Which new RSA signatures can be computed from some given RSA signatures?
(extended abstract)

Jan-Hendrik Evertse [‡]

Department of Mathematics and Computer Science, University of Leiden
P.O. Box 9512, 2300 RA Leiden, The Netherlands

Eugène van Heyst

CWI Centre for Mathematics and Computer Science
Kruislaan 413, 1098 SJ Amsterdam, The Netherlands

Abstract. We consider protocols in which a signature authority issues RSA-signatures to an individual. These signatures are in general products of rational powers of residue classes modulo the composite number of the underlying RSA-system. These residue classes are chosen at random by the signature authority. Assuming that it is infeasible for the individual to compute RSA-roots on randomly chosen residue classes by himself, we give, as a consequence of our main theorem, necessary and sufficient conditions describing whether it is feasible for the individual to compute RSA-signatures of a prescribed type from signatures of other types that he received before from the authority.

Key words. RSA scheme, RSA signature, cryptographic protocol.

1. Introduction

A cryptographic *protocol* can be taken to be a set of rules according to which messages are transmitted between parties. Generally the parties apply cryptographic operations (such as computation of digital signatures and encryption) to the messages sent and received, in order to protect their interests.

In this paper we consider *signature protocols* in which only one party, called the *signature authority*, can create signatures. The signature authority issues these signatures to an other party, called the *individual*. Such protocols are used, for instance, in credential systems (e.g. [CE86]) and payment systems (e.g. [CBHMS89]), in which a signature represents a credential or money.

[‡] This research has been made possible by a fellowship of the Royal Netherlands Academy of Arts and Sciences (K.N.A.W.)

I.B. Damgard (Ed.): Advances in Cryptology - EUROCRYPT '90, LNCS 473, pp. 83-97, 1991.

Figure 1 shows a simple version based on the RSA-system with modulus N. Let e_1,e_2 be public exponents, known to both the signature authority Z and the individual A, and $1/e_2$ the secret exponent, known only to Z. Here $1/e_2$ is some integer such that $(x^{1/e_2})^{e_2} \equiv x (\mathrm{mod}\ N)$, for all x coprime to N. (Note. that this implies that only Z knows the factorization of the RSA modulus).

Individual A		**Signature authority Z**
chooses x (mod N) randomly, computes $S \equiv x^{e_1/e_2}$ (mod N)	$\xrightarrow{x,S}$	verifies if $S^{e_2} \equiv x^{e_1}$ (mod N)

Fig. 1. A signature issuing protocol in which the individual has no influence on the choice of the integer.

The protocols we shall consider, are variations on or generalizations of the scheme in Figure 1. It will appear to be useful to consider variations in which Z does not send x to A, but only the signature (so then A can not verify the signature). In our most general protocols, the *RSA-signatures* are products of rational powers of residue classes modulo N, for instance $x_1^{2/5} \cdot x_2^{3/7}$ (mod N). It is reasonable to assume that an individual, not knowing the factorization of N, can not compute RSA-roots $x^{1/d}$ (mod N) on a randomly chosen x for $d>1$ by himself. Yet it is possible that the individual learns some RSA-signatures computed by Z (e.g. by participating in some protocol or by eavesdropping) and can use these to compute some new signatures of a type not issued by Z. The purpose of this paper is to investigate which new types of RSA-signatures an individual can compute from the ones obtained from Z.

We give an example of the kind of problems we shall consider. Suppose A has received, by participating in some protocol (or by eavesdropping) two random integers x_1,x_2 and a signature $S \equiv x_1^{2/5} \cdot x_2^{3/7}$ (mod N). Then A can compute $x_1^{1/5}$, using that $x_1^{1/5} \equiv x_1^3 \cdot x_2^3 / S^7$ (mod N). On the other hand we shall prove that for all positive integers d different from 1 and 5 (and relatively prime to $\varphi(N)$), it is infeasible for A to compute $x_1^{1/d}$ from (x_1, x_2, S). Another consequence of our results is a result of Shamir [Sh83] which states that it is feasible for A to compute $x^{1/m}$ from $(x, x^{1/a_1}, \ldots, x^{1/a_s})$ if and only if m divides the least common multiple of $(a_1,\ldots,a_s)$. In section 3 we give more detailed examples related to coin systems.

This paper is organized as follows. In section 2 the notation used in this paper is introduced. Section 3 contains descriptions of the RSA scheme and the four protocols that we want to investigate. We shall state four propositions related to the respective protocols and give some examples and applications to illustrate these propositions. With the lemmas of section 4, the four propositions will be proven in section 5.

The propositions of section 3 can not be considered as mathematical statements since they involve an intuitive notion of computational feasibility which we shall not formalize. Therefore in our main theorem in section 6, we will not use any assumption on the computational feasibility of RSA-roots by individuals. In this extended abstract we shall only state this theorem in words without using the formalism of Probabilistic Turing Machines, and we shall not prove this theorem here.

2. Notation

The following notation is used throughout this paper:

$\mathbb{N},\mathbb{Z},\mathbb{Q}$ the sets of positive integers, all integers and rational numbers respectively.

$(a_1,\ldots,a_t)$ the greatest common divisor of $a_1,\ldots,a_t$; also defined for rational numbers by $(a_1,\ldots,a_t):=\frac{(a_1d,\ldots,a_td)}{d}$, where $d\in\mathbb{N}$ such that $a_1d,\ldots,a_td\in\mathbb{Z}$; this definition is independent of the choice of d.

