EFFICIENT SIGNATURE SCHEMES BASED ON POLYNOMIAL EQUATIONS

(preliminary version)

H. Ong[1], C.P. Schnorr[1], A. Shamir[2]

[1]Fachbereich Mathematik
Universität Frankfurt

[2]Applied Mathematics Department
The Weizman Institute of Science
Rehovot 76100, Israel

ABSTRACT

Signatures based on polynomial equations modulo n have been introduced by Ong, Schnorr, Shamir [3]. We extend the original binary quadratic OSS-scheme to algebraic integers. So far the generalised scheme is not vulnerable by the recent algorithm of Pollard for solving $s_1^2 + k s_2^2 = m \pmod n$ which has broken the original scheme.

1. INTRODUCTION

Diffie and Hellman [1] introduced the concept of digital signature and that of public key cryptosystem. The RSA system [6] is currently believed to be the most secure scheme for both purposes. A new type of signature scheme based on the quadratic equation $s_1^2 + k s_2^2 = m \pmod n$ has been proposed by Ong, Schnorr, Shamir [3]. Here m is the message, s_1 and s_2 are the signature, and k and n are the publicly known key. The new scheme would be much easier to implement than the RSA-scheme, but it has been broken by a recent algorithm of Pollard which solves the equation $x^2 + k y^2 = m \pmod n$ without factoring n.

G.R. Blakley and D. Chaum (Eds.): Advances in Cryptology - CRYPTO '84, LNCS 196, pp. 37-46, 1985.

In this paper we consider signature schemes based on more general polynomial equations modulo n. In particular we extend the original OSS-scheme from rational integers to algebraic integers. This leads to a signature scheme based on the quadric equation $(m_2 - 2ks_{12}s_{21})^2 + 4s_{22}^2(ds_{12}^2 + k(s_{21}^2 + ds_{22}^2) - m_1) = 0 \pmod n$ where m_1 and m_2 are the message, s_{12}, s_{21} and s_{22} are the signature, and the public key consists of the integers k,d,n with $1 \leq k,d < n$. The private key is the square root $\sqrt{-k} \pmod n$. Signature verification can be done with 10 multiplications on integers modulo n, signature generation requires 9 multiplications and 1 division modulo n.

All participants of the system may share the (d,n)-part of their public key provided that the factorisation of n is completely unknown.

2. SIGNATURES BASED ON POLYNOMIAL EQUATIONS

When Alice joins the communication network she publishes a key consisting of two parts: a modulus n and the integer coefficients of a polynomial $P(s_1,\dots,s_d) \in \mathbb{Z}[s_1,\dots,s_d]$ with indeterminates $s_1,\dots,s_d$. The modulus n is the product of two large random primes p,q. The factorization of n should be unknown, except possibly to Alice. In order to prevent factoring of n by known factoring algorithms n should be at least 600 bits long. The coefficients of P are integers in the range $\mathbb{Z}_n := \{c \in \mathbb{Z} : 0 \leq c < n\}$. The elements in $\mathbb{Z}_n$ are used as representatives for the ring $\mathbb{Z}/n\mathbb{Z}$ of integers modulo n. Typically P will only have a few coefficients.

The messages m are numbers in $\mathbb{Z}_n$. A tuple $\underline{s} = (s_1,\dots,s_d)$ of numbers in the same range is a signature for m if it satisfies the equation

$$(1) \qquad P(s_1,\dots,s_d) = m \pmod n .$$

Given the coefficients of P and n it is easy to verify Alice's signatures by evaluating $P(s_1,\dots,s_d)$ with a few modular multiplications and additions.

Unlike the RSA system, signatures are not uniquely associated with messages. Since the number of possible messages is n while the number of possible signature tuples is n^d, each message has about n^{d-1} different signatures. However, the probability that a randomly chosen tuple $\underline{s} = (s_1,\dots,s_d)$ will be a valid signature of a given m is negligible, and thus the multiplicity of signatures does not imply that they are easy to find.

