Probabilistic Encryption & How To Play Mental Poker Keeping Secret All Partial Information

Shafi Goldwasser * and Silvio Micali **
Computer Science Department
University of California - Berkeley

1. Introduction

This paper proposes an Encryption Scheme that possess the following property:

> An adversary, who knows the encryption algorithm and is given the cyphertext, cannot obtain any information about the cleartext.

Any implementation of a Public Key Cryptosystem, as proposed by Diffie and Hellman in [8], should possess this property.

Our Encryption Scheme follows the ideas in the number theoretic implementations of a Public Key Cryptosystem due to Rivest, Shamir and Adleman [13], and Rabin [12].

Security is based on Complexity Theory and the intractability of some problems in number theory such as factoring, index finding and deciding whether numbers are quadratic residues with respect to composite moduli is assumed. In this context, impossibility means computational infeasibility and proving that a problem is hard means to show it equivalent to one of the above mentioned problems.

The key idea in both the RSA scheme and the Rabin scheme is the selection of an appropriate trapdoor function; an easy to evaluate function f such that x is not easily computable from $f(x)$, unless some extra information is known. To encrypt a message m, one simply evaluates $f(m)$.

This research was supported by
* NSF Grant MCS-79-037667
** fellowship from Consiglio Nazionale delle Ricerche - Italy and in part by NSF Grant MCS-79-037667

We would like to point out two basic weaknesses of this approach:

1) **The fact that f is a trapdoor function does not rule out the possibility of computing x from $f(x)$ when x is of a special form. Usually messages do not consist of numbers** chosen at random but possess more structure. Such structural information may help in decoding. For example, a function f, which is hard to invert on a generic input, could conceivably be easy to invert on the ASCII representations of English sentences.
2) **The fact that f is a trapdoor function does not rule out the possibility of easily computing some partial information about x (even every other bit of x) from $f(x)$.** The danger in the case that x is the ASCII representation of an English sentence is self evident. Encrypting messages in a way that ensures the secrecy of all partial information is an extremely important goal in Cryptography. The importance of this point of view is particularly apparent if we want to use encryption to play card games over the telephone. If the suit or color of a card could be compromised the whole game could be invalid.

Though no one knows how to break the RSA or the Rabin scheme, in none of these schemes is it **proved** that decoding is hard without any assumptions made on the message space. Rabin shows that, in his scheme, decoding is hard for an adversary if the set of possible messages has some density property.

The novelty of our contribution consists of

1. The notion of Trapdoor Functions is replaced by **Probabilistic Encryption.** To encrypt each message we make use of a fair coin. The encoding of each message will depend on the message plus the result of a sequence of coin tosses. Consequently, there are many possible encodings for each message. However, messages are always uniquely decodable.[1]

[1] Probabilistic Encryption is completely different from the technique of appending random bits to a message as suggested in [12] and [16].

2. Decoding is easy for the legal receiver of a message, but **provably** hard for an adversary. Therefore the spirit of a trapdoor function is maintained. In addition, in our scheme, without imposing any restrictions on the message space, we can prove that decoding is equivalent to deciding quadratic residuosity modulo composite numbers.

3. No Partial Information about an encrypted message could be obtained by an adversary. Assume that the message space has an associated probability distribution and that, with respect to this distribution, an easy to compute predicate P (such as "the exclusive or of all the bits in the message is 1") has probability p to be true. Let $p \geq .5$ without any loss of generality. Then, without any special ability, an adversary, given the cyphertext, can always guess that P is true for the cleartext, and be correct with probability p.

   Based on the assumption that deciding quadratic residuosity modulo composite numbers is hard, we prove that an adversary cannot guess correctly with probability $p+\varepsilon$,from the cyphertext, whether the cleartext satisfies the predicate P, where ε is a non negligible positive real number.

Probabilistic Encryption has been useful for the solution of Mental Poker. The problem whether it is possible to play a "fair" game of Mental Poker has been raised by Robert Floyd. Shamir, Rivest and Adleman proposed an elegant solution to this problem in [14] using commutative encryption functions, but they could not prove that partial information could not be compromised using their scheme. Indeed, several problems in the implementation of their scheme have been pointed out by Lipton in [10].

We present a solution for Mental Poker, for which we can **prove**, based on the assumption that factoring and deciding quadratic residuosity modulo composite numbers is hard, that not a single bit of information about a card which should remain hidden can be discovered. Our solution does not use commutative encryption functions.

2. The Security of a Public Key Cryptosystem.

All the number theoretic notation used in this section will be defined in section 3.1.

2.1 What is a Public Key Cryptosystem?

The concept of a Public Key Cryptosystem was introduced by Diffie and Hellman in their ingenious paper [8]. Let M be a finite message space, A, B,... be users, and let $m \in M$ denote a message. Let $E_A: M \to M$ be A's encryption function, which is ideally bijective, and D_A be A's decryption function such that $D_A(E_A(m)) = m$ for all $m \in M$. In a Public Key Cryptosystem E_A is placed in a public file, and user A keeps D_A private. D_A should be difficult to compute knowing only E_A. To send message m to A, B takes E_A from the public file, computes $E_A(m)$ and sends this message to A. A easily computes $D_A(E_A(m))$ to obtain m.

2.2 The RSA scheme and the Rabin scheme

The two implementations of a Public Key Cryptosystem most relevant and inspiring for this paper are the RSA scheme [13], due to Rivest, Shamir and Adleman, and its particularization suggested by Rabin [12].

The key idea in both the RSA scheme and the Rabin scheme consists in the selection of an appropriate number theoretic trapdoor function. In the RSA scheme, user A selects N, the product of two large primes p_1 and p_2 and a number s such that s and $\varphi(N)$ are relatively prime , where φ is the Euler totient function. A puts N and s in a public file and keeps the factorization of N private. Let $Z_N^* = \{ x \mid 1 \leq x \leq N-1$ and x and N are relatively prime$\}$. For every message $m \in Z_N^*$, $E_A(m) = m^s \bmod N$. Clearly, the ability to take sth roots mod N implies the ability to decode. A, who knows the factorization of N, can easily take sth roots mod N. No efficient way to take sth roots mod $N$is known when the factorization of N is unknown.

About the RSA scheme Rabin remarks that, for all we know, inverting the function $x^s \bmod N$ may be a hard problem in general, and yet easy for a large percentage of the x's.

He suggests to modify the RSA scheme by choosing $s=2$. Thus, for all users A, $E_A(x) = x^2 \bmod N$. Notice that E_A is a 4-1 function because our N is the product of two primes. In fact, every quadratic residue mod N, i.e every q such that $q \equiv x^2 \bmod N$ for some $x \in Z_N^*$, has four square roots mod N: $\pm x \bmod N$ and $\pm y \bmod N$. As A knows the factorization of N, upon receiving the encrypted message $m^2 \bmod N$, he could compute its four square roots and get the message m. The ambiguity in decoding could be eliminated, for example, by sending the first 20 digits of m in addition to $m^2 \bmod N$. Such extra information cannot effectively help in decoding: we could always guess the first 20 digits of m.

The following theorem shows how hard is it to invert Rabin's function $x^2 \bmod N$.

Theorem (Rabin): If for 1% of the q's quadratic

residues mod N one could find one square root of q, then one could factor N in Random Polynomial Time.

The theorem follows from the following lemma that we state without proof.

Lemma 1: Given x, $y \in Z_N^*$ such that $x^2 \equiv y^2 \bmod N$ and $x \neq \pm y \bmod N$, there is a polynomial time algorithm to factor N. (In fact the greatest common divisor of N and $x \pm y$ is a factor of N).

