A "PARADOXICAL" SOLUTION TO THE SIGNATURE PROBLEM*

(Extended abstract)

Shafi Goldwasser**
Silvio Micali**
Ronald L. Rivest**

Brief Abstract

We present a general signature scheme which uses any pair of trap-door permutations (f_0, f_1) for which it is infeasible to find any x, y with $f_0(x) = f_1(y)$. The scheme possesses the novel property of being robust against an adaptive chosen message attack: no adversary who first asks for and then receives signatures for messages of his choice (which may depend on previous signatures seen) can later forge the signature of even a single additional message.

For a specific instance of our general scheme, we prove that

(1) forging signatures is provably equivalent to factoring, while

(2) adaptive chosen message attacks are of no help to an "enemy" who wishes to forge a signature.

Such a scheme is "paradoxical" since the above two properties were believed (and even "proven" in the folklore) to be contradictory.

The new scheme is potentially practical: signing and verifying signatures are reasonably fast, and signatures are not too long.

Keywords: Cryptography, digital signatures, factoring, chosen message attacks, authentication, claw-free pairs of functions, randomization.

I. INTRODUCTION.

The idea of a "digital signature" first appeared in Diffie and Hellman's seminal paper, "New Directions in Cryptography"[DH76]. They propose that user A's signature for a message M should be a value which depends on M and on information held secret by A such that anyone can verify the validity of A's signature (using information published by A) but no one can forge A's signature on any messages. They also proposed a way of implementing signatures based on "trap-door functions" (see section II.A).

While the notion of a digital signature is robust, useful, and even legal [LM78, Ma79], a number of technical problems arise if they are implemented as suggested using trap-door functions; these problems have been addressed in part elsewhere. For example, [GMY83] showed how to handle arbitrary or sparse messages sets and how to ensure that if an enemy sees previous signatures it does not help him to forge new signatures (this is a so-called "non-adaptive chosen message attack"). For further discussion see section IV.

One difficult problem with simple trap-door signature schemes is proving they are secure against *adaptive* chosen message attacks, where the enemy can request signatures of messages which depend on previously obtained signatures.

We present a new digital signature scheme that is seemingly "paradoxical", in that we prove that forgery is equivalent to factoring, even if the enemy uses an *adaptive* chosen message attack.

We can restate the paradox as follows:

- Any general technique for forging signatures can be used as a "black box" in a construction that enables the enemy to factor one of the signer's public moduli (he has two in our scheme),

but

- The technique of "forging" signatures by getting the real signer to play the role of the "black box" (i.e. getting the real signer to produce some desired genuine signatures) does not help the enemy to factor either of the signer's moduli.

Resolving this paradox was previously believed to be impossible and contradictory [Wi80, misled by Rivest].

From a cryptographer's viewpoint, the following points might be judged to be even more significant than resolving the apparent paradox:

- What we prove to be difficult is *forgery*, and not merely obtaining the secret trap-door information embedded in the signing algorithm (or obtaining an efficient equivalent algorithm).
- Forgery is proven to be difficult for a "most general" enemy who can mount an "adaptive chosen message attack": an enemy who can use the real signer as "an oracle" can not in time polynomial in the size

* This research was supported by NSF grant MCS-80-06938, an IBM/MIT Faculty Development Award, and DARPA contract N00014-85-K-0125.

** MIT Laboratory for Computer Science, Cambridge, Mass. 02139

0272-5428/84/0000/0441$01.00 © 1984 IEEE

of the public keys forge a signature for any message whose signature was not obtained from the oracle. In contrast to all previous published work on this problem, we prove the scheme invulnerable against such an "adaptive" attack (where each message whose signature is requested may depend on all signatures previously obtained from the oracle). We believe that such an "adaptive chosen message attack" to be the most powerful attack possible for an enemy who is restricted during his attack to using the signature scheme in a natural manner.

- The properties we prove about the new signature scheme do not depend in any way on the set of messages which can be signed or on any assumptions about an input probability distribution on the message set.
- Our scheme can be generalized so that it can be based on "hard" problems other than factoring whenever one can create (so-called "claw-free") pairs of trap-door permutations (f_0, f_1) such that the hard problem is equivalent to finding x, y with $f_0(x) = f_1(y)$ (a "claw" – see Figure 1). The paradoxical nature of the signature scheme remains.

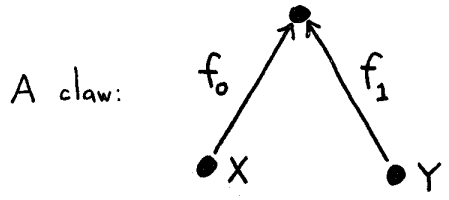

Figure 1.