The secret that helps Alice solve the equation (1) is an integer (d,d)-matrix A which modulo n is invertible. If the transformation

$\underline{x} = A\underline{s} \pmod n$ transforms P into a polynomial $x_1P'(x_2,\ldots,x_d) = P(s_1,\ldots,s_d) \pmod n$ then Alice can easily solve equation (1). She picks random values $x_2,\ldots,x_d \in \mathbb{Z}_n$, evaluates

$$(2) \qquad x_1 := m/P'(x_2,\ldots,x_d) \pmod n$$

and transforms

$$(3) \qquad \underline{s} := A^{-1}\underline{x} \pmod n .$$

So Alice can generate signatures of m by choosing random values $x_2,\ldots,x_d$ and evaluating (2), (3) using a few modular multiplications and additions and one modular division. If $P'(x_2,\ldots,x_d)$ is not relatively prime to n then $m/P'(x_2,\ldots,x_d) \pmod n$ may be not defined, but if all the factors of n are large Alice is unlikely to choose such values $x_2,\ldots,x_d$.

The relationship between messages and signatures are summarized in the following lemma. Let $\mathbb{Z}_n^*$ be the set of numbers in $\mathbb{Z}_n$ which are relatively prime to n. Note that $\mathbb{Z}_n$ represents the set $\mathbb{Z}/n\mathbb{Z}$ of integers modulo n, so $\mathbb{Z}_n$ is a commutative ring under addition and multiplication modulo n and $\mathbb{Z}_n^* \subset \mathbb{Z}_n$ is the group of invertible elements.

<u>LEMMA 1</u> For every $m \in \mathbb{Z}_n^*$ the set of signatures of m is in 1-1 correspondence with the set of values (3) as $x_2,\ldots,x_d$ range over $\mathbb{Z}_n$ and $x_1 = m/P'(x_2,\ldots,x_d) \pmod n$.

<u>PROOF</u> For every $(x_2,\ldots,x_d) \in (\mathbb{Z}_n)^{d-1}$ with $P'(x_2,\ldots,x_d) \in \mathbb{Z}_n^*$ (2), (3) clearly define a signature $\underline{s}$ of m. On the other hand for every signature $\underline{s} = (s_1,\ldots,s_d)$ there exists $\underline{x} := A\underline{s} \pmod n$. We have $P(s_1,\ldots,s_d) = x_1P'(x_2,\ldots,x_d) = m \pmod n$, and $P'(x_2,\ldots,x_d) \in \mathbb{Z}_n^*$ follows from the assumption $m \in \mathbb{Z}_n^*$. Since A is non singular only one value of $(x_2,\ldots,x_d) \in (\mathbb{Z}_n)^{d-1}$ can correspond to each signature. Q.E.D.

<u>REMARKS</u> (i) By using independent random values $x_2,\ldots,x_d$, Alice can choose an arbitrary signature of m with uniform probability distribution, and is not restricted to signatures of some special form.
(ii) If several messages m^i are signed with the same $x_2,\ldots,x_d$ then $\underline{x}^i = (x_1^i,\ldots,x_d^i)$ and the signature $\underline{s}^i$ are known for each message and A can be computed from the linear equations $\underline{x}^i = A\underline{s}^i \pmod n$. Thus Alice must choose independent random values $x_2,\ldots,x_d$ for each message.

How does Alice generate her public key? She first chooses the modulus n as a large composite number which is difficult to factor. By using a probabilistic primality testing algorithm on random integers with at least 300 bits, Alice can find after a few hundred tests two numbers p and q which are almost certainly primes. The product n of p and q is easy to compute, but even the fastest known factoring algorithm on the fastest available computer will take millions of years to factor it. The generation of n can be done within a few hours on a typical microcomputer. Such an overnight initialization is acceptable in most applications, but if the user cannot afford it, there is a faster alternative: If a trusted third party (the NBS?) computes n and then erases p and q, no one knows the factorization of n and thus everyone can use it as a standard modulus.