Informal proof of Rabin's theorem: Assume that we have a magic box B such that given q, a quadratic residue mod N, for 1% of the q's it outputs one square root of q mod N. Then we could factor N by iterating the following step:

> Pick i at random in Z_N^* and compute $q = i^2 \bmod N$. Feed the magic box B with q. If M outputs a square root of q different from i or -i mod N, then (by the above lemma) factor N.

The expected number of iterations is low, as at each step, we have a 0.5% chances to factor N.

2.3 Objections to Cryptosystems based on Trapdoor Functions

Covering ones face with a handkerchief certainly helps to hide personal identity. However:

1) It will not hide from me the identity of a special subset of people: my mother, my sister, close friends.
2) I can gather a lot of information about the people I cannot identify: their height, their hair color and so on.

Essentially, the same kind of problems may arise in the RSA scheme and in the Rabin scheme and, more generally, in any other Public Key Cryptosystem based on Trapdoor Functions:

1) The fact that f is a Trapdoor Function does not rule out the possibility of computing x from f(x) when x is of special form.

2) The fact that f is trapdoor function does not rule out the possibility of easily computing some partial information about x from f(x).

2.4 Discussion of Objection 1

One may argue that Rabin's Public Key Cryptosystem is as hard to break as factoring in the following way; whoever can get a messages m from their encryptions $m^2 \bmod N$ 1% of the time, is actually realizing the magic box of Rabin's theorem and thus could efficiently factor n.

We would like to point out the following fact.

Claim: If M, the set of messages, is "sparse" in Z_N^*, the ability to decode 1% of all messages does not yield a random polynomial time algorithm for factoring.

By "sparse" we mean that for a randomly chosen $x \in Z_N^*$, the probability that x is a message is virtually 0.

Let $f(x) = x^2 \bmod N$. Assume that we are able to invert the function f only on $f(M)$. Then we would have a magic box MB which, fed $m^2 \bmod N$, would output m whenever $m \in M$; and fed q, outputs nothing whenever $q \notin \{m^2 \bmod N \mid m \in M\}$, except, at most, for a negligible portion of the q's. With the use of such a magic box we could decode, but not factor N efficiently. Using such MB, let us look at the above informal proof of Rabin's theorem . If we pick $m \in M$ and feed $m^2 \bmod N$ into MB, then we get back m and we cannot factor. If we pick $i \notin M$ and feed $i^2 \bmod N$ to MB, then the probability that one square root of $i^2 \bmod N$ different from i, belongs to M is practically 0 and we get no answer.

2.5 Discussion of Objection 2

We would like to define a Public Key Cryptosystem to be secure if an adversary, given the cyphertext, cannot obtain any partial information about the cleartext. This latter notion needs to be formalized:

> Let P be any easy to evaluate, non constant, boolean predicate defined on the message space M. Let $m \in M$. If, given the encryption of m, an adversary can efficiently compute the value of $P(m)$, then **partial information** about m can be obtained from the encryption of m

Notice that, according to the above definition, no Public Key Cryptosystem *based on trapdoor functions* is secure. In fact, if E_A is a trapdoor function, the following predicate P, defined on the cleartext, is easy to evaluate from the cyphertext: $P(x)$ is true if and only if $E_A(x)$ is even. We can avoid such problems using Probabilistic Encryption.

We know that some decision problems may be hard to solve for particular inputs, but easy to solve for most of the inputs. In view of the special purpose of Cryptography, the requirement that obtaining partial information should be difficult needs to be strengthened.

Assume that the message space has an associated probability distribution and that, with respect to this distribution, a predicate P has a probability p to be true. Without loss of generality, let $p \geq 0.5$.

Definition: An adversary has an ε advantage in evaluating the predicate P, if he can correctly guess the value of P relative to the cleartext with probability greater than $p+\varepsilon$.

We are now able to restate the previous partial information definition.

Definition: A Public Key Cryptosystem is ε secure if an adversary does not have an ε advantage in evaluating, given the cyphertext, any easy to compute predicate relative to the cleartext.

Based on the assumption that deciding quadratic residuosity modulo composite numbers is hard, we introduce an ε-secure Public Key Cryptosystem, for every non negligible, positive, real number ε. Let us first deal with the question of sending securely a single bit in a Public Key Cryptosystem. This question, closely related to the security of Partial Information, has been raised by Brassard in [7].

2.6 Attempts to Send a Single Bit Securely in Public Key Cryptosystems based on TrapDoor Functions

Suppose that user B wants to send a single bit message to user A in great secrecy. The bit is equally likely to be a 0 or a 1. B wants no adversary to have a 1% advantage in guessing correctly his message. B knows that E_A is hard to invert and tries to make use of this fact in the following way.

Idea 1: All users in the system agree on an integer i. User B selects $r \in M$ at random, except for the ith bit of r, which will be his message. B sends E_A(r) to A.

A can decode and thus get the desired bit. But what can an adversary do ?

Danger: let y = E_A(x), where E_A is a one way function. Then, given y, it could be difficult to compute x but not a specific bit of x.

Example: let p be a large prime such that $p-1$ has at least one large prime factor. Let g be a generator for Z_p^*. Then $y \equiv g^x \bmod p$ is a well known one-way function. But, even though it is difficult to compute x from $g^x \bmod p$ (the index finding problem), it is easy to get the last bit of x. In fact, x ends in 0 if and only if y is a quadratic residue mod p, For p prime we have fast random polynomial time algorithms to test quadratic residuosity, see [10].

The following idea was suggested by Donald Johnson.

Idea 2: B selects $8 \le i \le 100$ at random, and sets the ith bit of x to the bit he wants to communicate. The remaining 93 bits of x are chosen at random, except for the first 7 bits of x, which specify location i. B sends E_A(x) to A.

Danger: If, given E_A(x), we can easily compute the first 7 bits of x and one of the last 93 bits of x, then we could guess B's message with a 1/93 advantage.

Summarizing: There are many ways in which a single bit could be "embedded" in a binary number x. Taking the "exclusive or" of all the digits of x is just one more example. However, given y =E_A(x), being able to discover some particular bits embedded in x DOES NOT CONTRADICT the fact that it is hard to compute x. Then, what is a secure way to send a single bit ? The answer to this problem is discussed in the next section.

3. DECIDING QUDRATIC RESIDUOSITY IS HARD ON THE AVERAGE

The symbol (x,N) will denote the greatest common divisor of x and N. We use $\Pr(X)$ to denote the probability of the event X. We let $Z_N^* = \{ x \mid 1 \le x \le N-1 \text{ and } (x,N)=1 \}$.

3.1 Background and Notation

Given $q \in Z_N^*$, is $q \equiv x^2 \bmod N$ solvable ? If N is prime, then the answer to this question is easily computed. If a solution exists, q is said to be a **quadratic residue mod N.** Otherwise q is said to be a **quadratic non-residue mod N.** From now on let p_1 and p_2 be odd, distinct primes and $N = p_1 p_2$. Then, $q \equiv x^2 \bmod N$ is solvable if and only if both $q \equiv x^2 \bmod p_1$ and $q \equiv x^2 \bmod p_2$ are solvable. If this is the case, q is said to be a **quadratic residue mod N,** otherwise q is said to be a **quadratic non-residue mod N.** We will call the problem of determining whether an element $q \in Z_N^*$ is a quadratic residue, the **quadratic residuosity** problem.

Let p be an odd prime and $q \in Z_p^*$, then the Jacobi symbol (q/p) equals 1 if q is a quadratic residue mod p and -1 otherwise. The Jacobi symbol (q/N), is defined as $(q/N) = (q/p_1)(q/p_2)$. Despite the fact that the Jacobi symbol (q/N) is defined through the factorization of N, (q/N) is computable in polynomial time even when the factorization of N is not known !

It is easy to see, from the above definitions that if $(q/N) = -1$ then q must be a quadratic non-residue mod N. In fact, q must be a quadratic non-residue either mod p_1 or mod p_2. However, if $(q/N)=+1$, then either q is a quadratic residue mod N or q is a quadratic non-residue for both the prime factors of N.