The scheme has a "pumping" nature: using any family of pairs of trap-door permutations we can produce a signature scheme that is *invulnerable* to a chosen message attack, even if the the trap-door permutations are *vulnerable* to a chosen message attack when used to make a trap-door signature scheme (see section II).

Fundamental ideas in the construction are the use of randomization, signing by using two authentication steps (the first step authenticates a random value which is used in the second step to authenticate the message), and the use of a tree-like branching authentication structure to produce short signatures.

We note that because our signature scheme is randomized it is not of the simple Diffie-Hellman "trap-door" type. (For example, a given message can have many signatures.)

The rest of the paper is organized as follows. In section II we review the fundamental notions of what it means to "break" a signature scheme and what it means to "attack" a signature scheme. In section III we review more closely the nature of the "paradox", and present the folklore "proof" that it is impossible to have a signature scheme for which forgery is provably equivalent to factoring and which is simultaneously invulnerable to an adaptive chosen message attack. In section IV we review previously proposed signature schemes. In section V we give the details of our proposed signature scheme, and in section VI we prove that it has the desired properties.

II. FUNDAMENTAL NOTIONS

To properly characterize the results of this paper, it is helpful to answer the following questions:

- What is a digital signature scheme?
- What kinds of attacks can the enemy mount against a digital signature scheme?
- What is meant by "breaking" the signature scheme?

II.A. WHAT IS A DIGITAL SIGNATURE SCHEME?

A *digital signature scheme* contains the following components:

- A *key generation algorithm* $\kappa(R, k)$ which any user A can use to produce a pair (P_A^k, S_A^k) of matching *public* and *secret* keys from inputs k and (random) input R. (The secret key is sometimes called the *trap-door information*. The parameter k is called the *security parameter*; a number of quantities (e.g. length of signatures, overall security) may depend on k.
- A *message space* M which is the set of messages to which the signature algorithm may be applied. We assume here that the messages are represented in some encoding suitable for the signature algorithm.
- A *signature algorithm* which produces a signature $\sigma(M, S_A, R)$ for a message M using the secret key S_A and random input R. (This is the *memoryless* model; it is also permissible to have the signature algorithm depend on the number of messages previously signed and even how they were signed. The scheme proposed in this paper is not memoryless.)
- A *verification predicate* $\tau(S, M, P_A)$ which tests whether S is valid signature for message M using the public key P_A.

We note that there are other kinds of "signature" problems which are not dealt with here; the most notable being the "contract signing problem" where two parties wish to exchange their signatures to an agreed-upon contract *simultaneously* (for example, see [EGL82]).

II.A.1 TRAP-DOOR SIGNATURES

To create a signature scheme Diffie and Hellman proposed that A use a "trap-door function" f: a function for which it is easy to evaluate $f(x)$ for any argument x but for which, given only $f(x)$, it is computationally infeasible to find *any* y with $f(y) = f(x)$ without the secret "trap-door" information. Then A publishes f and anyone can validate a signature by checking that $f(signature) = message$. Only A possesses the "trap-door" information allowing her to invert f: $f^{-1}(message) = signature$. A *trap-door permutation* is a trap-door function which is one-to-one and onto; then any message can be signed since the domain of f^{-1} is the entire message space. We call any signature scheme

that fits into this model (i.e. uses trap-door functions and signs by apply f^{-1} to the message) a *trap-door signature scheme*.

We note that not all signature schemes are trap-door schemes, although most of the proposals in the literature are of this type.

II.B. KINDS OF ATTACKS

The enemy may mount an attack knowing only the real signer's public key – what we call a *direct attack*. Of more concern, however, are what we call *known* or *chosen message attacks* where the enemy is able to examine some signatures corresponding to either known or chosen messages before his attempt to break the scheme. (These are analogous to "chosen ciphertext attacks" for encryption schemes.)

We identify the following four kinds of message attacks, which are characterized by how the messages whose signatures the enemy sees are constructed. (Here we let A denote the user whose signature method is being attacked.)

- **Known Message Attack:** The enemy sees signatures for a set of messages $M_1, \ldots, M_k$. The messages are known to the enemy but are not in any way chosen by him.
- **Generic Chosen Message Attack:** Here the enemy is allowed to obtain from A valid signatures for a chosen list of messages $M_1, \ldots, M_k$ before he attempts to break A's signature scheme. These messages are *chosen* by the enemy, but they are *fixed* and *independent* of A's public key (for example the M_i's may be chosen at random). This attack is *nonadaptive*: the entire message list is constructed before any signatures are seen. This attack is "generic" since it does not depend on the A's public key; the same attack is used against everyone.
- **Directed Chosen Message Attack:** This is similar to the generic chosen message attack, except that the list of messages to be signed may depend on A's public key. However, it is still nonadaptive as before. This attack is "directed" against a particular user A.
- **Adaptive Chosen Message Attack:** This is more general yet: here the enemy is also allowed to use A as an "oracle"; not only may he request from A signatures of messages which depend on A's public key but he may also request signatures of messages which depend additionally on previously obtained signatures.

