Breaking the Ong-Schnorr-Shamir Signature Scheme for Quadratic Number Fields

Dennis Estes [1]
Leonard M. Adleman [2](*)
Kireeti Kompella [2]
Kevin S. McCurley [1]
Gary L. Miller [2]

[1] Department of Mathematics
University of Southern California
Los Angeles, CA 90089-1113

[2] Department of Computer Science
University of Southern California
Los Angeles, CA 90089-0782

1. Introduction

Recently Ong, Schnorr, and Shamir [OSS1, OSS2] have presented new public key signature schemes based on quadratic equations. We will refer to these as the OSS schemes. The security of the schemes rest in part on the difficulty of finding solutions to

$$X^2 - KY^2 \equiv M \pmod{n}, \qquad (1)$$

where n is the product of two large rational primes. In the original OSS scheme [OSS1], K, M, X, and Y were to be rational integers. However, when this version succumbed to an attack by Pollard [PS,S1], a new version was introduced [OSS2], where M, X, and Y were to be quadratic integers, i. e. elements of the ring $Z[\sqrt{d}]$. In this paper we will show that the OSS system in $Z[\sqrt{d}]$ is also breakable. The method by which we do this is to

*Research sponsored by NSF Grant #53-4510-2651

H.C. Williams (Ed.): Advances in Cryptology - CRYPTO '85, LNCS 218, pp. 3-13, 1986.

reduce the problem of solving the congruence over the ring $\mathbf{Z}[\sqrt{d}]$ to the problem of solving the congruence over the integers, for which we can use Pollard's algorithm.

The OSS signature scheme described in [OSS2] was intended to provide a method by which a person can sign messages with the assurance that no one, including the receiver, can forge the signature, and so that anyone can easily verify the validity of both the signature and the message. It works as follows: Party A generates two rational primes p and q each about 300 bits long, using the same care as in the key generation for RSA to ensure that $n = pq$ cannot be easily factored by known methods. Party A also chooses random integers d, t_0, and t_1 such that $(n, t_0^2 - dt_1^2) = 1$, and publishes n, d, and $K \equiv (t_0 + t_1\sqrt{d})^2 \pmod{n}$, keeping t_0, t_1, p, and q secret. (In [OSS2], they took $K \in \mathbf{Z}$, but we will show that the scheme is insecure with $K \in \mathbf{Z}[\sqrt{d}]$). The messages consist of pairs of integers (M_0, M_1) from the interval $[1,n)$. In order to sign a message, party A uses the secret key (t_0, t_1) to construct a solution to the congruence $X^2 - KY^2 \equiv M_0 + M_1\sqrt{d} \pmod{n}$. The receiver of the message can easily verify that the message was the one signed by party A. In order for the scheme to be secure, the receiver should have some assurance that no one can forge the signature without knowledge of the secret key K.

It was presumed that it would be hard to solve congruence (1) without knowing the secret keys, in part because $\mathbf{Z}[\sqrt{d}]$ is not in general a Euclidean domain, and Pollard's algorithm resembles the Euclidean algorithm in some ways. In this paper we will show that the problem of solving (1) over $\mathbf{Z}[\sqrt{d}]$ can be reduced to the problem of solving (1) over $\mathbf{Z}$. Pollard's algorithm can then be used to solve the problem over $\mathbf{Z}$, giving then also a solution over $\mathbf{Z}[\sqrt{d}]$. Because we use Pollard's algorithm, the method constructs a solution to the congruence without necessarily producing the secret keys. The most general OSS scheme was based on a polynomial congruence modulo a composite integer. Even though both of the quadratic OSS shemes have now been broken, it remains an open question whether the most general form of the OSS scheme can be broken.

In this paper, when we write $x_0 + x_1\sqrt{d} \equiv y_0 + y_1\sqrt{d} \pmod{n}$ we mean that $x_0 \equiv y_0 \pmod{n}$ and $x_1 \equiv y_1 \pmod{n}$. In general, an element X of the ring $\mathbf{Z}[\sqrt{d}]$ will be written as $X = X_0 + X_1\sqrt{d}$, and we will write N(X) for the norm of X, namely

$N(X) = X_0^2 - dX_1^2$. If $X \in \mathbf{Z}[\sqrt{d}]$ and $(N(X),n) = 1$, then X is invertible modulo n, and we write $\overline{X} = \overline{X}_0 + \overline{X}_1\sqrt{d}$ for the inverse. Note that

$$\overline{X}_0 \equiv X_0 N(X)^{-1} \pmod{n},$$

$$\overline{X}_1 \equiv -X_1 N(X)^{-1} \pmod{n},$$

and these can be calculated using the Euclidean algorithm, even though $\mathbf{Z}[\sqrt{d}]$ may not be a Euclidean domain.

