During the preparation of the paper the author was supported by National Science Foundation under grant MCS-8305382.

# A CLASS OF CRYPTOSYSTEMS EQUIVALENT TO RSA

Evangelos Kranakis

Department of Computer Science, Yale University

Technical Report 316, April 1984

# A CLASS OF CRYPTOSYSTEMS EQUIVALENT TO RSA

**Evangelos Kranakis**[1]
Department of Computer Science
Yale University

April 1984

**Abstract**

*A new class of cryptosystems $\{RSA_m : m>0\}$ is defined which is inspired from the well known RSA cryptosystem ( i.e. the system due to Rivest, Shamir, and Adelman.) It is shown that for each m, $RSA_m$ is equivalent to RSA, in the sense that if there exists an efficient algorithm that can "break" every instance of RSA then there exists an efficient algorithm that can "break" every instance of $RSA_m$, and vice versa. Single $RSA_m$ bits are also studied.*

## 1. Introduction

The significance of the new results in public key cryptosystems is partly based on the fact that one makes the security of a cryptosystem dependant upon the difficulty of solving a problem in number theory. In other words one proves results of the following type: "if a certain problem in number theory is difficult to solve, then the given cryptosystem is difficult to break." Such an approach can be called "The Method of External Reduction", because one reduces the security of the given cryptosystem to an external problem (usually to one in number theory).

In the present paper a new approach is initiated in order to study the security of public key cryptosystems, called "The Method of Internal Reduction." According to this approach one tries to reduce the security of the given public key cryptosystem to another already existing one in order to determine which of the two systems is the most secure. For that purpose a new class of

---

[1]Many thanks to Mike Fischer, Dan Gusfield, Dana Angluin, and Eric Bach for many fruitful discussions on the contents of this paper.

cryptosystems is defined in section 2, denoted by $\{RSA_m\}$. In section 3 it is shown that each cryptosystem $RSA_m$ is equivalent to RSA. The security of single $RSA_m$ bits is studied in section 4. Security considerations of $RSA_m$ are further studied in section 5; in particular these include a generalization of Davida's cryptanalytic attack.

It would be interesting if one could find "internal reductions" between other well known public key cryptosystems. Throughout the present paper $Z_n^*$ will denote the set of all positive integers less than n which are relatively prime to n.

## 2. Encryption and Decryption of $RSA_m$

Each user U gets access to an integer $n_U$ which is the product of two odd primes $p_U$, $q_U$. The iteger $n_U$ is made public, but its factorization is not. For simplicity the subscript U will be omitted. Let $\phi(n) = (p-1)(q-1)$, where $\phi$ is the Euler function. Messages M are integers which are relatively prime to n. The **encryption key** is an m+1-tuple $n,e_1,...,e_m$ such that $\gcd(\phi(n),e_1,...,e_m) = 1$. Similarly, the **decryption key** is an m+1-tuple $n,d_1,...,d_m$ such that

$$d_1e_1+...+d_me_m = 1 \bmod \phi(n)$$

Each user makes his encryption key public, but keeps secret his decryption key.

**The encryption algorithm** $E_m$ is defined by

$$E_m(M) = <M^{e_1} \bmod n,...,M^{e_m} \bmod n>,$$

and **the decryption algorithm** $D_m$ is defined by

$$D_m(M_1,...,M_m) = M_1^{d_1} ... M_m^{d_m} \bmod n$$

An **instance** of $RSA_m$ is a 2m+1 tuple n, $e_1,...,e_m$, $M_1,...,M_m$, where n is the product of two distinct odd primes, $e_1,...,e_m < n$, $\gcd(n,e_1,...,e_m) = 1$, and there exists an integer N such that $N^{e_i} = M_i \bmod n$ for all i = 1,..,m. If m = 1, then $RSA_1$ is identical to the cryptosystem due to Rivest, Shamir, and Adelman and will be denoted by RSA (see [RSA].)