$\mathrm{lcm}(a_1,\ldots,a_t)$ the least common multiple of $a_1,\ldots,a_t\in\mathbb{Q}$ (this is defined for rational numbers analogously to the gcd).

$a|b$ there is an integer c such that $ac=b$; also defined for $a,b\in\mathbb{Q}$.

$a\equiv b \pmod m$ it holds that $m|(a-b)$, for $a,b\in\mathbb{Q}$, $m\in\mathbb{N}$; we shall omit the suffix $(\mathrm{mod}\ m)$, if no confusion is likely to arise.

S^k the set of k-dimensional column vectors with entries from the set S.

$\boldsymbol{a}$ column vector $(a_1,\ldots,a_k)^{\mathrm{T}}$; if $\boldsymbol{a}\in S^k$, then $a_1,\ldots,a_k\in S$.

$\boldsymbol{e}_i$ the i^{th} unit vector $(0,\ldots,0,1,0,\ldots,0)^{\mathrm{T}}$ which has a 1 on the i^{th} place and zeros elsewhere (the dimension of these vectors will follow from the context).

$<\boldsymbol{a},\boldsymbol{b}>$ the scalar product of two column vectors $\boldsymbol{a}=(a_1,\ldots,a_k)^{\mathrm{T}}$ and $\boldsymbol{b}=(b_1,\ldots,b_k)^{\mathrm{T}}$, which is defined by $<\boldsymbol{a},\boldsymbol{b}>=a_1b_1+\ldots+a_kb_k$.

$[\boldsymbol{a}_1\ \ldots\ \boldsymbol{a}_t]$ the matrix with columns $\boldsymbol{a}_1,\ldots,\boldsymbol{a}_t$.

$[C\ \boldsymbol{a}]$ the matrix with column vector $\boldsymbol{a}$ concatenated at the right to matrix C.

$\mathrm{def}(\boldsymbol{a}_1,\ldots,\boldsymbol{a}_t;\boldsymbol{b})$ the defect of $\boldsymbol{a}_1,\ldots,\boldsymbol{a}_t;\boldsymbol{b}\in\mathbb{Q}^k$; this is the smallest positive integer d such that $[\boldsymbol{a}_1\ \ldots\ \boldsymbol{a}_t]\boldsymbol{y}=d\boldsymbol{b}$ has a solution $\boldsymbol{y}\in\mathbb{Z}^t$ (well defined if $[\boldsymbol{a}_1\ \ldots\ \boldsymbol{a}_t]\boldsymbol{x}=\boldsymbol{b}$ has a solution $\boldsymbol{x}\in\mathbb{Q}^t$). Examples: def(3;1)=3, def(5;1)=5, def(3,5;1)=1.

N the RSA modulus used in all the protocols; N is a composite, odd number.

$\mathbb{Z}^*_N$ the set $\{a|\ a\in\mathbb{N},\ 1\leq a\leq N,\ (a,\ N)=1\}$.

$\varphi(N)$ Euler's Totient function; $\varphi(N)=|\mathbb{Z}^*_N|$.

$\mathbb{Q}_N$ the set $\{\frac{a}{b}\big|\,a,\ b\in\mathbb{Z},\ b>0,\ (b,\ \varphi(N))=1\}$.

$\mathbf{x}^{\mathbf{a}}$ $x_1^{a_1} x_2^{a_2} \ldots x_k^{a_k} \pmod{N}$, for $\mathbf{x}=(x_1,\ldots,x_k)^{\mathrm{T}} \in (\mathbb{Z}^*_N)^k$ and $\mathbf{a}=(a_1,\ldots,a_k)^{\mathrm{T}} \in (\mathbb{Q}_N)^k$. Examples: $\mathbf{x}^{e_i} \equiv x_i$, and if $\mathbf{x}=(x^{b_1},\ldots, x^{b_k})$, then $\mathbf{x}^{\mathbf{a}} = x^{<\mathbf{a},\mathbf{b}>}$.

$\mathbf{a}/\mathbf{b}$ $(\frac{a_1}{b_1},\ldots, \frac{a_k}{b_k})^{\mathrm{T}}$, if $b_i \neq 0$ for $i=1,\ldots,k$.

3. Protocols

In this paper we will consider 4 protocols, and each but the first is a generalization of the previous one. In each protocol, a signature authority Z issues one or more RSA-signatures of certain types to the individual A, who has no influence on the integers used. We deal with the problem to determine for which other types of RSA-signatures it is feasible for A to compute them from the types of signatures that he obtained from Z.

In order to avoid technical complications ,we shall not give a mathematically precise definition of the notion "*computational feasibility*", but only the following intuitive definition. If $a_1,\ldots,a_t$ are binary strings chosen according to some prescribed probability distribution and b is a binary string with $b=f(a_1,\ldots,a_t)$ for some function f, then we say that it is feasible to compute b from $a_1,\ldots,a_t$ if there is an efficient probabilistic algorithm that outputs b with non-negligible probability, when it is given $a_1,\ldots,a_t$ as input. In this section we shall freely use the notion of computational feasibility in statements of propositions, corollaries etc. We shall state four propositions, each related to a protocol.

First we briefly sketch the RSA scheme [RSA78]. The signature authority Z chooses two large "random" primes, each of 100 decimal digits say, and computes their product N, which will be used as RSA modulus.

