Fast irreducibility and subgroup membership testing in XTR

Arjen K. Lenstra[1] and Eric R. Verheul[2]

[1] Citibank, N.A., Technical University Eindhoven, 1 North Gate Road, Mendham, NJ 07945-3104, U.S.A.
arjen.lenstra@citicorp.com
[2] PricewaterhouseCoopers, GRMS Crypto Group, Goudsbloemstraat 14, 5644 KE Eindhoven, The Netherlands
Eric.Verheul@[nl.pwcglobal.com, pobox.com]

Abstract. We describe a new general method to perform part of the set-up stage of the XTR system introduced at Crypto 2000, namely finding the trace of a generator of the XTR group. Our method is substantially faster than the general method presented at Asiacrypt 2000. As a side result, we obtain an efficient method to test subgroup membership when using XTR.

1 Introduction

XTR is an efficient and compact method to work with order p^2-p+1 subgroups of the multiplicative group $\mathrm{GF}(p^6)^*$ of the finite field $\mathrm{GF}(p^6)$. It was introduced at Crypto 2000 (cf. [4]), followed by several practical improvements a Asiacrypt 2000 (cf. [5]). In this paper we present some further improvements of the methods from [4] and [5]. Given the rapidly growing interest in XTR our new methods are of immediate practical importance.

Let p and q be primes such that $p \equiv 2 \bmod 3$ and q divides p^2-p+1, let g be a generator of the order q subgroup of $\mathrm{GF}(p^6)^*$, and let $Tr(g) = g + g^{p^2} + g^{p^4} \in \mathrm{GF}(p^2)$ be the trace over $\mathrm{GF}(p^2)$ of g. In [4] it is shown that the conjugates over $\mathrm{GF}(p^2)$ of elements of the XTR group $\langle g \rangle$ can conveniently be represented by their trace over $\mathrm{GF}(p^2)$, and it is shown how this representation can efficiently be computed given $Tr(g)$.

Given p and q the trace of a generator of the XTR group can be found as follows, as shown in [4]. First one finds a value $c \in \mathrm{GF}(p^2)$ such that $F(c, X) = X^3 - cX^2 + c^pX - 1 \in \mathrm{GF}(p^2)[X]$ is irreducible over $\mathrm{GF}(p^2)$. Given an irreducible $F(c, X)$, there exists an element $h \in \mathrm{GF}(p^6)^*$ of order > 3 and dividing $p^2 - p + 1$ such that $Tr(h) = c$. Actually, h is a root of $F(c, X)$. This implies that $Tr(g)$ can be computed as $Tr(h^{(p^2-p+1)/q})$, assuming that this value is $\neq 3$; if $Tr(h^{(p^2-p+1)/q}) = 3$ another c has be to found such that $F(c, X)$ is irreducible. Because $F(c, X)$ is irreducible for about one third of the c's in $\mathrm{GF}(p^2)$, on average $3q/(q-1)$ different c's have to be tried before a proper c is found.

Thus, for the XTR parameter set-up process one needs to be able to test irreducibility of polynomials of the form $F(c, X) = X^3 - cX^2 + c^pX - 1 \in \mathrm{GF}(p^2)[X]$

for random $c \in \mathrm{GF}(p^2)$. The irreducibility test given in [4] takes $8\log_2(p)$ multiplications in $\mathrm{GF}(p)$; finding an irreducible $F(c,X)$ using this method thus takes an expected $24\log_2(p)$ multiplications in $\mathrm{GF}(p)$. In the follow-up paper [5] a method is described that tests irreducibility of $F(c,X)$ for random $c \in \mathrm{GF}(p^2)$ in $2.4\log_2(p)$ multiplications in $\mathrm{GF}(p)$ on average, so that an irreducible $F(c,X)$ can on average be found in $7.2\log_2(p)$ multiplications in $\mathrm{GF}(p)$. In this paper we present a further refinement of this last method that results in an $F(c,X)$-irreducibility test that takes, on average for random $c \in \mathrm{GF}(p^2)$, only $0.9\log_2(p)$ multiplications in $\mathrm{GF}(p)$. As a result, an irreducible $F(c,X)$ can be found in an expected $2.7\log_2(p)$ multiplications in $\mathrm{GF}(p)$.

The test from [4] takes $8\log_2(p)$ multiplications in $\mathrm{GF}(p)$, irrespective of the outcome. The test from [5], on the other hand, is effectively free for half the c's, and takes $4.8\log_2(p)$ multiplications in $\mathrm{GF}(p)$ for the other half (two thirds of which lead to an irreducible $F(c,X)$). Similarly, the refined test in the present paper is effectively free for half the c's, and takes $1.8\log_2(p)$ multiplications in $\mathrm{GF}(p)$ for the other half. Thus, if during a cryptographic application of XTR a value c is transmitted for which, if the protocol is carried out correctly, $F(c,X)$ is supposed to be irreducible, then the irreducibility of $F(c,X)$ can be verified at the cost of $1.8\log_2(p)$ multiplications in $\mathrm{GF}(p)$ using our new method. This is more than 60% faster than the method from [5] and implies that this verification by the recipient of XTR related values does not cause severe additional overhead. Note that such checks are required because many cryptographic protocols are vulnerable if 'wrong' data are used (cf. [1], [2], [6], [11], and Section 4).

As the irreducibility test from [5] our new irreducibility test is based on Scipione del Ferro's method. Instead of applying it directly to test $F(c,X) \in \mathrm{GF}(p^2)[X]$ for irreducibility, however, we reformulate the problem as an irreducibility problem for a third-degree polynomial $P(c,X) \in \mathrm{GF}(p)[X]$. This is done in Section 2. We then show in Section 3 how the irreducibility of the resulting polynomial $P(c,X)$ can be verified. In Section 4 we discuss subgroup membership testing, and in Section 5 we show how this can be done in XTR. We present a method that is based on the $F(c,X)$-irreducibility test and costs a small amount of additional computation but no additional communication, and another method that takes only a constant number of $\mathrm{GF}(p)$-operations but causes some additional communication overhead.

2 From $F(c,X) \in \mathrm{GF}(p^2)[X]$ to $P(c,X) \in \mathrm{GF}(p)[X]$

Let $c \in \mathrm{GF}(p^2)$ and let $h_j \in \mathrm{GF}(p^6)$ for $j = 0,1,2$ be the roots of $F(c,X) \in \mathrm{GF}(p^2)[X]$. Because $F(c,h_j^{-p}) = 0$ for $j = 0,1,2$ (cf. [4, Lemma 2.3.2.*iv*]) we can distinguish three cases:

I. $h_j = h_j^{-p}$ for $j = 0,1,2$.
II. $h_0 = h_0^{-p}$ and $h_j = h_{3-j}^{-p}$ for $j = 1,2$.
III. $h_j = h_{j+1 \bmod 3}^{-p}$ for $j = 0,1,2$.