In order to generate the polynomial P, Alice chooses a simple polynomial $P'(x_2,\dots,x_d)$ with integer coefficients and then picks a random integer (d,d)-matrix A. Alice keeps A secret, transforms the polynomial $x_1\, P'(x_2,\dots,x_d)$ with $\underline{x} := A\,\underline{s} \pmod{n}$ into a polynomial P, $P(s_1,\dots,s_d) = x_1\, P'(x_2,\dots,x_d) \pmod{n}$ and publishes the coefficients of the transformed polynomial P. P is no longer linear in any of the variables. The equation $P(s_1,\dots,s_d) = m \pmod{n}$ is apparently difficult. Alice also verifies that A is invertible modulo n. If the prime factors of n are large then singular matrices are unlikely to occur. It is important that Alice can generate P without knowing the factors of n. All the participants of the communication network may use the same simple polynomial $P'(x_2,\dots,x_d)$ and even the same modulus n (provided that the factors of n are unknown) and differ only in their choice of A.

The security of the scheme requires to choose particular transformation matrices A which cannot be easily computed from the coefficients of P and P'. We choose the polynomial P' and the matrix A so that recovering A from the polynomials P and P' is as hard as factoring n. Since Alice is not restricted to signatures of some special form it is impossible to obtain information on the secret parameters, A and the factors of n by analysing her signatures. Also Alice herself may be unaware of the factors of n. Since Bob cannot benefit from Alice's signatures and cannot use her method for solving equation (1), he must come up with an alternative way of solving this equation. So for each class of transformations A and for each polynomial P' one must carefully analyse whether equation (1) is sufficiently difficult for the corresponding polynomials P.

The security of the scheme is based on the difficulty of factoring n. When the factors p and q of n are known the equation (1) can be solved efficiently. The probabilistic root finding algorithm of Rabin[5] computes $\underline{s}', \underline{s}'' \in \mathbb{Z}^d$ such that $P(\underline{s}') = m \pmod{p}$ and $P(\underline{s}'') = m \pmod{q}$. By the Chinese remainder theorem $\underline{s}'$ and $\underline{s}''$ can be combined to a solution $\underline{s} = \sigma\underline{s}' + \tau\underline{s}'' \pmod{n}$. Here σ and τ are integers

satisfying $\sigma = \begin{cases} 1 \pmod{p} \\ 0 \pmod{q} \end{cases}, \qquad \tau = \begin{cases} 0 \pmod{p} \\ 1 \pmod{q} \end{cases}$.

<u>The binary quadratic scheme:</u> The simplest polynomial equation (1) appears for d = 2, we transform the equation

$$(5) \qquad x_1 \cdot x_2 = m \pmod n$$

using an arbitrary $u \in \mathbb{Z}_n^*$ by the linear substitution

$$x_1 := s_1 + u^{-1}s_2 \pmod n$$
$$x_2 := s_1 - u^{-1}s_2 \pmod n \ .$$

This yields $x_1 \cdot x_2 = s_1^2 - u^{-2}s_2^2 = m \pmod n$. So the trivial equation (5) is transformed into the less trivial polynomial equation

$$(6) \qquad s_1^2 + k\, s_2^2 = m \pmod n \quad \text{with} \quad k = -u^{-2} \pmod n .$$

The public key of the corresponding signature scheme consists of n and k, and the private key is u. A pair $(s_1, s_2) \in (\mathbb{Z}_n)^2$ is a valid signature for m if $s_1^2 + k\, s_2^2 = m \pmod n$. Recovering the private key u from the public key k requires the computation of $\sqrt{-k} \pmod n$ and thus is as hard as factoring n.

Unfortunately this case of our signature concept is insecure due to a recently discovered algorithm of Pollard[4] which efficiently solves quadratic equations $s_1^2 + k\, s_2^2 = m \pmod n$. Pollard's method does not solve general polynomial equations modulo n nor does it extend to systems of polynomial equations.

