Uniform Complexity and Digital Signatures[†]

Karl Lieberherr

Princeton University
Department of Electrical Engineering and Computer Science
Princeton NJ 08544

ABSTRACT

A concept of uniform complexity is defined and a class of functions is shown to have uniform complexity. A special case of these results is used to develop a new digital signature method, which makes forging signatures as hard as factoring a large number and which allows to sign all messages directly. The signature production involves only one exponentiation modulo a large number and the signature checking the comparison of a fourth and a second power modulo a large number. Therefore this new method is faster than known methods with the same degree of safety.

1. Introduction

Signatures should give the parties involved three kinds of protection. (Let A be the originator and B the receiver of a message.)

(i) (Authenticity)
Both party A and party B should be protected against forged messages, planted into the communication channel by party C which pretends to be party A.

(ii) (Unforgeability)
Party A should be protected against forged messages by party B, which B claims to have received (properly signed) from party A.

(iii) (No repudiation)
It must not be possible for party A to subsequently disclaim authorship of a signed message sent to B.

With digital signatures, the first kind of protection can be guaranteed by using appropriate coding techniques which are only known to A and B. The second kind of protection is harder to obtain, since B should know enough about the way A

[†] This research partially supported by National Science Foundation grant MCS80-04490 and ENG76-16808.

signs its messages in order to recognize them, and yet should be unable to generate them.

The third kind of protection is very hard to achieve with digital signatures since the validity of a digital signature on a message is only as safe as the entire future protection of the private key used to produce the signature. Unfortunately the author of a digital signature can effectively disavow and repudiate his signatures at any time, merely by causing his secret key to be made public or "compromised". When such an event occurs, either by accident or intention, all messages previously "signed" using the given private key are invalidated, since the only proof of validity has been destroyed (Saltzer(1978)). This paper does not provide a solution for this very important problem. It is assumed in the remainder that secret keys are not lost.

Proposals for digital signatures are given in Rabin (1976, see Rabin (1978)), Merkle et al. (1978), Rivest et al. (1978), Shamir (1978), Shamir (1979), Rabin (1979). The system proposed in Rabin (1978) is too complicated and it can be simplified considerably while making it safer at the same time. The simplified system is a version of a public-key system with multiple keys which are used only once. The key idea of the simplification is that if a party A has to reveal, say 20, keys to a party B for checking of 20 encodings, this does not imply that party B is able to produce efficiently 20 correct encodings for another message. This effect can easily be achieved with trapdoor one-way functions.

If the network of communicating parties is sufficiently big it is impractical to use a distinct and secret signature algorithm for every pair of potential users. In their innovative paper (Diffie *et al.* (1976)), Diffie and Hellman introduce the notion of a "public key cryptosystem," in which (among other things) each user makes public a quick method for recognizing his signatures. The resultant "signature directory" is available to anyone, and thus two participating parties can start sending signed messages without any special preparation.

As explained in (Rabin (1977)), the viability of a digital signature system requires that the relevant system-breaking computations be intractable in a sense stronger than the one usually defined in complexity theory. Even if the problem were proved to be exponentially complex (no such result was proved to date), this would only be a worst case or average case complexity result. It does not preclude the possibility that for one key in a thousand the system cannot be broken by an algorithm not known to the user.

Ideally, we want the signature forging function to be intractable almost *without exception*. Thus we want to capture the idea that signature forging should be uniformly intractable. Since complexity theory is far from an intractability proof of a problem in **NP**, it is worth-while to consider signature methods which can be proven to have uniform complexity for forging. Intuitively, a problem has uniform complexity, if even the solution of a few (random) instances would imply the solution of a problem which is expected (or proven) to be difficult. From another point of view, a problem for which no polynomial algorithm is known, has uniform complexity, if the solution of a few (random) instances is likely to imply that all instances can be solved in polynomial time.

In order to define uniformity we need to introduce the class of functions Δ_R (a generalization of a definition given in Adleman *et al.* (1977), see also Gill (1974)). Babai (1979) defines Δ_R to be the class of functions which have an efficient Las Vegas algorithm. A Las Vegas algorithm is a randomized algorithm which always, if it arrives at an answer, computes a correct output (with known success probability), but sometimes it gives no answer (with known failure probability).