In cases I and II we have that $h_j \in \mathrm{GF}(p^2)$ so that $F(c,X)$ is reducible over $\mathrm{GF}(p^2)$. In case III all h_j have order dividing p^2-p+1 and >3 so that $F(c,x)$ is irreducible over $\mathrm{GF}(p^2)$ (cf. [4, Lemma 2.3.2.*vi*]). Thus, if case III can quickly be distinguished from the other two cases, then the irreducibility of $F(c,X)$ can quickly be tested. Actually, we only have to be able to distinguish between cases I and III, because case II can quickly be recognized since it applies if and only if $\Delta \in \mathrm{GF}(p)$ as in [5, Step 2 of Algorithm 3.5] is a quadratic non-residue in $\mathrm{GF}(p)$ (cf. [5, Lemma 3.6]).

Definition 2.1 *Let $G(c,X) = F(c,X) \cdot F(c^p,X)$, and let*

$$P(c,X) = X^3 + (c^p + c)X^2 + (c^{p+1} + c^p + c - 3)X + c^{2p} + c^2 + 2 - 2c^p - 2c.$$

The following lemma describes some of the immediate properties of the polynomials $G(c,X)$ and $P(c,X)$ and their interrelation.

Lemma 2.2 *Both $G(c,X)$ and $P(c,X)$ are in $\mathrm{GF}(p)[X]$. Furthermore $P(c,X)$ can be written as the product $\prod_{j=0}^{2}(X - G_j)$ of three linear polynomials if and only if $G(c,X)$ can be written as the product $\prod_{j=0}^{2}(X^2 + G_jX + 1)$ of three quadratic polynomials, where $G_j \in \mathrm{GF}(p^6)$ for $j = 0,1,2$. In particular, this decomposition of $G(c,X)$ is unique modulo permutation and either all G_j are in $\mathrm{GF}(p^2)$ or all G_j are in $\mathrm{GF}(p^3)$.*

Proof. It follows from Definition 2.1 and a straightforward computation that $G(c,X)$ equals

$$X^6-(c^p+c)X^5+(c^{p+1}+c^p+c)X^4-(c^{2p}+c^2+2)X^3+(c^{p+1}+c^p+c)X^2-(c^p+c)X+1.$$

All coefficients of $G(c,X)$ and $P(c,X)$ equal their own p^{th} power, so that $G(c,X)$ and $P(c,X)$ are in $\mathrm{GF}(p)[X]$. Because $\prod_{j=0}^{2}(X^2 + G_jX + 1)$ equals

$$X^6 + (G_0 + G_1 + G_2)X^5 + (G_0G_1 + G_0G_2 + G_1G_2 + 3)X^4 + (2G_0 + 2G_1 + 2G_2$$
$$+G_0G_1G_2)X^3 + (G_0G_1 + G_0G_2 + G_1G_2 + 3)X^2 + (G_0 + G_1 + G_2)X + 1,$$

it follows that $G(c,X) = \prod_{j=0}^{2}(X^2 + G_jX + 1)$ is equivalent to $G_0 + G_1 + G_2 = -c^p - c \in \mathrm{GF}(p)$, $G_0G_1 + G_0G_2 + G_1G_2 = c^{p+1} + c^p + c - 3 \in \mathrm{GF}(p)$, and $G_0G_1G_2 = 2c^p + 2c - c^{2p} - c^2 - 2 \in \mathrm{GF}(p)$. That is, G_0, G_1, G_2 are the roots of $P(c,X)$. The proof now follows from the fact that $\prod_{j=0}^{2}(X - G_j) = X^3 - (G_0 + G_1 + G_2)X^2 + (G_0G_1 + G_0G_2 + G_1G_2)X - G_0G_1G_2$, Definition 2.1, and the well known result that the roots of a third degree polynomial over $\mathrm{GF}(p)$ are either in $\mathrm{GF}(p^2)$ or in $\mathrm{GF}(p^3)$.

Lemma 2.3 $G(c,X) = \prod_{j=0}^{2}(X^2 + G_jX + 1)$ *where, depending on cases I, II, and III as identified above, the following holds:*

I. *$G_j \in \mathrm{GF}(p)$ for $j = 1,2,3$.*
II. *$G_0 \in \mathrm{GF}(p)$ and $G_j \in \mathrm{GF}(p^2)$ for $j = 1,2$.*
III. *$G_j \in \mathrm{GF}(p^3)$ for $j = 0,1,2$ and $G(c,X)$ is irreducible over $\mathrm{GF}(p)$.*

Proof. Immediate. For completeness we present the details. It follows from [4, Lemmas 2.3.4.*ii* and 2.3.2.*v*] that $F(c^p, h_j^p) = 0$ for $j = 0, 1, 2$, so that h_j^i for $j = 0, 1, 2$ and $i = 1, p$ are the roots of $G(c, X)$, in cases I, II, and III.

In case III (i.e., $F(c, X)$ is irreducible over $\mathrm{GF}(p^2)$) the h_j are conjugates over $\mathrm{GF}(p^2)$, i.e., $h_j = h_{j+1 \bmod 3}^{p^2}$. It follows that h_0 and its conjugates over $\mathrm{GF}(p)$ are the zeros of $G(c, X)$ so that $G(c, X)$ is irreducible over $\mathrm{GF}(p)$. Furthermore, $h_j^{p^3} = h_j^{p^3 \bmod p^2-p+1} = h_j^{-1} = h_{j+1 \bmod 3}^{p}$ and $h_{j+1 \bmod 3}^{p^4} = h_{j+2 \bmod 3}^{-p^5} = h_j^{p^6} = h_j$. Therefore $(h_j + h_{j+1 \bmod 3}^p)^{p^3} = h_j + h_{j+1 \bmod 3}^p$, so that $h_j + h_{j+1 \bmod 3}^p \in \mathrm{GF}(p^3)$. With $h_j \cdot h_{j+1 \bmod 3}^p = h_{j+1 \bmod 3}^{-p} \cdot h_{j+1 \bmod 3}^p = 1$ and defining $G_j = -h_j - h_{j+1 \bmod 3}^p \in \mathrm{GF}(p^3)$ for $j = 0, 1, 2$ we find that in case III the polynomial $G(c, X)$ factors as $\prod_{j=0}^{2}(X^2 + G_j X + 1)$ over $\mathrm{GF}(p^3)[X]$.

