To appear in *Advances in Cryptology—Asiacrypt 2000, Lecture Notes in Computer Science 1976*, Springer-Verlag 2000, 220-223.

Key improvements to XTR

Arjen K. Lenstra[1], 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. This paper describes improved methods for XTR key representation and parameter generation (cf. [4]). If the field characteristic is properly chosen, the size of the XTR public key for signature applications can be reduced by a factor of three at the cost of a small one time computation for the recipient of the key. Furthermore, the parameter set-up for an XTR system can be simplified because the trace of a proper subgroup generator can, with very high probability, be computed directly, thus avoiding the probabilistic approach from [4]. These non-trivial extensions further enhance the practical potential of XTR.

1 Introduction

In [1] it was shown that conjugates of elements of a subgroup of $\mathrm{GF}(p^6)^*$ of order dividing $\phi_6(p) = p^2 - p + 1$ can be represented using $2\log_2(p)$ bits, as opposed to the $6\log_2(p)$ bits that would be required for their traditional representation. In [4] an improved version of the method from [1] was introduced that achieves the same communication advantage at a much lower computational cost. The resulting representation method is referred to as XTR, which stands for Efficient and Compact Subgroup Trace Representation. As shown in [4], solving the XTR version of a particular discrete logarithm related problem is equivalent to solving the same problem in its traditional $\mathrm{GF}(p^6)$ setting, which is as hard as solving the problem in the full multiplicative group $\mathrm{GF}(p^6)^*$.

It is argued in [4] that XTR is an excellent alternative to either RSA or Elliptic Curve Cryptosystems using random curves over prime fields (ECC), because it combines most of the advantages of RSA and ECC without having any of their disadvantages. More specifically, it is shown in [4] that, with the exception of signature applications, XTR keys are much smaller than RSA keys of equivalent security, and at most twice as big as ECC keys. Furthermore, parameter and key selection for XTR is very fast compared to RSA, and thus much faster than ECC. Finally, for almost all cryptographic applications XTR is faster than ECC when random curves over prime fields are used; the exception is signature verification where ECC is slightly faster than XTR.

In this paper we describe three improvements to XTR. We present a careful analysis of Scipione del Ferro's classical method to solve cubic equations. As a result we are able to reduce the XTR public key size for signature applications by a factor of three if the field characteristic is not equal to 8 modulo 9. Because that is not unduly restrictive, it follows that XTR public keys are at most twice as long as ECC public keys for *all* applications of XTR. This is, in our opinion, an important enhancement of XTR. As a side result we get a method to find the trace of a proper subgroup generator that is 50% faster than the method presented in [4]. Finally, we give a much faster deterministic method for the same problem that works only if the characteristic is not equal to 8 modulo 9. None of these two improved XTR parameter selection methods is of crucial importance for practical applications of XTR, but the last method in particular makes implementation of XTR even easier. The resulting algorithms are all very practical and allow easy implementation.

In Section 2 we review XTR. In Section 3 we present Scipione del Ferro's method and the resulting improved parameter selection method. An even faster parameter selection method is given in Section 4, and the key size reduction methods are given in Section 5.

2 XTR

In this section we review some of the results from [4]. Let p be prime and let $F(c, X)$ for $c \in \mathrm{GF}(p^2)$ be the polynomial $X^3 - cX^2 + c^pX - 1 \in \mathrm{GF}(p^2)[X]$. For $n \in \mathbf{Z}$ we denote by c_n the sum of the n^{th} powers of the roots of $F(c, X)$, i.e., if $F(c, h_j) = 0$ for $j = 0, 1, 2$, then $c_n = h_0^n + h_1^n + h_2^n$. Notice that $c_1 = c$. It is shown in [4] that $c_n \in \mathrm{GF}(p^2)$, that $c_{-n} = c_n^p$, and that $F(c_n, h_j^n) = 0$ for $j = 0, 1, 2$. Furthermore, if $p \equiv 2 \bmod 3$, then p^{th} powering in $\mathrm{GF}(p^2)$ is effectively free, and c_n can be computed given $c = c_1$ in $8\log_2(n)$ multiplications in $\mathrm{GF}(p)$ using a Fibonacci-like recurrence relation (cf. [4]). The values c_{n-1} and c_{n+1} are obtained at no extra cost as a side result of the computation of c_n.