We use the term "non-adaptive message attack" to mean a known, generic chosen, or directed chosen message attack.

II.C. WHAT DOES IT MEAN TO "BREAK" A SIGNATURE SCHEME?

One might say that the enemy has "broken" user A's signature scheme if his attack allows him to do any of the following with a non-negligible probability:

- **A Total Break:** Compute A's secret trap-door information.
- **Universal Forgery:** Find an efficient signing algorithm functionally equivalent to A's signing algorithm (based on possibly different but equivalent trap-door information).
- **Selective Forgery:** Forge a signature for a particular message chosen *a priori* by the enemy.
- **Existential Forgery:** Forge a signature for at least one message. The enemy has no control over the message whose signature he obtains, so it may be random or nonsensical. Consequently this forgery may only be a minor nuisance to A.

We say that a scheme is respectively *totally breakable*, *universally forgeable*, *selectively forgeable*, or *existentially forgeable* if it is breakable in one of the above senses. Note that it is more desirable to prove that a scheme is not even existentially forgeable than to prove that it is not totally breakable. The above list is not exhaustive; there may be other ways of "breaking" a signature scheme which fit in between those listed, or are somehow different in character.

Our notion of *forgery* means that the enemy must produce a signature for a message whose signature he was not given by A during his attack; it is not forgery to obtain from A a valid signature for a message and then claim that he has now "forged" that signature, any more than photocopying a signed document is an instance of forgery.

To say that the scheme is "broken", we insist that it be broken with a non-negligible probability – for at least some positive fraction ϵ of all possible public keys.

We note here that the characteristics of the signature scheme may depend on its message space in subtle ways. For example, a scheme may be existentially forgeable for a message space $\mathcal{M}_1$ but not existentially forgeable if restricted to a message space which is a sparse subset of $\mathcal{M}_1$.

For examples of the notions, see section IV (where we review previously proposed signature schemes).

III. THE PARADOXICAL PROBLEM OF PROVING SIGNATURE SCHEMES SECURE

The paradoxical nature of signature schemes which are provably secure against chosen message attacks made its first appearance in Rabin's paper, "Digitalized Signatures as Intractable as Factorization". The signature scheme he proposed there works as follows. User A publishes a number n which is the product of two large primes. To sign a message M, A computes as M's signature one of M's square roots modulo n. (When M is not a square modulo n, A modifies a few bits of M to find a nearby square.) Here signing is essentially just extracting square roots modulo n. Using the fact that extracting square roots modulo n enables one to factor n, it follows that selective forgery in Rabin's scheme is equivalent to factoring if the enemy is restricted to at most a known message attack.

However, it is true (and was noticed by Rabin) that an enemy might totally break the scheme using a directed chosen message attack. By asking A to sign a value $x^2 \pmod n$ (where x was picked at random), the enemy would

obtain with probability $\frac{1}{2}$ another square root y of x^2 such that $\gcd(x+y, n)$ was a prime factor of n.

Rabin suggested that one could overcome this problem by, for example, having the signer concatenate a fairly long randomly chosen pad U to the message before signing it. In this way the enemy can not force A to extract a square root of any particular number.

However, the reader may now observe that the proof of the equivalence of selective forgery to factoring no longer works for the modified scheme. That is, being able to selectively forge no longer enables the enemy to directly extract square roots and thus to factor. Of course, breaking this equivalence was really the whole point of making the modification.

III.A. THE PARADOX

We now "prove" that it is impossible to have a signature scheme for which it is both true that forgery is provably equivalent to factoring, and yet the scheme is invulnerable to adaptive chosen message attacks. (This is essentially the argument given in [Wi80].) By *forgery* we mean in this section any of universal, selective, or existential forgery – we assume that we are given a proof that forgery of the specified type is equivalent to factoring.

Let us begin by considering this given proof. The main part of the proof presumably goes as follows: given a subroutine for forging signatures, a constructive method is specified for factoring. (The other part of the equivalance, showing that factoring enables forgery, is usually easy, since factoring usually enables the enemy to totally break the scheme.)

But it is trivial then to show that an adaptive chosen message attack enables an enemy to totally break the scheme. The enemy merely executes the constructive method given in the proof. Whenever he needs to execute the forgery subroutine, he merely performs an "adaptive chosen message attack" step – getting the real user to sign a message. In the end the unwary user has enabled the enemy to factor his modulus! (If the proof relates to universal or selective forgery, we have to get real user to sign a particular message. If the proof relates to existential forgery, we can get him to sign anything at all.)