To begin, we consider four computational problems:

Problem I

INPUT $A, M, n \in \mathbf{Z}$, $(n,A) = (n,M) = 1$.

OUTPUT $X, Y \in \mathbf{Z}$ such that $X^2 + AY^2 \equiv M \pmod{n}$.

Problem II

INPUT $A, B, C, M, n \in \mathbf{Z}$, $n \nmid A$, $n \nmid B$, $n \nmid (C^2 - AB)$, $n \nmid M$.

OUTPUT Either a) or b):

a) $X, Y \in \mathbf{Z}$ such that $AX^2 + BY^2 + 2CXY \equiv M \pmod{n}$.

b) $m \in \mathbf{Z}$ such that $1 < m < n$ and $m \mid n$.

Problem III

INPUT $d, n \in \mathbf{Z}$, $K = K_0 + K_1\sqrt{d}$, $M = M_0 + M_1\sqrt{d}$, $(N(KM),n) = 1$, $n \nmid M_1$, $n \nmid d$.

OUTPUT Either a), b), or c):

a) $X, Y \in \mathbf{Z}[\sqrt{d}]$, $c \in \mathbf{Z}$ such that $(c,n) = 1$ and $X^2 - KY^2 \equiv cM \pmod{n}$.

b) $m \in \mathbf{Z}$ such that $1 < m < n$ and $m \mid n$.

c) $S \in \mathbf{Z}[\sqrt{d}]$ such that $S^2 \equiv K \pmod{n}$.

Problem IV

INPUT: $d, n \in \mathbf{Z}$, $K, M \in \mathbf{Z}[\sqrt{d}]$, $(N(KM),n) = 1$.

OUTPUT: $X, Y \in \mathbf{Z}[\sqrt{d}]$ with $X^2 - KY^2 \equiv M \pmod{n}$.

We shall prove:

Theorem 1 Problem II is solvable in polynomial time with an oracle for Problem I.

Theorem 2 Problem III is solvable in polynomial time with an oracle for Problem II.

Theorem 3 Problem IV is solvable in polynomial time with an oracle for Problem III.

The security of the original OSS scheme was based on the difficulty of solving Problem I when $N = pq$, the product of two large primes. Pollard produced an algorithm for Problem I which is believed to run in deterministic polynomial time, and as a consequence was able to break the original OSS signature scheme. The details of his algorithm should appear in a joint paper of Pollard and Schnorr [PS], where they will prove under the assumption of an extended Riemann hypothesis that Pollard's algorithm for Problem I runs in random polynomial time. It should be mentioned that in this paper they also outline a method similar to ours for solving Problem III, having made this discovery independently of the authors.

Three of the authors (A., E. and Mc.) have recently discovered a variation of Pollard's algorithm that allows us to prove that Problem I is solvable in random polynomial time, removing the assumption of the extended Riemann hypothesis in Pollard and Schnorr's result. Our variation of Pollard's algorithm is not a practical procedure for breaking the OSS scheme, but it has the advantage that one can rigorously analyse its running time without any hypothesis. The details of this will appear in a later paper.

As a consequence of these results, it follows that Problem IV is solvable in random polynomial time, and therefore that the OSS signature scheme over $Z[\sqrt{d}]$ is insecure.

Several remarks are in order here before we proceed.

1. The assumption that $(N(KM),n) = 1$ is made primarily for convenience. In the OSS signature scheme, n was taken to be the product of two large primes, and the scheme is compromised if the factorization of n can be discovered. Therefore values of $N(K)$ and $N(M)$ which have a nontrivial factor in common with n are not

of any interest.

2. If n is not squarefree, then our algorithm for solving (1) may not work if $1 < (N(M),n) < n$. The reason for this is illustrated by the example $n = t^2$, where t is composite, and $t|M$. Our algorithm might detect the factorization $n = t^2$, and try to use Hensel's lemma to construct a solution modulo t^2 from a solution modulo t. Modulo t, however, the congruence reduces to $X^2 - KY^2 \equiv 0 \pmod t$. Without knowing the factorization of t, the only solution we can construct in this case is the trivial one with $X = Y = 0$, and this solution will not work in Hensel's lemma. In fact, Rabin [R] has observed that any algorithm which produces solutions to $X^2 - KY^2 \equiv 0 \pmod n$ can be used as a probabalistic algorithm for factoring n. This provides a reason for believing that (1) may be hard to solve if $(N(KM),n) > 1$.

