

To the Graduate Council:

I am submitting herewith a thesis written by Janet Burstein Svirsky entitled "The Pseudo-RSA Scheme: A Suggested Improvement to the RSA Cryptography Scheme." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Computer Science.

David Sragk

We have read this thesis
and recommend its acceptance:

David C. Mutchler

Jan R. S. Blair

Accepted for the Council:

Cowminkel
Vice Provost
and Dean of The Graduate School

THE PSEUDO-RSA SCHEME: A SUGGESTED IMPROVEMENT
TO THE RSA CRYPTOGRAPHY SCHEME

A Thesis
Presented for the
Master of Science
Degree
The University of Tennessee, Knoxville

Janet Burstein Svirsky

May 1991

ACKNOWLEDGEMENT

The work described in this thesis was performed under the project:
Probabilistic Algorithms Used in Public Key Cryptosystems, directed by V.R.R.
Uppuluri, sponsored by the Seed Money Program at the Oak Ridge National
Laboratory.

ABSTRACT

In this thesis, some suggested improvements are made to the RSA scheme of Rivest, Shamir, and Adleman. In particular, it is suggested that the "primes", p , be selected using pseudoprimality testing with tests for small factors of $p - 1$ and $p + 1$ as opposed to the Solovay-Strassen or Rabin Tests. This requires additional greatest common divisor tests. It is also suggested that the Euler function $\phi(n)$ be replaced by the Carmichael function $\lambda(n)$ (or its generalization $L(n)$) in the production of encryption and decryption keys. In addition, further greatest common divisor tests are proposed when encoding to increase security.

TABLE OF CONTENTS

CHAPTER	PAGE
INTRODUCTION	1
1. TYPES OF PSEUDOPRIMES AND RELATED PRIMALITY TESTS.	5
0. Preliminary Definitions and Theorems.	5
1. Fermat Pseudoprimes.	8
2. Euler Pseudoprimes	11
3. Strong Pseudoprimes.	12
2. THE RSA ENCRYPTION SCHEME AND A SUGGESTED IMPROVEMENT.	24
1. Public-key Cryptosystems.	24
2. The RSA Scheme.	28
3. Suggested Improvements, the Pseudo-RSA Scheme.	36
3. SUGGESTED FURTHER WORK.	45
BIBLIOGRAPHY.	48
VITA.	50

SUMMARY OF STUDY

The RSA scheme of Rivest, Shamir, and Adelman, a public key cryptosystem, rests upon certain concepts and theorems of Algebraic Number Theory requiring some mathematical background mainly contained in Chapter 1 with a mathematical digression in Chapter 2. The description of public key cryptosystems, the RSA scheme, and its proposed modification, the pseudo-RSA scheme, can be found in Chapter 2.

The pseudo-RSA scheme is an attempt to remedy some of the problems with the RSA scheme. It rests on the definition of a function $L(n)$, a generalization of the Carmichael function, and the congruency relation it is shown to satisfy. Its use as a substitution for the Euler function in the RSA scheme reduces the risk of a message being recovered by re-encryption and copes with the possibility that the primes required in the creation of the coding keys are in fact pseudoprimes, provided that they are sufficiently 'primelike'. Some other modifications are proposed to ensure that the hypothesis of needed theorems are met, and to enhance security by reducing the risk of factoring.

INTRODUCTION

The RSA scheme of Rivest, Shamir, and Adleman is an implementation of a public-key cryptosystem, in which the coding keys, consisting of pairs of positive integers, say (e, n) for encryption and (d, n) for decryption, satisfy the following criteria. The n appearing in both pairs is the product of two very large "primes" p and q , d is chosen to be a large random integer relatively prime to the Euler function $\phi(n)$, which equals $(p - 1) \times (q - 1)$ if p and q are really prime, and e is then computed from p , q , and d to satisfy $e \times d \equiv 1 \pmod{\phi(n)}$. The message is first represented as a series of blocks, each represented by an integer between 0 and $n - 1$. It is encoded by raising each converted block to the power e modulo n and decoded by raising each enciphered block to the power d modulo n . Due to the symmetry, the methods for computing d and e may be interchanged. The difficulties, as described below, lie in the choices of p and q , $\phi(n)$, and d or e .

The first problem is that of finding sufficiently large prime numbers, p and q , so that it is infeasible for anyone to factor $n = p \cdot q$. The required size of p and q renders the use of a deterministic algorithm infeasible. Instead a probabilistic method is proposed known as the Rabin Test. This test reduces to the Solovay-Strassen Test, described in Chapter 1, Section 2, when $n \equiv 3 \pmod{4}$. These algorithms are series of compositeness tests that check to see if the candidate prime is a pseudoprime for various bases, i.e. that its behavior is "sufficiently primelike".¹ It has been popularly believed that the

¹Various types of pseudoprimes are described in Chapter 1.

probability of a composite number being incorrectly labeled "prime" by Rabin's algorithm is $\leq 4^{-k}$ if k test bases are used. However, this is the result of confusion between two conditional probabilities. The number of false primes returned by the algorithm is calculated in [1] and appears here as Theorem 1.3C.2. Finally, there is no attempt made to deal with the possibility that at least one of p and q is composite in the conventional RSA scheme.

The second problem is the choice of d or e . Rivest, Shamir, and Adleman suggest that d be chosen to be any prime greater than $\max(p, q)$. This is at least as hard as finding p and q .

The remaining problem to be discussed is that of fixed points. By a generalization of Euler's Theorem, $a^{\phi(n)+1} \equiv a \pmod{n}$; that is, a message block a is recovered by raising it to the $\phi(n) + 1$ power modulo n . For a specific a , a smaller exponent might work, and when at least one of p and q is composite, a smaller exponent will work. Thus the message may be recovered by re-encryption or the cipher text may be the same as the message, i.e., a fixed point. Some fixed points are unavoidable, but could be reduced by replacing $\phi(n)$ with a function that is a divisor satisfying a similar congruence relation.

This thesis tries to address these problems by proposing the pseudo-RSA scheme, described in Chapter 2, Sections 3B-E. Before it can be explained, the Carmichael function $\lambda(n)$ needs to be described. This function meets the requirements mentioned above, i.e. it is a generalization of the Euler function of which it is also a divisor. It also satisfies a similar congruence relation, namely $a^{\lambda(n)+1} \equiv a \pmod{n}$ if n is a product of distinct primes. The Carmichael Function has a further generalization $L(n)$ satisfying an analogous congruence

which appears as Theorem 2.3A.3. This theorem handles the possibility that one or more of the "prime" factors of n may in fact be a Fermat pseudoprime to the base a . The Fermat pseudoprime is described in Definition 1.1.1.

The pseudo-RSA scheme replaces $\phi(n)$ with $\lambda(n)$, or more generally, $L(n)$. This means that e and d are chosen so that $e \times d \equiv 1 \pmod{\lambda(n)}$. Some additional suggestions are made for the choice of e and d as well. For $n = p \cdot q$, p, q , odd primes, $L(n) = \lambda(n) = (p-1)(q-1)/\gcd(p-1, q-1)$ which is a proper divisor of $\phi(n)$ so this results in fewer fixed points and may result in less calculation since for a particular d , the smallest possible e chosen using $\lambda(n)$ is less than or equal to the one found using $\phi(n)$. In choosing p and q , it begins by using the Sieve of Erasthosthenes with small prime divisors on an interval of large numbers and testing the remaining ones for pseudoprimality using several bases. It also adds tests for small factors of $P \pm 1$ of a candidate prime P as well as tests for greatest common divisors. Small factors of $p-1$ and $q-1$ of the chosen pair (p, q) increase the number of fixed points, and small factors of $p \pm 1$, and $q \pm 1$ make it easier to factor n . One greatest common divisor test is that of p and q as both a check for primality and to ensure that the congruences for $\lambda(n)$ and $L(n)$ hold. Others are used to increase security by checking for factors common to message blocks and n . Such a common factor reveals a divisor of n .