To show that the cryptosystem is well-defined it must be shown that

**Theorem 1:** (i) For all messages M, $D_m(E_m(M)) = M$

(ii) for all messages $M_1,...,M_m$ if there exists an M such that $M^{e_i} = M_i \bmod n$ for all $i = 1,...,m$, then $E_m(D_m(M_1,...,M_m)) = <M_1,...,M_m>$

**Proof:** (i) Let M be a given message. Then $E_m(M) = <M^{e_1} \bmod n,...,M^{e_m} \bmod n>$. Hence,

$$D_m(E_m(M)) =$$

$$D_m(M^{e_1} \bmod n,...,M^{e_m} \bmod n) =$$

$$M^{e_1 d_1} ... M^{e_m d_m} \bmod n =$$

$$M^{e_1 d_1 + ... + e_m d_m} \bmod n = M,$$

because $e_1 d_1 + ... + e_m d_m = 1 \bmod \phi(n)$, and $b^{\phi(n)} = 1 \bmod n$ for all integers b relatively prime to n, which completes the proof of part (i). The proof of part (ii) is similar.

## 3. Equivalence of the Cryptosystems

The purpose of the present section is to show that the cryptosystems $RSA_m$ are equivalent for each $m>0$. This assertion follows from the following two theorems

**Theorem 2:** If there is an efficient algorithm A which given as input an instance $n,e_1,...,e_m,M_1,...,M_m$ of $RSA_m$ will output a message N such that $N^{e_i} = M_i \bmod n$, for all $i=1,..,m$, in $S(n,e_1,...,e_m,M_1,M_2,...,M_m)$ steps, then there is an efficient algorithm $A_m$ which given as input an instance n,e,M of RSA will output a message N such that $N^e = M \bmod n$, in $S(n,e,...,e,M,M,...,M)$ steps.

**Proof:** The idea is to transform instances of one cryptosystem into instances of the other, and then use the algorithm which "breaks" one cryptosystem to devise an algorithm that will "break" the other cryptosystem. Let $A_m$, S be as in the hypothesis of the theorem, and let n,e,M be an instance of RSA. Put $e_1 = ... = e_m = e$, and let $M_1 = ... = M_m = M$. It is then clear that

$n,e_1,...,e_m,M_1,...,M_m$ form an instance of $RSA_m$. Hence it is enough to define $A(n,e,M) = A_m(n,e_1,...,e_m,M_1,...,M_m)$. This completes the proof of the theorem.

**Remark 1:** Define the set $MS(n,e_1,...,e_m,A_m) = \{x \in Z_n^* : A_m(n,e_1,...,e_m,x^{e_1} \bmod n,...,x^{e_m} \bmod n) = x\}$. Using the notation of theorem 2 it follows that for all $x \in Z_n^*$, $x \in MS(n,e,...,e,A_m)$ if and only if $x \in MS(n,e,A)$.

Converseley,

**Theorem 3:** If there is an efficient algorithm $A_m$ which given as input an instance n, e, M of RSA will output a message N such that $N^e = M \bmod n$, in S(n,e,M) steps, then there is an efficient algorithm A which given as input an instance $n,e_1,...,e_m,M_1,...,M_m$ of $RSA_m$ will output a message N such that $N^{e_i} = M_i \bmod n$ holds for all i=1,..,m, and the number of steps required is $S(n,\gcd(e_1,...,e_m),P(e_1,...,e_m,M_1,M_2,...,M_m)) + Q(e_1,...,e_m)$, where P, Q are polynomials.

**Proof:** Let A, S be as in the hypothesis of the theorem, and let $n,e_1,...,e_m,M_1,...,M_m$ be an instance of $RSA_m$. The algorithm $A_m$ is defined as follows:

**Input:** $n,e_1,...,e_m,M_1,...,M_m$

**Step 1:** Compute $e = \gcd(e_1,...,e_m)$

**Step 2:** Compute $k_1,...,k_m$ such that $k_1e_1+...+k_me_m = e$

**Step 3:** Compute $M = M_1^{k_1} ... M_m^{k_m}$

**Output:** A(n,e,M)

It remains to show that the above algorithm works. Let $n,e_1,...,e_m,M_1,...,M_m$ be an instance of $RSA_m$. It will be shown that the above algorithm outputs an integer N such that $N^{e_i} = M_i \bmod n$ for all $i = 1,...,m$. Since $n,e_1,...,e_m,M_1,...,M_m$ is an instance of $RSA_m$, there exists an integer N

such that $N^{e_i} = M_i \bmod n$ for all i = 1,...,m. Compute $e,k_1,...,k_m$ as in the above algorithm. Then it can be shown that

$$N^e =$$
$$N^{k_1e_1+...+k_me_m} =$$
$$(N^{k_1})^{e_1}...(N^{k_m})^{e_m} =$$
$$M_1^{k_1}...M_m^{k_m} =$$
$$= M \bmod n.$$

It follows that n,e,M is an instance of RSA. Consequently A(n,e,M) = N, and the proof of the theorem is complete. The polynomials P, Q depend on standard algorithms for computing the greatest common divisor (e.g. see [A].)