In case I we have for $j = 0, 1, 2$ that $h_j \cdot h_j^p = h_j^{-p} \cdot h_j^p = 1$ and $(h_j + h_j^p)^p = h_j^p + h_j^{p^2} = h_j^p + h_j$ so that $h_j + h_j^p \in \mathrm{GF}(p)$. Defining $G_j = -h_j - h_j^p \in \mathrm{GF}(p)$ for $j = 0, 1, 2$, we find that in case I the polynomial $G(c, X)$ factors as $\prod_{j=0}^{2}(X^2 + G_j X + 1)$ over $\mathrm{GF}(p)[X]$.

In case II we define $G_0 = -h_0 - h_0^p$, so that $G_0 \in \mathrm{GF}(p)$ as in case I. Furthermore, we define $G_j = -h_j - h_{3-j}^p$ for $j = 1, 2$. In this case $(h_j + h_{3-j}^p)^{p^2} = h_j^{p^2} + h_{3-j}^{p^3} = h_j + h_{3-j}^p$ so that $G_j \in \mathrm{GF}(p^2)$ for $j = 1, 2$. Because furthermore $h_j \cdot h_{3-j}^p = h_{3-j}^{-p} \cdot h_{3-j}^p = 1$ we find that $G(c, X)$ is the product of $X^2 + G_0 X + 1 \in \mathrm{GF}(p)[X]$ and $X^2 + G_j X + 1 \in \mathrm{GF}(p^2)[X]$ for $j = 1, 2$. This concludes the proof of Lemma 2.3.

Corollary 2.4 *Depending on cases I, II, and III, the following holds:*

I. *$P(c, X)$ has three roots in $\mathrm{GF}(p)$.*
II. *$P(c, X)$ has one root in $\mathrm{GF}(p)$ and two roots in $\mathrm{GF}(p^2)$.*
III. *$P(c, X)$ has three roots in $\mathrm{GF}(p^3) \setminus \mathrm{GF}(p)$.*

Corollary 2.5 *$F(c, X)$ is irreducible over $\mathrm{GF}(p^2)$ if and only if $P(c, X)$ is irreducible over $\mathrm{GF}(p)$.*

In the next section we show that we can determine irreducibility for $P(c, X)$ faster than for $F(c, X)$. Note that $P(c, X)$ can be computed from $F(c, X)$ at the cost of a small constant number of multiplications in $\mathrm{GF}(p)$.

3 Testing $P(c, X) \in \mathrm{GF}(p)[X]$ for irreducibility

Let $P(c, X) \in \mathrm{GF}(p)[X]$ as in Definition 2.1. We base our method to test $P(c, X)$ for irreducibility over $\mathrm{GF}(p)$ on Scipione del Ferro's method, cf. [5, Algorithm 3.1]. We recall this algorithm as it applies to $P(c, X) \in \mathrm{GF}(p)[X]$.

Algorithm 3.1 To find the roots of $P(c, X) = X^3 + p_2 X^2 + p_1 X + p_0 \in \mathrm{GF}(p)[X]$ in a field of characteristic unequal to 2 or 3, do the following.

1. Compute the polynomial $P(c, X - p_2/3) = X^3 + f_1X + f_0 \in \mathrm{GF}(p)[X]$ with $f_1 = p_1 - p_2^2/3$ and $f_0 = (27p_0 - 9p_2p_1 + 2p_2^3)/27$.
2. Compute the discriminant $\Delta = f_0^2 + 4f_1^3/27 \in \mathrm{GF}(p)$ of the polynomial $X^2 + f_0X - (f_1/3)^3$, and compute its roots $r_{1,2} = (-f_0 \pm \sqrt{\Delta})/2$.
3. If $r_1 = r_2 = 0$, then let $u = v = 0$. Otherwise, let $r_1 \neq 0$, compute a cube root u if r_1, and let $v = -f_1/(3u)$.
4. The roots of $P(c, X)$ are $u+v-p_2/3$, $u\alpha+v\alpha^2-p_2/3$, and $u\alpha^2+v\alpha-p_2/3$, with α as in [4, Section 2.1].

Lemma 3.2 *With cases I, II, and III as identified in Section 2 and Algorithm 3.1 applied to the polynomial $P(c, X) \in \mathrm{GF}(p)[X]$, we have that case III applies if and only if Δ as in Step 2 of Algorithm 3.1 is a quadratic non-residue in $\mathrm{GF}(p)$ and r_1 as in Step 2 of Algorithm 3.1 is not a cube in $\mathrm{GF}(p^2)$.*

Proof. If Δ as in Step 2 of Algorithm 3.1 is a quadratic residue in $\mathrm{GF}(p)$, then r_1 is in $\mathrm{GF}(p)$. From $p \equiv 2 \bmod 3$ it follows that all elements of $\mathrm{GF}(p)$ are cubes, so u as in Step 3 of Algorithm 3.1 is in $\mathrm{GF}(p)$ as well. It follows that $P(c, X)$ has at least one root in $\mathrm{GF}(p)$ so that with Corollary 2.4 case III does not apply.

If Δ is a quadratic non-residue in $\mathrm{GF}(p)$, then $r_1 \in \mathrm{GF}(p^2) \setminus \mathrm{GF}(p)$. If r_1 is a cube in $\mathrm{GF}(p^2)$ then $P(c, X)$ cannot have three roots in $\mathrm{GF}(p^3) \setminus \mathrm{GF}(p)$ so that, with Corollary 2.4, case III does not apply. The proof now follows by observing that if Δ is a quadratic non-residue in $\mathrm{GF}(p)$ and r_1 is not a cube in $\mathrm{GF}(p^2)$, then $P(c, X)$ cannot have a root in $\mathrm{GF}(p)$ so that, with Corollary 2.4, case III must apply.

Lemma 3.2 reduces $P(c, X)$-irreducibility (and thus $F(c, X)$-irreducibility, cf. Corollary 2.5) to the computation of a quadratic residue symbol, possibly followed by an actual square-root computation and a cubic residuosity test. We show that the square-root computation can be avoided by combining it with the cubic residuosity test. We first sketch our approach.