Suppose this method selects a pseudoprime instead of an actual prime. The resulting n will be easier to factor since it has more and smaller divisors and there will be more fixed points. However there should be no more errors than in the RSA scheme and probably less. If the pseudoprime is a Fermat pseudoprime for all the message blocks and $\lambda(n)$ is replaced by $L(n)$, the

coding method will work properly. If the pseudoprime is a Carmichael number, as defined in Definition 1.1.1, Corollary 2.3A.1 shows that the coding method will work for all message blocks. In addition, this method is simpler. Of course, for complete confidence that a given message has been encrypted properly, the user can either check Fermat pseudoprimality using the message blocks as bases or decrypt the encrypted message, the easier choice being dependent on the relative sizes of n and d .

CHAPTER 1

TYPES OF PSEUDOPRIMES AND RELATED PRIMALITY TESTS

0. Preliminary Definitions and Theorems

This section contains definitions and theorems needed to understanding the following descriptions of various types of pseudoprimes and related primality tests. A pseudoprime is an integer that satisfies a necessary but not sufficient condition for primality. These conditions can be used to develop primality tests and are the bases for the compositeness tests used to select 'primes' in the RSA scheme. The pseudo-RSA scheme also includes tests for Fermat pseudoprimes and allows coding using Carmichael numbers.

The following can be found in many books on number theory; see [5] for example.

Definition 1.0.1: For a prime $p > 2$, and integer a , $0 < a < p$, we say that a is a quadratic residue if $x^2 \bmod p = a$ has a solution, and that a is a quadratic nonresidue otherwise.

If $x^2 \bmod p = a$ has one solution, then it has exactly two. Half of the elements a , $0 < a < p$, are quadratic residues and half are not.

Definition 1.0.2: The Legendre Symbol, $\left(\frac{a}{p}\right)$, is defined as follows:

$$\left(\frac{a}{p}\right) = \begin{cases} 1, & \text{if } a \text{ is a quadratic residue} \\ -1, & \text{if } a \text{ is a quadratic nonresidue.} \end{cases}$$

Theorem 1.0.1: (Euler): $\left(\frac{a}{p}\right) \equiv a^{(p-1)/2} \pmod{p}$.

More generally,

Theorem 1.0.2: Let $L > 0$, $p \mid n$. If $p > 2$, then the congruence $x^2 \equiv n \pmod{p^L}$ has $1 + \left(\frac{n}{p}\right)$ solutions.

Definition 1.0.3: The Jacobi Symbol $\left(\frac{a}{n}\right)$ is defined as follows:

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right) \left(\frac{a}{p_2}\right) \dots \left(\frac{a}{p_L}\right)$$

where the p_i are the prime, possibly repeating, factors of n ,

i.e., $n = p_1 p_2 \dots p_L$.

The Jacobi symbol can be evaluated recursively:

Writing $\left(\frac{a}{n}\right)$ as $J(a, n)$:

if $a = 1$ then $J := 1$

else if $a \pmod{2} = 0$ then begin ;

if $(n \cdot n - 1)/8 \pmod{2} = 0$

then $J := J(a/2, n)$

else $J := -J(a/2, n)$ end .

else if $(a - 1) \cdot (n - 1)/4 \pmod{2} = 0$

then $J := J(n \pmod{a}, a)$

else $J := -J(n \pmod{a}, a)$.

Definition 1.0.4: Let $\mathbb{Z}_n$ denote the ring of integers $\pmod{n}$. Let $\mathbb{Z}_n^\times$ denote the group of integers relatively prime to n under multiplication $\pmod{n}$.

Suppose p is prime. Suppose b is a generator of $\mathbb{Z}_p^\times$, a cyclic group of order $p - 1$. Then the index of $a \pmod{p}$ is defined as follows:

$$\text{ind}_p a = \text{ind}_p a(b) = \min\{m: b^m \equiv a \pmod{p}\} .$$

We say that a is a q^{th} residue $\pmod{p}$ if there exists a b such that $b^q \equiv a \pmod{p}$.

If p and q are prime and $q|(p - 1)$, then a is a q^{th} residue $\pmod{p}$ if and only if $q|\text{ind}_p a$.

1. Fermat Pseudoprimes

Fermat proved that for p , prime

$$a^{p-1} \equiv 1 \pmod{p}, \text{ if } (a, p) = 1.$$

(Note that here and elsewhere (a, b) is a shortened notation for $\gcd(a, b)$).

Thus, if for some a , $1 < a < n$,

$$a^{n-1} \not\equiv 1 \pmod{n} \tag{1}$$

then n must be composite. There are two problems with using Fermat's theorem as a primality test:

(a) A systematic search for an a satisfying (1) could be very lengthy.

(b) There are composite n which satisfy Fermat's congruence, i.e., there are composite n 's such that

$$a^{n-1} \equiv 1 \pmod{n} \text{ for all } (a, n) = 1.$$

Definition 1.1.1.: An odd composite number N for which

$$a^{N-1} \equiv 1 \pmod{N}$$

is called a (Fermat) pseudoprime for the base a [see 11]. If this congruence holds for all a relatively prime to N , N is called a Carmichael number.

Definition 1.1.2: Let $n = p_1^{v_1} p_2^{v_2} \dots p_n^{v_n}$ be the prime factorization of the odd number n . Then Carmichael's λ -function,

$$\lambda(n) = \text{lcm}\{p_1^{v_1-1} (p_1 - 1), p_2^{v_2-1} (p_2 - 1), \dots, p_n^{v_n-1} (p_n - 1)\}.$$

Theorem 1.1.1: N is a Carmichael number $\Leftrightarrow \lambda(N) \mid (N - 1)$.

Theorem 1.1.2: Such an N must be a product of at least three distinct prime factors.

Some examples of Carmichael numbers are listed below:

$$3 \cdot 11 \cdot 17, 5 \cdot 13 \cdot 17, 5 \cdot 17 \cdot 29, 7 \cdot 13 \cdot 19, 7 \cdot 19 \cdot 67, \\ 7 \cdot 31 \cdot 73$$

[See 2].

The following inequalities have been proved so far concerning the number of Fermat pseudoprimes of base $a \leq x$, $P_a(x)$, and the number of Carmichael numbers $\leq x$, $C(x)$:

$$e^{(\ln x)^{5/14}} \leq P_2(x) \leq x e^{-1/2 \ln x} \ln x / \ln_2 x \\ C(x) \leq x e^{-\frac{\ln x}{\ln_2 x} (\ln_3 x + \ln_4 x + \frac{\ln_4 x - 1}{\ln_3 x} + O(\frac{\ln_4 x}{\ln_3 x})^2)}$$

where $\ln_2 x = \ln \ln x$, etc.

Comparing these expressions with the density of prime numbers, $\Pi(x) \sim x / \ln x$,

[see 5], where $\Pi(x)$ denotes the number of primes not exceeding x , the pseudoprimes described above are rather rare.

Examples:

$$(i) \ x = 10^{20}, P_2(x)/\Pi(x) \leq 1.43 \cdot 10^{-2}$$

$$C(x)/\Pi(x) \leq 7.2 \cdot 10^{-5}$$

$$(ii) \ x = 10^{50}, P_2(x)/\Pi(x) \leq 7.2 \cdot 10^{-7}$$

$$C(x)/\Pi(x) \leq 5.7 \cdot 10^{-16}$$

[See 11, pp. 101-102] .

If all the pseudoprimes for some particular base were known, then using Fermat's congruence together with a list of pseudoprimes would give a primality test. This technique was used by D.H. Lehmer who made a table of all Fermat pseudoprimes below $2 \cdot 10^8$ for the base 2 with no factor < 317 . Lehmer tested numbers between 10^7 and $2 \cdot 10^8$ since the existing prime number tables stopped at 10^7 . His scheme was the following:

- (1) Perform trial divisions by the 65 primes ≤ 313 . If a divisor is found, N is composite.
- (2) Compute $2^{N-1} \pmod{N}$. If the residue is $\neq 1$, then N is composite.
- (3) For the residue 1, check Lehmer's table of pseudoprimes. If N is in the table, it is composite and its factors are given in the table. If N is not in the table, N is prime.

This test, of course, only gives a definite answer for numbers of limited size
[see 11, p. 92] .