III.B. BREAKING THE PARADOX

How can one hope to get around the apparent contradictory natures of equivalence to factoring and invulnerability to an adaptive chosen message attack?

A major idea in both the construction and the proof is the notion of "random rooting". Each user publishes not only his two composite moduli $n1$ and $n2$, but also a "random root" R_0. This value R_0 is used when validating the user's signatures. The paradox is resolved using this notion as follows:

- It is provably equivalent to factoring for an enemy to have a *uniform* algorithm for forging; uniform in the sense that for each pair of composite numbers $n1$ and $n2$, if the enemy can randomly forge signatures for a significant fraction of the possible random roots R_0, then he can factor either $n1$ or $n2$.
- The above proof *requires* that the enemy be able to pick R_0 himself – the forgery subroutine is fed triples $(n1, n2, R_0)$ where the R_0 part is chosen by the enemy according the procedure specified in the constructive proof. *However*, the user has picked a fixed R_0 at random to put in his public file, so an adaptive chosen message attack will not enable the enemy to "forge" signatures corresponding to any other values of R_0. Thus the constructive method given in the proof can not be applied!

IV. PREVIOUS SIGNATURE SCHEMES

In this section we list a number of previously proposed signature schemes and briefly review some facts about their security.

Trap-Door Signature Schemes [DH76]: Any trap-door signature scheme is existentially forgeable with a direct attack since a valid (message, signature) pair can be created by beginning with a random "signature" and applying the public verification algorithm to obtain the corresponding message. A common heuristic for handling this problem in practice is to require that the message space be sparse (e.g. by having each message contain a reasonably long checksum); in this case the proposed attack is not likely to result in a successful existential forgery.

Rivest-Shamir-Adleman [RSA78]: The RSA scheme is selectively forgeable using a directed chosen message attack, since RSA is *multiplicative*: the signature of a product is the product of the signatures. (This can be handled in practice as above using a sparse message space.)

Merkle-Hellman [MH78]: Shamir showed the basic Merkle-Hellman "knapsack" scheme to be universally forgeable using just a direct attack [Sh82]. (This scheme was perhaps more an encryption scheme than a signature scheme, but had been proposed for use as a signature scheme as well.)

Rabin [Ra79]: As noted earlier, Rabin's signature scheme is totally breakable if the enemy uses a directed chosen message attack. However, for non-sparse message spaces selective forgery is as hard as factoring if the enemy is restricted to a known message attack.

Williams [Wi80]: This scheme is similar to Rabin's. The proof that selective forgery is as hard as factoring is slightly stronger, since here only a single instance of selective forgery guarantees factoring (Rabin needed a probabilistic argument). Williams uses effectively (as we do) the properties of numbers which are the product of a prime $p \equiv 3 \pmod 8$ and a prime $q \equiv 7 \pmod 8$.

Lieberherr [Li81]: This scheme is similar to Rabin's and Williams'.

Shamir [Sh78]: This knapsack-type signature scheme has recently been shown by Tulpan [Tu84] to be universally forgeable with a direct attack for any practical values of the security parameter.

Goldwasser-Micali-Yao [GMY83]: This paper presents two signature schemes, which are not of the trap-door type. These schemes have the interesting property that their

characteristics hold for *any* message space (even a sparse one). The first signature scheme presented in [GMY83] was proven not to be even existentially forgeable against a *generic* chosen message attack unless factoring is easy. However, it is not known to what extent *directed* chosen message attacks or adaptive chosen message attacks might aid an enemy in "breaking" the scheme.

The second scheme presented there (based on the RSA function) was also proven not to be even existentially forgeable against a generic chosen message attack. This scheme may also resist existentially forgery against an adaptive chosen message attack, although this has not been proven. (A proof would probably require showing certain properties about the distribution of prime numbers and making a stronger intractability assumption about inverting RSA.)

By comparison, the scheme presented here is much faster, produces much more compact signatures, and is based on much simpler assumptions (only the difficulty of factoring or more generally the existence of sets of claw-free pairs of functions).

Several of the ideas and techniques presented in [GMY83], such as bit-by-bit authentication, are used in the present paper.

Ong-Schnorr-Shamir [OSS84]: Totally breaking this scheme using an adaptive chosen message attack has been shown to be as hard as factoring. However, Pollard [Po84] has recently been able to show that the "OSS" signature scheme is universally forgeable in practice using just a direct attack; he developed an algorithm to forge a signature for any given message without obtaining the secret trap-door information. A more recent "cubic" version has recently been shown to be universally forgeable in practice using just a direct attack (also by Pollard).