Let $d \in \mathbb{Z}^*_{\varphi(N)}$. The equation $d\bar{d} \equiv 1 \pmod{\varphi(N)}$ [†] has a unique solution $\bar{d} \in \mathbb{Z}^*_{\varphi(N)}$ which can be computed by Z, because Z knows the factorization of N (and thus $\varphi(N)$). We define $x^{a/d} \pmod{N}$ to be the unique solution $y \in \mathbb{Z}^*_N$ to $y^d \equiv x^a \pmod{N}$, for $x \in \mathbb{Z}^*_N$ and $\frac{a}{d} \in \mathbb{Q}_N$. This solution y can be computed by $y \equiv x^{a\bar{d}} \pmod{N}$. We call $x^{1/d} \pmod{N}$ the d^{th} *RSA-root* of $x \in \mathbb{Z}^*_N$.

Z makes N and d public, and keeps $\bar{d}$ and the factorization of N secret. The RSA-signatures issued by Z in the protocols are products of rational powers of residue classes. For all the signatures in this paper the same modulus is used. The case that an individual receives signatures with different moduli is partially solved in [Has85].

[†] The RSA-scheme can be made slightly more efficient by solving $\bar{d}$ from $d\bar{d} \equiv 1 \pmod{\lambda(N)}$, where $\lambda(N)$ is Carmichael's function. For instance, if $N=PQ$ for primes P,Q, then $\varphi(N)=(P-1)(Q-1)$ and $\lambda(N)=\varphi(N)/(P-1,Q-1)$.

We assume that it is computationally infeasible for an individual A to compute RSA-roots by himself: the only positive integer d with $(d,\varphi(N))=1$ for which A can feasibly compute $x^{1/d}$ (mod N) for uniformly chosen x from $\mathbb{Z}^*_N$, is d=1. In other words:

Assumption. *Let N be the used RSA-modulus. Then for every integer $d>1$ with $(d,\varphi(N))=1$ it is computational infeasible for A to compute $x^{1/d}$ (mod N) when given only N,d,x as input, where x is chosen uniformly from $\mathbb{Z}^*_N$*

We now describe the four protocols, the propositions and some examples (related to coin systems) to illustrate the propositions.

3.1. Protocol 1

Protocol 1. *Z makes public integers a,n with $a/n \in \mathbb{Q}_N$.*

(1) *Z chooses x uniformly from $\mathbb{Z}^*_N$ and computes the RSA-signature*
   $S \equiv x^{a/n} (\text{mod } N)$.

(2) *Z sends the pair (x,S) to A.*

(3) *A verifies the RSA-signature on x by checking if $S^n \equiv x^a (\text{mod } N)$.*

We consider the problem for which integers $m>0$ with $(m,\varphi(N))=1$, A is able to compute $x^{1/m}$(mod N) from the pair (x,S) that he received from Z. Necessary and sufficient conditions are given in the next proposition.

Proposition 1. *Fix integers a,n,m with $n,m>0$ and $(n,\varphi(N))=(m,\varphi(N))= (a,n)=1$. Then the following three statements are equivalent:*

(i) *It is feasible for A to compute $x^{1/m}$ from $(x, x^{a/n})$, if Z chooses x uniformly from $\mathbb{Z}^*_N$.*

(ii) *There are integers v,w such that $1/m=v\cdot a/n + w$.*

(iii) *$m|n$.*

Proposition 1 can be applied to coin systems, such as in Figure 2. Here f is a fixed, public, "pseudo-random" function. In a coin system, different exponents s are used, each representing another coin value. Suppose that the exponents s=3,5,7,9 (assumed to be coprime with $\varphi(N)$) are used, and that they correspond to the coin values 8,4,2,1 respectively. Now any user A can gain 7 money units simply by withdrawing a coin of value 1, which is of the form $C = f(y)^{1/9}$, and computing $C^3 = f(y)^{1/3}$, which is a coin of value 8. One can prevent users from gaining money by replacing s=9 for instance by s=11. Assume that A withdraws the coins $f(y)^{1/11}$, $f(y)^{1/7}$ and $f(y)^{1/5}$ of value 1,2 and 4 respectively. Then A can compute $f(y)^{1/(5\cdot 7\cdot 11)}$ by

$f(y)^{1/(5\cdot 7\cdot 11)} = (f(y)^{1/5})^3 (f(y)^{1/7})^{13} (f(y)^{1/11})^{-27}$. But by proposition 1, A cannot compute $f(y)^{1/3}$ from $f(y)^{1/(5\cdot 7\cdot 11)}$. So A cannot gain a money unit.

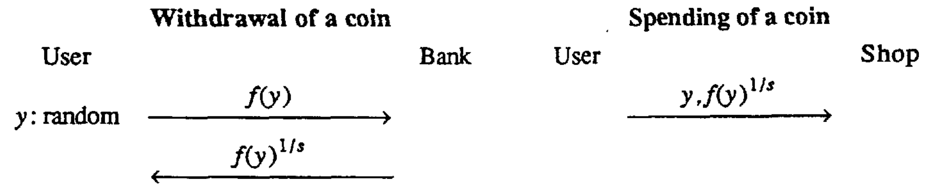

Fig. 2. A simple coin system

3.2. Protocol 2

In Protocol 2, Z issues to A *one* RSA-signature that is a product of powers of RSA-roots on integers (chosen by Z). Proposition 2 describes which new RSA-signatures are feasibly computable from the received ones.

Protocol 2. *Z makes public vectors* $a, n \in \mathbb{Z}^k$ *such that* $a/n \in (\mathbb{Q}_N)^k$. *Let* $n^* = \mathrm{lcm}(n_1, \ldots, n_k)$.