Let us count how many of the q's, such that $(q/N) = 1$, are actually quadratic residues.

Theorem: Let p be an odd prime. Then Z_p^* is a cyclic group.

Theorem: Let g be a generator for Z_p^*, then $g^s \bmod p$ is a quadratic residue if and only if s is even.

Corollary: Half of the numbers in Z_p^* are quadratic residues and half are quadratic non-residues.

Theorem: Let $N = p_1 p_2$ where p_1 and p_2 are distinct odd primes. Then half of the numbers in Z_N^* have Jacobi symbol equal to -1 and thus are quadratic non-residues. The Jacobi symbol of the rest of the numbers is 1. Exactly half of these latter ones are quadratic residues.

3.2 A Difficult Problem in Number Theory.

If the factorization of N is not known and (q/N) =1, then there is no known procedure for deciding whether q is a quadratic residue mod N. This decision problem is well known to be hard in Number Theory. It is one of the main four algorithmic problems discussed by Gauss in his "Disquisitiones Arithmeticae" (1801). A polynomial solution for it would imply a polynomial solution to other open problems in Number Theory, such as deciding whether a composite n, whose factorization is not known, is the product of 2 or 3 primes, see open problems 9 and 15 in Adleman [3].
Recently, Adleman[1] showed that a generalization of quadratic residuosity is equivalent to factoring. Using this generalized notion in our protocol, we could base the security of our cryptosystem on factoring. At present, we await the final version of Adelman's paper.

Assumption: Let $0<\varepsilon<1$. For each positive integer k, let $C_{k,\varepsilon}$ be the minimum size of circuits C that decide correctly quadratic residuosity mod n for a fraction ε of the k bit integers n. Then, for every $0<\varepsilon<1$ and every polynomial Q, there exists $\delta_{\varepsilon,Q}$ such that $k>\delta_{\varepsilon,Q}$ implies $C_{\varepsilon,k} > Q(k)$

3.4 A number theoretic result.

We want to show that deciding whether q is a quadratic residue mod N, is not hard in some special cases, but is **hard on the average** in a very strong sense. In order to do so, let us recall the weak law of large numbers:

> If $y_1, y_2, \ldots, y_k$ are k independent Bernoulli variables such that $y_i = 1$ with probability p, and $S_k = y_1+\ldots+y_k$, then for real numbers ψ, $\delta>0$, $k \geq \frac{1}{4\delta\psi^2}$ implies that
> $$Pr\left(\left| \frac{S_k}{k} - p \right| > \psi \right) < \delta.$$

Notice that k is bounded by a polynomial in ψ^{-1} and δ^{-1}.

Let $A_N^* = \{ x \mid x \in Z_N^* \text{ and } (x/N)=1 \}$.

Definition: For a composite number N, and for real number $0 < \varepsilon \leq \frac{1}{2}$, we say that we can guess with ε advantage whether q drawn at random from A_N^* is a quadratic residue mod N if we can, in polynomial($|N|$) time, guess quadratic residuosity mod N correctly for at least $\frac{1}{2}+\varepsilon$ of the elements of A_N^*.

Theorem 1: Let $0 < \varepsilon \leq \frac{1}{2}$, $0 < \delta \leq 1$ be non-negligible numbers. Suppose we could guess, with an ε advantage whether q, drawn at random from A_N^*, is a quadratic residue mod N. Then we could decide quadratic residuosity of any integer mod N with probability $1-\delta$ by means of a polynomial in $|N|$, ε^{-1} and δ^{-1} time probabilistic algorithm.

Proof: Assume, to the contrary, that we have a polynomial time magic box MB which guesses correctly whether $q \in A_N^*$ is a quadratic residue mod N, for $\frac{1}{2}+\varepsilon$ of the elements of A_N^*. Let,
α = Pr(MB answers "q is a quadratic residue" | q is a quadratic residue mod n)
β = Pr(MB answers "q is a quadratic residue" | q is a quadratic non-residue mod N, $q \in A_N^*$).

The fraction of A_N^* on which MB is correct equals $\frac{1}{2}\alpha+\frac{1}{2}(1-\beta)$. In order for MB to have a ε advantage, it must be that $\alpha - \beta \geq 2\varepsilon$. However, α need not be equal to $\varepsilon+\frac{1}{2}$. We will now show how to get a good estimate for α.

Construct a sample of k quadratic residues chosen at random in Z_N^* (the value of k will be defined later on). This can be easily done by picking $s_1,\ldots,s_k$ at random in Z_N^* and squaring them mod N.

Initialize two counters R and NR to 0.

Feed each s_i^2 to MB. Every time that MB answers "quadratic residue", increment the R counter. Every time that MB answer "quadratic non residue", increment the NR counter.

Let $\psi=\frac{2\varepsilon}{4}$. If k is chosen to be suitably large, $k \geq \frac{1}{\delta\psi^2}$, the weak law of large numbers assures that

$$Pr\left(\left| \alpha - \frac{R}{k} \right| > \psi \right) < \frac{\delta}{4},$$

i.e. R/k is a very good approximation to how well MB guesses if the inputs are only quadratic residues.

We are now ready to determine the quadratic residuosity of elements in A_N^*.

Let q be an element of A_N^* that we want to test for quadratic residuosity. Randomly generate k quadratic residues, $x_1, \ldots, x_k$, elements of Z_N^* and compute $y_i \equiv q x_i \bmod N$ for i =1,...,k. Notice that

a) if q is a quadratic residue, then the y_i's are random quadratic residues in Z_N^*
b) if q is a quadratic non-residue in A_N^*, then the y_i's are random quadratic non-residues in A_N^*.

Let us postpone the proof of (a) and (b) and assume, for the time being, that they are true.

Initialize two counters R^* and NR^* to 0. Feed the sample $\{y_i\}$ into MB. Increment R^* every time that MB answers "quadratic residue", and NR^* every time that MB answers "quadratic non-residue". We know, that if q is a quadratic residue, then the $Pr(|\frac{R^*}{k} - \frac{R}{k}| \le 2\psi) \ge (1-\frac{\delta}{4})^2$, and if q is a quadratic non-residue then $Pr(|\frac{R^*}{k} - \frac{R}{k}| \le 2\psi) < 1-(1-\frac{\delta}{4})^2$. Thus if $\left|\frac{R^*}{k} - \frac{R}{k}\right| \le 2\psi$ then with probability greater than $1-\delta$, q is a quadratic residue mod N, otherwise, again with probability greater than $1-\delta$, q was a quadratic non-residue mod N.

We still need to prove (a) and (b). We will only prove (a) as the proof for (b) is similar. It will suffice to prove that, given **any** quadratic residue q, **any** other quadratic residue y in Z_N^* can be uniquely written as $y = q\,x$ where x is a quadratic residue mod N. It is a well known theorem in algebra that $Z_N^* = Z_{p_1}^* \times Z_{p_2}^*$. Thus let a and b be generators for $Z_{p_1}^*$ and $Z_{p_2}^*$ such that $(a, p_2)=1$ and $(b, p_1) = 1$. Then any element of Z_N^* can be written uniquely as $a^i b^j$ where $1 \le i \le p_1-1$ and $1 \le j \le p_2-1$. Moreover, q is a quadratic residue mod N if and only if it can be written as $q = a^{2i} b^{2j}$ where $1 \le 2i \le p_1-1$ and $1 \le 2j \le p_2-1$. Thus if $y = a^{2s} b^{2t}$ is any quadratic residue and $x = a^{2(s-i)} b^{2(t-j)}$, then $y = qx$ part (a) is proved. []

Theorem 2: Let $r \in A_N^*$ be a publicized quadratic non-residue mod N. Let $0 < \varepsilon \le \frac{1}{2}$, $0 < \delta \le 1$ be non-negligible numbers. Suppose we could guess with an ε advantage whether q, drawn at random from A_N^*, is a quadratic residue mod N. Then we could decide quadratic residuosity of any integer mod N with probability $1-\delta$ by means of a polynomial in $|N|$, ε^{-1} and δ^{-1} time probabilistic algorithm.