El Gamal[EG84]: This scheme, based on the difficulty of computing discrete logarithms, is existentially forgeable with a generic message attack and selectively forgeable using a directed chosen message attack.

V. DESCRIPTION OF THE SCHEME

A General Scheme: It is convenient to present our scheme in a general manner that is divorced from any particular assumptions, such as that factoring is hard. This clarifies the exposition, and helps to establish the true generality of the proposed scheme.

Definition: We define a *claw-free family* to be a set of pairs of trap-door permutations such that:

- It is easy, given a security parameter k, to select members of the family at random which have the given security parameter together with the trap-door information allowing inversion of the permutations chosen. We note that the family may contain many pairs of permutations associated with a given security parameter, just as there are many composite numbers of a given length.
- For each such pair (f_0, f_1) we have $domain(f_0) = domain(f_1)$.
- Given a pair (f_0, f_1) of permutations from the family it is computationally infeasible (even by a probabilistic algorithm) given just a description of the pair to find any (x, y) with $f_0(x) = f_1(y)$ (a "claw" – specifically, an "f-claw") with a non-negligible probability.

We also call each pair of permutations in the family "claw-free".

Remark: Note that if it is infeasible to find claws, then it is infeasible to invert either permutation, since an inversion algorithm enables one to create claws easily. It is thus a *stronger* requirement that the pair of functions be claw-free than that they merely be one-way in the sense that inversion is infeasible. Note, for example, that the RSA functions $f_0(x) = x^s \pmod n$ and $f_1(x) = x^t \pmod n$ are not easily invertible but are also not claw-free, since their commutativity allows one to create claws easily.

Remark: This is a slight generalization of the notion of a "claw-free" function f (one for which both inversion is hard and finding x, y with $f(x) = f(y)$ is hard). This latter notion has previously been proposed in the literature, and has been proposed as the proper notion of a one-way function. (See [Yu79, Li81], for example.)

Notation: If (f_0, f_1) and (g_0, g_1) are claw-free pairs of functions, we extend the notation f_i and g_i to handle the case $i > 1$ by:

$$f_i(x) = f_{i_d}(f_{i_{d-1}}(f_{i_{d-2}}(\ldots(f_{i_1}(f_{i_0}(x)))\ldots))$$

if $i = i_d i_{d-1} \ldots i_1 i_0$ in binary.

Notation: f_i^{-1} is interpreted as $(f_i)^{-1}$ so that $f_i^{-1}(f_i(x)) = x$.

Prefix-Free Encodings

We will be using the mapping from i to $f_i^{-1}(x)$ as a one-way function, where the pair (f_0, f_1) and the value x were previously known or proven to have been produced by the real signer. Anyone will be able to check this result, since $f_i(f_i^{-1}(x)) = x$.

It is important for this use that the value i be chosen from a set whose elements have a prefix-free binary encoding. (An encoding scheme is prefix-free if no encoding of an element of the set is a prefix of the encoding of any other element of the set.) If a prefix-free encoding scheme were not used, an enemy could "forge" $f_j^{-1}(x)$ from $f_i^{-1}(x)$ if the encoding for j is a prefix of the encoding for i.

We do not care to fix a particular prefix-free encoding for use here, but note that such encodings are simple to devise (e.g. code each 0 as 00, each 1 as 11, and terminate the encoding with 01).

We do, however, introduce the notation $[x]$ to denote the chosen prefix-free encoding of the integer x. Thus, our basic one-way function can be represented as $f_{[i]}(x)$.

Message Space: The new signature scheme can use any countable set as a message space, as long as a prefix-free encoding is used. Like the schemes presented in [GMY83], the properties of the new scheme do not depend on the message space used (even if it is, say, sparse).

An Atomic Authentication Step: Given an "authenticated" quantity Q, we can authenticate *two* new quantities L and R if $f^{-1}_{[R]}(Q) = L$. This is done in a *bit by bit* manner: by examining the bits of $[R]$ one-by-one, we can easily compute L. Only someone who knowns how to invert the f_i's could have produced a valid (L, R) pair from Q. (In [GM82] and [GMY83] very similar ideas appeared.)

Randomization: The signer flips coins; there are many valid signatures for any one message.

Signing by Two-Step Authentication: Signing the i-th message M_i consists of first authenticating a random message R_i, and then authenticating the given message M from the random starting point R_i. (This is reminiscent of the routing scheme for the boolean n-cube proposed by Reif and Valiant [RV83].)