(1) *Z chooses* x *uniformly from* $(\mathbb{Z}^*_N)^k$, *and computes the signature*

$S \equiv x^{a/n} (\mathrm{mod}\ N)$.

(2) *Z sends* (x, S) *to A.*

(3) *A verifies the signature on* x *by checking whether*

$S^{n^*} \equiv x_1^{a_1 n^*/n_1} \cdot \ldots \cdot x_k^{a_k n^*/n_k} (\mathrm{mod}\ N)$.

Proposition 2. *Fix vectors* $a, n, b, m \in \mathbb{Z}^k$ *with* $(a_i, n_i) = (b_i, m_i) = (n_i, \varphi(N)) = (m_i, \varphi(N)) = 1$ *for* $i = 1, \ldots, k$. *Then the following three statements are equivalent:*

(i) *It is feasible for A to compute* $x^{b/m}$ *from* $(x, x^{a/n})$, *if Z chooses* x *uniformly from* $(\mathbb{Z}^*_N)^k$.

(ii) *There are* $v \in \mathbb{Z}$ *and a vector* $w \in \mathbb{Z}^k$ *such that* $b/m = v(a/n) + w$.

(iii) $m_i | n_i$ *for* $i = 1, \ldots, k$ *and*

$a_i b_j n_j / m_j \equiv a_j b_i n_i / m_i \mod (n_i, n_j)$ *for* $1 \le i, j \le k$.

To illustrate this proposition, we consider the product $\prod_{i=1}^{10} x_i^{3^{i-1}/17}$. We are interested in the question whether it is feasible for an individual to change the order of the terms in the product, i.e. is it feasible for an individual to find a non-identical permutation τ such that

$\prod_{i=1}^{10} x_i^{3^{i-1}/17} = \prod_{i=1}^{10} x_i^{3^{\tau(i)-1}/17}$? Using the next corollary (which can be derived from Proposition 2) we can prove that this is not feasible. So to each position in this product (i.e. to each exponent) we can assign a different coin value. This result is used in the offline check system of [CBHMS89].

Corollary 1. *Let p and q be different primes such that $(p,\varphi(N))=(q,\varphi(N))=1$ and let k,m be integers. Define the integral vectors $a=(q^m,\ldots,q^{m+k-1})^T$ and $n=(p,\ldots,p)^T$. The following statements are equivalent if x is chosen randomly from $(\mathbb{Z}^*_N)^k$.*

(i) *There is a non-identical permutation τ of $(0,\ldots,k-1)$, such that it is feasible for A to compute $x^{b/n}$ from $(x,x^{a/n})$ where $b=(q^{m+\tau(0)},\ldots,q^{m+\tau(k-1)})^T$.*

(ii) *There is an i_0 with $1\leq i_0\leq k$ such that $q^{i_0} \equiv 1 \pmod{p}$.*

3.3. Protocol 3

We now consider a general protocol, in which Z issues to A several signatures at once, together with the chosen vector x. Notice that sending x is exactly the same as sending $(x^{e_1},\ldots, x^{e_k})$, where $e_1,\ldots,e_k$ are the unit vectors of $(\mathbb{Q}_N)^k$.

Protocol 3. *Z makes public vectors $a_1,\ldots,a_s \in (\mathbb{Q}_N)^k$.*

(1) *Z chooses x uniformly from $(\mathbb{Z}^*_N)^k$, and computes $S_i \equiv x^{a_i} \pmod{N}$ for $i=1,\ldots,s$.*

(2) *Z sends $(x, S_1,\ldots,S_s)$ to A.*

(3) *A verifies that $S_i^d \equiv x^{da_i} \pmod{N}$ for $i=1,\ldots,s$, where d is a positive integer such that $da_1,\ldots,da_s \in \mathbb{Z}^k$.*

We want to know for which vectors $b \in (\mathbb{Q}_N)^k$, it is feasible for A to compute $x^b \pmod{N}$ from $(x, x^{a_1},\ldots, x^{a_s})$.

Proposition 3. *Fix vectors $a_1,\ldots,a_s,b \in (\mathbb{Q}_N)^k$. Then the following four statements are equivalent:*

(i) *It is feasible for A to compute x^b from $(x, x^{a_1},\ldots, x^{a_s})$, if Z chooses x uniformly from $(\mathbb{Z}^*_N)^k$.*

(ii) *There are $v_1,\ldots,v_s \in \mathbb{Z}$ and a vector $w \in \mathbb{Z}^k$ such that $b=v_1a_1+\ldots+v_sa_s+w$.*

(iii) $\mathrm{def}(a_1,\ldots,a_s,e_1,\ldots,e_k;b)=1$.

(iv) *Let $\lambda_1,\ldots,\lambda_m$ be all the subdeterminants of* $[a_1 \ldots a_s]$ *of order between* 1 *and* $\min(k,s)$, *and* $\lambda_{m+1},\ldots,\lambda_n$ *be all the subdeterminants of* $[a_1 \ldots a_s\, b]$ *of order between* 1 *and* $\min(k,s+1)$, *containing at least one entry from* **b**. *Then* $(1,\lambda_1,\ldots,\lambda_m)=(1,\lambda_1,\ldots,\lambda_n)$ *(i.e.* $(1,\lambda_1,\ldots,\lambda_m) \mid \lambda_i$, *for* $i=m+1,\ldots,n$*).*

To illustrate how this proposition can be used, we consider the off-line coin system of [OO89]. In this system the bank uses a signature scheme which we do not specify here. The user makes RSA-signatures using his own modulus N whose factorization he keeps secret; so here the user plays the role of a signature authority. Let L be a fixed integer, and define $I \equiv (\text{account number user})^L \bmod N$. In Figure 3 the basic idea of the withdrawal (in which the user is able to blind and the bank to sign messages, cf. [OO89]) and spending protocol of a coin is given. Each shop sends the numbers it received to the bank and the bank verifies that these numbers have not been used before. Since the system is off-line, usually each shop first collects the numbers from several payments before sending them to the bank.