Proof:

Assume first that given **any** r quadratic non-residue mod N, $r \in A_N^*$, someone could build a polynomial time magic box MB_r that has a ε advantage in distinguishing between quadratic residues and non-residues mod N. We will show that even if one is not given such an r, quadratic residuosity can still be decided.

Construct a set T consisting of 20 elements chosen at random from A_N^*. With probability $1-(1/2)^{20}$ one of the elements in T will be a quadratic non-residue mod N. For each $x \in T$ do the following:

Choose k as in theorem 1. Construct MB_x and test its performance on k random quadratic residues, $S=\{s_1, \ldots, s_k\}$, as we did in Theorem 1. Also pick $y_1, \ldots, y_{20}$ at random from A_N^*. Again, with very high probability, at least one of the y_i's will be a quadratic non-residue. Now, construct samples $H_i=\{y_i\, s \mid s \in S\}$, and feed them into MB_x.

a) If MB_x performs on all the H_i's as it performed on S, then go to the next element in T. Halt if all elements in T have been used.

b) If MB_x performs "significantly" differently on , say H_i, than it did on S, halt.

If case (b) occurs then y_i is a quadratic non-residue and, most importantly, we obtain a magic box, MB_x, which distinguishes between quadratic residues and non-residues in random polynomial time.

Case (b) occurs when there is an $x \in T$ which is a quadratic non-residue mod N, and at least one of its corresponding y_i's is a quadratic non-residue mod N. Thus case (b) occurs with probability $\left|1-\frac{1}{2}^{20}\right|^2$. This contradicts our assumption that deciding quadratic residuosity is hard.

In the above, we assumed that given **any** quadratic non residue $r \in A_N^*$, one could construct a magic box MB_r, having a ε advantage in deciding quadratic residuosity, and we derived a contradiction.

Suppose one is able to build a MB_r, having a ε advantage in deciding quadratic residuosity, only for 1% of the quadratic non-residues, $r \in A_N^*$. Then all that would be changed in the above proof would be the size of the set T, so that T will include a suitable r. []

4. HOW TO SEND MESSAGES IN A PUBLIC KEY CRYPTOSYSTEM IN A PROVABLY SECURE WAY

Every user in the system publicizes a large composite number N whose factorization, $N = p_1 p_2$, he alone knows, and $y \in A_N^*$ such that y is a quadratic non-residue mod N.

Let N be the public key of user A. Suppose user B wants to send A a binary message $m=(m_1, \ldots, m_k)$. Then, for each m_i, B randomly picks an $x_i \in Z_N^*$ and sets

$$e_i \leftarrow \begin{cases} x_i^2 \bmod N \text{ if } m_i \text{ is a 0} \\ y x_i^2 \bmod N \text{ if } m_i \text{ is a1} \end{cases}$$

B sends $(e_1, \ldots, e_k)$ to A.

To decode m, user A, who knows the factors of N, reconstructs m by letting

$$m_i \leftarrow \begin{cases} 1 \text{ if } e_i \text{ is a quadratic residue mod N} \\ 0 \text{ if } e_i \text{ a is quadratic non residue mod N} \end{cases}$$

Testing whether $q \in A_N^*$ is a quadratic residue mod N, when the factorization of N is known, is easy by the following lemma.

Lemma 2: If the factorization of N is known, we can test whether there exists an x such that $q \equiv x^2 \bmod N$ in polynomial time.

Proof: q is a quadratic residue mod N if and only if q is a quadratic residue mod p_1 AND p_2. For a prime p, q is a quadratic residue mod p if and only if $q^{(p-1)/2} = 1 \bmod p$. Thus, to test whether q is a quadratic residue mod N we need only compute $q^{(p_1-1)/2} \bmod p_1$ and $q^{(p_2-1)/2} \bmod p_2$.

We now address the question of the security of the newly proposed Public Key Cryptosystem. Let $E(x)$ stand for our new encryption function and let M be the set of all possible messages.

The definition of security in a Public Key Cryptosystem is very difficult. It depends on the model assumed of the possible behavior of an adversary. At present, we assume that an adversary may intercept $E(m)$ and try to extract information about m. He can make use only of a computer, the cyphertext and the a priori knowledge of the message space M. No restrictions on M are assumed.

Notice that in our scheme, differently from the RSA, an adversary, given $E(m)$, may be lucky in guessing correctly m and yet not able to prove the correctness of his guess. However, the possibility of *understanding* a message, without being able to prove what it is, is still dangerous for the security of the Public Key Cryptosystem.

We show that, given $E(m)$ for $m \in M$, if an adversary can do better than guessing m at random, then deciding quadratic residuosity of any integer mod N, is easy.

Recall that $A_N^* = \{x \in Z_N^* \mid (x/N)=1\}$.

Definition: Let $x \in A_N^*$. The **signature** of x, $\sigma_N(x)$ is defined as

$$\sigma_N(x) \leftarrow \begin{cases} 1 \text{ if } x \text{ is a quadratic residue mod N} \\ 0 \text{ if } x \text{ a is quadratic non residue mod N} \end{cases}$$

Let S_N^n be the set of all sequences of n elements from A_N^*.

Definition: Let $s = (x_1, \ldots, x_n) \in S_N^n$. The **n-signature** of s, $\Sigma_N(s)$, is defined to be the string $\Sigma_N(s) = \sigma_N(x_1)\,\sigma_N(x_2) \cdots \sigma_N(x_n)$

Definition: A **decision** function is a function $d: S_N^n \to \{0,1\}$.

Let $a = (a_1, \ldots, a_n)$ and $b = (b_1, \ldots, b_n)$ be n-signatures.

Definition: The **distance** between a and b is defined to be the number of positions in which a and b differ. We say that a and b are **adjacent** if the distance between them is 1.

For any decision function d and n-signature l, let $P_d(l): \{0,1\}^n \to [0,1]$ be defined as

$$P_d(l) = \Pr\,(\, d(x)=1 \mid \Sigma_N(x) = l \text{ for } x \in S_N^n)$$

Theorem 3: Let $0 < \varepsilon \le \frac{1}{2}$ and $0 < \delta \le 1$ be non-negligible numbers. If there exists a decision function d which is easy to compute and two n-signatures, u and v, have been found such that $|P_d(u) - P_d(v)| > \varepsilon$, then we can decide quadratic residuosity of any integer mod N with probability $1-\delta$ by means of a polynomial (in $|N|$, ε^{-1}, and δ^{-1}) time probabilistic algorithm.

Proof: Suppose there exists a decision function d and two n-signatures u and v such that $|P_d(u) - P_d(v)| > \varepsilon$. Let Δ be the distance between u and v. Let $a_0, a_1, \ldots, a_\Delta$ be a sequence of n-signatures such that $a_0 = u$, $a_\Delta = v$ and a_i is adjacent to a_{i+1} for $0 \le i < m$. As $|P_d(u) - P_d(v)| > \varepsilon$, there must exist i, $0 \le i \le \Delta - 1$, such that $|P_d(a_i) - P_d(a_{i+1})| \ge \varepsilon/n$. For convenience, let $s = a_i$ and $t = a_{i+1}$.