In [5] it was shown (just before Algorithm 3.5 in [5]) that an element x of $\mathrm{GF}(p^2)$ is a cube if and only if $x^{(p^2-1)/3} = 1$, i.e., if and only if $(x^{p-1})^{(p+1)/3} = 1$. It is easily shown that for $y \in \mathrm{GF}(p^2)$ of order dividing $p+1$ the trace over $\mathrm{GF}(p)$ of $y^{(p+1)/3}$ equals 2 if and only if $y^{(p+1)/3} = 1$. The trace over $\mathrm{GF}(p)$ of $y^{(p+1)/3}$ can be computed at the cost of $1.8 \log_2(p)$ multiplications in $\mathrm{GF}(p)$ if the trace over $\mathrm{GF}(p)$ of y is known (cf. Algorithm 3.4). In our application, $y = x^{p-1}$ and $x = r_1$ with $r_1 = -f_0/2 + \sqrt{\Delta}/2$ (cf. Step 2 of Algorithm 3.1) where Δ is a quadratic non-residue. We show that for x of this form the trace over $\mathrm{GF}(p)$ of x^{p-1} is given by an easy expression in which $\sqrt{\Delta}$ does not occur. Thus, the only substantial computation that remains to be done is the computation of the trace over $\mathrm{GF}(p)$ of $y^{(p+1)/3}$ at the cost of $1.8 \log_2(p)$ multiplications in $\mathrm{GF}(p)$. We now present this method in more detail.

Lemma 3.3 *Let $t \in \mathrm{GF}(p)$ be a quadratic non-residue in $\mathrm{GF}(p)$ and $a, b \in \mathrm{GF}(p)$. Then $a^2 - b^2t \neq 0$ and*

$$\left((a + bX)^{p-1} + (a + bX)^{1-p}\right) \bmod (X^2 - t) = 2\frac{a^2 + b^2t}{a^2 - b^2t}.$$

Proof. Because t is a quadratic non-residue we find that $a^2 - b^2 t \neq 0$ and that $t^{(p-1)/2} = -1$. The latter implies that $X^p = -X \bmod (X^2 - t)$, so that

$$\left(\frac{(a+bX)^p}{a+bX} + \frac{a+bX}{(a+bX)^p}\right) \bmod (X^2 - t) = \left(\frac{a-bX}{a+bX} + \frac{a+bX}{a-bX}\right) \bmod (X^2 - t).$$

The result follows with

$$\frac{a-bX}{a+bX} + \frac{a+bX}{a-bX} = \frac{(a-bX)^2 + (a+bX)^2}{(a+bX)(a-bX)} = 2\frac{a^2 + b^2X^2}{a^2 - b^2X^2}.$$

The following algorithm is well known in the context of primality testing, more specifically the $p+1$-test for primality (cf. [10, Section 4]).

Algorithm 3.4 To compute the trace $Tr(y^n) \in \mathrm{GF}(p)$ over $\mathrm{GF}(p)$ of $y^n \in \mathrm{GF}(p^2)$, given an integer $n > 0$ and the trace $Tr(y) \in \mathrm{GF}(p)$ over $\mathrm{GF}(p)$ of some $y \in \mathrm{GF}(p^2)$ of order dividing $p+1$. This algorithm takes $1.8 \log_2(p)$ multiplications in $\mathrm{GF}(p)$ assuming a squaring in $\mathrm{GF}(p)$ takes 80% of the time of a multiplication in $\mathrm{GF}(p)$.

1. Let $n = \sum_{i=0}^{k} n_i 2^i$ with $n_i \in \{0, 1\}$ and $n_k \neq 0$, let $v = Tr(y) \in \mathrm{GF}(p)$ and compute $w = (x^2 - 2) \in \mathrm{GF}(p)$.
2. For $i = k-1, k-2, \ldots, 0$ in succession, do the following.
   - If $n_i = 1$, then first replace v by $vw - Tr(y)$ and next replace w by $w^2 - 2$.
   - If $n_i = 0$, then first replace w by $vw - Tr(y)$ and next replace v by $v^2 - 2$.
3. Return $Tr(y^n) = v$.

Algorithm 3.5 To test $P(c, X) = X^3 + p_2X^2 + p_1X + p_0 \in \mathrm{GF}(p)[X]$ for irreducibility over $\mathrm{GF}(p)$, with p unequal to 2 or 3, do the following.

1. Compute the polynomial $P(c, X - p_2/3) = X^3 + f_1X + f_0 \in \mathrm{GF}(p)[X]$ with $f_1 = p_1 - p_2^2/3 \in \mathrm{GF}(p)$ and $f_0 = (27p_0 - 9p_2p_1 + 2p_2^3)/27 \in \mathrm{GF}(p)$.
2. Compute the discriminant $\Delta = f_0^2 + 4f_1^3/27 \in \mathrm{GF}(p)$ of the polynomial $X^2 + f_0X - (f_1/3)^3$.
3. Compute the Jacobi symbol of Δ. If Δ is a quadratic residue in $\mathrm{GF}(p)$, then $P(c, X)$ is not irreducible (cf. Lemma 3.2).
4. Otherwise, if Δ is a quadratic non-residue in $\mathrm{GF}(p)$, compute the trace of r_1^{p-1} over $\mathrm{GF}(p)$ as $s = 2\frac{f_0^2+\Delta}{f_0^2-\Delta} \in \mathrm{GF}(p)$, where $r_1 = -f_0/2 + \sqrt{\Delta}/2$ (cf. Lemma 3.3).
5. Apply Algorithm 3.4 to $Tr(y) = s$ and $n = (p+1)/3$ to compute the trace over $\mathrm{GF}(p)$ of $(r_1^{p-1})^{(p+1)/3}$. If the result equals 2, then r_1 is a cube in $\mathrm{GF}(p^2)$ and thus $P(c, X)$ is not irreducible (cf. Lemma 3.2).
6. Otherwise, Δ is a quadratic non-residue and r_1 is not a cube in $\mathrm{GF}(p^2)$ so that $P(c, X)$ is irreducible over $\mathrm{GF}(p)$ (cf. Lemma 3.2).

Theorem 3.6 *For $c \in \mathrm{GF}(p^2)$ the irreducibility of the polynomial $F(c, X) = X^3 - cX^2 + c^pX - 1$ over $\mathrm{GF}(p^2)$ can be tested at the cost of $m + 1.8 \log_2(p)$ multiplications in $\mathrm{GF}(p)$, for some small constant m.*

Proof. The proof follows from Section 2, Algorithm 3.5, and Algorithm 3.4.

Corollary 3.7 *Finding the trace of a generator of the XTR group can be expected to take about* $\frac{q}{q-1}(2.7\log_2(p) + 8\log_2((p^2 - p + 1)/q))$ *multiplications in* $\mathrm{GF}(p)$ *(cf. [5, Theorem 3.7]).*

Proof. Immediate from the proof of [5, Theorem 3.7] and Theorem 3.6 above.

Note that the result from Corollary 3.7 is only about $2.7\log_2(p)$ multiplications in $\mathrm{GF}(p)$ slower than [5, Algorithm 4.5], but more general since it applies to all $p \equiv 2 \bmod 3$ and not only to $p \equiv 2, 5 \bmod 9$.