3. THE BINARY QUADRATIC SCHEME OVER ALGEBRAIC INTEGERS

The binary quadratic scheme may still yield a good signature scheme if we replace rational integers x_1, x_2, s_1, s_2, m by algebraic integers X_1, X_2, S_1, S_2, M which range over the set

$$\mathbb{Z}_{n,d} := \{a + b\sqrt{d} \mid a,b \in \mathbb{Z} \ , \ 0 \le a,b < n\} \ .$$

The set $\mathbb{Z}_{n,d}$ can play a similar role as the set $\mathbb{Z}_n$ of integers modulo n. There is a natural way of adding and multiplying elements in $\mathbb{Z}_{n,d}$:

$$(a' + b'\sqrt{d}) + (a'' + b''\sqrt{d}) := a + b\sqrt{d}$$
$$\text{with} \quad a := a' + a'' \pmod n \ , \ b := b' + b'' \pmod n$$

$$(a' + b'\sqrt{d})(a'' + b''\sqrt{d}) = a + b\sqrt{d}$$
$$\text{with} \quad a := a'a'' + db'b'' \pmod n \ , \ b := a'b'' + a''b' \pmod n .$$

So all arithmetic operations in $\mathbb{Z}_{n,d}$ are done modulo $n\,\mathbb{Z}[\sqrt{d}]$ and in standard algebraic notation $\mathbb{Z}_{n,d}$ is the ring $\mathbb{Z}[\sqrt{d}]/n\,\mathbb{Z}[\sqrt{d}]$. An element $a + b\sqrt{d}$ is invertible iff $a^2 - b^2 d \in \mathbb{Z}_n^*$, and in this case

$$(a + b\sqrt{d})^{-1} = a' - b'\sqrt{n}$$

with $a' = a(a^2 - b^2 d)^{-1} \pmod n$, $b' = b(a^2 - b^2 d)^{-1} \pmod n$. Let $\mathbb{Z}^*_{n,d} \subset \mathbb{Z}_{n,d}$ be the subgroup of invertible elements.

LEMMA 2 With the above arithmetic operations $\mathbb{Z}_{n,d}$ forms a commutative ring with $\mathbb{Z}_n \subset \mathbb{Z}_{n,d}$, $\mathbb{Z}^*_n \subset \mathbb{Z}^*_{n,d}$.

In the sequel we let the variables X_1, X_2, S_1, S_2, M range over $\mathbb{Z}_{n,d}$. For an arbitrary $u \in \mathbb{Z}^\bullet_n$ the substitution

$$X_1 := S_1 - u^{-1} S_2 \qquad (7)$$

$$X_2 := S_1 + u^{-1} S_2$$

yields $X_1 X_2 = S_1^2 + k S_2^2$ with $k := -u^{-2} \pmod n$. So given u the equation

$$X_1 X_2 = M \qquad (8)$$

which can easily be solved for any $M \in \mathbb{Z}_{n,d}$, is equivalent to the less trivial equation

$$S_1^2 + k S_2^2 = M . \qquad (9)$$

This observation yields an efficient signature scheme. For key generation Alice picks a random element $u \in \mathbb{Z}^*_n$ publishes $k := -u^{-2} \pmod n$, and keeps u secret. For any M Alice can easily solve the equation (9). She picks $X_1 \in \mathbb{Z}^*_{n,d}$ at random, computes $X_2 := M X_1^{-1}$ and inverts the linear substitution (7)

$$S_1 := (X_2 + X_1)/2$$

$$S_2 := (X_2 - X_1) u/2 .$$

Once k is published, Bob (or anyone else) cannot compute u, and cannot follow the method of solving equation (9) that Alice is using.