Let us choose $\psi = \frac{\varepsilon}{4n}$. Also, let $k \ge \frac{1}{\delta\psi^2}$. Choose k elements, $x_1, \ldots, x_k$ at random from $\Omega_s = \{x \in S_N^n \mid \Sigma_N(x) = s\}$ and k elements, $y_1, \ldots, y_k$ at random from $\Omega_t = \{x \in S_N^n \mid \Sigma_N(x) = t\}$. Then, by the weak law of large numbers,

$$Pr\left(\left|P_d(s) - \frac{d(x_1)+\ldots+d(x_k)}{k}\right| > \psi\right) < \frac{\delta}{4}$$

and

$$Pr\left(\left|P_d(t) - \frac{d(y_1)+\ldots+d(y_k)}{k}\right| > \psi\right) < \frac{\delta}{4}$$

Set,

$$\alpha = \frac{d(x_1)+\ldots+d(x_k)}{k}, \quad \beta = \frac{d(y_1)+\ldots+d(y_k)}{k}$$

As $s = (s_1, \ldots, s_n)$ and $t = (t_1, \ldots, t_n)$ are adjacent, they differ in exactly one location. Call this location r. Let us assume, without loss of generality, that $s_r = 1$ and $t_r = 0$.

We will now show that we can decide quadratic residuosity mod N with probability greater than $1-\delta$. Let q be an element of A_N^* that we want to test for residuosity. Choose k random quadratic residues in A_N^*: $x_1^2, \ldots, x_k^2$ and compute $\mathbf{y}_j = q \cdot x_j^2 \bmod N$ for $1 \le j \le k$. By theorem 1, the $\mathbf{y}_j$'s are all quadratic residues if

q is a quadratic residue and all quadratic non-residues in A_N^*, otherwise.

In theorem 2 we showed that knowing a non-residue in A_N^* does not help in deciding quadratic residuosity. Therefore we can assume that such a non-residue, h, is known. This allows us to pick quadratic non-residues at random from A_N^* (by computing hx^2).

We are now ready to decide whether q is a quadratic residue.

(* Construct a random sample of k elements $(y_{1,1}, \ldots, y_{1,n}), \ldots, (y_{k,1}, \ldots, y_{k,n}) \in {S_N}^n$ such that for all $1 \le i \le n$, $i \ne r$, $1 \le j \le k$, $\sigma_N(y_{j,i}) = s_i$ and for all $1 \le j \le k$, $y_{j,r} = y_j$. *)

For i = 1,..., r-1, r+1 ,... n do
  begin
  For j = 1,...,k do
    draw $x \in A_N^*$ at random.
    if $s_i = 1$ then $y_{j,i} := x^2 \bmod N$
    else if $s_i = 0$ then $y_{j,i} := hx^2 \bmod N$
  end.

(* Evaluate the decision function d on each member of the sample *)

For j = 1,...,k do
  $X_j = d(y_{j,1}, \ldots, y_{j,r-1}, y_j, y_{j,r+1}, \ldots, y_{j,n})$

Notice that the entire sample $\{y_{j,1}, \ldots, y_{j,r-1}, y_j, y_{j,r+1}, \ldots, y_{j,n} \mid 1 \le j \le k\}$ is either a subset of Ω_s or a subset of Ω_t. Thus with probability greater than $1-\delta$ one of the following two mutually exclusive events will occur:

$$(1) \quad \left| \frac{(X_1 + \ldots + X_k)}{k} - \alpha \right| < \frac{\varepsilon}{2n}$$

or

$$(2) \quad \left| \frac{(X_1 + \ldots + X_k)}{k} - \beta \right| < \frac{\varepsilon}{2n}.$$

If case (1) occurs, we conclude, with probability greater than $1-\delta$, that q is a quadratic residue. Otherwise, we conclude, again with probability greater than $1-\delta$ that q is a quadratic non-residue. □

The notion of a decision function is immediately generalized to that of a discriminating function. This is a decision function which can take on more than 2 values. For any non empty set Ω, let $D: {S_N}^n \to \Omega$. Let $a \in \Omega$, then $P_{D,a}(l) = Pr(D(x) = a \mid \Sigma_N(x) = l \text{ for } x \in {S_N}^n)$. The following theorem is an easy extension of theorem 3 and we will state it without proof.

Theorem 4: Let $0 < \varepsilon \le \frac{1}{2}$ and $0 < \delta \le 1$ be non-negligible numbers. If there exists a discriminating function $D: {S_N}^n \to A$, which is easy to compute and two n-signatures, u and v, have been found such that $|P_{D,a}(u) - P_{D,a}(v)| > \varepsilon$, then we can decide quadratic residuosity of any integer mod N with probability $1-\delta$ by means of a polynomial (in $|N|$, ε^{-1}, and δ^{-1}) time probabilistic algorithm.

Let us introduce some more notation. Let, $M^n = \{m_1, m_2, \ldots, \}$ be the set of messages whose length is n, where n is bounded by a polynomial function in $|N|$. Set $k = |M^n|$. Let M_i be the set of all possible encodings of message $m_i \in M^n$, using the scheme described at the beginning of this section. Clearly, $M_i \subset {S_N}^n$ and for all i and j, $|M_i| = |M_j|$. Set $\chi = |M_i|$.

4.1 The Security of Partial Information

In the present version of the paper, we assume that all messages in M^n are equally likely. Let P be an easy to evaluate predicate, defined on M^n. Let p be the probability that $P(x)$ is true for a random $x \in M^n$. Since M^n is uniformly distributed, and $|M^n| = k$, P must evaluate to 1 on pk messages in M^n.

Let MB be a magic box that receives as input the cyphertext $E(m) \in {S_N}^n$, where $m \in M^n$, and outputs 0 or 1, its guess for the value of $P(m)$. Let 0_j be the number of 0's and let 1_j be the number of 1's that MB guesses on encodings of m_j. Clearly, $0_j + 1_j = \chi$. Let

$$C_j = \begin{cases} 1_j & \text{if } P(m_j) = 1 \\ 0_j & \text{if } P(m_j) = 0 . \end{cases}$$

C_j represents the number of encodings of message m_j on which MB correctly guesses the value of $P(m_j)$.

Theorem 5: Let $0 < \delta < 1$ be a non negligible real number. If $\frac{1}{k\chi} \sum_{j=1}^{k} C_j \ge p + \varepsilon$, for some non-negligible real $\varepsilon > 0$, then we could decide quadratic residuosity of any integer mod N with probability $1-\delta$ by means of a polynomial in $|N|$, ε^{-1}, and δ^{-1} time probabilistic algorithm.

Proof: Let us partition M^n into $10/\varepsilon$ buckets, $M^n = \bigcup_{i=1}^{10/\varepsilon} B_i$, such that $m \in B_i$ if and only if $(i-1)\frac{\varepsilon}{10} \le \frac{1_m}{\chi} < i\frac{\varepsilon}{10}$. We show that there exist two non-adjacent buckets, each containing a non-negligible portion of the messages. More formally, we show there exist g, h where $1 < h+1 < g \le 10/\varepsilon$ such that $|B_g|, |B_h| > \frac{1}{(10\varepsilon^{-1})^2} k$. Say, that B_i is big if $|B_i| > \frac{1}{(10\varepsilon^{-1})^2} k$ and small otherwise. Then we want to show that there are two non adjacent big buckets. Assume, for contradiction, that this is not the case. Then one of the following cases must apply:

1) There are no big buckets.
2) There is only one big bucket: B_i

3) There are exactly two adjacent big buckets: B_i and B_{i-1}

Note that case 1 can never be true; otherwise $k = \sum_{i=1}^{10\varepsilon^{-1}} |B_i| \le \frac{k}{10\varepsilon^{-1}} < k$. In case 2, $\sum_{m_j \in B_i} C_j$ is maximum for $i = \frac{\varepsilon}{10}$, and if all messages m_j for which $P(m_j) = 1$ belong to $B_{\frac{\varepsilon}{10}}$, i.e when MB guesses 1 for all the encodings of all the messages for which the predicate is true.