4 Subgroup attacks

Many cryptographic protocols can be tricked into undesirable behavior if data is used that does not have the properties prescribed by the protocol. For instance, elements of a certain group may be exchanged, but if membership of the proper group is not tested before the elements are operated upon, security may be endangered. A prominent example is the following. Let G be a cyclic, multiplicative group of prime order q (of size ≥ 160 bits) where the discrete logarithm problem is believed to be intractable, and let g be an element of order q in G. In practice, G is often constructed as a subgroup of an abelian *supergroup* H, such that membership of H is easily verified.

For example, if $H = \mathrm{GF}(p)^*$ for a 1024-bit prime number p and the set $\{0, 1, ..., p-1\}$ is used to represent $\mathrm{GF}(p)$, then $x \in H$ if and only if $0 < x < p$, which can trivially be tested. Similarly, if H is the group of points (written multiplicatively) of a properly chosen elliptic curve over a finite field, then $x \in H$ can simply be verified by testing that the coordinates of the 'point' x belong to the finite field and that x satisfies the curve equation. In both examples G may be chosen as $\langle g \rangle$ for an element g of prime order q dividing the order $|H|$ of H. But testing if $x \in G$ is less trivial and consists of verifying that $x \in H$ and $x^q = 1$. In the first example $|H|/|G|$ is usually very large compared to q, whereas in the second example this ratio is commonly chosen to be very small.

To review why membership testing of G is crucial to maintain security we consider the Diffie-Hellman protocol. Assume that Alice calculates $v_A = g^{k_A} \in G$ where k_A is secret and sends the result to Bob. Likewise, Bob calculates and sends $v_B = g^{k_B} \in G$ to Alice, where k_B is supposed to be secret for Alice. The shared secret key $g^{k_A k_B}$ can then easily be computed by both Alice and Bob. The security is based on the assumption that k_A or k_B cannot be inferred from g, v_A, and v_B. This assumption may be incorrect if v_A or v_B is replaced by an element not in G, inadvertently or on purpose. As a first illustration, suppose that $\alpha \in H$ is of small order, say 2, and suppose that an active eavesdropper changes v_A into $v_A \cdot \alpha$ in transit. It follows that in this scenario the Diffie-Hellman protocol runs successfully if and only if v_B is even (or, more in general, if the order of α divides k_B). In other words, the eavesdropper obtains information on k_B, which is not supposed to happen.

As a second illustration, suppose that $|H|/|G|$ is a product of small primes (cf. [8]), and that h is an element of order $|H|/|G|$. If Alice somehow convinces Bob to use gh instead of g, and receives $(gh)^{k_B}$ instead of $g^{k_B} \in G$ from Bob, then Alice can easily determine h^{k_B} and thus $k_B \bmod (|H|/|G|)$ by using the Pohlig-Hellman algorithm (cf. [9]). That is, Alice obtains secret information on k_B if Bob naïvely uses a 'wrong' generator provided by Alice and does not check subgroup membership of the results of his own computations either. Another example is the Cramer-Shoup cryptosystem (cf. [3]) whose provable resistance against chosen ciphertext attacks relies on subgroup membership for a substantial number of elements that are exchanged in the course of the protocol.

In this paper, *subgroup attacks* refer to attacks that take advantage of the omission to verify membership of the subgroup G: they attack the security provided by the subgroup by replacing subgroup elements by elements from the supergroup H that do not belong to the proper subgroup. Examples of subgroup and related attacks can be found in [1], [2], [6], and [11]. We implicitly assume that membership of H is verified, i.e., that all alleged elements of H are indeed elements of H, and that this verification can easily be done.

Subgroup attacks can be prevented in roughly three ways:

1. By assuring that alleged subgroup members are indeed subgroup members, i.e., performing a membership test.
2. By ensuring that the ratio $|H|/|G|$ is small, e.g. 2.
3. By slightly adapting protocols.

We discuss these three prevention methods in more detail.

Membership test
In most practical circumstances the supergroup H is cyclic as well (as in systems based on the multiplicative group of a finite field), or the order q of G is a prime number such that H is not divisible by q^2 (as in elliptic curve cryptography, when using non-cyclic curve groups). The following result states that in these cases it suffices to do an *order check*, i.e., checking that $x \in H$ satisfies $x^q = 1$, to test membership of G.

Lemma 4.1 *Let G be multiplicative subgroup of prime order q of a supergroup H. If there exists an element $x \in H \setminus G$ for which $x^q = 1$, then H is not a cyclic group and the order of H is divisible by q^2.*

Proof: Assume to the contrary that H is cyclic. Then the number of elements of order dividing q is equal to q. The set $G \bigcup \{x\}$, however, contains at least $q+1$ elements of order dividing q; it follows that H cannot be cyclic. Furthermore, $\langle x, g \rangle$ is a subgroup of H of q^2 elements; it follows that q^2 divides $|H|$.

Thus, testing membership of G may entail an operation of cost comparable to the regular operations of the protocol. To illustrate that an order check is not sufficient in all cases, let $\tilde{G}$ be any cyclic group of order q and consider the cyclic subgroup $G = \langle (g_1, g_2) \rangle$ of the supergroup $H = \tilde{G}^2$, where g_1, g_2 are randomly chosen in $\tilde{G}$. In this case H is not cyclic and has order q^2. To test membership of G it is not sufficient to check that $(h_1, h_2)^q = (1, 1)$, but one needs to prove that $\log_{g_1}(h_1) = \log_{g_2}(h_2)$ which usually is computationally infeasible. This is known as an instance of the Decision Diffie-Hellman problem which usually is computationally infeasible. The latter example is not common in cryptographic applications, but simply serves as an illustration. From now on we will restrict ourselves to the situation that an order check is sufficient, i.e., H is cyclic or of order not divisible by q^2.

Choosing a small ratio $|H|/|G|$
If one chooses the ratio $r = |H|/|G|$ small then there exist only very few possibilities to perform subgroup based attacks. It seems widely accepted that at most $\log_2(r)$ secret bits are leaked if membership of H is checked but membership of G is not. In ordinary multiplicative groups r can only be small if q is very large, thereby losing the 'short exponents' advantage of working in a small subgroup. The computational overhead of full size exponents can, however, be reduced by using exponents that are only as long as one typically would choose the size of a subgroup of prime order q, i.e., ≥ 160 bits (cf. [8, Lemma 2]). Note that a small $|H|/|G|$ ratio is common in elliptic curve cryptosystems. In XTR the supergroup H is the order p^2-p+1 subgroup of $\mathrm{GF}(p^6)^*$, and the XTR group G is a subgroup of order q of H. In Section 5 below it is shown how membership of H can quickly be tested. Although the possibility of small values for $|H|/|G| = (p^2 - p + 1)/q$ is not explicitly mentioned in [4] or [5] it can without difficulty be used in the XTR versions of common cryptographic protocols, thereby limiting the risk of XTR subgroup attacks. Note that the risk of subgroup attacks against XTR is also very limited if $|H|/|G|$ is chosen as $3q_2$ for a prime number q_2 of the same order of magnitude as q.

