RSA for poor men: a cryptosystem based on probable primes to base 2 numbers

Marek Wójtowicz[0000−0003−4644−6713]

Kazimierz Wielki University, Institute of Mathematics
Powstańców Wielkopolskich 2, 85-090 Bydgoszcz, Poland
mwojt@ukw.edu.pl

Abstract. We show it is possible to build an RSA-type cryptosystem by utilizing *probable primes to base 2* numbers. Our modulus N is the product $n \cdot m$ of such numbers (so here both prime and some composite, e.g. Carmichael or Fermat, numbers are acceptable) instead of prime numbers. Moreover, we require for n and m to be co-prime only, and so we don't have to worry about whether any of the numbers n, m is composite or not.
The encryption and decryption processes are similar as those in the RSA. Hence, in this cryptosystem we may apply the above kind of numbers of arbitrary length being still sure that the system works well. The price for that is the size of words permitted by the new system: any message M, as a number, must be smaller than log (in base 2) of $n \cdot m$.

Keywords: RSA · Pseudoprimes in base 2 · Carmichael numbers.

1 Introduction and the result

In this article, we present a variant of the RSA cryptosystem, based on *probable prime to base 2* numbers instead of prime numbers. We assume that the reader is familiar with the classic RSA and knows the basic facts of cryptography, see e.g. [4]. For $K > 1$ an integer, $\mathbf{Z}_K^*$ denotes the multiplicative group of the ring $\mathbf{Z}_K$, i.e., $\mathbf{Z}_K^*$ consists of positive integers $k < K$ co-prime to K, endowed with multiplication modulo K.

1.1 Motivation and background

Let p, q be two co-prime prime numbers. Set $N := p \cdot q$, and let φ and λ be the Euler and Carmichael, respectively, functions on N: $\varphi(N) = (p-1) \cdot (q-1)$, and $\lambda(N) = \mathrm{lcm}(p-1, q-1)$. The classic RSA cryptosystem, built on p, q and N, encrypts and decrypts a message $M \in \mathbf{Z}_N^*$ using functions E and D, respectively, of the form:

$$E(M) = M^e (\mathrm{mod}\ N), \text{ and } D(C) = C^d (\mathrm{mod}\ N)), \tag{1}$$

where $e, d \in \mathbf{Z}_{\lambda(N)}^*$ fulfill the congruence

$$e \cdot d \equiv 1 (\mathrm{mod}\ \lambda(N)). \tag{2}$$

Since, by Euler's formula

$$x^{\varphi(N)} \equiv 1 \pmod{N} \text{ for all } x \in \mathbf{Z}_N^*, \tag{3}$$

$\lambda(N)$ in (2) can be replaced by $\varphi(N)$ because $\lambda(N)$ is the least positive integer t fulfilling the congruence

$$x^t \equiv 1 \pmod{N} \text{ for all } x \in \mathbf{Z}_N^*. \tag{4}$$

(and also $\varphi(N)$ is trivially a multiple of $\lambda(N)$). The fact that $E(D(M)) = M$ is the result of congruences (1), (2) and (4)/(3).

The basis of this cryptosystem are two large prime numbers p, q. The problem of primality of a given *odd* positive integer n is a fundamental issue in building cryptosystems utilizing prime numbers. For this purpose, we can use either deterministic primality tests (based mainly on the Pockington test [6], or AKS [7, Section 21]), or check the primality of n by a probabilistic test. (For a recent review of the effectiveness of all known methods of such tests see the paper by Albrecht, Massimo, Paterson, and Smorovsky [1].)

Each of these tests has both advantages and disadvantages. For example, deterministic tests are effective for particular kind of numbers or have other constrains, and probability tests may give erroneous results: in 2005, Bleichenbacher [3] showed that the most popular probabilistic primality test, the Miller-Rabbin test, if not well implemented, may pass composite numbers with probability 1.

Before proper testing the primality of n, we should do the "zero test": to check that n has no small divisors, i.e., whether

$$\gcd(n, T) = 1, \tag{5}$$

where T is the product of consecutive prime numbers with $T \approx n$.

The simplest probabilistic test is based on Fermat's little theorem: if the number n is prime, then each integer $a > 1$ coprime to n fulfills the congruence:

$$a^{n-1} \equiv 1 (\text{mod } n); \tag{6}$$

in particular (as n is odd by assumption),

$$2^{n-1} \equiv 1 (\text{mod } n). \tag{7}$$

Hence, congruence (6), as well as its particular form (7), is a *necessary condition* for n to be prime.