3. In the OSS scheme, K was assumed to be a square modulo n, and part of the secret key used to sign messages was $\sqrt{K} \pmod n$. It turns out that this information is not necessary for signing messages in polynomial time.

4. If n is odd, then a solution to (1) exists if $(N(KM),n) = 1$. If n is even, then not all messages M can be signed, even if $(N(KM),n) = 1$. In particular the message $M = \sqrt{d}$ is not signable if K_1 is even, (where $K = K_0 + K_1\sqrt{d}$), so $\sqrt{d}$ is not signable if K is a square. Our method will produce a solution to (1) if such a solution exists.

2. Proof of Theorem 1

The proof of Theorem 1 is elementary, requiring only that we complete the square. To begin, if $1 < (A,n)$ or $1 < (B,n)$ or $1 < (M,n)$ or $1 < (C,n) < n$, then the Euclidean algorithm will produce a nontrivial factor of n. If $1 = (A,n) = (B,n) = (M,n)$ and $n|C$, then solving the congruence in question is equivalent to solving

$$X^2 + BA^{-1}Y^2 \equiv MA^{-1} \pmod n.$$

An oracle for Problem I now produces a solution. The only case remaining is if $(A,n) = (B,n) = (C,n) = (M,n) = 1$. By completing the square we get

$$(X+CA^{-1}Y)^2 - [(CA^{-1})^2 - (BA^{-1})]Y^2 \equiv MA^{-1} \pmod{n}. \qquad (2)$$

Substituting $Z = Y$ and $W = X + CA^{-1}Y$ gives

$$W^2 - [(CA^{-1})^2 - (BA^{-1})]Z^2 \equiv MA^{-1} \pmod{n}. \qquad (3)$$

By assumption $n \nmid [(CA^{-1})^2-(BA^{-1})]$, so either (n, C^2-AB) gives a nontrivial factor of n or else an oracle for Problem I produces a solution W, Z to (3). In the latter case, $Y = Z$ and $X = W - CA^{-1}Y$ is a solution to the original congruence.

3. Proof of Theorem 2

If $1 < (M_1, n) < n$, then the Euclidean algorithm gives a nontrivial factor of n, so we may assume that $(M_1,n) = 1$. Since $(N(M),n) = 1$, it follows that M is invertible modulo n, and we can use the Euclidean algorithm to calculate $\overline{M} = \overline{M}_0+\overline{M}_1\sqrt{d}$ such that $M\overline{M} \equiv 1 \pmod{n}$. If we now want to solve

$$(X_0+X_1\sqrt{d})^2 - (K_0+K_1\sqrt{d})(Y_0+Y_1\sqrt{d})^2 \equiv M_0+M_1\sqrt{d} \pmod{n}, \qquad (4)$$

then it suffices to solve

$$(\overline{M}_0+\overline{M}_1\sqrt{d})(X_0+X_1\sqrt{d})^2 - (r_0+r_1\sqrt{d})(Y_0+Y_1\sqrt{d})^2 \equiv 1 \pmod{n}, \qquad (5)$$

where $r_0+r_1\sqrt{d} = (\overline{M}_0+\overline{M}_1\sqrt{d})(K_0+K_1\sqrt{d})$. Setting $Y_0 = 1$ and $Y_1 = 0$, the left hand side of the congruence (5) becomes

$$\{\overline{M}_0X_0^2 + \overline{M}_0dX_1^2 + 2d\overline{M}_1X_0X_1 - r_0\} + \{\overline{M}_1X_0^2 + \overline{M}_1dX_1^2 + 2\overline{M}_0X_0X_1 - r_1\}\sqrt{d}.$$

By our assumptions we have that $(\overline{M}_1,n) = 1$, $n \nmid \overline{M}_1 d$, and $n \nmid (\overline{M}_0^2 - d\overline{M}_1^2)$. Therefore using an oracle for Problem II, we either get a nontrivial factor of n or a solution X_0, X_1 to the congruence

$$\overline{M}_1 X_0^2 + \overline{M}_1 d X_1^2 + 2\overline{M}_0 X_0 X_1 \equiv r_1 \pmod{n}.$$

Let $c = \overline{M}_0 X_0^2 + \overline{M}_0 d X_1^2 + 2d\overline{M}_1 X_0 X_1 - r_0$. If $(n,c) = 1$, then

$$(X_0 + X_1\sqrt{d})^2 - (K_0 + K_1\sqrt{d})(1 + 0\sqrt{d}) \equiv c(M_0 + M_1\sqrt{d}) \pmod{n},$$

giving an output of type a). If $n|c$, then

$$(X_0 + X_1\sqrt{d})^2 \equiv K_0 + K_1\sqrt{d} \pmod{n},$$

giving an output of type c). If $1 < (n,c) < n$, then we get a nontrivial factor of n.