Slightly Adapting Protocols
By adding an additional step to protocols using subgroups it can be ensured that the alleged subgroup element is retracted into the subgroup before secret information is employed to it. We illustrate this for the Diffie-Hellman protocol, using the notation as introduced above. Instead of using $g^{k_A k_B}$ as the shared secret key, one uses $g^{r k_A k_B}$, where $r = |H|/|G|$, which is computed in the following way. Upon receipt of v_B from Bob, Alice calculates $(v_B^r)^{k_A}$ instead of $v_B^{k_A}$. Similarly, Bob calculates $(v_A^r)^{k_B}$. Note that v_A^r is an element of G and that $v_A^{r k_B \bmod q}$ can only be equal to $(v_A^r)^{k_B}$ if $v_A \in G$. That is, performing the operations successively is crucial and, since an attacker may have chosen $v_A \notin G$, it is also crucial not to compute $v_A^{r \bmod q}$ but v_A^r for the 'original' $r = |H|/|G|$. Since, as we assumed, the co-factor $r = |H|/|G|$ is relatively prime with q, breaking this variant of the Diffie-Hellman protocol is as secure as the original one with a membership test incorporated into it. Many other DL based protocols and schemes that are susceptible to subgroup attacks, like the ElGamal scheme, can be adapted in a similar fashion.

Obviously, adaptation of protocols is typically a practical solution only if r is smaller than the prime order q of G, because otherwise a membership test would be more efficient. For instance, in traditional Schnorr-type subgroups systems

H is the multiplicative group of a large finite field GF(p), the subgroup G has substantially smaller size q, and r is often quite large: if $\log_2(p) = 1024$ and $\log_2(q) = 160$ then $\log_2(r) \approx 864$. If $r > q$, as in this example, then the best method we are aware of to verify subgroup membership is to check that the qth power of the element to be tested equals one (after one has verified, of course, that it is an element of H). Else, if $r < q$, then one may choose to slightly adapt the protocols used.

5 Prevention of subgroup attacks in XTR

In this section we focus on preventing subgroup attacks for XTR. Let G denote the XTR group and H the XTR supergroup of all elements of $\mathrm{GF}(p^6)^*$ of order > 3 and dividing $p^2 - p + 1$. We describe efficient ways to determine if an element in $\mathrm{GF}(p^2)$ is the trace of an element of H. The results from the previous section, e.g. choosing $|H|/|G|$ small and using short exponents, can then be used to obtain variants of XTR that are not susceptible to subgroups attacks.

Let d be the element of $\mathrm{GF}(p^2) \setminus \mathrm{GF}(p)$ to be verified. The first method consists simply of checking that $F(d, X)$ is irreducible over $\mathrm{GF}(p^2)$ (cf. [4, Remark 2.3.3]), which can be done at the cost of $1.8 \log_2(p)$ plus a small constant number of multiplications in GF(p) (cf. Theorem 3.6).

Our second method is effectively free from a computational point of view because it requires only a small constant number of operations in GF(p), but it requires a small amount of additional communication. Let p, q, and $Tr(g)$ be as above and let $d \in \mathrm{GF}(p^2)$ be the element to be verified, i.e., the element that is supposedly the trace of an element, say h, of the XTR group $\langle g \rangle$. Corollary 5.9 below shows that if one sends $Tr(h \cdot g)$ along with $d (= Tr(h))$, then one can efficiently verify that d corresponds to the trace of an element of the XTR supergroup H.

Definition 5.1 *Let $R(X), S(X) \in \mathrm{GF}(p^2)[X]$ be two monic third-degree polynomials with non-zero constant term. If the roots of R and S are $\alpha_0, \alpha_1, \alpha_2 \in \mathrm{GF}(p^6)$ and $\beta_0, \beta_1, \beta_2 \in \mathrm{GF}(p^6)$, respectively, then the root-product $\Re(R, S)$ is defined as the monic polynomial with the nine roots $\alpha_i \cdot \beta_j$ for $0 \leq i, j \leq 2$.*

Lemma 5.2 *For $R(X), S(X) \in \mathrm{GF}(p^2)[X]$ the root-product $\Re(R, S)$ is a ninth-degree polynomial over $\mathrm{GF}(p^2)$ with non-zero constant term.*

Proof: Fixing $R(X)$ and varying $S(X)$ one finds that the coefficients of the polynomial $\Re(R, S)$ are symmetric polynomials in the roots $\beta_0, \beta_1, \beta_2$ of $S(X)$, and that they can be written (cf. [7]) as linear sums of elementary symmetric polynomials in $\beta_0, \beta_1, \beta_2$ with fixed coefficients depending on $\alpha_0, \alpha_1, \alpha_2$. It also follows that these fixed coefficients are symmetric polynomials in the roots $\alpha_0, \alpha_1, \alpha_2$. The values of the elementary symmetric polynomials in $\alpha_0, \alpha_1, \alpha_2$ and $\beta_0, \beta_1, \beta_2$ are in $\mathrm{GF}(p^2)$ because $R(X), S(X) \in \mathrm{GF}(p^2)[X]$, so that the coefficients of the polynomial $\Re(R, S)$ are in $\mathrm{GF}(p^2)$. The remainder of the lemma is straightforward.

Lemma 5.3 *Let $R(X), S(X) \in \mathrm{GF}(p^2)[X]$ and let $\beta_0, \beta_1, \beta_2 \in \mathrm{GF}(p^6)$ be the roots of $S(X)$. Then*

$$\Re(R,S) = (\beta_0 \cdot \beta_1 \cdot \beta_2)^3 R(X \cdot \beta_0^{-1}) \cdot R(X \cdot \beta_1^{-1}) \cdot R(X \cdot \beta_2^{-1}).$$

If $S(X)$ is irreducible over $\mathrm{GF}(p^2)$ then

$$\Re(R,S) = \beta_0^{3(p^4+p^2+1)} R(X \cdot \beta_0^{-1}) \cdot R(X \cdot \beta_0^{-p^2}) \cdot R(X \cdot \beta_0^{-p^4}).$$

Proof: The first part result is a straightforward verification and the second part follows from the fact that the roots of $S(X)$ are conjugate over $\mathrm{GF}(p^2)$ if $S(X)$ is irreducible over $\mathrm{GF}(p^2)$.