2. Euler Pseudoprimes

According to Euler's criterion for quadratic residues,

$$a^{(N-1)/2} \equiv \left(\frac{a}{N}\right) \equiv \pm 1 \pmod{N} \text{ for } (a, N) = 1 \quad (2)$$

if N is an odd prime. Hence (2), as well as Fermat's theorem, can be used as a test for compositeness on N .

Theorem 1.2.1: If N is odd, $(a, N) = 1$ and $a^{(N-1)/2} \not\equiv \pm 1 \pmod{N}$, then N is composite.

If $a^{(N-1)/2} \equiv \pm 1 \pmod{N}$, then if $a^{(N-1)/2} \not\equiv \left(\frac{a}{N}\right) \pmod{N}$, where $\left(\frac{a}{N}\right)$

is the Jacobi Symbol, N is composite.

If $a^{(N-1)/2} \equiv \left(\frac{a}{N}\right) \pmod{N}$, then the test is not conclusive.

Definition 1.2.1: An odd composite number N such that $a^{(N-1)/2} \equiv \left(\frac{a}{N}\right) \pmod{N}$, with $(a, N) = 1$, is called an Euler pseudoprime for the base a .

Some Carmichael numbers and other Fermat pseudoprimes can be shown to be composite using Euler's Criterion so it distinguishes primes from composites more accurately than does the Fermat congruence. Some examples are:

$$(i) \ 2^{170} \equiv 1 \pmod{341}, \left(\frac{2}{341}\right) = -1$$

$$(ii) \ 5^{280} \equiv 67 \pmod{561} \not\equiv \pm 1 \pmod{561}$$

$$(iii) \ 11^{864} \equiv 1 \pmod{1729}, \left(\frac{11}{1729}\right) = -1.$$

The numbers 561 and 1729 are Carmichael numbers. [See 11, p. 97].

Euler's Criterion was the motivation for the Solovay-Strassen Test [see 13]:

If $(a, n) = 1$, compute $\epsilon = a^{(n-1)/2} \pmod{n}$, $-1 \leq \epsilon < n - 2$.

Let $\delta = \left(\frac{a}{n}\right)$, the Jacobi Symbol.

If $\epsilon = \delta$, i.e., $\left(\frac{a}{n}\right) = a^{(n-1)/2} \pmod{n}$, decide that n is prime.

Decide that n is composite if either $\gcd(a, n) > 1$, or $\epsilon \neq \delta$.

Solovay and Strassen in [3] analyze the error probability by showing that the set

$$G_n = \{a \in \mathbb{Z}_n \mid (a, n) = 1, \text{ and } a^{(n-1)/2} \equiv \left(\frac{a}{n}\right) \pmod{n}\} \neq \mathbb{Z}_n^\times,$$

the subgroup of units of $\mathbb{Z}_n$. Thus $|G_n| \leq \frac{1}{2} |\mathbb{Z}_n^\times| \leq (n-1)/2$ so that at most

$1/2$ of the numbers between 1 and $n-1$ will lead to the decision that n is prime.

3. Strong Pseudoprimes

Definition 1.3.1: An odd composite number N with $N-1 = d \cdot 2^s$, d odd, is called a strong pseudoprime for the base a if either

$$a^d \equiv 1 \pmod{N} \quad \text{or} \quad a^{d \cdot 2^r} \equiv -1 \pmod{N},$$

for some $r = 0, 1, 2, \dots, s - 1$.

Note: There are only 13 numbers below $25 \cdot 10^9$ which are strong pseudoprimes for bases 2, 3, and 5 simultaneously. [See 11, pg. 98].

This fact can be used to determine whether any number below $25 \cdot 10^9$ is either composite or prime as follows:

1. Check whether N is a strong pseudoprime for the base 2. If not, then N is composite.
2. Check whether N is a strong pseudoprime for the base 3. If not, then N is composite.
3. Check whether N is a strong pseudoprime for the base 5. If not, then N is composite.
4. If N is one of the thirteen pseudoprimes, then N is composite; otherwise determine that N is prime.

If the strong pseudoprime test also includes bases 7 and 11, then no table look-up is needed below $25 \cdot 10^9$. [See 11, p. 98]. The strong pseudoprime test is more stringent than the Euler pseudoprime test since in [9], it is shown that every strong pseudoprime for the base a is also an Euler pseudoprime for the base a .

For large primes, the more pseudoprime tests that are performed, the more likely the prime candidate is prime but this in itself is not a mathematical proof. As yet there is no method to select a combination of pseudoprime tests so that a

particular candidate is not a pseudoprime in all the tests. The closest to such a method is Miller's Test. Assuming the truth of the Extended Riemann Hypothesis, Miller in [8] shows that consecutive, strong pseudoprime tests give a deterministic primality test taking polynomial time. The following is a description of his approach.

3.A. Miller's Test

Definition 1.3A.1: Let $n = p_1^{v_1} p_2^{v_2} \dots p_m^{v_m}$ be the prime factorization of the odd number n .

- (i) Euler's ϕ -function: $\phi(n) = p_1^{v_1-1} (p_1 - 1) p_2^{v_2-1} (p_2 - 1) \dots p_m^{v_m-1} (p_m - 1)$.
- (ii) Miller's λ' -function: $\lambda'(n) = \text{lcm}\{(p_1 - 1), (p_2 - 2), \dots, (p_m - 1)\}$.

As was discussed in Section 1, composite numbers n fall into two cases, those such that $\lambda(n) \nmid (n - 1)$ which can be found using Fermat's Theorem, and those such that $\lambda(n) \mid (n - 1)$ which cannot. In Miller's paper, $\lambda'(n)$ is used to classify composite numbers into classes instead of $\lambda(n)$. In fact, he shows that if n is composite then either

- (i) $n = p^\alpha$
- (ii) $\lambda'(n) \mid (n - 1)$, or
- (iii) $\lambda'(n) \nmid (n - 1)$ and not a prime power.

First consider the case when $\lambda'(n) \mid (n - 1)$.

Lemma 1.3A.1: If $\lambda'(n) \nmid (n-1)$ then there exist primes p, q such that

(1) $p|n$, $(p-1) \nmid (n-1)$, $q^m | (p-1)$ and $q^m \nmid (n-1)$ for some integer $m \geq 1$;

(2) If a is any q^{th} nonresidue mod p then $a^{n-1} \not\equiv 1 \pmod{n}$.

Definition 1.3A.2: Let $N(p, q)$ be the least a such that a is a q^{th} nonresidue mod p defined only when $q | (p-1)$. Note: Miller claims that $N(p, q)$ is prime.

Theorem 1.3A.1 (Ankeny): With the Extended Riemann Hypothesis,

$$N(p, q) = O(|p|^2).$$

So, if $\lambda'(n) \mid (n-1)$, there exists an $a \leq O(|n|^2)$ such that $a^{n-1} \not\equiv 1 \pmod{n}$.

Next, consider the case when $\lambda(n) \mid (n-1)$. Let $q_1, q_2, \dots, q_m$ be the distinct prime factors of n .

Definition 1.3A.3: Let $\#_2(n) = \max\{k : 2^k | n\}$, the number of factors of two in n .

Note: By the definition of $\lambda'(n)$,

$\#_2(\lambda'(n)) = \max\{\#_2(q_1 - 1), \#_2(q_2 - 1), \dots, \#_2(q_m - 1)\}$. Thus for some $i, 1 \leq i \leq m$,

$\#_2(\lambda'(n)) = \#_2(q_i - 1)$.

Definition 1.3A.4: We say n is of type A if there exists j such that

$$\#_2(\lambda'(n)) > \#_2(q_j - 1) ,$$

and that n is of type B if

$$\#_2(\lambda'(n)) = \#_2(q_i - 1) , \forall i , 1 \leq i \leq m .$$

The strategy of the next three lemmas is motivated by the following observation.

Suppose $n = pq$ and there exists an m such that

$$m \equiv 1 \pmod{q} \text{ and } m \equiv -1 \pmod{p} . \quad (3)$$

Then $m \not\equiv 1 \pmod{n}$ and then $q = (m - 1, n)$. So if we could compute some m satisfying (3), we would know a divisor of n .