**Remark 2:** Using the notation of theorem 2 and remark 1 it follows that for all $x \in Z_n^*$, $x \in MS(n,e_1,...,e_m,A_m)$ if and only if $x \in MS(n,\gcd(e_1,...,e_m),A)$.

## 4. Single $RSA_m$ bits

The results of Goldwasser, Micali, and Tong about the security of single RSA bits, can now be generalized to the context of the $RSA_m$ cryptosystems (see [GMT].) In particular it will be shown that inverting the $RSA_m$ system is "equivalent" to computing certain functions f with domain $Z_n^* \times ... \times Z_n^*$ and values in {0,1}. The proof of the main theorem of this section is very similar to the corresponding result on RSA, and the reader is advised to consult [GMT] for further details.

Define the following functions

**Last Bit Function:** $LB_{n,e_1,...,e_m}(x^{e_1} \bmod n,...,x^{e_m} \bmod n) = 0$ if x is even, and 1 if x is odd.

**Location Function:** $LOC_{n,e_1,...,e_m}(x^{e_1} \bmod n,...,x^{e_m} \bmod n) = 0$ if $x < n/2$, and 1 if $x > n/2$.

**Significant Bit Function:** Let $B(n) = b_{k-1}...b_1b_0$ be the representation of the number n in the binary system, and let t(n) = the largest value of t such that $b_{t+1} = 0$, $b_t = 1$. The

significant bit function corresponding to $t \leq t(n)$ is defined by $SB_{t,n,e_1,...,e_m}(x^{e_1}\text{mod}n,...,x^{e_m}\text{mod}n)$ = the t-th bit in the binary representation of x. Notice that $SB_{0,n,e_1,...,e_m} = LB_{n,e_1,...,e_m}$.

As in [GMT] one can prove the following

**Theorem 4:** Given an m-tuple $e_1,...,e_m$ of odd integers $< \phi(n)$ such that $\gcd(e_1,...,e_m,\phi(n)) = 1$, and any $t \leq t(n)$ the following statements are equivalent

(i)There is an efficient algorithm $A_m$ such that for all x in $Z_n^*$, $A_m(x^{e_1}\text{mod}n,...,x^{e_m}\text{mod}n) = x$.

(ii)There is an efficient algorithm computing the function $LB_{n,e_1,...,e_m}$.

(iii)There is an efficient algorithm computing the function $LOC_{n,e_1,...,e_m}$.

(iv)There is an efficient algorithm computing the function $SB_{t,n,e_1,...,e_m}$.

**Proof:**(outline) It is obvious that (i) implies each of the statements (ii), (iii) and (iv). For each $i = 1,...,m$ let $I_i$ = the unique integer such that $I_i 2^{e_i} = 1\text{mod}n$. The equivalence of (ii) and (iii) is a consequence of the following easily proved

**Claim 1:** For all x in $Z_n^*$, $x < n/2$ if and only if $2x\text{mod}n$ is even.

It follows that $LB_{n,e_1,...,e_m}(x^{e_1}\text{mod}n,...,x^{e_m}\text{mod}n) = LOC_{n,e_1,...,e_m}(I_1x^{e_1}\text{mod}n,...,I_mx^{e_m}\text{mod}n)$. This proves the equivalence of (ii) and (iii).

To prove that (ii) implies (i) the following claim will be needed

**Claim 2:** For all x in $Z_n^*$, and all $i = 1,...,m$ the following hold:

(a) $n-x^{e_i} = (n-x)^{e_i}$ modn

(b) $I_ix^{e_i} = (x/2)^{e_i}$ modn, if x is even.

(c) $I_i(n-x^{e_i}) = ((n-x)/2)^{e_i}$ modn, if x is odd.