1.2 Probable primes in base *a*

Simple probabilistic arguments show that if the number a is chosen randomly, then the probability that n composite will pass the test (6) is $\leq 1/2$. Thus, if n

fulfills congruence (6) for $a_1, \ldots, a_k$ chosen randomly, then the probability that n is prime is $\geq 1 - (1/2)^k$ and tends to 1 as $k \to \infty$.

Let $n, a > 1$ be two co-prime positive integers. Then n is said to be a *probable prime to base* a (PRPa for short) if it fulfills the congruence

$$a^{n-1} \equiv 1 (\text{mod } n)\,. \tag{8}$$

By the Fermat little theorem, congruence (8) holds true if n is prime; in particular, for $a = 2$ and *every* n odd prime. As it is well known, the set of all PRP2-integers contains an infinite number of composite elements (e.g., Carmichael numbers [2]), along with all Fermat numbers $F_k = 2^{2^k} + 1$, $k = 1, 2, \ldots$ (see [5, Theorem 4.10]).

1.3 Construction of the new cryptosystem

Now, in a few simple steps, we shall present our idea of the new cryptosystem; it is easy to see, it has most of the elements from the RSA cryptosystem.

Let us define a Carmichael-type function μ on pairs (n, m) of *distinct* odd integers $n, m > 1$ by the formula

$$\mu(n, m) = \text{lcm}(n - 1, m - 1) \tag{9}$$

(hence μ equals the Carmichael function λ for n, m distinct primes). Now let n, m be two co-prime PRP2-integers, and set $N := n \cdot m$. Since $\mu(n, m) = a \cdot (n-1) = b \cdot (m - 1)$ for some integers $a, b \geq 1$, from the congruences

$$2^{n-1} \equiv 1 (\text{mod } n) \;\text{ and }\; 2^{m-1} \equiv 1 (\text{mod } m) \tag{10}$$

we obtain $2^{\mu(n,m)} \equiv 1 (\text{mod } n)$ and $2^{\mu(n,m)} \equiv 1 (\text{mod } m)$, i.e., the number $2^{\mu(n,m)} - 1$ is divided by both n and m, and hence by $N = n \cdot m$:

$$2^{\mu(n,m)} \equiv 1 (\text{mod } N). \tag{11}$$

Now we define two parameters e and d – the encryption and decryption keys, respectively – similarly as in the classic RSA system: we choose $e, d > 1$ from the multiplicative group $\mathbf{Z}^*_{\mu(n,m)}$ fulfilling the congruence

$$e \cdot d \equiv 1 (\text{mod } \mu(n, m)), \tag{12}$$

i.e.,

$$e \cdot d = 1 + k \cdot \mu(n, m) \;\text{ for some integer } k. \tag{13}$$

Further, with N as above, we define two functions E and D acting from the set of positive integers into positive real numbers:

$$E(x) = 2^{x \cdot e} (\text{mod } N), \text{ and } D(y) = \log_2(y^d (\text{mod } N)).$$

We claim that E and D are well defined encryption and decryption functions for all messages M less than $\log_2 N$. (Notice, however, that $D(y)$ is an integer if and only if $y^d (\mathrm{mod}\ N)$ is a power of 2, hence the proposed cryptosystem cannot be applied to digital signing, in general.) This is stated in the theorem below, and its proof is given in Section 3 of this paper.

Theorem. *In the notation as above, for every integer/message M with $1 < M < \log_2 N$, we have $E(D(M)) = M$.*

2 The Algorithm

In the description of the new algorithm we follow all steps of the RSA algorithm.

(KGA) Key Generation Algorithm

1. Generate two large co-prime PRP2-integers, n and m, of approximately equal size such that their product $N = n \cdot m$ is of the required bit length.
2. Compute $N = m \cdot m$ and $\mu(n, m) = \mathrm{lcm}(n - 1, m - 1)$.
3. Choose an integer $1 < e < \mu(n, m)$ such that $\gcd(e, \mu(n, m)) = 1$.
4. Compute $1 < d < \mu(n, m)$ such that $ed \equiv 1 (\mathrm{mod}\ \mu(n, m))$.
5. The public key is (e, N) and the private key is (d, N).