In the following lemmas, Miller develops a method for finding such an m .

Lemma 1.3A.2A: Let n be a composite number of type A where, say, $pq \mid n$ and $\#_2(\lambda'(n)) = \#_2(p - 1) > \#_2(q - 1)$. Assume further that there is an a , $0 < a < n$ such that $\left(\frac{a}{p}\right) = -1$, where $\left(\frac{a}{p}\right)$ is the Jacobi symbol. Then either (a, n) or $((a^{\lambda'(n)/2} \pmod{n}) - 1, n)$ is nontrivial (i.e., not equal to 1).

Lemma 1.3A.2B: Let n be a composite number such that $pq \mid n$. Suppose n is of type B and $1 < a < n$ is such that

$$\left(\frac{a}{pq}\right) = \left(\frac{a}{p}\right) \left(\frac{a}{q}\right) = -1.$$

Then either (a, n) or $((a^{\lambda'(n)/2} \pmod{n}) - 1, n)$ is nontrivial.

Lemma 1.3A.3: If $p \mid n$, $\lambda'(n) \mid m$ and $k = \#_2[m/\lambda'(n)] + 1$, then $a^{\lambda'(n)/2} \equiv a^{m/2^k} \pmod{p}$.

Remark: This lemma allows us to find m without knowing $\lambda'(n)$ which requires knowing the prime factors of n . When $\lambda'(n) \mid (n-1)$, put $m = n-1$.

Thus if n is type A and composite, $\lambda'(n) \mid (n-1)$ and $a = N(p, 2)$, the least a such that a is a quadratic nonresidue mod p , defined only when p is odd, then either $a \mid n$ or $((a^{(n-1)/2} \pmod{n}) - 1, n) \neq 1, n$.

For type B, we need the following definition.

Definition 1.3A.5: Let $N(pq)$ be the minimum a such that $\left(\frac{a}{pq}\right) \neq 1$, where $\left(\frac{a}{pq}\right)$ is the Jacobi symbol, and $N(pq)$ is defined only when $p \neq q$.

Note: Miller claims that $N(pq)$ is prime.

Theorem 1.3A.2 (Ankeny): Under the Extended Riemann Hypothesis,

$$N(pq) = O(|pq|^2) .$$

This results in the following deterministic algorithm.

Let f be a computable function on the natural numbers.

1. Check if n is a perfect power, i.e. $n = m^s$, $s \geq 2$. If n is a perfect power, return composite and halt.
2. Carry out steps (i) - (iii) for each $a \leq f(n)$. If at any stage (i), (ii), or (iii) gives composite, halt.
 - (i) $a|n$
 - (ii) $a^{n-1} \not\equiv 1 \pmod{n}$
 - (iii) $1 < ((a^{n-1})2^k \pmod{n} - 1, n) < n$ for some k , $1 \leq k \leq \#2|(n-1)$,
i.e., $2^k|(n-1)$.
3. Output "prime" and halt.

Note: In (iii), $k > 1 \Leftrightarrow n \equiv 1 \pmod{4}$.

Miller also has an "improved" algorithm that says that only primes need be tested. However, this requires knowing the primes $\leq \phi(n)$. [See 8].

3B. Rabin's Algorithm

Rabin's test:

1. Check for $p|n$, p , prime, $p < N = 1000$.

2. Choose k and randomly pick $1 \leq b_1, b_2, \dots, b_k < n$. Let $n - 1 = 2^t m$, m odd. For b_i , compute $b_i^m \pmod{n}$, then $b_i^{2^m} \pmod{n}$, $b_i^{2^{2m}}$, ..., $b_i^{2^{tm}} = b_i^{n-1} \pmod{n}$. Check for $b_i^{n-1} \equiv 1 \pmod{n}$.

3. If not, it is composite, violating Fermat's congruence.

Note: Fermat proved that for p , prime,

$$a^{p-1} \equiv 1 \pmod{p} \text{ if } (a, p) = 1.$$

4. If so, then compute $((b_i^{2^j} \pmod{n-1}), n)$ for $1 \leq j < t$. If any one of these gcd's is neither 1 nor n , then n has a proper divisor.

If

(i) $1 \leq b_i < n$,

(ii) a) $b_i^{n-1} \not\equiv 1 \pmod{n}$, or

b) there is a j such that

$$2^j | (n-1) \text{ and } 1 < (b_i^{(n-1)/2^j} - 1, n) < n,$$

then we say that $W_n(b_i)$ holds.

Such a b is called a "witness to compositeness". If $W_n(b_i)$ is true, then n is composite. If, for all i , $1 \leq i \leq k$, $W_n(b_i)$ is not true, then the test asserts that n is prime.

Remarks:

1. As noted before, $i > 1 \Leftrightarrow n \equiv 1 \pmod{4}$ so for $n \equiv 3 \pmod{4}$, this reduces to the Solovay-Strassen test.

2. The computation of the greatest common divisor may be replaced as follows:

Let $n - 1 = 2^t \times m$, m odd, $0 \leq x < n$. Let $x_0 = x^m \pmod{n}$, $x_i \equiv x_{i-1}^2 \pmod{n}$, $1 \leq i \leq t$. Note: $x_t = x^{n-1} \pmod{n}$. Calculate $x_0 \equiv x^m \pmod{n}$, and then square mod n repeatedly until $x_{i-1} \neq n - 1$ and $x_i = 1$, or until x_t is reached. This is equivalent since if $W_n(x)$ holds, then either $x_t \neq 1$, or $x_t = 1$ and for some j , $0 \leq j \leq t$, we have $1 < (x_j - 1, n) < n$. Suppose $x_t \neq 1$. Let $i - 1, j \leq i - 1 < t$ be the last index such that $x_{i-1} \neq 1$. Thus $x_i = 1$. We have $x_j \equiv 1 \pmod{p}$ where $p \mid n$. Hence $x_{i-1} \equiv 1 \pmod{p}$ since $x_{i-1} \equiv x_j^{2^{i-j-1}} \pmod{n}$ and $i - j - 1 > 0$. Also $x_{i-1} = n - 1$ is impossible since $p \mid (x_{i-1} - 1) = (n - 2) \Rightarrow p \mid 2$. Thus $x_i = 1$ and $x_{i-1} \neq n - 1$. If either $x_i \neq 1$, or for some i , $1 \leq i \leq t$, we have $x_i = 1$ and $x_{i-1} \neq n - 1$, n must be composite. For if $x_t \neq 1$, then Fermat's congruence does not hold, and if $x_{i-1} \neq n - 1$, $x_i \equiv x_{i-1}^2 \equiv 1 \pmod{n}$, then 1 has more than two square roots mod n . This implies that n is composite since the number of square roots of 1 mod n is equal to 2 to the power the number of distinct prime factors of n .

3. The total required number of steps required for each b_i is $3 \log_2 n + t \log_2 n$.

Rabin also gives an estimate of the number of strong pseudoprimes in W_n .

Theorem 1.3B.1: If $4 < n$ is composite, then

$$\frac{3(n-1)}{4} \leq c|\{b|W_n(b)\}| \quad (4)$$

where $c(s)$ is the number of elements in s . That is, no more than $1/4$ of the numbers b , $1 \leq b \leq n$ are not witnesses.

In the course of this proof, he shows that

$$c|\{b|W_n(b)\}| \leq \left(\frac{3}{4} + \frac{3}{p}\right) (n-1)$$

if there is a Carmichael number $n = p \cdot q \cdot r$ such that p, q, r are prime, $p < q < r$, $(p-1)(q-1)(r-1)|(n-1)$. So if there are such Carmichael numbers for arbitrarily large p , then (4) is asymptotically the best possible. [See 10].