For convenience we write polynomial equations over $\mathbb{Z}_{n,d}$ as systems of polynomial equations over $\mathbb{Z}_n$. Let

$$X_i = x_{i1} + \sqrt{d}\, x_{i2} \qquad i = 1,2$$

$$S_i = s_{i1} + \sqrt{d}\, s_{i2} \qquad i = 1,2$$

$$M = m_1 + \sqrt{d}\, m_2$$

with x_{ij}, s_{ij}, $m_i \in \mathbb{Z}_n$. The equation $X_1 \cdot X_2 = M$ can be written as

$$x_{11} x_{21} + d x_{12} x_{22} = m_1 \pmod n \qquad (10)$$

$$x_{11} x_{22} + x_{12} x_{21} = m_2 \pmod n .$$

The equation $S_1^2 + k S_2^2 = M$ can be written as

$$s_{11}^2 + d s_{12}^2 + k(s_{21}^2 + d s_{22}^2) = m_1 \pmod n \qquad (11)$$

$$2(s_{11} s_{22} + k s_{12} s_{21}) = m_2 \pmod n .$$

Elimination of s_{11} in the latter equation yields

$$s_{11} = (m_2 - 2k\, s_{12}\, s_{21})/(2\, s_{22}) \pmod n$$

Therefore the system of equations (11) is equivalent to the ternary, quadric equation (12) provided that $s_{22} \in \mathbb{Z}_n^*$:

$$(12) \qquad (m_2 - 2k\, s_{12} s_{21})^2 + 4\, s_{22}^2 (d\, s_{12}^2 + k(s_{21}^2 + d\, s_{22}^2) - m_1) = 0 \pmod n \ .$$

So this equation can be taken as verification condition for the binary quadratic signature scheme over $\mathbb{Z}_{n,d}$.

The signature scheme based on equation (12) consists of the following components:

<u>Key generation</u>

1. choose two random primes p,q so that p·q is difficult to factor, put $n := p \cdot q$.
2. pick random integers u,d which are relatively prime to n.
3. publish $k := -u^{-2} \pmod n$, d, n, and keep u secret.

<u>Messages</u> are pairs (m_1, m_2) of integers in the range $0 < m_1, m_2 < n$, i.e. $m_1, m_2 \in \mathbb{Z}_n - 0$.

<u>Signature verification</u>

A triple (s_{12}, s_{21}, s_{22}) of integers in $\mathbb{Z}_n$ is a valid signature for the message (m_1, m_2) if it satisfies the equation (12)

$$(m_2 - 2k\, s_{12} s_{21})^2 + 4\, s_{22}^2 (d\, s_{12}^2 + k(s_{21}^2 + d\, s_{22}^2) - m_1) = 0 \pmod n \ .$$

This equation can easily be checked using k,d,n with 10 multiplications, 4 additions/subtractions modulo n. We do not count the trivial multiplication by 4.

<u>Signature generation</u>

(We solve the easy system (10), and using the private key u we transform its solution into a solution of (12) by inverting the linear substitution (7).)

1. pick random elements $x_{11}, x_{12} \in \mathbb{Z}_n$ so that $x_{11}^2 - d\, x_{12}^2$ is relatively prime to n.
2. $x_{22} := \dfrac{m_2 x_{11} - m_1 x_{12}}{x_{11}^2 - d\, x_{12}^2} \pmod n$,
3. $x_{21} := \dfrac{(m_1 x_{11} - d\, m_2 x_{22})}{x_{11}^2 - d\, x_{12}^2} \pmod n$
4. $s_{12} := (x_{22} + x_{12})/2 \pmod n$

   $s_{21} := (x_{21} - x_{11})\, u/2 \pmod n$

   $s_{22} := (x_{22} - x_{12})\, u/2 \pmod n$

<u>LEMMA 3</u> Signature generation can be done with 9 multiplications, 1 division modulo n. (The division by 2 is trivial).

PROOF Compute x_{11}^2, $d\,x_{12}$, $d\,x_{12}^2$, $d\,m_2 x_{12}$ with only 4 multiplications modulo n. Obviously the rest of the computation can be done with 5 multiplications and 1 division modulo n. Q.E.D.