A randomized algorithm is an algorithm which can use coin-tossing in some of its decisions. Let $\mathbf{N}$ denote the natural numbers and let $F_1:\mathbf{N}\rightarrow\mathbf{N}$ be a function for which $F_1(x)=y$ can be checked in polynomial time. F_1 is said to be in Δ_R if there is a polynomial randomized algorithm A_1 which computes F_1 with success probability $\gamma>0$. By iterating A_1 we can make its success probability arbitrary close to 1.

E.g. a function $F_1:D_1\rightarrow R_1$ is in Δ_R if there is a polynomial p and a polynomial time computable function

$$F_2 : (D_1,D_2) \rightarrow R_1\cup\{?\}$$

(? for no answer) such that

(1)

$$\forall x\in D_1: \frac{|\{w\in D_2 \mid |w|\leqslant p(|x|) \text{ and } F_2(x,w)=F_1(x)\}|}{|\{w\in D_2 \mid |w|\leqslant p(|x|)\}|}\geqslant\gamma$$

for some constant $\gamma>0$ and

(2) there is a polynomial algorithm to check, given x,y whether $F_1(x)=y$.

With $|x|$ we denote the length of the string x over a binary alphabet. It is assumed that $F_1(x)=O(|x|^{k_0})$ for some constant k_0.

Example:

To compute square roots modulo a prime is in Δ_R Lehmer (1969), Berlekamp (1970), Shanks (1972), Rabin (1980).

For a Las Vegas algorithm A_1 which computes a function $F_1\in\Delta_R$ the expected number of iterations until $F_1(x)$ is found is $<\frac{1}{\gamma}$ since

$$\sum_{i=0}^{\infty} i\cdot\gamma\cdot(1-\gamma)^{i-1}=\frac{1}{\gamma}.$$

The probability that $F_1(x)$ is not found after $\frac{c}{\gamma}$ iterations of A_1 is $<e^{-c}$.

Let

$$X: (\mathbf{N},\Omega)\rightarrow\mathbf{N}, \quad Y: \mathbf{N}\rightarrow\mathbf{N}$$

be two functions computable in exponential time. X and Y are allowed to have several function values for the same argument, i.e. they are relations. $X(j,n)$ is defined for $n\in\Omega\sqsubseteq\mathbf{N}$ (equal the domain of the second argument of X) and all j $(0<j<n)$. Assume that $|\{X(j,n)\}|=O(|n|)$ and $|\{Y(n)\}|=O(|n|)$.

X is said to be *Y-uniform,* if for all $n\in\Omega$ and for all j $(0<j<n)$,

(1) if all values of $X(j,n)$ are given by an oracle then $Y(n)$ can be computed in polynomial time in $|n|$ and

(2) if one value of $Y(n)$ is given by an oracle then $X(j,n)$ can be computed by a polynomial randomized algorithm with success probability $\geqslant½$, i.e. $X\in\Delta_R$.

X is said to be *randomly Y-uniform,* if

(1) if one value of $X(j,n)$ is given by an oracle then $Y(n)$ can be computed by a polynomial randomized algorithm with success probability $\geqslant\gamma$, i.e. $Y\in\Delta_R$. (γ a constant depending on X, $(0<\gamma\leqslant1)$).

(2) as (2) above.

Immediate consequences of these definitions are

a) If X and Y are polynomial time computable then X is Y-uniform.

b) If X is randomly uniform (for some Y which is not important here) and if $\pi(n)$ is an algorithm which computes $X(j,n)$ in time $T(n)$ for the fraction $\epsilon>0$ of the j's in the interval $0<j<n$ then there is a

$$\frac{1}{\epsilon\cdot\gamma}\cdot T(n)+q(|n|)$$

randomized algorithm to compute $X(j,n)$ for any j in the interval $0<j<n$. q is a polynomial guaranteed by condition (2) of the definition of random uniformity. (Rabin (1979), Shamir (1979)).

Rabin (1979) presents a problem which he shows to be randomly factoring-uniform, namely essentially square root extraction modulo a composite number. The resulting signature method has the property that about ¼ of all messages can be signed directly (without randomization). In this paper a new signature system is proposed which has several advantages

a) with respect to Rabin's system (Rabin (1979)):

1) *All* messages can be signed directly without randomization. (This saves about a factor of four for signature production.)