$$\text{Thus, } p+\varepsilon \le \frac{1}{k\chi} \sum_{m_j \in M^n} C_j$$

$$= \frac{1}{k\chi}\left(\sum_{m_j \in B_i} C_j + \sum_{m_j \in B_k, k \neq i} C_j \right) \le p + \frac{\varepsilon}{10} < p+\varepsilon$$

In case 3, $\sum_{m_j \in B_i} C_j + \sum_{m_j \in B_{i-1}} C_j$ is maximum when $i = \frac{\varepsilon}{10}$ and all the messages for which P is true belong to $B_{\frac{\varepsilon}{10}}$ and all the messages for which P is false belong to $B_{\frac{\varepsilon}{10}-1}$

$$\text{Thus, } p+\varepsilon \le \frac{1}{k\chi} \sum_{m_j \in M^n} C_j =$$

$$\frac{1}{k\chi}\left\{ \left(\sum_{m_j \in B_i} C_j + \sum_{m_j \in B_{i-1}} C_j \right) + \sum_{m_j \in B_k, k \neq i, i+1} C_j \right\}$$

$$\le \frac{1}{k\chi}\left\{ [\, pk\chi + (1-p)2\varepsilon 10^{-1} k\chi \,] + k\chi\varepsilon 10^{-1} \right\}$$

$$\le \frac{1}{k\chi}(pk\chi + 3\varepsilon 10^{-1} k\chi) < p + \frac{\varepsilon}{2}$$

In all three cases we reach a contradiction.

Thus there exist two non adjacent buckets B_g and B_h each containing at least $\frac{\varepsilon}{10}\frac{2}{k}$ messages. By sampling, we can find, in a small expected time, two messages u and v in B_g and B_h, respectively. We view MB as a decision function $D: S_N{}^n \to [0,1]$. Then, $P_D(u) - P_D(v) > \frac{\varepsilon}{10}$ and theorem 3 applies. □

Next, we will see that an adversary cannot decode more than a negligible fraction of the encodings of all messages.

4.2 An Adversary Cannot Decode.

Let MB be a magic box that receives as input $E(m)$ for $m \in M^n$, and outputs m_i. MB's output can be interpreted as MB's guess of what m is.

Let $r_{j,i}$ denote the number of encodings of message m_j, on which MB answers m_i. Clearly, $r_{i,i}$ will denote the number of times, over all possible encodings of m_i, that MB answers correctly.

Theorem 6: Let $0 < \delta < 1$ be a non negligible real number. If $\sum_{i=1}^{k} \frac{r_{i,i}}{k\chi} > \varepsilon + \frac{1}{k}$ for some non-negligible $\varepsilon < 1 - \frac{1}{k}$, then we can decide quadratic residuosity mod N with probability $1-\delta$ by means of a polynomial in $|N|$, ε^{-1} and δ^{-1} time probabilistic algorithm.

Proof: Say that a message m_i is **well decoded** if $r_{i,i} > (\frac{1}{2}\varepsilon)\chi$. Let, W be the set of well-decoded messages and $W' = M^n - W$.

Claim 1: There exist at least $\frac{\varepsilon k}{2}$ well-decoded messages.

Proof:

$$\varepsilon k\chi < \varepsilon k + \chi < \sum_{i=1}^{k} r_{i,i} = \sum_{i \in W} r_{i,i} + \sum_{i \in W'} r_{i,i}$$

$$\le \chi |W| + (k - |W|)\frac{1}{2}\varepsilon\chi = \chi\left[(1 - \frac{1}{2}\varepsilon)|W| + k\frac{1}{2}\varepsilon \right]$$

$$\text{Hence, } \frac{|W|}{k} > \frac{\varepsilon/2}{(1-\varepsilon/2)} > \frac{\varepsilon}{2}. \text{ (claim 1) } \square$$

Clearly, if we pick messages at random from M^n, we expect to find a well-decoded message in $2\varepsilon^{-1}$ trials. Let $\Omega \subset W$ such that $|\Omega| > 2\varepsilon^{-1}$ and let $\rho > \frac{1}{2\varepsilon^{-1}(2\varepsilon^{-1}+1)}$.

Claim 2: There exists two well-decoded messages $m_i, m_j \in \Omega$ such that $\left| \frac{r_{i,i}}{\chi} - \frac{r_{j,i}}{\chi} \right| > \rho$

proof: Fix $m_j \in \Omega$. How many messages $m_i \in \Omega$ can be such that $\left| \frac{r_{i,i}}{\chi} \frac{r_{j,i}}{\chi} \right| \le \rho$? There are at most $\frac{1}{(\frac{1}{2}\varepsilon - \rho)} < 2\varepsilon^{-1} + 1$ such messages. Thus there exists an $m_i \in \Omega$ that satisfies the claim. (claim 2) □

Let us transform MB into a discriminating function $D: S_N{}^n \to M^n \cup \{\gamma\}$. If $x \in S_N{}^n$ and MB, on input x, outputs m_j, then set $D(x) = m_j$. If y is not the encoding of any message, then one of 3 cases must occur:

1) MB outputs m_i for $1 \le i \le t$. Set $D(y) = m_i$.
2) MB outputs m_i for $i < 1$ or $i > t$. Set $D(y) = \gamma$.
3) MB does not answer within a certain time limit. Set $D(y) = \gamma$.

Now, note that in claims 1 and 2 just proved above, we showed that we can quickly find two well-decoded messages m_i and m_j such that $|P_{D,m_i}(m_i) - P_{D,m_i}(m_j)| > \rho$. Thus the hypothesis of theorem 4 holds and deciding quadratic residuosity mod N is polynomial in $|N|$, ε^{-1} and δ^{-1}. □

Theorem 6 shows that inverting the function E on the encrypted messages is as hard as deciding quadratic residuosity, independently of the sparsity of M^n.

5. MENTAL POKER

Mental Poker is played like regular poker except that there are no cards and no deck The game is played over the telephone lines, or over a computer network. Since we cannot send physical cards over the phone lines, dealing and playing must be simulated by exchanging messages between the players. The players do not trust each other more than ordinary players do. A **fair game on the telephone** should ensure that:

1) Neither player can have any partial information about the cards in his opponent's hand or in the deck,
2) There is no overlap in the cards dealt to players,
3) All possible hands are equally probable for both players.
4) At the end of the game each player can verify that the game was played according to the rules and no cheating occurred.

Note that in a fair game of Mental Poker it is not enough to show that it is computationally difficult to get the exact value of a card. We must also show that no partial information about the card can fall into the hands of an adversary.

We present a protocol for two people to play a fair game of Mental Poker, using encryption. We prove that there is no way a player can get any information about cards not in his hand under the assumption that deciding quadratic residuosity is hard.

There are two main tools used in our implementation of Mental Poker. One is a method for coin-flipping over the telephone[5] and the other is the method for sending a single bit securely in a Public Key Cryptosystem presented here.

A different solution to the problem of Mental Poker has been obtained independently by Manuel Blum in [6]. His solution is based on the assumption that factoring is hard and that completely secure one way functions exist.

5.1 Background For Coin Flipping

To *flip a coin in the well* - A and B stand far apart from each other. B is standing next to a deep well. A throws a coin into the well from a distance. Now, B knows the outcome of the flip (by looking into the well) but can not change it, and A has no way of knowing the outcome. Later on when B would like to prove to A that he won (or lost), he lets A come closer and look into the well.

Essentially, if we can simulate a flip in the well by exchanging messages over the telephone, A can send a **random** bit to B, where A does not know what he sent, but B can, if necessary, prove to A what the bit was. This is especially applicable to cryptographical games.

The notion of coin flipping in the well has been introduced by Blum and Micali in [5], in which, based on the assumption that index finding is hard, they show how to flip a coin in the well over the telephone lines. Another method based on the assumption that factorization is hard has been found by Blum in [4]. We sketch a third method, based on the difficulty of distinguishing quadratic residues from non-residues with respect to composite moduli.