For a message (m_1,m_2) let $M := m_1 + m_2\sqrt{d}$ be the corresponding element in $\mathbb{Z}_{n,d}$, obviously $m_1^2 - d\,m_2^2 \in \mathbb{Z}_n^*$ iff $M \in \mathbb{Z}_{n,d}^*$. For messages (m_1,m_2) with $m_1^2 - d\,m_2^2 \in \mathbb{Z}_n^*$ the above signature procedure generates arbitrary signatures of (m_1,m_2) with uniform probability distribution.

LEMMA 4 For every message (m_1,m_2) with $m_1 + m_2\sqrt{d} \in \mathbb{Z}_{n,d}^*$ the set of signatures of (m_1,m_2) is in 1-1 correspondence with the set of values (s_{12},s_{21},s_{22}) in step 4, as $x_{11} + x_{12}\sqrt{d}$ ranges over $\mathbb{Z}_{n,d}^*$.

PROOF The set of signatures of (m_1,m_2) is in 1-1 correspondence to the set of solutions (S_1,S_2) of $S_1^2 + k\,S_2^2 = M$. By the linear transformation (7) the set of solutions (S_1,S_2) of $S_1^2 + k\,S_2^2 = M$ is in 1-1 correspondence to the set of solutions (X_1,X_2) of $X_1 \cdot X_2 = M$. Since $M \in \mathbb{Z}_{n,d}^*$ these solutions are in 1-1 correspondence with the set of elements $X_1 \in \mathbb{Z}_{n,d}^*$ (remember that $X_1 = x_{11} + x_{12}\sqrt{d} \in \mathbb{Z}_{n,d}^*$ iff $x_{11}^2 - d\,x_{12}^2$ is relatively prime to n). Q.E.D.

As a consequence of Lemma 4 messages (m_1,m_2) for which $m_1^2 - d\,m_2^2$ is not relatively prime to n should be avoided. We have excluded messages with $m_1 = 0$ or $m_2 = 0$ anyway, see remark 7 (iv),(v). No other message (m_1,m_2) with $\gcd(m_1^2 - d\,m_2^2, n) \neq 1, n$ is likely to occur.

REMARKS 5 The characteristical properties of the original binary quadratic OSS-scheme remain intact: i) The generation of the keys u, $k := -u^2 \pmod n$, d can be done without knowing the factorization of n. All public keys may share the (d,n)-part provided that the factorization of n is unknown to all participants of the system. ii) Computing the private key u from the public key k,n requires to compute $\sqrt{-k} \pmod n$, and thus is as hard as factoring n. iii) The signature scheme is multiplicative over $\mathbb{Z}_{n,d}$. Solutions S_1', S_2' and S_1'', S_2'' of

$$S_1'^2 + k\,S_2'^2 = M' \quad , \quad S_1''^2 + k\,S_2''^2 = M''$$

yield a solution S_1, S_2 of $S_1^2 + k\,S_2^2 = M'M''$ as

$$S_1 = S_1'S_1'' - k\,S_2'S_2'' \quad , \quad S_2 = S_1'S_1'' - k\,S_2'S_2''$$

iv) The roles of k,M in the equation $S_1^2 + k\,S_2^2 = M$ can be interchanged since $S_1^2 + k\,S_2^2 = M$ is equivalent to $(S_1/S_2)^2 - M\,S_2^{-2} = -k$.

With these remarks the following theorem can be proved in the same way as its counterpart in [3].

THEOREM 6 Any algorithm for computing u from random signatures of messages of its choice can be transformed into a probabilistic factoring algorithm with similar complexity.

PROOF see proof of theorem 2 [3].