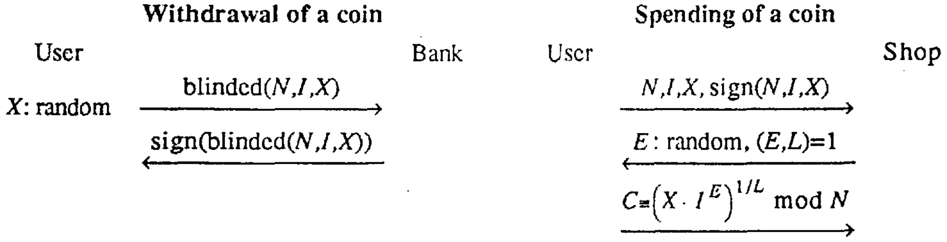

Fig. 3. The (simplified) off-line coin system of [OO89]

From Proposition 3 it follows that it is not feasible for the shop/bank to compute the identity of the user (i.e. $I^{1/L} \bmod N$) from N,I,X,E and $C = X^{1/L} \cdot I^{E/L}$. But if the user spends the same coin at two shops, then the bank receives the integers $N,I,X,\text{sign}(N,I,X)$, E_1,E_2 (coprime with L), $(X\cdot I^{E_1})^{1/L} \bmod N$ and $(X\cdot I^{E_2})^{1/L} \bmod N$. From Proposition 3 it follows that the bank can compute the users identity $I^{1/L} \bmod N$ from this if and only if $(E_1-E_2,L)=1$. Hence the probability that a double spender is caught by the bank is approximately $\varphi(L)/L$. This probability is close to 1 if L is a large prime, and close to 0 if L is the product of small primes. Therefore it is not wise to let the user choose L himself (which was the original suggestion of [OO89]), but to fix L as a large prime.

Suppose we modify protocol 3 in such a way that an individual A receives s signatures on *different vectors*, so suppose A has received $(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_s$,

$S_1 \equiv x_1^{a_1}, \ldots, S_s \equiv x_s^{a_s}$) and wants to compute $S_{s+1} \equiv x_{s+1}^{a_{s+1}}$, where $a_i \in (\mathbb{Q}_N)^{k_i}$ and $x_i \in (\mathbb{Z}^*_N)^{k_i}$,(i=1,...,s+1). Define y to be the vector obtained by concatenating all the different entries of the vectors $x_1,\ldots,x_{s+1}$. We can write S_i as y^{b_i}, where the exponent on the "new" y_j's (i.e. those y_j which were no entry of x_i) is zero. Now Proposition 3 can be used to determine for which a_{s+1} it is feasible for A to compute S_{s+1} from $(y,S_1,\ldots,S_s)$.

3.4. Protocol 4

We now consider the most general protocol, in which Z issues to A several signatures at once, but without sending the used vector.

Protocol 4. *Z makes public vectors* $a_1,\ldots,a_s \in (\mathbb{Q}_N)^k$.

(1) *Z chooses* x *uniformly from* $(\mathbb{Z}^*_N)^k$, *and computes* $S_i \equiv x^{a_i} (\mathrm{mod}\ N)$ *for* $i=1,\ldots,s$.
(2) *Z sends* $(S_1,\ldots,S_s)$ *to A.*

If one does not accept this as a useful protocol (because A can in general not verify the signatures), then assume A has received $(x^{a_1},\ldots,x^{a_s})$ during eavesdropping. We want to know for which vectors $b \in (\mathbb{Q}_N)^k$, it is feasible for A to compute $x^b (\mathrm{mod}\ N)$ from $(x^{a_1},\ldots,x^{a_s})$, and prove that the only b's for which x^b is computable from $(x^{a_1},\ldots,x^{a_s})$, is the lattice generated by $a_1,\ldots,a_s$.

Proposition 4. *Fix vectors* $a_1,\ldots,a_s,b \in (\mathbb{Q}_N)^k$, *and assume that the equation* $[a_1 \ldots a_s]y=b$ *is solvable in* $y \in \mathbb{Q}^s$. *Then the following four statements are equivalent:*

(i) *It is feasible for A to compute* x^b *from* $(x^{a_1},\ldots,x^{a_s})$, *if Z chooses* x *uniformly from* $(\mathbb{Z}^*_N)^k$.
(ii) *There are* $v_1,\ldots,v_s \in \mathbb{Z}$ *such that* $b = v_1a_1 + \ldots + v_sa_s$.
(iii) def$(a_1,\ldots,a_s;b)$=1.
(iv) *Let* $\mu_1,\ldots,\mu_m$ *be the subdeterminants of* $[a_1 \ldots a_s]$ *of order* k *and* $\mu_{m+1},\ldots,\mu_n$ *be the subdeterminants of* $[a_1 \ldots a_s\ b]$ *of order* k*, containing at least one entry from* b*.*
Then $(1,\mu_1,\ldots,\mu_m)=(1,\mu_1,\ldots,\mu_n)$ *(i.e.* $(1,\mu_1,\ldots,\mu_m) | \mu_i$*, for* $i=m+1,\ldots,n$*).*

This proposition implies that only the exponents must be investigated, and that the only reasonable computations an individual can do, in order to create a new signature from some received signatures, are the basic computations (add, subtract, multiply and divide). So applying the cosine or DES can not help an individual in creating more signatures.