Tree Authentication: We begin with an authenticated root R_0 (authenticated by being in the public file), and from each authenticated point R_i (resp. L_i) we authenticate two new values (L_{2i+1}, R_{2i+1}) (resp. (L_{2i}, R_{2i})). Each R_i is randomly chosen and the L_i values are determined from them. This defines a tree structure on the L_i and R_i values. (This tree can either be grown as new signatures are needed or can have a suitably large size defined initially.) A path from any node to the root is an "authentication chain" which authenticates the node, assuming the root has been authenticated.

Random Rooting: The initial value R_0, which is placed in the public directory, is randomly chosen.

Signatures: The signature for the j-th message M_j consists of

- The message M_j itself.
- A random quantity R_j and an authentication chain for it.
- An atomic authentication for M_j beginning at R_j.

Thus, each message M_i is authenticated by producing a pair (S_i, M_i) authenticated from R_i (which in turn is authenticated in the tree structure defined above).

V.A. HOW TO GENERATE KEYS

Each user publishes his public key, consisting of:

- two claw-free pairs of permutations (f_0, f_1) and (g_0, g_1), and
- a random number R_0 in the range of f_0 and f_1.

V.B. HOW TO SIGN

Implicit: User Alice has an infinite list $R_0, R_1, R_2, \ldots$ of random numbers in the range of f_0, f_1. She will use one such number per signature, begining with R_1. In practice, Alice will create these as needed rather than all at the beginning.

Authenticators: Alice will include R_j as part of her $j-th$ signature,and provide an "authenticator" that it is valid (really created by Alice). Define

$$L_j = \begin{cases} f^{-1}_{[R_j]}(L_{j/2}), & \text{if } j \text{ is even;} \\ f^{-1}_{[R_j]}(R_{j-1/2}), & \text{if } j \text{ is odd.} \end{cases}$$

and

$$A_j = \begin{cases} (1, R_1, L_1), & \text{if } j = 1; \\ (j, R_j, L_j, A_{\lfloor j/2 \rfloor}), & \text{if } j > 1. \end{cases}$$

Here "A_j" is the "authenticator" for R_j; only Alice could have created it but anyone can check it. The authenticators form a "tree-like" structure (see figure 2).

Signature: Alice's signature for the j-th message M_j is $(M_j, A_j, g^{-1}_{[M_j]}(R_j))$.

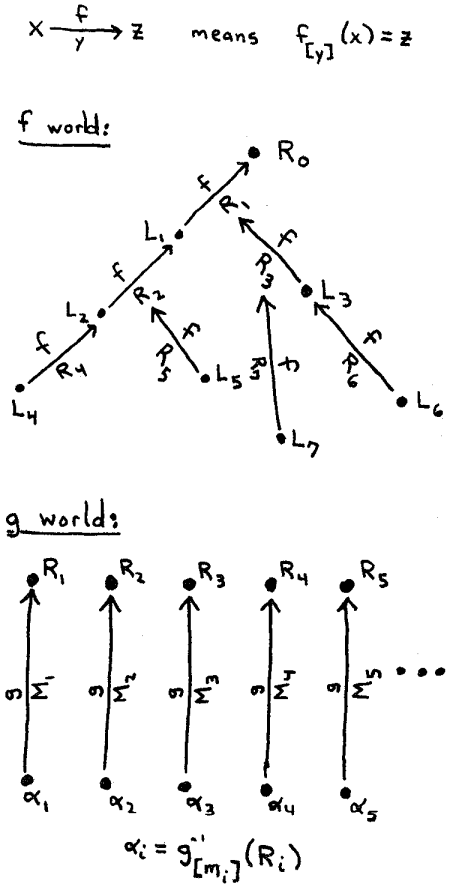

Figure 2.

V.C. How to Verify a Signature

First, authenticate R_j using the published f_i's.

Then, authenticate M_j using the published g_i's.

V.D. Efficiency of the Proposed Signature Scheme

Let us assume that all numbers and messages have length $O(k)$, where k is the "security parameter" for the system. Then the time to compute a signature is $O(k)$ function inversions (i.e. inversions of f_0 or f_1).

The length of the j-th signature is

$$O(\log(j) \cdot k).$$

VI. PROOF OF SECURITY

We recall that a signature scheme is existentially forgeable if the enemy is able to forge any valid messsage/signature pairs at all. We also recall that in an adaptive chosen message attack the enemy can use the real signer as an "oracle" for a while before attempting to forge a new signature.

Theorem. The proposed signature scheme is not existentially forgeable, even if the enemy uses an adaptive chosen message attack.

Proof: Assume that there exists an adaptive chosen message attack which enables the enemy to later forge valid signatures. We prove that this would enable an enemy to create an f-claw or a g-claw, or to invert one of the f_i's or the g_i's.

We assume that the security parameter k is given.