REMARKS 7 i) The theorem can easily be extended to the case of an algorithm that succeeds for only some of the u-values provided that the fraction of these u-values is non negligible. ii) In Rabin's signature scheme an opponent can factor n by analysing the signature of specific messages. In our scheme the factorization of n and the secret parameter u cannot be revealed by chosen message attacks. iii) If Bob could compute one of the x_{ij}-values $i,j \in \{0,1\}$ corresponding to a signature s_{ij} $i,j \in \{0,1\}$, he could compute u. For instance given x_{11}, s_{11} and s_{21}, Bob can compute u from $x_{11} = s_{11} - u^{-1} s_{21} \pmod n$. A single x_{ij}-value is thus as hard to compute as u. iv) Messages (m_1, m_2) with $m_2 = 0$ can be signed without the private key u. It is sufficient to solve

$$s_{11}^2 + k\, s_{21}^2 = m_1 \pmod n$$

by Pollard's algorithm [4]. v) Messages (m_1, m_2) with $m_1 = 0$ can also be signed without the private key u. This easily follows from (iii) and the multiplicativity of the scheme (remark 5, iii).

THE COMPLEXITY OF SOLVING $S_1^2 + k\,S_2^2 = M$ over $\mathbb{Z}_{n,d}$

Pollard [4] solves the equation $s_1^2 + k\,s_2^2 = m \pmod n$ by successively reducing m and k. He reduces m to $m' \le \sqrt{k}$, interchanges m and k, and continues until both m and k are 1. His basic reduction step uses the euclidean algorithm over $\mathbb{Z}$.

Pollard's method does not solve $S_1^2 + k\,S_2^2 = M$ since $\mathbb{Z}[\sqrt{d}]$ is not euclidean domain provided that $d > 73$ or $d < -11$. In particular there exist $A, B \in \mathbb{Z}[\sqrt{d}]$ such that $|N(A - C\cdot B)| > |N(B)|$ for all $C \in \mathbb{Z}[\sqrt{d}]$, (where N is the norm, $N(x + \sqrt{d}\,y) = x^2 - d\,y^2$). It is unlikely that the missing euclidean algorithm for $\mathbb{Z}[\sqrt{d}]$ can be replaced by some other norm reducing procedure. For large $|d|$ almost all elements $A \in \mathbb{Z}[\sqrt{d}]$ with $|N(A)| << d$ are rational integers and these are unlikely to appear in a general procedure over $\mathbb{Z}[\sqrt{d}]$.

The methods for solving $s_1^2 + k\,s_2^2 = m \pmod n$ which use the class group of quadratic form with discriminant -4k, see [3], do not solve $S_1^2 + k\,S_2^2 = M$. The reason is that equivalence classes of quadratic forms with coefficients in $\mathbb{Z}[\sqrt{d}]$ cannot be represented in a canonical way by reduced forms.

The fastest known method for solving $S_1^2 + k\,S_2^2 = M$ is by factoring n. This method becomes infeasible if n is at least 600 bits long.

The complexity of solving general polynomial equations modulo n is an open problem and it may become an important subject for further cryptographic research.

REFERENCES

1. Diffie, W. and Hellman, M.: New Directions in Cryptography. IEEE, IT-22, (1976), 644-654.
2. Ong, H. and Schnorr, C.P.: Signatures Through Approximate Representations by Quadratic Forms. Advances in Cryptology: Proceedings of Crypto 83. Plenum Publ. New York 1984, 117-132.
3. Ong, H., Schnorr, C.P., and Shamir, A.: An Efficient Signature Scheme Based on Quadratic Equations. Proceedings of 16th ACM-Symp. of Theory of Computing, Washington (1984), p. 208-216.
4. Pollard, J.M.: Solution of $x^2 + k y^2 \equiv m \pmod{n}$, with Application to Digital Signatures. Preprint 1984.
5. Rabin, M.O.: Probabilistic Algorithms in Finite Fields. SIAM J. on Computing 9 (1980), p. 273-280.
6. Rivest, R.L., Shamir, A. and Adleman, L.: A Method for Obtaining Digital Signatures and Public Key Cryptosystems. Comm. ACM 21 (1978) 120-126.
7. Shamir, A.: Identity Based Cryptosystems & Signature Schemes. Proceedings of Crypto 84.