**Proof of the claim:** To prove (a) expand $(n-x)^{e_i}$ using the binomial theorem. It follows that $(n-x)^{e_i} = (-1)^{e_i}x^{e_i} = -x^{e_i} = (n-x^{e_i})$ modn, because $e_i$ is odd. To prove (b) write $x = 2u$. Then $I_i x^{e_i} = I_i 2^{e_i}u^{e_i} = u^{e_i}$ modn. The proof of (c) is an immediate consequence of (a) and (b).

The main idea needed in the proof of the implication: (ii) implies (i) is included in the following procedure

**Input:** $x_1 = x^{e_1}$modn,...,$x_m = x^{e_m}$modn

**Step 1:** Compute $LB_{n,e_1,...,e_m}(x_1,...,x_m)$

**Step 2:** If the result of the computation in step 1 is 0 then set $x_i := I_i x_i$modn for i = 1,...,m and **goto** step 1. If the result of the computation in step 1 is 1 then set $x_i := I_i(n-x_i)$modn for i = 1,...,m and **goto** step 1.

If k is the binary length of n then k applications of the above loop will output from step 1 a sequence of k bits, which gives the binary representation of x, if the original input were $x^{e_1}$modn,...,$x^{e_m}$modn. Details of a more formal proof can be derived by an argument similar to that in [GMT].

(iv) is in essence a generalization of (ii). The proof of (iv) implies (i) is similar to the proof of (ii) implies (i) above. Further formal details can be found in [GMT]. This completes the outline of the proof of theorem 4.

## 5. Security of $RSA_m$

The reader should take into account the security considerations of RSA regarding factoring n, computing $\phi(n)$ without factoring n, and computing $d_1,...,d_m$ without factoring n or computing $\phi(n)$, as those are described in [RSA].

The proof of theorem 3 shows that if $\gcd(e_1,...,e_m) = 1$, then one can efficiently compute M

from the given $M^{e_1},...,M^{e_m}$. Indeed, compute $k_1,...,k_m$ such that $e_1k_1+...+e_mk_m = 1$. Then it is clear that $M = (M^{e_1})^{k_1}...(M^{e_1})^{k_1} \bmod n$.

Since the systems RSA and $RSA_m$ are equivalent it is expected that both systems will suffer from the same shortcomings. The following cryptanalytic attack on $RSA_m$ is a generalization of a corresponding cryptanalytic attack on RSA, due to Davida (see [DA], [DE].) It is based on the fact that a cryptanalyst can intercept the transmission between the Sender and the Receiver. The cryptanalyst can compute M by following the procedure below:

**Step 1:** Intercept $M^{e_1},...,M^{e_m}$.

**Step 2:** Pick any message X such that gcd(X,n) = 1.

**Step 3:** Compute $X^{-1} \bmod n, X^{e_1} \bmod n,...,X^{e_m} \bmod n$.

**Step 4:** Ask the Receiver to sign the messages $(XM)^{e_1} \bmod n,...,(XM)^{e_m} \bmod n$.

**Step 5:** Intercept the Receiver's response: $((XM)^{e_1})^{d_1} \bmod n,...,((XM)^{e_m})^{d_m} \bmod n$.

**Step 6:** Obtain M through the equation:

$M = X^{-1}((XM)^{e_1})^{d_1} \bmod n...((XM)^{e_m})^{d_m} \bmod n$.

For a further discussion see [DE].

## 6. References


[D] **D. Angluin**, Lectures Notes on the Complexity of some problems in Number Theory, Technical Report 243, Yale University, August 1982.

[DA] **G.L. Davida,** Chosen Signature Cryptanalysis of the RSA (MIT) Public Key Cryptosystem, Dept of EE and CS, Univ. of Wisconsin, Milwaukee, TR-CS-82-2, October 1982.

[DE] **D.E. Denning,** A note on Strengthening RSA and Other Public-Key Cryptosystems, Dept. of CS, Purdue Univ., TR-CSD-419, October 1982.

[GMT] **S. Goldwasser, S. Micali, and P. Tong,** Why and how to establish a Private Code on a Public Network, in 23rd IEEE Symp. on Foundations of Computer Science, pp. 134-144, IEEE 1982.

[RSA] **R.L. Rivest, A. Shamir, and L. Adleman,** A method for obtaining Digital Signatures and Public Key Cryptosystems, Communications of the ACM, vol 21, number 2(February 1978), pp. 120-126.