2) It is faster. To produce a signature only requires one exponentiation modulo a large number as in the RSA public key cryptosystem (Rivest et al.(1978)). Signature checking is slightly slower.

b) with respect to other systems (for an overview see Lempel (1979), Popek et al. (1979), Simmons (1979)):
The signature forging function is proven to be randomly factoring uniform, which means that signature forging is as difficult as factoring a large number.

The following two technical lemmata are used later.

Lemma A

Let s_1, s_2, $(s_1 > s_2 > 0)$ be positive integers, and let $p_1, p_2, \ldots, p_m$ be m distinct primes, so that

$$\begin{aligned} p_1-1 &\mid s_1-s_2 \\ p_2-1 &\mid s_1-s_2 \\ &\vdots \\ p_m-1 &\mid s_1-s_2 . \end{aligned}$$

Then

$$\forall\, j \geq 0 : j^{s_1} \equiv j^{s_2} \ (mod\ p_1\, p_2 \cdots p_m).$$

Proof:
Let $n = p_1 p_2 \cdots p_m$.
Case 1: $gcd(j,n) = 1$.
By Fermat's theorem

$$\begin{aligned} j^{s_1} &\equiv j^{s_2+k_1(p_1-1)} \equiv j^{s_2} \ (mod\ p_1) \\ j^{s_1} &\equiv j^{s_2+k_2(p_2-1)} \equiv j^{s_2} \ (mod\ p_2) \\ &\vdots \\ j^{s_1} &\equiv j^{s_2+k_m(p_m-1)} \equiv j^{s_2} \ (mod\ p_m) \end{aligned}$$

for certain $k_1, k_2, \ldots, k_m$.
Case 2: $gcd(j,n) > 1$.

a) $gcd(j,n) = n \rightarrow j^{s_1} \equiv j^{s_2} \equiv 0 \ (mod\ n)$.

b) $gcd(j,n) = p_1 \rightarrow j \equiv 0 \ (mod\ p_1) \ \rightarrow j^{s_1} \equiv j^{s_2} \equiv 0 \ (mod\ p_1)$.
For the other primes we have by case 1:

$$j^{s_1} \equiv s^{s_2} \ (mod\ p_i) \ (2 \leqslant i \leqslant m).$$

The result follows by the Chinese remainder theorem.
□

The following lemma will be important for the digital signature method described later.

Lemma B

Let n be the product of 2 distinct primes of the form $p_1=4k_1-1$, $p_2=4k_2-1$ and let

$$k=\frac{n-1}{4}-2k_1k_2+1 \ (=2k_1k_2-k_1-k_2+1).$$

Then

a) $\forall j \geqslant 0$: $j^{4k} \equiv j^2$ $(mod\ n)$

b) $\forall c$ (c a quadratic residue): $c^k \equiv \sqrt{c}$ $(mod\ n)$.

Proof:

a) is a special case of lemma A.

b) c is a quadratic residue mod n, iff $\left(\frac{c}{p_1}\right)=\left(\frac{c}{p_2}\right)=1$, where $\left(\frac{c}{p}\right)$ is the Legendre symbol. $\left(\frac{c}{p_1}\right)=1$, iff $c^{(p_1-1)/2} \equiv 1$ $(mod\ p_1)$. This implies that $c^{2k-1} \equiv 1$ $(mod\ p_1)$, hence $c^{2k} \equiv c$ $(mod\ p_1)$.

A similar argument holds for p_2 and therefore (by the Chinese remainder theorem) for all quadratic residues c modulo n:

$$c^{2k} \equiv c \ (mod\ n),$$

which implies

$$c^k \equiv \sqrt{c} \ (mod\ n).$$

□

2. Uniformity Results

In this section four uniformity results are given which are relevant to the unforgeability of digital signatures. Proposition 1 is a generalization of an observation made by Legendre (Legendre (1798), or see Knuth (1969), page 351) and Corallary 2 generalizes a result by Rabin (Rabin (1979)). Theorem 3 and Corallary 4 are the main results of this paper.

First six functions are defined which will be employed in the uniformity results. The arguments of a function are denoted as "input" and the function value as "output."

$FACTORING_d$, $(d \geqslant 2)$

input: $n=p_1^{v_1}p_2^{v_2} \cdots p_r^{v_r}$, $r \geqslant 2$, the primes $p_1, p_2, \ldots, p_r$ all distinct, odd and of the form $dk+1$.

output: f, a nontrivial factor of n.