> A and B want to flip a coin. A generates two large odd primes at random, P and Q and sets $N=P*Q$. A publicizes N and $y \in A_N^*$ such that y is a quadratic non-residue mod N. A picks a number q at random from A_N^* and asks B, who does not know the factorization of N, whether q is a quadratic residue mod N or not. B tells A what his guess his. A now knows whether B won (lost), and can later prove to B that he indeed won(lost) by releasing the factorization of N.

To avoid adding new assumptions to the ones that we already have, we propose to use one of these latter two coin flipping methods in our protocol for Mental Poker.

The next section will list some known results that will be used in the proof of the protocol.

5.2 Useful Results

Let p_1, p_2 be odd primes and $N = p_1 p_2$.

Lemma 3: If the factorization of N is known, we can find $q \in Z_N^*$ such that (q/N) =1 and q is a quadratic non-residue, in random polynomial time.

Proof: Pick $a \in Z_{p_1}$ such that $(a/p_1)=-1$. This can be done in 2 expected trials. Similarly, pick $b \in Z_{p_2}$ such that $(b/p_2)=-1$. Using the Chinese Remainder theorem compute the unique $q \in Z_N^*$ such that $q \equiv a \ (mod\ p_1)$ and $q \equiv b \ (mod\ p_2)$. Now, q is a quadratic non-residue and $(q/N) = (q/p_1p_2) =$

$$(q/p_1)\cdot(q/p_2) = (a/p_1)\cdot(b/p_2)=1.$$

Lemma 4: Let $N=p_1p_2$ such that $p_1 \equiv p_2 \equiv 3 \ mod\ 4$. For all $x, y \in Z_N^*$, if $x^2 \equiv y^2 \ mod\ N$ and $x \neq \pm y \ mod\ N$ then $(x/N) = -(y/N)$.

Proof: Let

$$c \leftarrow \begin{cases} 1 \ (mod\ p_1) \\ 0 \ (mod\ p_2) \end{cases}$$

$$d \leftarrow \begin{cases} 1 \ (mod\ p_2) \\ 0 \ (mod\ p_1) \end{cases}$$

We can find c and d through the Chinese Remainder Theorem. Let $a^2 \equiv x^2 \pmod{p_1}$ and $b^2 \equiv x^2 \pmod{p_2}$. Then the four square roots (mod N) are given by $ac+db$, $-ac+db$, $-(ac+db)$ and $(ac-db)$. Let $x = ac+db$, and $y = -ac+bd$. Since $N \equiv 1 \bmod 4$ implies $(x/N)=(-x/N)$, we need only prove that $(+x/N)=-(+y/N)$. Thus, $(x/N) = (ac+bd/N) = (ac+bd/p_1)(ac+bd/p_2) = (ac/p_1)(bd/p_2)$. And $(y/N) = (-ac+bd/N)=(-ac+bd/p_1)(-ac+bd/p_2)= (-ac/p_1)(bd/p_2)=(-1/p_1)(x/N)$. Since $p_1 = 3 \pmod 4$, $(-1/p_1)=-1$.

□

By a theorem of de la Vallee Poussin[15], approximately half of all primes of a given length are congruent to 3 mod 4. Thus, composite numbers of the form $N=p_1p_2$ where $p_1 \equiv p_2 \equiv 3 \bmod 4$ constitute approximately 1/4 of all composite numbers which are a product of two odd primes of a given length. Thus factoring and deciding quadratic residuosity modulos such special N's remains a hard problem. Another method, which does not use special composite numbers, but increases the number of messages exchanged in the protocol, will appear in the final paper.

5.3 THE PROTOCOL

To represent 52 cards in binary we must use at least 6 bits per card. Thus at first A and B agree on 52 different bit patterns which correspond to the 52 cards.

From now on, when we say that A flips k to B, we mean that B receives a number k at random from A, and A has no information whatsoever about k. k is actually sent bit by bit through a sequence of *coin flips into a well.*

5.3.1 The Algorithm

STEP 1: B chooses at random 52 pairs of large prime numbers: (p_1, q_1), (p_2, q_2), (p_3, q_3) ,..., (p_{52}, q_{52}) such that $p_i \equiv q_i \equiv 3 \bmod 4$ for $1 \le i \le 52$, and produces 52 large composite numbers whose factorization she knows, i.e $N_1 := p_1 \cdot q_1$, $N_2 := p_2 \cdot q_2$,..., $N_{52} := p_{52} \cdot q_{52}$. Next, she shuffles the deck of cards in her hands and assigns N_1,..., N_{52} to the shuffled deck, an N_i per the i th card. She publicizes the ordered 52 tuple $< N_1, N_2, ..., N_{52} >$.

STEP 2: A does the same. Let us denote the primes chosen by him as (s_1, t_1), (s_2, t_2), (s_3, t_3),..., (s_{52}, t_{52}) such that $s_i \equiv t_i \equiv 3 \bmod 4$ for $1 \le i \le 52$, and his 52 composite numbers by $M_1 := s_1 \cdot t_1$, $M_2 := s_2 \cdot t_2$,..., $M_{52} := s_{52} \cdot t_{52}$. He shuffles the deck of cards and assigns M_1,..., M_{52} to the shuffled deck, an M_i per the i th card. He publicizes the ordered 52 tuple $< M_1, M_2, ..., M_{52} >$.

STEP 3: B publicizes his entire deck. The deck is encrypted in the following way. For every card C_i (with public key N_i), B publicizes an ordered list of 6 numbers in $A_{N_i}^*$, $(q_1,...,q_6)$ such that for $1 \le j \le 6$, q_j is a quadratic residue if and only if the jth bit of C_i is a 1.

For example, let the first card in B's deck be 010010. Then B publicizes $(q_1, q_2, q_3, q_4, q_5, q_6)$ where q_1, q_3, q_4 and q_6 are quadratic non-residues mod N_i, and q_2, q_5 are quadratic residues mod N_i with Jacobi symbol 1. The q_i's are chosen at random among the elements of $A_{N_i}^*$ with the desired properties. This can be done in random polynomial time, by Lemma 3.

NOTE that, by Lemma 2, if A can factor N_i, he can also determine whether the numbers that B posed as corresponding to the bits in the encoding of C_i are quadratic residues or not and therefore determine what the card is. If A can not factor N_i, he can not tell whether the numbers corresponding to bits in the cards encoding are quadratic residues or not, and therefore can not tell what the remaining cards are.

STEP 4: A publicizes his deck in the exact same way that B did.

STEP 5 [B deals a card to A]: Suppose A decided to pick the K- th card from B's deck. Repeat the following procedure for each card in B's encrypted deck. We describe it for the i-th card, to which N_i corresponds. B flips $x \in Z_{N_i}^*$ to A. A computes $x^2 \bmod N_i$ and (x/N_i). At this point A must follow one of two procedures: P1 if $i=K$ and P2 otherwise.

P1: A sends x^2 mod N_i and $-(x/N_i)$ to B.
P2: A sends x^2 mod N_i and (x/N_i) to B.

B computes the square roots of x^2 mod N_i. Let the square roots be x, $n-x$, y and $n-y$. Next, B sends the root whose Jacobi symbol she received from A : y if she received $-(x/N_i)$ from A, and x otherwise. By lemma 4, (x/N_i) uniquely identifies x, and $-(x/N_i)$ uniquely identifies y. Thus if A followed P1 then he will receive 4 square roots of x^2 mod N_i, and by lemma 1 can factor. If A followed P2, he will get no new information as to the value of C_i. B

from her side has no information as to which card A selected. Later, B can verify what he flipped to A, and hence verify that B has only found out the factorization of a single card.