Choose at random a claw-free pair of functions f_0, f_1 with the correct security parameter from the given family of pairs of claw-free functions, so we don't know f_i^{-1} $(i = 0, 1)$. We will show that the existence of the effective attack by the enemy would violate the claw-freeness assumption for the f_i's.

We choose g_i at random with corresponding trapdoor information $(i = 0, 1)$. We can therefore invert each g_i.

We consider two cases and apply the presumed attack to each:

CASE 1: Apply the attack to the (f, g) signature scheme – (i.e. as described above). Note that we can "simulate" the attack (i.e. play the role of the actual signer when asked to sign messages) even though we don't know f_i^{-1}, since we can *a priori* create the necessary tree in the "f-world" using f in the forward direction only (since all nodes in the "f-world" are randomly chosen). So the attack can be executed resulting in the forgery of a new message.

CASE 2: Apply the attack as in case 1, but switching the roles of f and g (but not their names). Here it is easy to simulate the attack by simulating the signing of messages as needed, without using f_i^{-1}. To do this, given a message M_j to sign, we can compute $f_{[M_j]}(S)$ where S is randomly chosen, resulting in a value R_j. We can then "authenticate" R_j in the "g-world" by using g^{-1} as needed.

Lemma: A successful attack will, when it forges its signature, either create an f-claw, a g-claw.

Proof sketch: We can view the authentication structure produced by the legitimate signer during a chosen message attack as a collection of atomic authentication steps, each of which authenticate two values from one previously authenticated value. (Some of these steps are in g-world and some in f-world, but it doesn't matter here.) To forge a new signature means to produce new atomic authentication steps (otherwise nothing new has been signed) which "link in" to values previously authenticated by the real signer. If it "links in" in g-world we get a g-claw and if it "links in" in f-world we get an f-claw. ∎

By assumption about the ways in which the f_i's and the g_i's were chosen, the attack could not tell if it was in case 1 or case 2. Therefore the attack will with probability at least 1/2 (if it succeeds) "break" the given f_i's by creating an f-claw. By assumption, however, (f_0, f_1) was a claw-free pair for which we did not know the trap-door information. This contradiction proves that it is impossible to have a uniform method of forging signatures with an adaptive chosen signature attack. ∎

VI.A. An Implementation of our scheme as intractable as factoring

The assumption of the existence of "claw-free" pairs was made in a general manner, and not based on any particular number theoretic assumptions. Thus, the above proof of security holds even if factoring turns out to be in polynomial time. However for concretely implementing our scheme the following is suggested.

We first make an assumption about the intractability of factoring, and then exhibit a family of claw free pairs whose existence is thereby implied.

Notation: Let $H_k = \{ n = p \cdot q \mid |p| = |q| = k \}$ (the set of composite numbers which are the product of two k-bit primes), and let $H = \bigcup_k H_k$.

Remark: Randomly selected members of H seem to be among the "hardest" inputs for all known factoring algorithms.

The following assumption about the intractability of factoring is made throughout this section.

The Intractability Assumption for Factoring (IAF): Let $0 < \epsilon < 1$, let Q be an arbitrary polynomial, and let $C_{\epsilon,k}$ denote the minimum size of a boolean circuit that can factor at least a fraction ϵ of the numbers in H_k. Then $C_{\epsilon,k} > Q(k)$ for all sufficiently large k.

Consider the subset B of H whose elements are the product of a prime $p \equiv 3 \pmod 8$ and prime $q \equiv 7 \pmod 8$. (These numbers were used in [Wi80, Bl82].) We note that for $n \in B_n$:

-1 has Jacobi symbol $+1$ but is not a quadratic residue (mod n).

2 has Jacobi symbol -1 (and is not a quadratic residue (mod n)).

Let Q_n denote the set of quadratic residues (modulo n). Define f_0^n and f_1^n as permutations of Q_n as follows:

$$f_0^n(x) = x^2 \pmod n$$

$$f_1^n(x) = 4x^2 \pmod n.$$

(It is not too difficult to prove that f_0^n and f_1^n are permutations of Q_n when $n \in B_n$. See [Bl82] for example.)

Claim: Under the IAF, $F = \{(f_0^n, f_1^n) \mid n \in B\}$ is a claw-free family of permutations.

Proof: Every $x \in Q_n$ has exactly one square root $y \in Q_n$, but has four square roots $y, -y, w, -w$ altogether. Roots w and $-w$ have Jacobi symbol -1, while y and $-y$ have Jacobi symbol $+1$.