Note that $\beta_0 \cdot \beta_1 \cdot \beta_2$ in Lemma 5.3 equals the constant term of $S(X)$. The crucial aspect of the second part of the lemma is that it describes $\Re(R,S)$ using only $R(X)$ and the conjugates of the roots of $S(X)$. That is, if we consider the representation of $\mathrm{GF}(p^6)$ that follows by adjoining a root of $S(X)$ to $\mathrm{GF}(p^2)$, we can efficiently determine the root-product of $R(X)$ and $S(X)$, assuming we can efficiently determine the $(p^2)^{\mathrm{th}}$ and $(p^4)^{\mathrm{th}}$ powers of a root of $S(X)$ in this representation.

In our application $S(X)$ is $F(c,X)$ where $c = Tr(g)$ for some element g in the XTR supergroup H. That is, $F(c,X)$ is irreducible by [4, Remark 2.3.3], and we represent $\mathrm{GF}(p^6)$ as $\mathrm{GF}(p^2)(g)$, i.e., by adjoining the root g of $F(c,X)$ to $\mathrm{GF}(p^2)$. Since $g^{p^2} = g^{p-1}$ and $g^{p^4} = g^{-p}$ and g^{p-1} and g^{-p} easily follow given a representation of g^p, in order to be able to compute the root-product $\Re(R, F(c,X))$ it suffices to have a representation for g^p in $\mathrm{GF}(p^2)(g)$. The following result shows how such a representation can be obtained. We abbreviate $Tr(g^i)$ as c_i.

Proposition 5.4 *Let $c = Tr(g)$ for some element $g \in H$. Given $c_{m-2} = Tr(g^{m-2}), c_{m-1} = Tr(g^{m-1})$, and $c_m = Tr(g^m)$, values $K, L, M \in \mathrm{GF}(p^2)$ such that $g^m = Kg^2 + Lg + M \bmod F(c,X)$ can be computed at the cost of a small constant number of operations in $\mathrm{GF}(p)$.*

Proof: By raising $g^m = Kg^2 + Lg + M$ to the $(p^i)^{\mathrm{th}}$ power for $i = 0, 2, 4$, and by adding the three resulting identities, we find that $c_m = Kc_2 + Lc_1 + Mc_0$. Similarly, from $g^{m-1} = Kg + L + Mg^{-1}$ and $g^{m-2} = K + Lg^{-1} + Mg^{-2}$ it follows that $c_{m-1} = Kc_1 + Lc_0 + Mc_{-1}$ and $c_{m-2} = Kc_0 + Lc_{-1} + Mc_{-2}$ respectively. This leads to the following system of equations over $\mathrm{GF}(p^2)$:

$$\begin{pmatrix} c_{m-2} \\ c_{m-1} \\ c_m \end{pmatrix} = \begin{pmatrix} c_{-2} & c_{-1} & c_0 \\ c_{-1} & c_0 & c_1 \\ c_0 & c_1 & c_2 \end{pmatrix} \cdot \begin{pmatrix} M \\ L \\ K \end{pmatrix}.$$

Because c_m, c_{m-1}, and c_{m-2} are given and the matrix on the right hand side is invertible (cf. [4, Lemma 2.4.4]) the proof follows.

Corollary 5.5 *Let $c = Tr(g)$ for some element $g \in H$. Given $Tr(g^{p-2})$, a representation of $g^p \bmod F(c,X)$ can be computed at the cost of a small constant number of operations in $\mathrm{GF}(p)$.*

Proof: This follows from Proposition 5.4 and the fact that $c_p = c_1^p = c^p$ and $c_{p-1} = c_{p^2} = c_1^{p^2} = c_1 = c$.

Theorem 5.6 *Let $R(X) \in \mathrm{GF}(p^2)[X]$ be a monic third-degree polynomial with non-zero constant term. Let $c = Tr(g)$ for some element $g \in H$. Given $Tr(g^{p-2})$, the root-product $\Re(R(X), F(c, X))$ can be computed at the cost of a small constant number of operations in $\mathrm{GF}(p)$.*

Proof: This follows immediately from Lemma 5.3 and Corollary 5.5.

We remark that $c_{p-2} = c_{p+1}$, as $c_{p+1} = c \cdot c_p - c^p \cdot c_{p-1} + c_{p-2}$ (cf. [4, Lemma 2.3.4.*i*]), $c_p = c^p$, and $c_{p-1} = c$. As the value c_{p-2} plays an important role it could be pre-computed and stored. The following result states that c_{p-2} can quickly be recovered from a single bit.

Proposition 5.7 *Let $c = Tr(g)$ for some element $g \in H$. Then $Tr(g^{p-2}) = c_{p-2}$ can be computed at the cost of a square-root computation in $\mathrm{GF}(p^2)$, assuming one bit of information to resolve the square-root ambiguity.*

Proof: Write $c_{p-2} = x_1\alpha + x_2\alpha^2$ in the representation of $\mathrm{GF}(p^2)$ introduced in [4, Section 2.1]. A straightforward verification shows that $(c_{p-2} - c_{p-2}^p)^2 = -3(x_1 - x_2)^2$. Combining this with the identity for $(c_{p-2} - c_{p-2}^p)^2$ given in [4, Lemmas 2.4.4 and 2.4.5] (and using that $c_{p-2} = c_{p+1}$), we find that $-3(x_1 - x_2)^2 = c^{2p+2} + 18c^{p+1} - 4(c^{3p} + c^3) - 27 \in \mathrm{GF}(p)$.

On the other hand $c_{p-2} + c_{p-2}^p = -(x_1 + x_2)$. Using that $c_{p-2} = g^{p-2} + g^{(p-2)p^2} + g^{(p-2)p^4} = g^{p-2} + g^{-2p+1} + g^{p+1}$, it follows that $c_{p-2}^p = g^{-p-1} + g^{-p+2} + g^{2p-1}$. Now,

$$\begin{aligned} c^{p+1} &= c \cdot c^p = (g + g^{p-1} + g^{-p})(g^p + g^{-1} + g^{-p+1}) \\ &= g^{p+1} + g^{p-2} + g^{-2p+1} + g^{-p-1} + g^{-p+2} + g^{2p-1} + 3 \\ &= c_{p-2} + c_{p-2}^p + 3. \end{aligned}$$

That is, $x_1 + x_2 = 3 - c^{p+1} \in \mathrm{GF}(p)$. Combining the two identities involving $x_1 - x_2$ and $x_1 + x_2$ it follows that c_{p-2} and its conjugate over $\mathrm{GF}(p)$ can be computed at the cost of a square-root calculation in $\mathrm{GF}(p^2)$. To distinguish $c_{p-2} = x_1\alpha + x_2\alpha^2$ from its conjugate $x_2\alpha + x_1\alpha^2$ a single bit that is on if and only if $x_1 > x_2$ suffices.