3C. Further Developments

As was previously observed, the strong pseudoprime test is only stronger than the Euler pseudoprime test if $n-1$ contains more than one factor of 2, i.e. if $n \equiv 1 \pmod{4}$. So to come up with a stronger test for the $n \equiv 3 \pmod{4}$ case, a method to find factors of $n \equiv 3 \pmod{4}$ numbers should be included. One such method is proposed in [1]. The authors consider the set

$$H_n = \{b \in \mathbb{Z}_n^\times | b \neq R_n \text{ and } \exists a \in R_n \text{ s.t. } a^2 \equiv b^2 \pmod{n}\}$$

where

$$R_n = \{a \in \mathbb{Z}_n^* \mid a^{(n-1)/2} \equiv \pm 1 \pmod{n}\}.$$

They prove the following theorem:

Theorem 1.3C.1:

- (i) $\forall b \in H_n$, $\gcd(n, 1 + b^{(n-1)/2} \pmod{n})$ is a nontrivial divisor of n ;
- (ii) $\# H_n \geq \# R_n$.

That is, each element of H_n gives a nontrivial divisor of n . The proof of (i) rests on the fact that if $b \in H_n$, $x = b^{(n-1)/2} \not\equiv \pm 1 \pmod{n}$, since $b \notin R_n$, but $x^2 \equiv 1 \pmod{n}$. So x is a nontrivial square root of 1 and hence $\gcd(n, 1 + x)$ is nontrivial. The second part follows from the fact that the number of solutions of $x^2 \equiv a^2 \pmod{n}$ equals 2 to the power of the lower number of distinct prime factors of n and is thus at least four for composite n .

They also include a probability analysis of Rabin's algorithm. They make the following important observation. The popular belief has been that the probability that a number called "prime" by Rabin's algorithm is composite for k test numbers $\leq 4^{-k}$ because each of the k iterations of the algorithm has a probability smaller than $1/4$ of failing on any given composite number. However, this is a result of confusing two conditional probabilities. The probability, "given that the algorithm returns 'prime' that n is in fact composite", is equal to the number of false primes. This is not the same as the probability,

"given that n is composite, the algorithm returns 'prime' ". This is the probability which is $\leq 4^{-k}$. They prove the following:

Theorem 1.3C.2: Let p be the probability that a uniformly randomly selected odd L -digit number is prime. The probability that Rabin's algorithm returns a composite number, i.e. mistakenly labels a composite number as 'prime', is given by

$$\frac{1}{1 + \left(\frac{p}{1-p}\right) 4^k}.$$

This is about $(L \alpha / 2) \times 4^{-k}$ provided that L is substantially smaller than 4^k and is about $1/2$ when $L = 2 \times 4^k / \alpha$, where $\alpha \approx 2.3$, $\alpha = \ln 10$. [See 1].

CHAPTER 2

THE RSA ENCRYPTION SCHEME AND A SUGGESTED IMPROVEMENT

This chapter contains a description of public key cryptosystems, the RSA scheme, and, after a mathematical digression, some modifications of the RSA scheme called the pseudo-RSA scheme.

R.L. Rivest, A. Shamir, and L. Adleman presented an implementation of a "public-key cryptosystem" in 1978 [see 12, or 3, pp. 104-109], a concept invented by Diffie and Hellman in 1976. [See 4] .

1. Public-key Cryptosystems

In a "public-key cryptosystem," each user makes his encryption procedure E public and keeps his decryption procedure D private. These procedures have the following properties:

- 1.a. Deciphering the enciphered form of a message M yields M , i.e.,

$$D(E(M)) = M .$$

- 1.b. Both E and D are "easy" to compute.
- 1.c. By publicly revealing E , the user does not reveal an easy way to compute D .

1.d. If a message M is first deciphered and then enciphered, M is received, i.e.,

$$E(D(M)) = M.$$

Each user can use the same general method with his own encryption key. When the user reveals E , he reveals a very inefficient method of computing $D(C)$, where $C = E(M)$, the ciphertext. If property 1.c holds, the number of possible messages M that must be tested until one such that $E(M) = C$ is found will be prohibitively large.

Definition 2.1.1: A function E satisfying 1.a-1.c is called a trap-door one-way function.

Remark: These functions are called "one-way" because they are easy to compute in one direction and (apparently) hard to compute in the other direction. They are called "trap-door" functions since the inverse functions are easy to compute once certain private "trap-door" information is known.

Definition 2.1.2: A function E satisfying 1.a-1.d is called a trap-door one-way permutation.

Remark: A trap-door one-way function that also satisfies 1.d. must be a permutation since every message is the ciphertext of some other message and every ciphertext is a permissible message, i.e., the mapping is bijective.

As will be seen in the following two sections, a trap-door one-way permutation allows both privacy and signatures that cannot be forged.

1.A. Privacy

The purpose of encryption is to provide privacy. The sender enciphers each message before transmitting it to the receiver. The receiver, and hopefully no unauthorized person, knows how to decipher it. An eavesdropper who hears the transmitted message should hear only "garbage", i.e. the ciphertext.

The most widely used encryption technique, adopted by the National Bureau of Standards which is now the National Institute of Standards and Technology, is DES. DES grew out of the Vulcan project at IBM. It does not have property 1.c. All classical encryption methods, including DES, require the encryption and decryption keys to be distributed privately before private communication can begin. In a public-key cryptosystem, the encryption keys can be distributed over an insecure communications channel.

In a public-key cryptosystem, a private message M can be sent from person A to person B as follows:

Suppose (E_A, D_A) are the encryption and decryption keys for A and (E_B, D_B) are the corresponding keys for B . Both E_A and E_B are public. Sender A takes E_B and sends $E_B(M)$. Receiver B deciphers the message by computing $D_B(E_B(M))$. By property 1.c, only person B can decipher the message. Even if E_A and E_B are not public, A and B can send each other their encryption keys over an insecure channel.

A public-key cryptosystem could be used to "bootstrap" into a standard encryption scheme by using the public-key cryptosystem to transmit the encryption and decryption keys. This might be useful if the standard encryption scheme is significantly faster. In the case of DES, it is hardware implemented. However, there is now an RSA chip, and since the RSA scheme involves multiprecision arithmetic operations as opposed to the complicated bit manipulations of DES, the RSA scheme may be faster on a general-purpose computer. Also, National Security Agency has refused to renew their certification of the security of DES. Some people view this as an indication that National Security Agency no longer feels DES is secure.

1.B. Signatures

The recipient of a message often needs to have proof that the message originated from the sender. In particular, the recipient must be able to show that he did not create the message himself. Thus, an electronic signature must be message-dependent as well as signer dependent to avoid the possibility of tampering by the recipient. Otherwise the recipient could modify the message or tack on a signature. A digital signature is implemented as follows:

A sends a signed message M to B by transmitting

$D_A(M)$.

B authenticates the message by using E_A :

$$E_A(D_A(M)) = M .$$

Since only A can apply D_A , it cannot be forged. If both secrecy and authenticity are required, A sends

$$E_B(D_A(M)) .$$

B authenticates and decodes the message using

$$E_A(D_B(E_B(D_A(M)))) .$$

2. The RSA Scheme

The RSA scheme is an example of a trap-door one-way permutation. In this section, a description of its coding keys, means to provide both privacy and a signature, the underlying mathematical theory, the method for encoding and decoding, how the large primes required are produced and how reliably, how the coding keys are produced, and how secure it is, is discussed.

In the RSA scheme, an encryption key consists of a pair of positive integers (e, n) . The message is first represented as an integer between 0 and $n - 1$ by breaking the message into a series of blocks and representing each block as such an integer (using, for example, its ASCII representation). Then it is encoded by raising each converted block to the power e modulo n , i.e.

$$C = E(M) = M^e \pmod{n} \text{ for a message block, } M .$$

Each encoded block C is decoded with the corresponding decryption key (d, n) where d is also a positive integer, i.e.,

$$D(C) \equiv C^d \pmod{n} \text{ for a ciphertext } C.$$

The keys satisfy the following criteria. The modulus $n = pq$ where p and q are very large "random" primes. The decoding exponent d is chosen to be a large, random integer relatively prime to $\phi(n) = (p - 1) \times (q - 1)$. The encoding exponent e is then computed from p, q and d to satisfy

$$e \times d \equiv 1 \pmod{\phi(n)}.$$

Due to the symmetry, the methods for computing d and e may be interchanged.

A small difficulty arises when both secrecy and authenticity are required because it is necessary to apply successive transformations with different moduli. For example, if A sends

$$C = E_B(D_A(M)),$$

and if $n_A > n_B$, the blocks comprising $D_A(M)$ may not be in the range $[0, n_B - 1]$, the range of B 's transformation. Reducing modulo n_B does not solve the problem because then the original message is not recoverable.