It is shown in [4] that if $F(c, X)$ is irreducible, then the roots of $F(c, X)$ take the form h, h^{p^2}, h^{p^4} for some $h \in \mathrm{GF}(p^6)$ of order dividing $p^2 - p + 1$ and > 3. This implies that in these circumstances c_n is of the form $Tr(h^n)$, where $Tr(y) = y + y^{p^2} + y^{p^4} \in \mathrm{GF}(p^2)$ is the trace over $\mathrm{GF}(p^2)$ of $y \in \mathrm{GF}(p^6)$, i.e., the sum of the conjugates over $\mathrm{GF}(p^2)$ of y. The trace over $\mathrm{GF}(p^2)$ is $\mathrm{GF}(p^2)$-linear. Vice versa, it is shown that the minimal polynomial of any $h \in \mathrm{GF}(p^6)$ of order dividing $p^2 - p + 1$ and > 3 is equal to $F(Tr(h), X)$, illustrating the fundamental idea of XTR that for such h the trace value fully specifies h's minimal polynomial, and thus the conjugates of h.

Let $g \in \mathrm{GF}(p^6)$ have order q for a prime $q > 3$ dividing $p^2 - p + 1$. It follows from the results cited above that $Tr(g^n) \in \mathrm{GF}(p^2)$ and $F(Tr(g^n), g^n) = 0$ for any n. Furthermore, if $p \equiv 2 \bmod 3$ then $Tr(g^n)$ can be computed given $Tr(g)$ in $8\log_2(n)$ multiplications in $\mathrm{GF}(p)$, which is almost three times faster than computing g^n from g using traditional exponentiation methods. Thus, in XTR we replace powers of g by their traces, thereby saving a factor of three both in storage and in computing time. Note that an actual representation of g is not required, and that it suffices to have its trace $Tr(g)$. Given $Tr(g)$, the order q subgroup generated by (the unknown) g is called the XTR group.

XTR parameter selection is the problem of finding primes p and q such that q divides $p^2 - p + 1$, $q > 3$, $p \equiv 2 \bmod 3$, and $p \equiv 3 \bmod 4$, and the trace $Tr(g)$ of a generator of the XTR group. The primes p and q of appropriate sizes can be found using either of the two methods given in [4]. To find a proper $Tr(g)$ it suffices to find $c \in \mathrm{GF}(p^2) \setminus \mathrm{GF}(p)$ such that $F(c, X) \in \mathrm{GF}(p^2)[X]$ is irreducible, such that $c_{(p^2-p+1)/q} \neq 3$, and to put $Tr(g) = c_{(p^2-p+1)/q}$ (cf. [4]). The probability that $c_{(p^2-p+1)/q} = 3$ if $F(c, X)$ is irreducible is only $1/q$, so usually the first irreducible $F(c, X)$ works. In Section 3 we describe a fast way to test $F(c, X)$ for irreducibility (assuming a randomly selected $c \in \mathrm{GF}(p^2)$), and in Section 4 we show how irreducible polynomials of the form $F(c, X)$ can be written down directly if $p \not\equiv 8 \bmod 9$.

The ability to quickly compute $Tr(g^n)$ given $Tr(g)$ suffices for efficient implementation of many cryptographic protocols. But in some cryptographic applications, most notably verification of digital signatures and authentication responses, values of the form $Tr(g^{a+kb})$ have to be computed, for $a, b \in \mathbf{Z}$, given $Tr(g)$ and $Tr(g^k)$ for some secret integer k (the private key). It is shown in [4] that computation of $Tr(g^{a+kb})$ can efficiently be done if additionally $Tr(g^{k-1})$ and $Tr(g^{k+1})$ are known. Thus, whereas for many applications the XTR public key data consist of just p, q, $Tr(g)$, and $Tr(g^k)$ (for unknown k), in some applications $Tr(g^{k-1})$ and $Tr(g^{k+1})$ must be included in the XTR public key data as well. This considerably increases the transmission overhead for the XTR public key data. In Section 4 we show how this problem can be dealt with. First we show that $Tr(g^{k-1})$ (or $Tr(g^{k+1})$) can easily be determined as a function of $Tr(g)$, $Tr(g^k)$ and $Tr(g^{k+1})$ (or $Tr(g^{k-1})$). And next we show how $Tr(g^{k+1})$ (or $Tr(g^{k-1})$) can be quickly computed based on just $Tr(g)$ and $Tr(g^k)$, assuming that $p \not\equiv 8 \bmod 9$. Both methods impose very mild restrictions on the choice of the private key k and have no negative impact on the security of XTR.