The previous propositions can be used to prove the following results.

Corollary 2. *Let $a, a_1, \ldots, a_s, b, d$ be positive integers coprime with $\varphi(N)$, c be an integer, and x, y be chosen randomly from $\mathbb{Z}^*_N$. Then the following five results hold for A.*

(i) *It is feasible to compute* $x^{1/d}$ *from* $(x, x^{c/a})$ $\Leftrightarrow$ $d \mid \frac{a}{(a, c)}$.

(ii) *It is feasible to compute* $x^{1/d}$ *from* $(x, y, x^{1/a} \cdot y^{1/b})$ $\Leftrightarrow$ $d \mid \frac{a}{(a, b)}$.

(iii) *It is feasible to compute* $x^{1/d}$ *from* $(x, x^{1/a_1}, \ldots, x^{1/a_s})$ $\Leftrightarrow$ $d \mid \mathrm{lcm}\,(a_1, \ldots, a_s)$ [Sh83].

(iv) *It is feasible to compute* $(xy)^{1/d}$ *from* $(x, y, x^{1/a}, y^{1/b})$ $\Leftrightarrow$ $d \mid (a, b)$.

(v) *It is feasible to compute* x^d *from* $(x^{a_1}, \ldots, x^{a_s})$ $\Leftrightarrow$ $\gcd(a_1, \ldots, a_s) \mid d$.

4. Auxiliary results

When we say that something is computable in polynomial time, we mean that it is computable by a polynomial time deterministic algorithm.

Lemma 1. *The following operations can be done in polynomial time:*

(1) *computing* $\gcd(a,b)$ *from a and b,*

(2) *computing the inverse of a* (mod b) *from a and b, if* $(a,b)=1$,

(3) *computing* a^b (mod c) *from a,b and c, if* $(a,c)=1$,

(4) *the Gaussian elimination method for a system of linear equations with rational coefficients,*

(5) *determining the rank of a rational matrix,*

(6) *determining the determinant of a given rational square matrix,*

(7) *determining the inverse of a nonsingular rational square matrix,*

(8) *testing rational vectors for linear independence,*

(9) *computing the Hermite Normal Form of a matrix* [KaBa79],

(10) *computing a unimodular matrix U, such that AU is the Hermite Normal Form of A, for a rational matrix A of full row rank,*

(11) *deciding if a system of rational linear equations has an integral solution, and if so, finding one.*

References for the proofs can be found in Chapter 3 and 5 in [Schr86].

Lemma 2. ([Heg1858] page 111)

Let A be a rational matrix of full row rank, with k rows, and let $\boldsymbol{b}$ *be a rational column k-vector. Then* $A\boldsymbol{x}=\boldsymbol{b}$ *has an integral solution* $\boldsymbol{x}$, *if and only if the gcd of all subdeterminants of A of order k divides each subdeterminant of* $[A\ \boldsymbol{b}]$ *of order k.*

Lemma 3. *Let $a_1,\dots,a_s,b\in(\mathbb{Q}_N)^k$, $A=[a_1\ \dots\ a_s]$ of full row rank, and let d=def($a_1,\dots,a_s;b$); hence*

$$Av=db \tag{4.1}$$

is solvable in $v\in\mathbb{Z}^s$.

Further, let $\mu_1,\dots,\mu_m$ be the subdeterminants of A of order k and $\mu_{m+1},\dots,\mu_n$ be the subdeterminants of $[A\ \ b]$ of order k, containing at least one entry from b.

Then:

(i) $$d=\frac{(\mu_1,\dots,\mu_m)}{(\mu_1,\dots,\mu_m,\mu_{m+1},\dots,\mu_n)}. \tag{4.2}$$

(ii) *There is a polynomial time deterministic algorithm that computes d and a solution of* (4.1).

(iii) *There is a polynomial time deterministic algorithm that computes a $z\in\mathbb{Q}^k$ such that*

$$(d,<z,db>)=1 \text{ and } A^{\mathrm{T}}z\in\mathbb{Z}^s. \tag{4.3}$$

Remark: Note that expression (4.2) does *not* yield a polynomial time algorithm to compute def($a_1,\dots,a_s;b$), because $m=\binom{s}{k}$, $n-m=\binom{s}{k-1}$, and $s\geq k$.

Proof. Matrix A has full row rank, so according to Lemmas 1.7 and 1.10, we can compute in polynomial time a matrix $[D0]$ (in Hermite Normal Form in which D is a nonsingular square matrix and 0 is a matrix consisting of zeros) and a unimodular matrix U such that $A=[D\,0]U$. The matrices $U,U^{-1},U^{\mathrm{T}},(U^{\mathrm{T}})^{-1}$ have integral entries and in this lemma matrix D is rational. Since $A^{\mathrm{T}}z=U^T\begin{pmatrix}D^T\\0\end{pmatrix}z=U^T\begin{pmatrix}D^Tz\\0\end{pmatrix}$, we have that $A^{\mathrm{T}}z\in\mathbb{Z}^s$ if and only if $D^{\mathrm{T}}z\in\mathbb{Z}^k$. Equation (4.1) has an integral solution if and only if