Rivest, Shamir, and Adleman suggest using a threshold value h . Each user has two sets of transformations, say (E_{A1}, D_{A1}) for signatures, and

(E_{A2}, D_{A2}) for secrecy where $n_{A1} < h < n_{A2}$. To send a signed, secret message to B, A transmits

$$C = E_{B2}(D_{A1}(M))$$

which is computable because $n_{A1} < h < n_{B2}$. User B recovers M and checks A's signature by computing

$$\begin{aligned} E_{A1}(D_{B2}(C)) &= E_{A1}(D_{B2}(E_{B2}(D_{A1}(M)))) \\ &= E_{A1}(D_{A1}(M)) \\ &= M. \end{aligned}$$

Another method, due to Kohnfelder, [see 6], uses the observation that if $C = E_B(D_A(M))$ is not computable because $n_A > n_B$, then $C^1 = D_A(E_B(M))$ is computable. User B, knowing both n_A and n_B , will know which was computable by A, and can recover M as follows:

For $n_A < n_B$:

$$\begin{aligned} E_A(D_B(C)) &= E_A(D_B(E_B(D_A(M)))) \\ &= E_A(D_A(M)) \\ &= M. \end{aligned}$$

For $n_A > n_B$:

$$D_B(E_A(C^1)) = D_B(E_A(D_A(E_B(M))))$$

$$= D_B(E_B(M))$$

$$= M.$$

The RSA scheme rests on the following theorem due to Euler.

Theorem 2.2.1: For every a and n such that a and n are relatively prime,

$$a^{\phi(n)} \pmod{n} = 1$$

where, for $n = \prod_{i=1}^t P_i^{e_i}$, $\phi(n) = \prod_{i=1}^t P_i^{e_i-1} (P_i - 1)$.

Theorem 2.2.1 has the following generalization.

Theorem 2.2.2: If n is a product of distinct primes, then for all a ,

$$a^{\phi(n)+1} \equiv a \pmod{n}.$$

Note: Thus, in particular, Theorem 2.2.2 says that a message block, a , is recoverable by raising it to the $\phi(n) + 1$ power and reducing modulo n for $n = p \cdot q$.

When $n = pq$, Theorem 2.2.1 reduces to $a^{(p-1)(q-1)} \pmod n = 1$. The fact that d and $\phi(n)$ are relatively prime guarantees a solution of $e \cdot d \equiv 1 \pmod{\phi(n)}$. Properties 1.a and 1.d follow from Theorem 2.2.1.

$$D(E(M)) \equiv (E(M))^d \equiv (M^e)^d \equiv M^{e \cdot d} \pmod n$$

$$\equiv M^{h\phi(n)+1} \equiv M^1 \equiv M \pmod n ,$$

$$E(D(M)) \equiv (D(M))^e \equiv (M^d)^e \equiv M^{d \cdot e} \pmod n$$

$$\equiv M^{h\phi(n)+1} \equiv M^1 \equiv M \pmod n , \text{ for } 0 \leq M < n + \gcd(M, n) = 1 .$$

Thus E and D are inverse permutations.

2.A. Algorithms

In this section, algorithms for computing $M^e \pmod n$ producing large prime numbers, and choosing the coding keys, are described.

2.A.a. A Method to Encrypt and Decrypt

To encrypt in the RSA scheme, $M^e \pmod n$ is computed.

This computation requires at most $2 \log_2(e)$ multiplications and $2 \log_2(e)$ divisions using the following procedure:

Step 1. Let $e_k e_{k-1} \dots e_1 e_0$ be the binary representation of e .

Step 2. Let $C = 1$.

Step 3. Repeat for $i = k, k-1, \dots, 0$:

Set C equal to $C^2 \pmod{n}$.

If $e_i = 1$, then set C equal to $C \times M \pmod{n}$.

At the end of the above procedure C is the ciphertext of M . The same procedure can be used to decrypt, replacing e by d , C by M , and M by C . This, and other methods, are discussed in Knuth [see 7].

The fact that encryption and decryption are essentially identical simplifies implementation, reducing the complexity of special-purpose chips. The encryption time per block increases no faster than the cube of the number of digits in n .

2.A.b. Finding Large Prime Numbers

The RSA method requires each user to find two large prime numbers, p, q to create his encryption and decryption keys. These numbers must be sufficiently large so that it is computationally infeasible for anyone to factor $n = p \cdot q$. Rivest, Shamir and Adleman suggest using primes of 100 decimal digits. However the size of the numbers makes finding them using a deterministic algorithm infeasible. Then there is the question of what is meant by a random prime number. In their original paper, Rivest, Shamir and Adleman suggest the Solovay-Strassen test, discussed in Chapter 1, Section 2. Later Rabin developed an algorithm based on Miller's deterministic algorithm,

as discussed in Chapter 1, Sections 3.A-B . The methods do not produce prime numbers. They produce numbers that have passed compositeness tests and are "likely primes".

Some additional security questions should be considered when selecting these "primes" to reduce the risk of factoring. The two primes should differ in length so that they are not close to $\sqrt{n}$. Both $p \pm 1$ and $q \pm 1$ should not have small factors, since factoring any of them leads to factoring of $\phi(n)$ and n . Also $p - 1$ and $q - 1$ should not have large common factors.

Another reason for $p - 1$ and $q - 1$ not to have too many factors is the fixed point problem, not addressed by Rivest, Shamir and Adleman. By Theorem 2.2.2 of Section 2, we have $a^{\phi(n)+1} \equiv a \pmod{n}$. For a specific a , an exponent smaller than $\phi(n) + 1$ may do. If such values are abundant, the message may be recovered by re-encryption, or the ciphertext may be the same as the message, i.e., a fixed point of the transformation. If the message is revealed by re-encrypting m times, we call it a fixed point of order m . Finding the fixed points of order m is equivalent to solving

$$Te^m \equiv T \pmod{N} .$$

According to Riesel, [see 11, p. 234], the number of solutions is

$$g(m) = (1 + (k^m - 1, p - 1)) \times (1 + (k^m - 1, q - 1)) .$$

Since e is always odd, the total number of fixed points of all orders is at least 9.

2.A.c. The choice of d and e

In the previous section the relevant issues for choosing p , q , and n were discussed. In this section, Rivest, Shamir, and Adleman suggest that d can be chosen to be any prime greater than $\max(p, q)$. This appears to be rather difficult since it seems at least as hard as finding p or q . However, any number greater than $\max(p, q)$ is a reasonable candidate although for ease of computation, a smaller one may be preferable. To be sure that it is relatively prime to $\phi(n)$, the greatest common divisor algorithm is required. The range of possible d 's should be large enough so that d cannot be found by direct search.

The exponent e is a by-product of using the extended Euclidean algorithm to compute the greatest common divisor of d and $\phi(n)$. Thus, if this algorithm is used to check possible d 's, a separate calculation is not required to compute e . By including the computation of e , each loop of the extended Euclidean algorithm is only slightly more time consuming and involves no more arithmetic operations. If e turns out to be smaller than $\log_2(n)$, another d must be chosen so that the encrypted message is actually reduced modulo n . This restriction might make it more reasonable to first pick e in the appropriate range using the method described for d , and then solve for d using the method described for e .

2B. The Security of the Method

Excluding theft, the problem of cracking the RSA code with modulus n can be shown to be mathematically equivalent to factoring n . So one way to check the security of a particular choice of n is to try to factor $n = pq$ with all known methods. Increasing the size of n increases the security. Unfortunately it also increases the block size, and the encryption and decryption algorithms will become more time-consuming and expensive. The flexibility of the choice of blocksize is one of the advantages of the method.

Our estimates of a reasonable blocksize of between 100 to 200 decimal places assumes that we can estimate an adversary's ability to factor. Using presently known methods requires an effort proportional to $n^{0.15}$ to factor n so an adversary will not be aided substantially by increased computing power. Also, as computing capacity increases, the cost of large size blocks decreases allowing larger n 's. However, it is conceivable that some breakthrough in more efficient factorization will occur, threatening the security of the method [see 11, pp. 235-236].