4. Proof of Theorem 3

We now show how to solve Problem IV, i.e. how to find solutions of the congruence

$$(X_0 + X_1\sqrt{d})^2 - (K_0 + K_1\sqrt{d})(Y_0 + Y_1\sqrt{d})^2 \equiv M_0 + M_1\sqrt{d} \pmod{n}, \qquad (6)$$

where $(N(KM),n) = 1$. The method uses two appeals to an oracle for Problem III, essentially to replace K and M by integers. There are however several possible outputs from Problem III, and we must show how to solve the congruence (6) in each case.

Let us first observe that if $n = 2^a n_1$, where n_1 is odd, then it suffices to solve the congruence separately modulo 2^a and modulo n_1, since we can then use the Chinese Remainder Theorem to construct a solution modulo n. In order to construct a solution modulo 2^a when $a \leq 3$, we can simply try all of the finite number of possible values for X and Y.

If $a > 3$, then we first construct a solution modulo 8 (if it exists). We will now show how to use Hensel's lemma to lift the solution modulo 8 to a solution modulo 2^a. Let $X = X_0 + X_1\sqrt{d}$ and $Y = Y_0 + Y_1\sqrt{d}$ be a solution of the congruence $X^2 - KY^2 \equiv M \pmod{2^b}$, where $b \geq 3$. We want to choose $Z, W \in \mathbb{Z}[\sqrt{d}]$ such that

$$(X + 2^{b-1}Z)^2 - K(Y + 2^{b-1}W)^2 \equiv M \pmod{2^{b+1}}. \qquad (7)$$

Since $b \geq 3$, this is equivalent to

$$X^2 - KY^2 - M + 2^b(XZ - KYW) \equiv 0 \pmod{2^{b+1}}.$$

Let $X^2 - KY^2 - M = 2^bR$, with $R \in \mathbb{Z}[\sqrt{d}]$. Then it suffices to find Z and W satisfying

$$XZ - KYW \equiv -R \pmod 2. \qquad (8)$$

Since $(N(KM),2) = 1$, $X^2 \equiv N(X) \pmod 2$ and $Y^2 \equiv N(Y) \pmod 2$, it follows that either $(N(X),2) = 1$ or $(N(KY),2) = 1$, so that either X or KY is invertible modulo 2. If X is invertible modulo 2, then a solution of (8) is given by $Z = -\overline{X}R$ and $W = 0$. If KY is invertible, then we take $Z = 0$ and $W = \overline{KY}R$. Since (7) is solvable, we can lift the solution modulo 2^b to a solution modulo 2^{b+1}.

It now suffices to show how to solve the congruence (6) in the case n is odd. Consider first the case that $n|d$. In this case (6) reduces to the system of congruences

$$X_0^2 - K_0Y_0^2 \equiv M_0 \pmod n$$
$$2X_0X_1 - K_1Y_0^2 - 2K_0Y_0Y_1 \equiv M_1 \pmod n.$$

Since $(N(K),n) = 1$ and $(N(M),n) = 1$, it follows that $(K_0,n) = 1$ and $(M_0,n) = 1$, so that an oracle for Problem I will produce a solution X_0, Y_0 to the first of these congruences. Furthermore, at least one of $2X_0$ and $2K_0Y_0$ will be relatively prime to n since $(M_0,n) = 1$ and n is odd. Hence the second congruence above can be solved using the

Euclidean algorithm.