Lemma 5.8 *Let $c = Tr(g)$ for some element $g \in H$ and let $d, d' \in \mathrm{GF}(p^2)$. Given the value c_{p-2} the correctness of the following statement can be checked at the cost of a small, constant number of operations in $\mathrm{GF}(p)$: there exists an element $h \in H$ such that $d = Tr(h)$ and $d' = Tr(h \cdot g)$.*

Proof: Consider the following algorithm:

1. By a simple verification check whether 1, α or α^2 are roots of the polynomial $F(d, X)$. If so, then the statement is not true.

2. Otherwise, calculate the root-product $\Re(F(d,X),F(c,X))$ and determine if this is divisible by the polynomial $F(d',X)$. If so, the statement is true, otherwise it is not.

The conclusion of the first step of the algorithm is trivial. For a proof of the conclusion of Step 2 of the algorithm, assume that d is not equal to the trace of an element of H. It follows from [4, Lemma 2.3.2] that the roots of $F(d,X)$ are in $\mathrm{GF}(p^2)$. According to Step 1 the roots are not equal to 1, α or α^2, so that the roots of $F(d,X)$ are not members of H either, as the greatest common divisor of p^2-p+1 and p^2-1 is 3 and H has order > 3. It easily follows that none of the roots of the root-product $\Re(F(d,X),F(c,X))$ lies in H. Moreover, as the roots of $F(c,X)$ do not lie in $\mathrm{GF}(p^2)$, it follows that the roots of the root-product do not lie in $\mathrm{GF}(p^2)$ either.

Applying [4, Lemma 2.3.2] once more, the roots of the polynomial $F(d',X)$ are either in $\mathrm{GF}(p^2)$ or in H. It follows that $F(d',X)$ cannot divide the root-product $\Re(F(d,X),F(c,X))$. Thus if $F(d',X)$ divides $\Re(F(d,X),F(c,X))$, then d is equal to the trace of an element $h \in H$. In this situation, the roots of the root-product are equal to $h^{p^i} \cdot g^{p^j}$ for $i,j = 0,2,4$. It follows that $F(d', h^{p^i} \cdot g) = 0$ for some i in $\{0,2,4\}$ and hence that $d = Tr(h^{p^i})$ and $d' = Tr(h^{p^i} \cdot g)$. That is, the statement is true.

We finally observe that the algorithm requires a small constant number of operations in $\mathrm{GF}(p^2)$.

Corollary 5.9 *Let $c = Tr(g)$ for some element $g \in H$ and suppose that $Tr(g^{p-2})$ is known. Then accompanying the trace value of an element $h \in H$ by the trace of its 'successor' $h \cdot g$ enables an efficient proof of membership of h in H.*

Corollary 5.10 *Let $c = Tr(g)$ where g is (known to be) a generator of the XTR group, let d be the trace of an element that is (known to be) in the XTR group $\langle g \rangle$, and let d' be some element of $\mathrm{GF}(p^2)$. Then it can efficiently be verified if d and d' are of the form $Tr(g^x)$, $Tr(g^{x+1})$, respectively, for some integer x, $0 < x < q$.*

An XTR public key meant for digital signatures takes the form p, q, c, d, and d' where p and q are primes satisfying the usual XTR conditions and where $c = Tr(g)$ for a generator g of the XTR group, $d = Tr(g^k)$ for a secret key k, and $d' = Tr(g^{k+1})$ (cf. [4]). The above corollary implies that a Certificate Authority can efficiently verify the consistency of an XTR signature public key presented by a client, before issuing a certificate on it. More specifically, suppose a client provides a Certificate Authority with XTR public key data containing p, q, c, d, and d' where p and q are primes satisfying the usual XTR conditions and where, supposedly, $c = Tr(g)$ for a generator g of the XTR group, $d = Tr(g^k)$ for a secret key k, and $d' = Tr(g^{k+1})$. The Certificate Authority can easily check that this is indeed the case, in two steps. First the Certificate Authority checks that p and q are well-formed and that c and d are indeed traces of elements of the XTR group using standard XTR arithmetic (cf. [4, Lemma 2.3.4 and Theorem 2.3.8]). Secondly, the Certificate Authority uses Corollary 5.10 to verify that d

and d' are traces of consecutive (and unknown, to the Certificate Authority) powers of the generator corresponding to c.

Acknowledgment. Acknowledgments are due to Wei Dai for inspiring us to improve the $F(c, X)$-irreducibility test from [5].

References

1. I. Biehl, B. Meyer, V. Müller, *Differential fault attacks on elliptic curve cryptosystems*, Proceedings of Crypto 2000, LNCS 1880, Springer-Verlag, 2000, 131-146.
2. M.V.D. Burmester, *A remark on the efficiency of identification schemes*, Proceedings of Eurocrypt'90, LNCS 473, Springer-Verlag 1990, 493-495.
3. R. Cramer, V. Shoup, *A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack*, Proceedings of Crypto'98, LNCS 1462, Springer-Verlag 1998, 13-25.
4. A.K. Lenstra, E.R. Verheul, *The XTR public key system*, Proceedings of Crypto 2000, LNCS 1880, Springer-Verlag, 2000, 1-19; available from www.ecstr.com.
5. A.K. Lenstra, E.R. Verheul, *Key improvements to XTR*, Proceedings of Asiacrypt 2000, LNCS 1976, Springer-Verlag, 2000, 220-233; available from www.ecstr.com.
6. C.H. Lim, P.J. Lee, *A key recovery attack on discrete log-based schemes using a prime order subgroup*, Proceedings of Crypto'97, LNCS 1294, Springer-Verlag 1997, 249-263.
7. W.K. Nicholson, *Introduction to abstract algebra*, PWS-Kent Publishing Company, Boston, 1993.
8. P.C. van Oorschot, M.J. Wiener, *On Diffie-Hellman key agreement with short exponents*, Proceedings of Eurocrypt '96, LNCS 1070, Springer-Verlag 1996, 332-343.
9. S.C. Pohlig, M.E. Hellman, *An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance*, IEEE Trans. on IT, 24 (1978), 106-110.
10. H. Riesel, *Prime numbers and computer methods for factorization*, Birkhäuser, Boston, 1985.
11. E.R. Verheul, M.P. Hoyle, *Tricking the Chaum-Pedersen protocol*, manuscript, 1998.