3. Suggested Improvements, the Pseudo-RSA Scheme

Public-key cryptosystems, in general, and the RSA scheme in particular, have been described to provide the proper background for understanding the modifications made to the RSA scheme resulting in the pseudo-RSA scheme. The details of the pseudo-RSA scheme and its underlying mathematical theory are described in this section.

3.A.a. Mathematical Digression

The RSA scheme relies on Euler's Theorem, mentioned as Theorem 2.2.1 in section 2, and e and d are chosen so that $e \cdot d \equiv 1 \pmod{\phi(n)}$. There is a generalization of Euler's Theorem due to Carmichael.

Definition 2.3A.1: Let $\lambda(n)$ be defined as follows:

$$\lambda(p^\alpha) = \phi(p^\alpha), \text{ where } p = 2 \text{ and } \alpha \leq 2 \text{ and when } p \geq 3$$

$$\lambda(2^\alpha) = 1/2 \phi(2^\alpha), \text{ where } \alpha \geq 3$$

$$\lambda(n) = \text{least common multiple of } (\lambda(p_i^{\alpha_i}))_i, \text{ if } n = \prod_i p_i^{\alpha_i}$$

Theorem 2.3A.1: If $(a, n) = 1$, then $a^{\lambda(n)} \equiv 1 \pmod{n}$.

Theorem 2.3A.1 has the following generalization:

Theorem 2.3A.2: If n is a product of distinct primes, then for all a ,

$$a^{\lambda(n)+1} \equiv a \pmod{n}.$$

Note. The assumption that n is a product of distinct primes is necessary. Also for a particular a , an exponent smaller than $\lambda(n) + 1$ may suffice. [See 11, pp. 276-277].

The following theorem and its corollary are the main mathematical contributions of this thesis and allow the basic method of the RSA scheme to be extended to the case when one or both factors of $n = pq$ are in fact Fermat pseudoprimes. The theorem says that the congruence relation needed to permit encoding and decoding holds when the factors of n are primes or pseudoprimes to the base a , for a particular message block, a . The corollary says that in case the pseudoprimality holds for all message blocks, the congruence relation holds for all message blocks. This allows the replacement of $\phi(n)$ by $L(n)$ which is central to the pseudo-RSA scheme.

Theorem 2.3A.3: Let $n = \prod_i N_i$, where N_i is a prime or a Fermat pseudoprime to the base a , and the N_i are pairwise relatively prime. Then

$$a^{L(n)+1} \equiv a \pmod{n}, \text{ where } L(n) = \text{l.c.m. } \{(N_i - 1)\}_i.$$

Proof: If N_i is prime, then by Fermat's Theorem, $a^{N_i-1} \equiv 1 \pmod{N_i}$.

If N_i is a pseudoprime to the base a , we also have

$$a^{N_i-1} \equiv 1 \pmod{N_i}, \text{ and } \lambda(N_i) | (N_i - 1).$$

Thus, for all i , $a^{m(N_i-1)} \equiv 1 \pmod{N_i}$ for $m \in \mathbb{Z}^+$.

Multiplying by a , we have $a^{s_i} \equiv a \pmod{N_i}$ if $s_i \equiv 1 \pmod{N_i - 1}$ since $m(N_i - 1) + 1 \equiv 1 \pmod{N_i - 1}$.

If $(a, N_i) = 1$, this also holds since both sides are congruent to $0 \pmod{N_i}$.

Choosing a value of s such that $s \equiv s_i \equiv 1 \pmod{N_i - 1}$ for every value of i , we have $a^s \equiv a \pmod{N_i}$, for all a .

The Chinese Remainder Theorem allows these congruences to be combined into one congruence $\pmod{n}$, namely,

$$a^s \equiv a \pmod{n} \text{ for all } a$$

provided that the exponent

$$s \equiv 1 \pmod{N_i - 1} \text{ for all } N_i | n.$$

Since $L(n) = \text{l.c.m. } \{N_i - 1\}_i$, $s = L(n) + 1$ will work.

Note: If all the N_i are prime, $L(n) = \lambda(n)$.

Corollary 2.3A.1: If each N_i that is a pseudoprime in Theorem 2.3A.3 is a Carmichael number, then for $n = \prod_i N_i$,

$$a^{L(n)+1} \equiv a \pmod{n}, \text{ for all } a.$$

Proof: If N_i is a Carmichael number,

$$a^{N_i-1} \equiv 1 \pmod{N_i}, \text{ for all } a \text{ such that } (a, N_i) = 1.$$

Also, $N_i \equiv \prod_{j=1}^t p_j$, where the p_j are distinct primes and $t \geq 3$, has the property that $\prod_{j=1}^t (p_j - 1)$ divides $(N_i - 1)$, that is, $\lambda(N_i)$ divides $\phi(N_i)$ which divides $(N_i - 1)$. Applying the argument of Theorem 3A.2 to each N_i , and observing that $\lambda(N_i)$ divides $(N_i - 1)$, we have

$$a^s \equiv a \pmod{N_i} \text{ if } s_i \equiv 1 \pmod{(N_i - 1)}$$

as before, although this may not be the smallest such exponent. Proceeding as in Theorem 2.3A.2, the result follows.

3.B. The Choice of e and d

As a result of the above, we suggest that $\phi(n)$ be replaced by $\lambda(n)$ (or $L(n)$), which divides $\phi(n)$. That is, e and d should be selected so that $e \cdot d \equiv 1 \pmod{\lambda(n)}$. Using $\lambda(n)$ may involve less calculation and has fewer fixed points. When $n = pq$, p, q prime, $\phi(n) = (p - 1)(q - 1)$ and $L(n) = \lambda(n) = (p - 1)(q - 1) / \gcd(p - 1, q - 1)$ which is a proper divisor of $\phi(n)$. Thus for a particular d , the smallest possible e chosen using $\lambda(n)$ (or $L(n)$) is less than or equal to that using $\phi(n)$. To find e and d , given n , the previously described method can be used, or else simply compute an integer m such that

$m\lambda(n) + 1$ is divisible by d (or e) chosen in advance and then solved for e (or d), so that $m\lambda(n) + 1 = e \cdot d$.

As before, the choice of n , e , and d depends on the size of the coding block. If the coding block contains N decimal places, then n must be at least 10^N so that all distinct messages possible within the coding block have different representations (mod n). The choice of e should also be restricted as discussed previously.

3.C. The Choice of $n = p \cdot q$

As a consequence of the formula describing scarcity of primes, mentioned on page 9, picking numbers at random as possible primes would require on average $\ln p \approx 138$ trials for $p \approx 10^{60}$ before a prime is selected, according to Riesel, [11, pg. 232]. Suppose we try the Sieve of Eratosthenes as follows:

Look for a prime near 10^{60} by picking a starting point near 10^{60} and a length, say 5000 consecutive odd integers, for the interval to sieve. Then sieve with all odd primes, say < 1000 . These choices will give about 81 numbers untouched by the sieve of which approximately 72 are prime [see 11, p. 232].

Next, we propose to test the remaining numbers for pseudoprimality with several bases. Each pseudoprimality test can be carried out in at most $2 \log_2 N$ steps.

Finally we check $p + 1$ and $p - 1$ for small factors, say $< 10^5$, to reduce the number of fixed points and to reduce the possibility of factoring using, say

Pollard's $(p - 1)$ method, [see 11, p. 172] or the $(p + 1)$ method [see 11, p. 174] .

Suppose our method selects a pseudoprime instead of an actual prime. Our resulting n will be easier to factor and there will be more fixed points. If our "prime" is, in fact, a Carmichael number, Corollary 2.3A.1 shows that encoding and decoding will work properly (replacing $\lambda(n)$ by $L(n)$) . So we do not believe that our method causes any more errors than the errors that would result from Rabin's method choosing a number as one of the factors of n which is in fact not prime. In addition, this method is simpler.

3.D. A Summary of Key Production

3.D.1. Sieve an Interval.