$FACTORING_{R_d}$, $(d \geqslant 2)$

input: $n=p_1^{v_1}p_2^{v_2} \cdots p_r^{v_r}$, $r \geqslant 2$, the primes $p_1, p_2, \ldots, p_r$ all distinct, odd and having the property that $d|(p-1)$ and $gcd((p-1)/d,d)=1$. (E.g for $d=2$ the primes $p_1, p_2, \ldots, p_r$ must be of the form $4k-1$; for $d=3$ of the form $18k+13$ or $18k+7$.)

output: f, a nontrivial factor of n.

$ROOT^d{}_{ALL}$, $(d \geqslant 2)$

input: $n=p_1^{v_1}p_2^{v_2} \cdots p_r^{v_r}$, $r \geqslant 2$, the primes $p_1^{v_1}p_2^{v_2} \cdots p_r^{v_r}$ all distinct, odd and of the form $dk+1$.

c, $(0<c<n)$ a d^{th} power residue $mod\ n$.

output: All d^{th} roots of $c\ mod\ n$.

Example:

$d=2$; the 4 square roots of 4 (mod 21) are ± 2, ± 5.

$ROOT^d{}_1$, $(d \geqslant 2)$

As $ROOT^d{}_{ALL}$, but only one d^{th} root is required as output.

$S^{d^2}J^d{}_{ALL}$, $(d \geqslant 2)$

input: $n=p_1^{v_1}p_2^{v_2} \cdots p_r^{v_r}, r \geqslant 2$, the primes $p_1^{v_1}p_2^{v_2} \cdots p_r^{v_r}$ all distinct, odd and having the property that $d|(p-1)$ and $gcd((p-1)/d,d)=1$.

j, $(0<j<n)$.

output: All solutions for s of $s^{d^2} \equiv j^d\ (mod\ n)$.

Example:

$d=2$; the 4 solutions of $s^4 \equiv j^2\ (mod\ 21)$ for $j=2$ are $s=\pm 4$, $s=\pm 10$.

$S^{d^2}J^d{}_1$, $(d \geqslant 2)$ As $S^{d^2}J^d{}_{ALL}$, but only one solution is required as output.

Proposition 1

If $ROOT^d_{ALL}(n,c)$ is known then $FACTORING_d(n)$ can be computed efficiently.

Proof:

If $gcd(c,n)\neq 1$ $(gcd(d,n)\neq 1)$ then $gcd(c,n)$ $(gcd(d,n))$ is a nontrivial factor of n. If gcd$(c,n)=1$, then if n contains r distinct odd primes then there are d^r distinct d^{th} roots of c modulo n. Namely, $j^d\equiv c\ (mod\ p_i^{v_i}),(i=1,2,\ldots,r)$ has d distinct solutions since $gcd(d,p_i-1)=d$. Each r-tuple of solutions for the r distinct prime powers gives a distinct solution modulo n and therefore there are d^r distinct solutions modulo n. (Note that if p is an odd prime, p does not divide c and p does not divide d, then if $j^d\equiv c\ (mod\ p)$ is solvable, so is $j^d\equiv c\ (mod\ p^v)$ for all $v\geqslant 1$ and there is the same number of solutions (see e.g. Ireland et al. (1972), section 4.2.3)).

Now let $t\equiv c^{1/d}\ (mod\ n)$ and consider

$$j^d-t^d\equiv(j-t)(j^{d-1}+j^{d-2}t+\cdots+jt^{d-2}+t^{d-1})\ (mod\ n).$$

The number of distinct solutions of

$$(j-t)(j^{d-1}+j^{d-2}t+\cdots+jt^{d-2}+t^{d-1})\equiv 0\ (mod\ n)$$

for j (t fixed) is at most $(d-1)^r$, since there are at most $d-1$ solutions modulo $p_i^{v_i}$, $(i=1,2,\ldots,r)$. (Namely, if $f(x)$ is a polynomial of degree d and the coëfficient of x^d is 1 then the congruence $f(x)\equiv 0\ (mod\ p)$ has no more than d distinct solutions modulo p, where p is a prime.) $j-t\equiv 0\ (mod\ n)$ has one solution and so at least $\omega(d,r)=d^r-((d-1)^r+1)$ solutions j yield zero divisors modulo n, i.e. for at least $\omega(d,r)$ solutions j:

$$(j-t)(j^{d-1}+j^{d-2}t+\cdots+jt^{d-2}+t^{d-1})\equiv 0\ (mod\ n)$$

without the factors being zero. Hence $gcd(j-t,n)$ yields a nontrivial factor of n for such a j.