STEP 6: At this point A knows the factorization of N_K. To reconstruct the actual card C_K, A applies the polynomial time test of Lemma 2 to the encrypted representation of C_K, $(q_1, \ldots, q_6)$. Next, A must delete C_K from his encrypted deck. B can see which encrypted element in A's deck is being erased, but this does not enable her to decrypt it.

STEP 7[A deals a card to B]: Clearly, the same procedure as in Step 5 and 6 is done with the roles of A and B reversed. Now B will discover the factorization of one of $M_1,..,M_{52}$.

STEP 8: If any more cards need to be dealt throughout the game, a similar protocol takes place. Whenever A needs a card, he will pick a card from B's deck, by following the procedure in step 5 and 6. And similarly whenever B needs a card, she will pick it from A's deck.

STEP 9 [after game verification]: After the game is over, A can prove to B that everything he claims she flipped him, was indeed flipped by her and in what order. B can do the same. A releases the factorization of each of the M_i for all $1 \le i \le 52$, and B releases the factorization of each of the N_i for all $1 \le i \le 52$. They can both prove to each other whatever claim they made in the game such as "N is a product of two primes", "all cards where present at the deck at all times", "these are the quadratic residues you flipped to me", or "I won".

5.3.2 Proof Of Correctness:

Claim 1: all hands are equally probable.
Proof: In step 9, A and B verify that both encrypted decks contained all 52 cards. In step 5, A himself chooses which encrypted value from B's deck he wants, thus he is equally likely to get any card in the deck. Similar reasoning holds for B.

Claim 2: no overlapping or repeating hands.
Proof: When A is dealt a card, he erases that card from his encrypted deck. Thus B can never be dealt the same card. A knows which cards he picked from B's deck, and thus will never pick the same card twice.

Claim 3: If player A knows the factorization of N_i he can reconstruct C_i in $O(|N|^3)$ time.
Proof: We are given $N_i = p_1 p_2$, and $(q_1,\ldots,q_6)$ such that for all j, $q_j \in Z_N^*$ and $(q_j / N_i) = 1$. To reconstruct C_i, we must test whether q_j is a quadratic residue mod N_i for all j. That can be done in $O(|N|^3)$ steps by Lemma 2.

It still remains to be shown that neither player can have, at any stage of the game, any partial information about a single encrypted card not in his hand, or any subset of encrypted cards not in his hand. A complete proof will be found in the final paper. Here we restrict ourselves to proving that when two players A and B publicize their respective encrypted decks, neither A nor B can answer quickly with 1% advantage a 1 bit question about a single card in the opponents deck. Examples of such 1 bit questions are: is the i-th card in the deck black?, Are the first and third bit of the i-th card equal? Is the mod 2 sum of the bits in the i-th card 0 or 1?

Theorem 7: If A, when B publicizes her encrypted deck, can answer, in polynomial time, a 1-bit question Q about a single card in B's deck with 1% advantage, then he can decide quadratic residuosity modulo a random composite N with probability 1, by means of a polynomial(|N|) time probabilistic algorithm.
Proof: Suppose A can answer a 1-bit question Q about card i, to which composite N_i corresponds. A's ability to answer Q with a 1% advantage can be viewed as a decision function $d: S^6 \to 0,1$ (S^6 = all 6-long sequences of elements from $A_{N_i}^*$). Since A answers Q correctly 51 times out of a 100, we can efficiently find two 6-signatures u and v such that $|P_d(u) - P_d(v)| \ge 1/100$. Thus we can apply theorem 3 and decide quadratic residuosity modulo N_i in polynomial time. Contradiction!

□

5.3.3 Implementation Details

In order to perform the protocol we must be able to do the following:

1. Generate large prime numbers. This can be done using Gary Miller's test for primality[11].
2. Find square roots of $x^2 \bmod N$ when the factorization of N is known. Use Adleman, Manders and Millers polynomial time algorithm[2] for finding square roots.

6. Remarks and Further Improvements

In this paper we showed that it is possible to encrypt messages in such a way, that an adversary, given the cyphertext, cannot extract information about the cleartext. This is sufficient for protocols such as Mental Poker or for encrypting one's private files. An adversary can read these files but cannot understand them.

We also showed that Probabilistic Encryption can be used in a Public Key Environment. However, in a Public Key Cryptosystem, getting hold of the cyphertext and trying to understand it is the most obvious attack to the secu-

rity of the scheme.

- An adversary could, as a user, try to break the scheme by communicating.
- He could try to break the scheme by intercepting some other user's messages and changing them.
- Finally, he may try to break the scheme by making use of the decoding equipment !

The Public Key Cryptosystem presented in this paper is not secure against these possible attacks. However, by forcing the users to follow a particular protocol for exchanging messages, we have built a Public Key Cryptosystem which is provably secure against the above mentioned attacks. These results will appear in a future paper.

Acknowledgements

Our most sincere thanks go to Richard Karp, who supervised this research, for his contributions, encouragement and great patience, and to Manuel Blum for a wonderful course in Number Theory, many insightful discussions and for having found a way to reduce the numbers of messages exchanged in the protocol.

We are particularly indebted to Faith Fich, Mike Luby, Jeff Shallit and Po Tong. Without their generous help this paper would have never been written.

Andrew Yao pointed out to us some general difficulties arising with commutative encryption functions. The claim in section 2.4 was obtained with Vijay Vazirani. We thank them both.

We are grateful to Ron Rivest and Mike Sipser for a very inspiring discussion. It improved this paper a great deal.

References

[1] Adleman, L., *Private Communication*, 1981.

[2] Adleman, L., Manders K. and Miller G., *On Taking Roots In Finite Fields*, Proceedings of the 18th Annual IEEE Symposium on Foundations of Computer Science (FOCS), 1977, 175-177.

[3] Adleman, L., *On Distinguishing Prime Numbers from Composite Numbers*, Proceedings of the 21st IEEE Symposium on the Foundations of Computer Science (FOCS), Syracuse, N.Y., 1980, 387-408.

[4] Blum, M., *Three Applications of The Oblivious Transfer*, to appear, 1981.

[5] Blum, M., and Micali, S., *How to Flip A Coin Through the Telephone*, to appear, 1982.

[6] Blum, M., *Mental Poker*, to appear, 1982.

[7] Brassard, G., *Relativized Cryptography*, Proceedings of the 20st IEEE Symposium on the Foundations of Computer Science (FOCS) , San Juan, Puerto Rico, 1979, 383-391.

[8] Diffie, W., and M. E. Hellman, *New Direction in Cryptography*, IEEE Trans. on Inform. Th. IT-22, 6 (1976), 644-654.

[9] Goldwasser S., and Micali S., *A Bit by Bit Secure Public Key Cryptosystem*, Memorandum NO. UCB/ERL M81/88, University of California, Berkeley, December 1981.

[10] Lipton, R., *How to Cheat at Mental Poker*, Proceeding of the AMS short course on Cryptology, January 1981.

[11] Miller, G., *Riemann's Hypothesis and Tests for Primality*, Ph.D. Thesis, U.C. Berkeley, 1975.

[12] Rabin, M., *Digitalized Signatures and Public-Key Functions As Intractable As Factorization*, MIT/LCS/TR-212, Technical Memo MIT, 1979.

[13] Rivest, R., Shamir, A., Adleman, L., *A Method for Obtaining Digital Signatures and Public Key Cryptosystems*, Communications of the ACM, February 1978.

[14] Shamir, Rivest, and Adleman, *Mental Poker*, MIT Technical Report, 1978.

[15] Shanks, D., *Solved and Unsolved Problems in Number Theory*, Chelsea Publishing Co. (1978).

Added in proof:

[16] Chaum, D. L., *Untraceable Electronic Mail, Return Addresses, and Digital Pseudonymus*, Communications of the ACM, 24,2 (1981) 84-88.