- 3.D.1a. Get a list of primes.
- 3.D.1b. Randomly generate an offset.
- 3.D.1c. Pick the interval to be offset.
- 3.D.1d. Use the list of primes to sieve the resulting interval.

To test for pseudoprimality, we go through the following steps:

3.D.2. Test for Pseudoprimality.

- 3.D.2a. Test for pseudoprimality using a fixed list of small primes, say 2, 3, 5, 7, 11, and 13 .
- 3.D.2b. Select bases at random to test for pseudoprimality. If they are selected from the list of primes used to sieve, no greatest common divisor test is

necessary. If any positive integer greater than 1 is allowed as a base, the base a must satisfy $(a, p) = 1$ for all candidates p .

3.D.3. Test the remaining candidates p to see if $p - 1$ and $p + 1$ have small factors (other than 2 and perhaps 3). One method is trial division.

3.D.4. Select two of the remaining candidates at random to be p and q .

3.D.5. Calculate (p, q) . If the result is not 1, select another pair p, q and repeat the calculation for the new pair. Repeat this step until the result is 1.

3.D.6. Calculate $n = p \times q$ and $L(n)$, where $L(n) = (p - 1)(q - 1)/\gcd(p - 1, q - 1)$.

3.D.7. Calculate e and d as described in 3.B.

3.E. An Additional Security Consideration

For the secrecy and authenticity protocols, we suggest some additional greatest common divisor calculations as follows.

Suppose you are User A. Take a plaintext message and put it into blocks of integers. Let M be such a block.

To send a secret message to B:

Compute $\text{g.c.d.}(M, n_B)$ for each M . If the result is 1 for all M , then send $E_B(M)$ for all M . If $\text{g.c.d.}(M, n_B) \neq 1$ for some M you have found a factor of n_B and B's key is less secure.

To send an authenticated message to B :

Compute $\text{g.c.d.}(M, n_A)$ for each M . If the result is 1 for all M , then send $D_A(M)$. If not, $\text{g.c.d.}(M, n_A)$ reveals a factor of n_A for some M . You could either modify the message or change your key.

To send a secret and authenticated message to B :

First use the procedure for the authenticated message to compute $D_A(M)$. Then use the procedure for the secret message with $D_A(M)$ in place of M for all M .

CHAPTER 3

SUGGESTED FURTHER WORK

The pseudo-RSA scheme requires some strenuous testing, checking for accuracy and varying parameters such as:

1. block size and construction,
2. the list of primes used to sieve,
3. the size and starting point of the interval to be sieved,
4. the list of small primes used to initially test for pseudoprimality,
5. the number of bases chosen at random to test for pseudoprimality, whether they should be required to be prime, and how large,
6. methods for looking for small factors of $p \pm 1$ and what small means,
7. selection of pairs of candidate primes,
8. the relative efficiency of using $\lambda(n)$ (or $L(n)$) versus $\phi(n)$ in practice,
9. choices of d or e ,
10. how often the checks for greatest common divisors find common factors,
11. the selection of algorithms used in repeated calculations, and
12. methods to make it faster such as parallelization and chip design.

To do this, a computer program that can handle large numbers is required.

There is an existing sample program that can only handle small numbers. It operates as follows:

A. A list of primes used to sieve is provided as well as a small list of primes used to initially test for pseudoprimality.

B. It prompts the user to select the size of the interval to be sieved and its starting point.

C. It prompts the user for the number of bases to be chosen at random for additional pseudoprimality tests.

D. After it performs the sieve and the pseudoprimality tests, it displays those numbers remaining in the interval that have not yet been shown to be composite.

E. It checks to see if a candidate prime p has small factors of $p \pm 1$ and displays the surviving candidates.

F. It picks two "primes", p and q , to be used in the coding procedure.

G. It calculates $n = p \cdot q$ and $L(n) = (p - 1) \cdot (q - 1) / \gcd(p - 1, q - 1)$.

H. Then d and e are selected as follows:

a. First e is selected at random.

b. Then a preliminary smallest possible d is selected by solving the congruence $m \cdot L(n) + 1 \equiv 0 \pmod{e}$ for m and solving $m \cdot L(n) + 1 = e \cdot d$ for d .

c. This d is replaced by a randomly selected d produced by picking a randomly selected congruent m .

I. The resulting coding keys are displayed.

J. The user is prompted for a plaintext message.

K. The message is converted to ASCII and the user is prompted for the block size.

L. The user is asked how many times the message should be encrypted and whether re-encryption should stop at the first fixed point.

M. These requests are implemented and the results are displayed.

N. The user is asked if the message should be decrypted. If decryption is requested, the result is translated back into a character string.

The program has a fair amount of flexibility, limited mainly by the size of the numbers it can handle without overflow. There are many independent operations performed such as pseudoprimality testing once the bases are selected, checking each candidate p for small factors of $p \pm 1$, and encrypting and decrypting each block provided that a method for keeping track of their order is included. Thus, there is a lot of opportunity for parallelization.

BIBLIOGRAPHY

BIBLIOGRAPHY

1. P. Beauchemin, G. Brassard, C. Crepeau, C. Goutier, "Two Observations on Probabilistic Primality Testing", *Advances in Cryptography - CRYPTO 1986*, edited by A.M. Odlyzko, Springer-Verlag, Berlin, pp. 443-450.
2. R.D. Carmichael, "On Composite numbers p which satisfy the Fermat congruence $a^{p-1} \equiv 1 \pmod{p}$ ", *Amer. Math. Monthly* 19 (1912), pp. 22-27.
3. D.E.R. Denning, *Cryptography and Data Security*, Addison-Wesley, 1982.
4. W. Diffie and M. Hellman, "New Directions in Cryptography," *IEEE Trans. Inform. Theory* IT-22, 6, pp. 644-654 (November 1978).
5. Hua Loo Keng, *Introduction to Number Theory*, Springer-Verlag, 1982.
6. L.M. Kohnfelder, "On the Signature Reblocking Problem in Public Key Cryptosystems," *Comm. ACM* 21(2), p. 179 (February 1978).
7. D.E. Knuth, *The Art of Computer Programming, Vol. 2: Seminumerical Algorithms*, Addison-Wesley, Reading, Mass., 1969.
8. Gary L. Miller, "Riemann's Hypothesis and Tests for Primality", *Journal of Computer and System Services* 13 (1976), pp. 300-317.
9. C. Pomerance, J. Selfridge, and S. Wagstaff, Jr., "The Pseudoprimes to $25 \cdot 10^9$ ", *Math. Comp.* 35 (1980), pp. 1003-1026.
10. Michael O. Rabin, "Probabilistic Algorithm for Testing Primality", *Journal of Number Theory* 12, (1980), pp. 128-138.
11. Hans Riesel, *Prime Numbers and Computer Methods for Factorization*, Birkhauser, 1985.
12. R.L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", *Comm. ACM* 21, pp. 120-126, 1978.
13. Solovay, R. and V. Strassen, "A fast Monte Carlo test for primality", *SIAM Journal on Computing*, vol. 6, pp. 84-85, 1977; erratum in vol. 7, p. 118, 1978.

VITA

Janet Burstein Svirsky, formerly Janet Suzanne Burstein, was born December 22, 1952 in Washington, D.C. She graduated from Interlochen Arts Academy in 1970. She attended Reed College in Portland, Oregon from September 1970 to May 1974, where she received a B.A. in Mathematics in May 1974. She began graduate study in Mathematics at The Johns Hopkins University in September 1974, and received an M.A. in May 1977. She married Roman Svirsky on December 28, 1980. She received her Ph.D. in February 1985. Her dissertation, "On the Class Numbers of Imaginary Quadratic Fields" was completed under the direction of Dr. Tahashi Ono. She was a visiting assistant professor at Tulane University from August 1985 to May 1986 and a lecturer at The University of Tennessee in the fall of 1986. She began graduate study in Computer Science at The University of Tennessee in the spring of 1987. She was a consultant at the Oak Ridge National Laboratory, Oak Ridge, Tennessee, as a member of the project: Probabilistic Algorithms Used in Public Key Cryptosystems, directed by V.R.R. Uppuluri, from March to July 1988. Her son, Leo Benjamin Svirsky, was born December 31, 1988.