Let $n \in B$ and $(f_0^n, f_1^n) \in F$. First f_0 and f_1 are permutations. Second they are trapdoor under IAF, by Rabin's proof. Finally, we show that if there exists a fast algorithm that finds x and y in Q_n such that $y^2 \equiv 4x^2 \pmod n$ then factoring is easy. Suppose such an x and y have been found. Then, $x^2 \equiv (2y)^2 \pmod n$. Since $x \in Q_n, y \in Q_n, 2 \notin Q_n$, we have $2y \notin Q_n$ so that $x \not\equiv \pm 2y \pmod n$. Thus $gcd(x \pm 2y, n)$ will produce a nontrivial factor of n. ∎

VII. Conclusions and Open Problems

- Can a signature scheme be developed with the properties of the new scheme proposed here, except that it is "memoryless" in the sense that the signature algorithm does not depend on the number of messages previously signed or how they were signed?
- It is an open question whether the RSA scheme is universally forgeable under an adaptive chosen message attack.
- Can an encryption scheme be developed for which decryption is provably equivalent to factoring yet for which an adaptive chosen ciphertext attack is of no help to the enemy?

VIII. References

[Bl82] Blum, M. "Coin Flipping by Telephone," *Proc. IEEE Spring COMPCOM* (1982), 133-137.

[DH76] Diffie, W. and M. E. Hellman, "New Directions in Cryptography", *IEEE Trans. Info. Theory* **IT-22** (Nov. 1976), 644-654.

[EG84] "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms", To appear in *Proceedings of Crypto 84.* (by El Gamal, Taher).

[EGL82] Even, S., O. Goldreich, and A. Lempel, "A Randomized Protocol for Signing Contracts", *Advances in Cryptology - Proceedings of Crypto 82,* (Plenum Press, New York, 1983), 205-210.

[GM82] Goldwasser, S., and S. Micali, "Probabilistic Encryption," *JCSS* **28** (April 1984), 270-299.

[GMY83] Goldwasser, S., S. Micali, and A. Yao, "Strong Signature Schemes," *Proc. 15th Annual ACM Symposium on Theory of Computing,* (Boston Massachusetts, April 1983), 431-439.

[La79] Lamport, Leslie. "Constructing Digital Signatures from a One-Way Function," SRI Intl. CSL-98. (Oct. 1979)

[Li81] Lieberherr, K. "Uniform Complexity and Digital Signatures," *Theoretical Computer Science* **16**,1 (Oct. 1981), 99-110.

[LM78] Lipton, S., and S. Matyas, "Making the Digital Signature Legal - and Safeguarded," *Data Communications* (Feb. 1978), 41-52.

[Ma79] Matyas, S. "Digital Signatures - An Overview," *Computer Networks* **3** (April 1979) 87-94.

[MH78] Merkle, R., and M. Hellman, "Hiding Information and Signatures in Trap-Door Knapsacks," *IEEE Trans. Infor. Theory* **IT-24** (Sept. 1978), 525-530.

[OSS84] Ong, H., C. Schnorr, and A. Shamir, "An Efficient Signature Scheme Based on Quadratic Equations," *Proc. 16th Annual ACM Symposium on Theory of Computing,* (Washington, D.C., April 1984), 208-217.

[Po84] Pollard, J. "How to Break The 'OSS' Signature Scheme", Private Communication (1984).

[Ra78] Rabin, Michael, "Digitalized Signatures," In FOUNDATIONS OF SECURE COMPUTATION, (Edited by R. A. DeMillo, D. Dobkin, A. Jones, and R. Lipton), (Academic Press, New York, 1978), 133-153.

[Ra79] Rabin, Michael. "Digitalized Signatures as Intractable as Factorization," MIT Laboratory for Computer Science Technical Report MIT/LCS/TR-212 (Jan. 1979).

[RV83] Reif, J. and L. Valiant, "A logarithmic time sort for linear size networks," *Proceedings 15th Annual ACM Symposium on Theory of Computing,* (Boston Massachusetts, April 1983), 10-16.

[RSA78] Rivest, R., A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Comm. of the ACM* (Feb. 1978), 120-126.

[Sh78] Shamir, A., "A Fast Signature Scheme," MIT Laboratory for Computer Science Technical Memo MIT/LCS/TM-107 (July 1978).

[Sh82] Shamir, A., "A Polynomial Time Algorithm for Breaking the Basic Merkle-Hellman Cryptosystem," *Proc. 23rd Annual IEEE FOCS Conference* (Nov. 1982), 145-152.

[Tu84] Tulpan, Y., "Fast Cryptanalysis of a Fast Signature System," Master's Thesis in Applied Mathematics, Weizmann Institute. (1984)

[Wi80] Williams, H. C., "A Modification of the RSA Public-Key Cryptosystem," *IEEE Trans. Info. Theory* **IT-26** (Nov. 1980), 726-729.

[Yu79] Yuval, G., "How to Swindle Rabin," *Cryptologia* **3** (July 1979), 187-189.