□

Corollary 2

If there is an oracle which computes $ROOT^d_1(n,c)$ for a random c then $FACTORING_d(n)$ can be computed efficiently with probability $\geqslant\gamma(d)>0$.

Proof:

Let $c=j^d\ mod\ n$ be a d^{th} power residue, j $(0<j<n)$ randomly chosen. Let t be the d^{th} root returned by $ROOT^d_1$ on input c. A simple counting argument shows that at least with probability

$$\alpha(d,r)=\frac{d^r-((d-1)^r+1)}{d^r}$$

the number $gcd(t-j,n)$ is a nontrivial factor of n.

□

Theorem 3

$S^{d^2}J^d{}_{ALL}$ is $FACTORING_{R_d}$-uniform for $r=2$.

Proof:

First it is shown that property (1) of the definition of uniformity holds for arbitrary r. W.l.o.g it is assumed that $gcd(d,n)\neq 1$ and $gcd(j,n)\neq 1$. First it is shown that, given a j $(0<j<n)$, there is an s, so that

$$s^{d^2}\equiv j^d \ (mod \ n).$$

This congruence holds, iff it holds modulo the prime powers $p_i^{v_i}$,$(i=1,2, \ldots ,r)$.

If p is a prime, then c is a d^2 power residue $(mod \ p)$, iff

$$c^{\frac{p-1}{g}}\equiv 1 \ (mod \ p),$$

where $g=gcd(d^2,p-1)$. c is a d^{th} power residue $(mod \ p)$, iff

$$c^{\frac{p-1}{h}}\equiv 1 \ (mod \ p),$$

where $h=gcd(d,p-1)$ (see e.g. Ireland et al. (1972)). Hence any d^{th} power residue is a d^2 power residue, iff $g=h$, which holds if $p-1=dk_1$, where k_1 satisfies $gcd(d,k_1)=1$. The primes prescribed in the definition of $FACTORING_{R_d}$ have this property.

Let $s_1, s_2, \ldots, s_{d^r}$ be the d^r solutions of $s^{d^2}\equiv j^d \ (mod \ n)$, where r is the number of distinct primes in n. (To determine the number of distinct solutions, use the fact that if $s^d\equiv c \ (mod \ p)$ is solvable there are exactly $gcd(d,p-1)$ solutions, if p is a prime (see e.g. Ireland et al. (1972), page 47)).

First it is shown that the d^{th} powers of these solutions are congruent modulo n, i.e.

$$s_1^d\equiv s_2^d\equiv \cdots \equiv s_{d^r}{}^d \ (mod \ n).$$

Let $c=j^d \ mod \ n$. Then $c\equiv s^{d^2} \ (mod \ n)$ and $c\equiv s^d \ (mod \ n)$ have both d^r distinct solutions. The claim follows by a simple counting argument.

Now, since all d^{th} roots of s^d are known, Proposition 1 shows that a factor can be determined efficiently.

It remains to be shown that property (2) of the definition of uniformity holds. Berlekamp (1970) gives an efficient randomized algorithm to compute d^{th} roots modulo primes or prime powers (see also Rabin (1980)). For square roots Lehmer

(1969) and Shanks (1972) give other efficient randomized algorithms. Square roots modulo primes of the form $4k-1$ can be computed by one exponentiation (this fact is exploited in Lemma B). Therefore, by Chinese remaindering, the functions which compute d^{th} roots modulo composite numbers (for which the factorization is given) are in Δ_R.

□

Corollary 4

$S^{d^2}J^d{}_1$ is randomly $FACTORING_{R_d}$-uniform for $r=2$.