Next we consider the case n odd and $n \nmid d$. One of the possible outputs from Problem III is a factorization $n = n_1 n_2$. If $(n_1, n_2) = 1$, then we can solve the congruences $X^2 - KY^2 \equiv M \pmod{n_i}$ and combine the results with the Chinese Remainder Theorem to get a solution of (6). This splitting procedure will be required at most $O(\log n)$ times. If in the factorization $n = n_1 n_2$ we have $(n_1, n_2) > 1$, then let $G = (n_1, n_2)$, $n = G^2 H_1$, and $G_1 = (G, H_1)$. If $G_1 = 1$ and $H_1 \neq 1$, then we have a relatively prime factorization and can use the Chinese Remainder Theorem. If $G_1 > 1$, then write $n = G^2 G_1 H_2$, and let $G_2 = (G, H_2)$. Continuing in this manner, since the H_i's are decreasing, we either arrive at a value $H_j = 1$, or else we find $G_i = 1$ which produces a relatively prime factorization of n. If $H_j = 1$, then it is easy to see that $p \mid n$ if and only if $p \mid G$. Hence we can run the algorithm with n replaced by G, and later use Hensel's Lemma to construct a solution modulo a sufficiently large power of G that is divisible by n. It should be remarked that the computations required to apply both Hensel's Lemma and the Chinese Remainder Theorem can be carried out in deterministic polynomial time.

Another possible output from Problem III is a square root of K modulo n. If we know $S \in \mathbb{Z}[\sqrt{d}]$ with $S^2 \equiv K \pmod{n}$, then as in [OSS2] we get the factorization $X^2 - KY^2 \equiv (X - SY)(X + SY)$. It then suffices to solve the linear system

$$X - SY \equiv 1 \pmod{n},$$
$$X + SY \equiv M \pmod{n}.$$

Notice that S is invertible mod n, and also that 2 is invertible mod n since we have assumed that n is odd. Hence the solution to the linear system is provided by

$$X \equiv (M+1)/2 \pmod{n}$$
$$Y \equiv (M-1)/(2S) \pmod{n}.$$

We may now disregard the cases in which the output from the oracle for Problem III is not of type a). The first step in solving (6) is to reduce to solving

$$(X + Y\sqrt{d})^2 - (K_0 + K_1\sqrt{d})(W + Z\sqrt{d})^2 \equiv c \pmod{n}, \qquad (9)$$

where $c \in \mathbf{Z}$. If $n|M_1$, then (6) is already in the desired form. If $n \nmid M_1$, then use an oracle for Problem III to obtain $X_0, X_1, Y_0, Y_1, c \in \mathbf{Z}$, such that $(c,n) = 1$ and

$$(X_0+X_1\sqrt{d})^2 - (K_0+K_1\sqrt{d})(Y_0+Y_1\sqrt{d})^2 \equiv c(M_0+M_1\sqrt{d}) \pmod{n}.$$

(The procedure if the oracle returns a type b) or c) output has already been dealt with.) Using an idea from Pollard's original algorithm (see [S] or [PS]) it is now enough to solve (9), since we can use the composition of binary quadratic forms to construct a solution to (6). By the observation of Lenstra (see [OSS2]), the roles of K and c are interchangeable, so to solve (9) it suffices to solve

$$(X + Y\sqrt{d})^2 - c(W + Z\sqrt{d})^2 \equiv (K_0 + K_1\sqrt{d}) \pmod{n}. \qquad (10)$$

By the same reasoning that led us to the problem of solving (9), we can use an oracle for Problem III in order to reduce (10) to the problem of solving

$$(X + Y\sqrt{d})^2 - c(W + Z\sqrt{d})^2 \equiv b \pmod{n}, \qquad (11)$$

where $b \in \mathbf{Z}$ satisfies $(b,n) = 1$. Finally we use an oracle for Problem I to solve (11) over the rationals.

References

OSS1 H. Ong, C. P. Schnorr, and A. Shamir, "An Efficient Signature Scheme Based on Quadratic Equations," Proc. 16th ACM Symp. Theor. Comput. (1984) 208-216.

OSS2 H. Ong, C. P. Schnorr, and A. Shamir, "Efficient Signature Schemes based on Polynomial Equations," to appear in Crypto 84, Lecture Notes in Computer Science, Springer-Verlag, N. Y., 1984.

PS J. M. Pollard and C.-P. Schnorr, "Solution of $x^2 + ky^2 \equiv m \pmod{n}$, with applications to digital signatures", preprint, 1985.

SI J. Shallit, "An Exposition of Pollard's Algorithm for Quadratic Congruences," Technical Report 84-006, Department of Computer Science, University of Chicago, Dec. 1984.

R M. O. Rabin, "Digitalized signatures and public-key functions as intractable as factorization," M.I.T. Laboratory for Computer Science, Technical report LCS/TR-212, 1979.