$$Dw=db \text{ is solvable in } w\in\mathbb{Z}^k, \tag{4.4}$$

because there is a 1-1 relationship between the solutions $v\in\mathbb{Z}^s$ of (4.1) and $w\in\mathbb{Z}^k$ of (4.4), defined by $Uv=\begin{pmatrix}w\\0\end{pmatrix}$. Hence d is also the smallest positive integer such that $Dw=db$ has a solution in $w\in\mathbb{Z}^k$, in other words def($A;b$)=def($D;b$). Combining the previous equations gives: $<z,db>=<z,Av>=<A^{\mathrm{T}}z,v>=<U^T\begin{pmatrix}D^Tz\\0\end{pmatrix},v>=$ $<\begin{pmatrix}D^Tz\\0\end{pmatrix},Uv>=<\begin{pmatrix}D^Tz\\0\end{pmatrix},\begin{pmatrix}w\\0\end{pmatrix}>=<D^{\mathrm{T}}z,w>$ Hence (4.3) is equivalent to

$$D^{\mathrm{T}}z\in\mathbb{Z}^k \text{ and}$$

$$(d,<D^{\mathrm{T}}z,w>)=1, \text{ for every solution } w \text{ of (4.4).} \tag{4.5}$$

(i) For every integer δ, the subdeterminants of $[A\ \delta b]$ of order k are $\mu_1,.., \mu_m, \delta\mu_{m+1},..., \delta\mu_n$. Now Lemma 2 implies that the equation $Av = \delta b$ has a solution in $v \in \mathbf{Z}^s$ if and only if

$$(\mu_1,...,\mu_m) \mid \delta\mu_i, \text{ for } i=m+1,...,n.$$

This holds if and only if $(\mu_1,.., \mu_m) \big| \delta \cdot (\mu_1,.., \mu_m, \mu_{m+1},..., \mu_n)$. Because d is the smallest positive integer for which (4.1) has an integral solution in v, we have $d = \frac{(\mu_1,..., \mu_m)}{(\mu_1,..., \mu_m, \mu_{m+1},..., \mu_n)}$.

(ii) Matrix D is a $k \times k$-matrix, so $\det(D)$ is the only subdeterminant of order k of D, and the matrix $[D\ b]$ has k subdeterminants $\eta_1,...,\eta_k$ of order k containing an entry from b. If we apply Lemma 3.(i) on matrix D, we get $d = \frac{\det(D)}{(\det(D),\ \eta_1,..., \eta_k)}$. The matrix D and the subdeterminants $\eta_1,...,\eta_k$ can be computed in polynomial time from A. Hence d can be computed in polynomial time and with this d, a solution w of (4.4) can be computed by Gauss elimination. With this w a solution v of (4.1) can also be computed in polynomial time.

(iii) It is sufficient to prove that there is a polynomial time deterministic algorithm to compute a $z \in \mathbb{Q}^k$ such that (4.5) holds. Let w be a solution of (4.4) and define $d_1 = \gcd(w_1,...,w_k)$. The equation $Dx = \frac{d}{(d,d_1)} b$ has $x = \frac{1}{(d,d_1)} w$ as an integral solution. But d was the smallest positive integer for which (4.4) is solvable, so we must have $(d,d_1)=1$. With (the extended) Euclid's algorithm we find in polynomial time an $y \in \mathbf{Z}^k$ such that $<y,w>=d_1$. If we define $z := (D^T)^{-1} y$, then $z \in \mathbb{Q}^k$, $D^T z \in \mathbf{Z}^k$ and $(d,<D^T z,w>) = (d,<y,w>) = (d,d_1) = 1$. Hence this z satisfies (4.5). □

5. Proofs of the propositions

We derive the four propositions of section 3 from the previous lemmas (or from Theorem 1, using the assumption on the computability of RSA-roots by individuals).

Proof of Proposition 4.

(ii)⇔(iii)⇔(iv)

This follows from Lemma 3 and the definition of defect.

(i)⇒(iii)

Suppose that it is feasible for the individual to compute x^b from $(x^{a_1},..., x^{a_s})$ for uniformly chosen x. Put $A=[a_1 \ldots a_s]$ and $d=\mathrm{def}(a_1, ...,a_s;b)$.

By Lemma 3 we can compute in polynomial time a vector $z=(z_1,...,z_k) \in \mathbb{Q}^k$ such that $(d,<z,db>)=1$ and $A^T z \in \mathbf{Z}^s$. Hence $A^T z = (c_1,...,c_s)^T$, where $c_i = <a_i,z> \in \mathbf{Z}$ for

$i=1,\dots,s$. Using (the extended) Euclid's algorithm we can feasibly compute $\alpha,\beta\in\mathbf{Z}$ with $\alpha<z,db>+\beta d=1$, so $\alpha<z,b>+\beta=1/d$. Choose x uniformly from $\mathbf{Z}^*_N$ and put $x=(x^{z_1},\dots,x^{z_k})$. Hence $x^{a_i}\equiv x^{<z,a_i>}=x^{c_i}$ for $i=1,\dots,s$. So the individual can feasibly compute those x^{a_i}, and thus by assumption he can compute x^b. But then it is feasible to compute $(x^b)^\alpha x^\beta = x^{\alpha<z,b>+\beta} = x^{1/d}$. From the assumption on RSA-roots, it follows that $d=1$. This proves (iii).