Proof:

Property (1) of the definition of uniformity holds for arbitrary r. The proof is similar to the proof of Corollary 2. Let s·be a solution and let

$$t=s^d \equiv (j^d)^{1/d} \ (mod \ n)$$

be an easily computed d^{th} root of j^d. Since the prime factors of n are of the form $dk+1$, again a counting argument shows that the probability is at least $\alpha(d,r)$ that $gcd(t-j,n)$ is a nontrivial factor of n for a random j. (E.g. $\alpha(2,2)=1/2$, $\alpha(3,2)=4/9$, $\alpha(2,3)=3/4$, $\alpha(3,3)=2/3$.)

Property (2), for which the assumption $r=2$ is necessary, is shown in the same way as in the proof of theorem 3.

□

3. A new signature method

A new signature method (suitable for a public key crypto system) is given in which all messages can be signed directly (without randomization) and for which the signature forging function is randomly factoring uniform. This implies that even if only a few signatures (say 3) could be forged, then it would be that a large number could be efficiently factored. Since factoring is apparently a difficult task for certain carefully chosen composite numbers (Rivest *et al.* (1978), Williams *et al.* (1979)) and since signature producing and checking can be done efficiently, it is believed that the proposed method and its variants are of practical value.

Williams (1979b) (also influenced by Rabin (1979)) has given a digital signature method which has similar properties as the one presented here. But it is considerably more complicated and less convenient to use.

The proposed signature method allows four distinct legal signatures for the same message. Furthermore the signature method allows an opponent to produce arbitrary many message/signature pairs, but the messages he can sign are very likely messages he is not interested in. Two messages are said to be *equivalent,* if they express the same meaning.

The messages which are signed must have a maximal length of say c=256 bits. Usually messages are longer and are compressed by a hashing function h known to parties A and B (Rabin (1978, 1979)). h must be a "good" hashing function as expressed in the following engineering assumption:

Assumption 1:
During the lifetime of the hashing function it is *never* observed that two messages are mapped on the same compressed message.

Signature algorithm $S^4J^4_{n,h}(M)$

input: Message M.
n and 2 distinct primes p_1, p_2, where $p_i = 4k_i-1$ (i=1,2), such that $p_1p_2=n$.
Hashing function h.

output: Signature s for M.

secret: The factorization of n.

public: (or at least known to other party)
n,h and an efficient algorithm for signature checking.

1. Compress message M to a smaller message j, so that $|j| \leqslant c$ by using the hashing function h.
2. Compute s, so that $s^4 \equiv j^2 \ (mod\ n)$. Lemma B shows that the s computed by

$$s := j^k \ mod\ n,$$

where $k = 2k_1k_2-k_1-k_2+1$ has the desired property.

If $gcd(j,n)>1$ (which is very unlikely), then message M should be slightly changed before it is signed.

Complexity of signature production

Signature production involves applying the hashing function h, plus one k^{th} power exponentiation modulo n, $k<n$. (For low level complexity see Knuth (1969).)

Complexity of signature checking

s is the signature of message M, iff

$$s^4 \equiv h(M)^2 \ (mod\ n).$$

Therefore signature checking costs:
1 application of the hashing function h
1 4^{th} power modulo n
1 2^{nd} power modulo n

1 comparison of numbers $\leqslant n$.

Complexity of signature forging

Before we can apply the uniformity result (Corollary 4), we have to make the following assumption:

> Assumption 2:
> Let M be a message. To find a signature for a message which is equivalent to M is as hard as finding a signature for a random message of length c.

Theorem 5

Signature forging when using method S^4J^2 is randomly $FACTORING_{R_2}$-uniform.

Proof:

Let M be a message which an opponent would like to sign. The opponent has two possibilities:

a) He can exploit the existence of the hashing function and use a signature which he knows or which he can easily generate.
This attack is ruled out by assumption 1.

b) Without using the existence of the hashing function.
By assumption 2 this is as hard as finding a signature for a random message of length c and therefore (by Corollary 4, with $d = 2$) signature forging for this method is randomly $FACTORING_{R_2}$-uniform.

□

Security requirement

> A signature for a message M is never sent to a person who already has another signature for M.

In order to satisfy this, it is necessary that e.g. a message which was prepared by somebody else who might already have a signature for that message is slightly changed in an unpredictable way (without changing the meaning) before it is signed. It has to be avoided that the signature algorithm can be misused as a square root extraction algorithm. (Recall that s^2 is a square root of j^2 $(mod\ n)$.)

Acknowledgment

I would like to thank my wife Dr. R. Lieberherr for improving the style and organization of the paper and my secretary S. Mairs for preparing it on the computer.

