in progress.

6 Other properties

In this section, we describe some attacks against our proposed cryptosystem and some other properties of it.

Permutation: Let S be a finite set, and let $F(x)$ be a function from S to S. The function $F(x)$ is called a permutation function if every pair $x, y \in S$ that satisfies $F(x) = F(y)$ also satisfies $x = y$. The encryption function must be a permutation function in order to have unique decryption. The encryption function of the proposed cryptosystem is $F(X) \equiv X^e \pmod{p^k q}$. This function is a permutation function if and only if $\mathrm{GCD}(p-1, e) = \mathrm{GCD}(q-1, e) = \mathrm{GCD}(p, e) = 1$. The last condition is always satisfied for small e, so this condition becomes the same as that for the original RSA cryptosystem.

Message concealing: A function $F(x)$ is called unconcealed when $F(x) = x$ holds for some x. If the encryption function is unconcealed, some plaintexts are not encrypted. Blakley and Borosh showed that the encryption function of the RSA cryptosystem is unconcealed [2]. And they also estimated the number of unconcealed messages for a modulus having the form $p^k q$. They proved

$$N = (1 + \mathrm{GCD}(e-1, p^{k-1}(p-1)))(1 + \mathrm{GCD}(e-1, (q-1))).$$

This number is negligible because we choose e to be small in our proposed cryptosystem.

Cycling attack: The cycling attack is to find an integer s such that $C^{e^s} \equiv C \pmod{p^k q}$ [12] [25]. If we find such an integer, then the modulus $p^k q$ can be factored with probability greater than 1/2. From a recent result by Rivest and Silverman, it is known that the probability of the cycling attack success is negligible [20]. This analysis is also true for our proposed cryptosystem, because p and q must be chosen to be more than 256-bit primes. Here, denote by $\mathrm{ord}_m(Q)$ the order of the point Q in the group $\mathbf{Z}_m$ for some integer m, and $\mathrm{ord}_{\mathrm{ord}_n(C)}(e) | s$ holds. Note that $\mathrm{ord}_m(Q) | \mathrm{ord}_n(Q)$ for $m | n$ and Q in $\mathbf{Z}_n$. The probability that $p | \mathrm{ord}_{p^k}(Q)$ for a random point Q in $\mathbf{Z}_{p^k}$ is $1 - 1/p$, so $p | \mathrm{ord}_n(C)$ holds for a random ciphertext C in $\mathbf{Z}_n$ with high probability, and $\mathrm{ord}_p(e)$ is greater than the largest prime of $p - 1$, which is more than 50 bits with high probability. Therefore, the integer s is greater than 50 bits with high probability.

Other attacks: All other attacks are applicable, for example, the low exponent attacks [3] [4] [6], the common modulus attack, and the chosen message attack (See, for example, [7] [13]).

Digital signature: Of course, the proposed algorithm can be used for a digital signature.[1] The prominent property of our proposed cryptosystem is the running time for generating the signature, which it is faster than that of the RSA cryptosystem using Chinese remainder theorem.

Rabin-type cryptosystem: We can construct a Rabin-type cryptosystem by applying the algorithm proposed in this paper. We can also prove that the extended Rabin-type cryptosystem is as intractable as factoring the modulus $p^k q$.

[1] Shamir proposed a variation of RSA cryptosystem with an unbalanced modulus [21]. As he stated in the paper, Shamir's RSA can not be used for digital signatures.

Acknowledgments

I wish to thank Shozo Naito for his helpful discussion. I would also like to thank the anonymous referees for their valuable comments.