(ii)⇒(i)

Suppose there are integers $v_1,\dots,v_s$ such that $v_1a_1+\dots+v_sa_s=db$. Lemma 3 states that it is feasible to compute such $v_1,\dots,v_s$. Now x^b can be computed in polynomial time from $x^b=(x^{a_1})^{v_1}\dots(x^{a_s})^{v_s}$. □

Proof of Proposition 3.

(i)⇔(ii)⇔(iii)

This follows from Proposition 4 with $A=[a_1 \dots a_s\ e_1 \dots e_k]$ (note that A has full row rank whence the equation $Ay=b$ is solvable in $y\in\mathbb{Q}^k$).

(iii)⇔(iv)

Define $\tilde{A}=[a_1 \dots a_s]$ and $I=[e_1 \dots e_k]$. Since each column of I has exactly *one* entry $\neq 0$, each subdeterminant of $[\tilde{A}\ \ I]$ containing q columns from I is a subdeterminant of $[\tilde{A}]$ of order s-q. Further $\det(I)=1$. Similarly, each subdeterminant of $[\tilde{A}\ \ I\ \ b]$ containing q columns from I and at least one entry from b is a subdeterminant of $[\tilde{A}\ \ b]$ of order s-q, containing at least one entry from b. □

We leave the proofs of Propositions 1 and 2 to the reader.

6. Main theorem

In our propositions of section 3, we used the assumption on the computability of RSA-roots by individuals. These propositions can be generalized into a theorem, in which that assumption is not required anymore. To state this theorem we need the formalism of Probabilistic Turing Machines. But in this extended abstract we shall only state that theorem in words and therefore not prove it here.

Theorem 1. *Let $a_1,\dots,a_s,b\in(\mathbb{Q}_N)^k$, and assume that the equation $[a_1 \dots a_s]y=b$ is solvable in $y\in\mathbb{Q}^k$. Hence $d=\mathrm{def}(a_1,\dots,a_s;b)$ is defined.*

(1) *Suppose we have a "black box" which outputs x^b from the input $(x^{a_1},\dots,x^{a_s})$ with average probability (over x) $\geq\varepsilon_1>0$.*

With this black box we can build an algorithm which computes $u^{\frac{1}{d}}$ from input u, with probability $\geq\frac{1}{2}$, for every fixed $u\in\mathbf{Z}^*_N$. The running time of this algorithm is

$\frac{1}{\varepsilon_1}P_1$, where P_1 depends polynomially on the length of the input (which consists of N and the numerators and the denominators of the coordinates of $a_1,\ldots,a_s,b$).

(2) *Suppose we have a "black box" which outputs $u^{\frac{1}{d}}$ from the input with average probability (over u) $\geq \varepsilon_2 > 0$.*

With this black box we can build an algorithm which computes x^b from input $(x^{a_1},\ldots,x^{a_s})$, with probability $\geq \frac{1}{2}$, for every fixed $x \in (\mathbb{Z}^*_N)^k$. The running time of this algorithm is $\frac{1}{\varepsilon_3}P_2$, where P_2 depends polynomially on the same input as in (1).

More informally, this theorem states that computing x^b from $(x^{a_1},\ldots,x^{a_s})$ is polynomial time reducible to computing $u^{\frac{1}{d}}$ from u, where d=def$(a_1,\ldots,a_s;b)$ and x,u random.

Under the assumption that it is infeasible for an individual to compute RSA-roots for random numbers, we can derive Proposition 4 (i)⇔(ii)⇔(iii) from Theorem 1.

References

[CBHMS89] David Chaum, Bert den Boer, Eugène van Heyst, Stig Mjølsnes and Adri Steenbeek, "Efficient Offline Electronic Checks", to appear in *Advances in Cryptology-EUROCRYPT '89*, Lecture Notes in Computer Science, Springer-Verlag.

[CE86] David Chaum and Jan-Hendrik Evertse, "A secure and privacy-protecting protocol for transmitting personal information between organizations", *Advances in Cryptology -CRYPTO '86*, A.M. Odlyzko ed., Lecture Notes in Computer Science 263, Springer-Verlag,.pp 118-167.

[Gill77] John Gill, "Computational Complexity of Probabilistic Turing Machines", *SIAM L. Comp.* **6** (1977) pp. 675-695.

[Has85] Johan Hastad, "On using RSA with low exponent in a public key network", *Advances in Cryptology -CRYPTO '85*, H.C. Williams ed., Lecture Notes in Computer Science 218, Springer-Verlag,.pp 403-408.

[Heg1858] I. Heger, "Über die Auflösung eines Systemes von mehreren unbestimmten Gleichungen des ersten Grades in ganzen Zahlen", *Denkschriften der Königlichen Akademie der Wissenschaften (Wien), Mathematisch-naturwissenschaftliche Klasse* **14** (2. Abth.) (1858) pp1-122.

[KaBa79] R. Kannan and A. Bachem, "Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix", *SIAM Journal on Computing*, **8** (1979) pp 499-507.

[OO89] Tatsuaki Okamoto and Kazuo Ohta, "Disposable Zero-Knowledge Authentications and Their Applications to Untraceable Electronic Cash", to appear in *Advances in Cryptology -CRYPTO '89*, Lecture Notes in Computer Science, Springer-Verlag.

[RSA78] R.L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems", *Comm. of the ACM* **21** (1978) pp 120-126.

[Schr86] Alexander Schrijver, *Theory of Linear and Integer Programming*, John Wiley & Sons, 1986.

[Sh83] Adi Shamir, "On the Generation of Cryptographically Strong Pseudorandom Sequences", *ACM Trans. on Computer Systems*, **1** (1983) pp 38-44.