Correction: Jim Finn pointed out that the signature method is only safe to use for compressed messages j for which the Jaçobi symbol (j/n)=1 . Hence only one half of the compressed messages (not all) can be signed directly.

References

Adleman1977a. L. Adleman and K. Manders, "Reducibility, Randomness and Intractability," *Proc. 9th Annual ACM Symposium on Theory of Computing*, pp.151-163 (1977).

Babai1979a. L. Babai, "Monte Carlo algorithms in graph isomorphism testing," *Submitted to SIAM J. on Computing* (1979).

Berlekamp1970a. E.R. Berlekamp, "Factoring Polynomials Over Large Finite Fields," *Mathematics of Computation* **24**(111), pp.713-735 (July 1970).

Diffie1976a. W. Diffie and M. Hellman, "New Directions in Cryptography," *IEEE Transactions on Information Theory* **IT-22**(6), pp.644-654 (Nov. 1976).

Gill1974a. J. Gill, "Computational Complexity of Probabilistic Turing Machines," *Proc. 6th Annual ACM Symposium on Theory of Computing*, pp.91-95 (1974).

Ireland1972a. K. Ireland and M. Rosen, *Elements of Number Theory*, Bogden and Quigley, New York (1972).

Knuth1969a. D.E. Knuth, *The Art of Computer Programming*, Addison-Wesley, Reading, Mass. (1969). Vol. 2

Legendre1798a. A.M. Legendre, *Theorie des Nombres*, Issuer unknown, Paris (1798).

Lehmer1969a. D. Lehmer, "Computer Technology Applied to the Theory of Numbers," pp. 117-151 in *Studies in Number Theory*, Mathematics Association of America (1969).

Lempel1979a. A. Lempel, "Cryptology in Transition," *ACM Computing Surveys* **11**(4), pp.285-304 (Dec. 1979).

Merkle1978a. R. Merkle and M.E. Hellman, "Hiding Information and Receipts in Trapdoor Knapsacks," *IEEE Trans. Inform. Theory IT-24* (Sept. 1978).

Popek1979a. G.J. Popek and C.S. Kline, "Encryption and Secure Computer Networks," *ACM computing surveys* **11**(4), pp.331-356 (Dec. 1979).

Rabin1977a. M.O. Rabin, "Complexity of Computations," *Communications of the ACM* **20**, pp.625-633 (1977).

Rabin1978a. M.O. Rabin, "Digitalized Signatures," in *Foundations of Secure Computation*, ed. Demillo et al., Academic Press (1978).

Rabin1979a. M. O. Rabin, "Digitalized Signatures and Public-Key Functions as Intractable as Factorization," *MIT/LCS/TR-212*, Massachusetts Institute of Technology, Laboratory for Computer Science. (Jan. 1979).

Rabin1980a. M.O. Rabin, "Probabilistic Algorithms in Finite Fields," *Siam J. Comput.* **9**(2), pp.273-280 (May 1980).

Rivest1978a. R.L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Comm. ACM* **21**(2), pp.120-126 (Feb. 1978).

Saltzer1978a. J. Saltzer, "On digital signatures," *ACM Operating Syst. Rev.* **12**(2), pp.12-14 (Apr. 1978).

Shamir1978a. A. Shamir, "A Fast Signature Scheme," *mIT/LCS/TM-107* (July 1978).

Shamir1979a. A. Shamir, "On the Cryptocomplexity of Knapsack Systems," *Proc. 11th Annual ACM Symposium on Theory of Computing*, pp.118-129 (1979).

Shanks1972a. D. Shanks, "Five Number Theoretic Algorithms," *Second Manitoba Conference on Numerical Mathematics*, pp.51-70 (1972).

Simmons1979a. G.J. Simmons, "Symmetric and Asymmetric Encryption," *ACM Computing Surveys*(4), pp.305-330 (Dec. 1979).

Williams1979a. H.C. Williams and B. Schmid, "Some Remarks Concerning The MIT Public-Key Cryptosystem," *BIT* **19**, pp.528-538 (1979).

Williams1979b. H.C. Williams, "A modification of the RSA public-key encryption procedure," *Report 92, Department of Computer Science, University of Manitoba, Winnipeg, Canada* (1979).