(E) Encryption

Sender X does the following:

1. Obtains the recipient Y's public key (N, e).
2. Represents the message as a positive integer M with $1 < M < \log_2 N$.
3. Computes $C = E(M) = 2^{e \cdot M} (\mathrm{mod}\ N)$.
4. Sends C to Y.

(D) Decryption

Recipient Y does the following:

1. Uses the private key (d, N) and computes the number $M_{(2)} = C^d (\mathrm{mod}\ N)$.
2. Computes $M = \log_2 M_{(2)}$.

Example. We give an example to show how the algorithm works in a concrete case.

Step (KGA). We have generated two small composite co-prime PRP2-numbers $n = 341$ and $m = 645$.

Hence $N = 219\ 945$ and $\mu(341, 645) = \mathrm{lcm}(340, 644) = 54\ 740$.

For $e = 257$, we obtain that $d = 213$ fulfills the congruence $ed \equiv 1 (\mathrm{mod}\ 54\ 740)$. Hence the public and private keys are $(257, 219\ 945)$ and $(213, 219\ 945)$, respectively. The system accepts messages $1 < M < \log_2 219\ 945) = 17.74...$

Step (E). Let $M = 15$. We encrypt M and compute $C = E(M) = 2^{257 \cdot 15} (\mathrm{mod}\ 219\ 945) = 175988$.

Step (D). We compute $M_{(2)} = 175988^{213} (\mathrm{mod}\ 219\ 945) = 32768$.

Finally, we compute $\log_2 M_{(2)} = \log_2 32768$ and obtain the sent message $M = 15$.

3 Correctness of the Algorithm – proof of the Theorem

Because the numbers $M_{(2)} = 2^M$ and N are co-prime, the 'message' $M_{(2)}$ lies in the multiplicative group $\mathbf{Z}_N^*$. Therefore every power of $M_{(2)}$ modulo N lies in $\mathbf{Z}_N^*$ too. Hence, the result of E, $C := E(M) = M_{(2)}^e (\mathrm{mod}\ N)$, belongs to $\mathbf{Z}_N^*$. Then the formula $C \to C^d (\mathrm{mod}\ N)$ sends C into an element of $\mathbf{Z}_N^*$, and the final element equals 2^M: by (11) and (13), we obtain

$$C^d \equiv M_{(2)}^{ed} \equiv M_{(2)}^{1+k\cdot\mu(n,m)} \equiv 2^{M+k\cdot M\cdot\mu(n,m)} \equiv$$

$$2^M \cdot (2^{\mu(n,m)}) \equiv 2^M \ (\mathrm{mod}\ N),$$

and the latter equals just 2^M because $2^M < N$. Therefore $D(E(M)) = \log_2 2^M = M$, as claimed.

References

1. Albrecht, R.M., Massimo, J., Paterson, K.G., Smorovsky, J.: *Prime and Prejudice: Primality Testing Under Adversarial Conditions*, In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, Toronto, Canada, October 15–19, 2018, 2018. https://doi.org/10.1145/3243734.3243787
2. Alford, W.R., Granville, A., and Pomerance, C.: *There are Infinitely Many Carmichael Numbers*, Annals of Math. **139** (1994), 703–722. https://doi.org/10.2307/2118576
3. Bleichenbacher, D.: *Breaking a Cryptographic Protocol with Pseudoprimes*, In: Vaudenay S. (eds) Public Key Cryptography - PKC 2005. PKC 2005. Lecture Notes in Computer Science, vol 3386. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-30580-4-2
4. Koblitz, N.: A Course in Number Theory and Cryptography, 2nd ed., Springer-Verlag, New York, 1994. https://doi.org/10.1007/978-1-4419-8592-7
5. Krizek, M., Luca, F., and Somer, L.: 17 Lectures on Fermat Numbers, Springer-Verlag, New York, 2001. https://doi.org/10.1007/978-0-387-21850-2
6. Maurer, U. M.: *Fast Generation of Prime Numbers and Secure Public-Key Cryptographic Parameters*, J. Cryptology, 8 (1995), 123–155. https://doi.org/10.1007/BF00202269
7. Shoup, V.: A Computational Introduction to Number Theory and Algebra, 2nd ed., Cambridge University Press, Cambridge, 2009. https://doi.org/10.1017/CBO9780511814549