References

1. L. M. Adleman and K. S. McCurley, "Open problems in number theoretic complexity, II" proceedings of ANTS-I, LNCS 877, (1994), pp.291-322.
2. G. R. Blakley and I. Borosh, "Rivest-Shamir-Adleman public key cryptosystems do not always conceal messages," Comput. & Maths. with Appls., 5, (1979), pp.169-178.
3. D. Coppersmith, M. Franklin, J. Patarin and M. Reiter, "Low-exponent RSA with related messages," Advances in Cryptology – EUROCRYPT '96, LNCS 1070, (1996), pp.1-9.
4. D. Coppersmith, "Finding a small root of a univariate modular equation," Advances in Cryptology – EUROCRYPT '96, LNCS 1070, (1996), pp.155-165.
5. J. Cowie, B. Dodson, R. Elkenbracht-Huizing, A. K. Lenstra, P. L. Montgomery, J. Zayer; "A world wide number field sieve factoring record: on to 512 bits," Advances in Cryptology – ASIACRYPT '96, LNCS 1163, (1996), pp.382-394.
6. J. Håstad, "Solving simultaneous modular equations of low degree," SIAM Journal of Computing, 17, (1988), pp.336-341.
7. B. S. Kaliski Jr. and M. Robshaw, "Secure use of RSA," CRYPTOBYTES, 1 (3), (1995), pp.7-13.
8. ECMNET Project; `http://www.loria.fr/~zimmerma/records/ecmnet.html`
9. D. Hühnlein, M. J. Jacobson, S. Paulus, and T. Takagi, "A cryptosystem based on non-maximal imaginary quadratic orders with fast decryption." Advances in Cryptology – EUROCRYPT '98, LNCS 1403, (1998), pp.294-307.
10. H. W. Lenstra, Jr., "Factoring integers with elliptic curves", Annals of Mathematics, 126, (1987), pp.649-673.
11. A. K. Lenstra and H. W. Lenstra, Jr. (Eds.), "The development of the number field sieve," Lecture Notes in Mathematics, 1554, Springer, (1991).
12. U. M. Maurer; "Fast generation of prime numbers and secure public-key cryptographic parameters," Journal of Cryptology, Vol.8, (1995), pp.123-155.
13. A. J. Menezes, P. C. van Oorschot and S. A. Vanstone, "Handbook of applied cryptography," CRC Press, (1996).
14. T. Okamoto, "A fast signature scheme based on congruential polynomial operations," IEEE Transactions on Information Theory, IT-36, (1990), pp.47-53.
15. T. Okamoto and S. Uchiyama; "A new public-key cryptosystem as secure as factoring," Advances in Cryptology – EUROCRYPT '98, LNCS 1403, (1998), pp.308-318.
16. R. Peralta and E. Okamoto, "Faster factoring of integers of a special form," IEICE Trans. Fundamentals, Vol.E79-A, No.4, (1996), pp.489-493.
17. J. -J. Quisquater and C. Couvreur, "Fast decipherment algorithm for RSA publickey cryptosystem," Electronic Letters, 18, (1982), pp.905-907.
18. M. O. Rabin, "Digitalized signatures and public-key functions as intractable as factorization", Technical Report No.212, MIT, Laboratory of Computer Science, Cambridge (1979), pp.1-16.

19. R. Rivest, A. Shamir and L. M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," Communications of the ACM, 21(2), (1978), pp.120-126.
20. R. Rivest and R. D. Silverman, "Are 'strong' primes needed for RSA," The 1997 RSA Laboratories Seminar Series, Seminars Proceedings, (1997).
21. A. Shamir; "RSA for paranoids," CryptoBytes, 1, Autumn, (1995), pp. 1-4.
22. T. Takagi, "Fast RSA-type cryptosystem using n-adic expansion," Advances in Cryptology – CRYPTO '97, LNCS 1294, (1997), pp.372–384.
23. E. R. Verheul and H. C. A. van Tilborg, "Cryptanalysis of 'less short' RSA secret exponents," Applicable Algebra in Engineering, Communication and Computing, 8, (1997), pp.425-435.
24. M. J. Wiener, "Cryptanalysis of short RSA secret exponents," IEEE Transactions on Information Theory, IT-36, (1990), pp.553-558.
25. H. C. Williams and B. Schmid, "Some remarks concerning the M.I.T. public-key cryptosystem," BIT 19, (1979), pp.525-538.

A Decryption algorithm

In this appendix, we describe the decryption program written in pidgin ALGOL. For $x \in \mathbf{Z}$ and a positive integer N, $[x]_N$ will denote the remainder of x modulo N, which is in $\{0, 1, \ldots, N-1\}$. The plaintext M is encrypted by $C \equiv M^e \pmod{p^k q}$. The relation between the encryption exponent e and the decryption exponent d is $ed \equiv 1 \pmod{\mathrm{LCM}(p-1, q-1)}$.

procedure **DECRYPTION**:
INPUT: d, p, q, e, k, C
OUTPUT: M

(1) $d_p := [d]_{p-1}, d_q := [d]_{q-1}$;
(2) $K_0 := [C^{d_p}]_p, M_q := [C^{d_q}]_q$;
(3) $A_0 := K_0$;
FOR $i = 1$ **to** $(k-1)$ **do**
$F_i := [{A_{i-1}}^e]_{p^{i+1}}$;
$E_i := [C - F_i]_{p^{i+1}}$;
$B_i := E_i / p^i$ **in** $\mathbf{Z}$;
$K_i := [(eF_i)^{-1} A_{i-1} B_i]_p$;
$A_i := A_{i-1} + p^i K_i$ **in** $\mathbf{Z}$;
(4) $M_p := A_{k-1}$;
(5) $p_1 := [(p^k)^{-1}]_q$, $q_1 := [q^{-1}]_{p^k}$;
(6) $M := [q_1 q M_p + p_1 p^k M_q]_{p^k q}$.