3 Finding a root of a cubic equation

We describe Scipione del Ferro's classical method (cf. [6], page 559) to compute the roots of a third-degree equation in its full generality, after which we apply it to test the third-degree polynomial $F(c, X) \in \mathrm{GF}(p^2)[X]$ as in Section 2 for irreducibility.

Algorithm 3.1 (Scipione del Ferro, ∼1465-1526) To find the roots of the third-degree polynomial $f(X) = aX^3 + bX^2 + dX + e$ in a field of characteristic p unequal to 2 or 3, perform the following steps.

1. Compute the polynomial $f(X - b/(3a))/a = X^3 + f_1 X + f_0$ with $f_1 = (3ad - b^2)/(3a^2)$ and $f_0 = (27a^2e - 9abd + 2b^3)/(27a^3)$.
2. Compute the discriminant $\Delta = f_0^2 + 4f_1^3/27$ of the polynomial $X^2 + f_0 X - f_1^3/27$, 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 of r_1, and let $v = -f_1/(3u)$. Note that v is a cube root of r_2.

4. The roots of $f(X)$ are $u+v-b/(3a)$, $uw+vw^2-b/(3a)$, and $uw^2+vw-b/(3a)$, where $w \in \mathrm{GF}(p^2)$ is a non-trivial cube root of unity, i.e., $w^3 = 1$ and $w^2 + w + 1 = 0$.

Theorem 3.2 *Let $f(X) \in \mathrm{GF}(p^2)[X]$ be such that Δ as in Step 2 of Algorithm 3.1 is in $\mathrm{GF}(p)$. The following four statements are equivalent.*

1. *$f(X)$ is reducible over $\mathrm{GF}(p^2)$.*
2. *$f(X)$ has a root in $\mathrm{GF}(p^2)$.*
3. *$f(X)$ has three roots in $\mathrm{GF}(p^2)$.*
4. *The roots r_1 and r_2 as in Step 2 of Algorithm 3.1 are cubes in $\mathrm{GF}(p^2)$.*

Proof. $1 \Leftrightarrow 2$ and $3 \Rightarrow 2$ are trivial. We prove $2 \Leftrightarrow 4$ and $4 \Rightarrow 3$.

'$4 \Rightarrow 2$'. If there is a u in $\mathrm{GF}(p^2)$ such that $u^3 = r_1$, then $u - f_1/(3u) - b/(3a)$ is a root of $f(X)$ in $\mathrm{GF}(p^2)$ (cf. Step 4 of Algorithm 3.1).

'$2 \Rightarrow 4$'. If $f(X)$ has a root in $\mathrm{GF}(p^2)$, then there is a cube root u of r_1 such that $u+v-b/(3a) \in \mathrm{GF}(p^2)$, with $v = -f_1/(3u)$, so that $u+v$ is in $\mathrm{GF}(p^2)$. Since also $uv = -f_1/3$ is in $\mathrm{GF}(p^2)$, it follows that $u \in \mathrm{GF}(p^4)$. On the other hand, $r_1, r_2 \in \mathrm{GF}(p^2)$ because $\Delta \in \mathrm{GF}(p)$. Since $u^3 = r_1$ it follows that $u \in \mathrm{GF}(p^6)$. From $u \in \mathrm{GF}(p^4) \bigcap \mathrm{GF}(p^6)$ it follows that $u \in \mathrm{GF}(p^2)$ so that r_1 is a cube in $\mathrm{GF}(p^2)$. It follows from $r_2 = (-f_1/(3u))^3$ that r_2 is a cube in $\mathrm{GF}(p^2)$ as well.

'$4 \Rightarrow 3$'. If $u - f_1/(3u) - b/(3a)$ is a root of $f(X)$ with u in $\mathrm{GF}(p^2)$ then $uw - f_1w^2/(3u) - b/(3a)$ and $uw^2 - f_1w/(3u) - b/(3a)$, with $w \in \mathrm{GF}(p^2)$ as in Step 4 of Algorithm 3.1, are the two other roots of $f(X)$ (cf. Step 4 of Algorithm 3.1), and all three roots are in $\mathrm{GF}(p^2)$.

Lemma 3.3 *For any $c \in \mathrm{GF}(p^2)$ the discriminant Δ as in Step 2 of Algorithm 3.1 of $f(X) = F(c, X)$ is in $\mathrm{GF}(p)$.*

Proof. It follows from a straightforward computation that $\Delta = 1 - 2c^{p+1}/3 - c^{2p+2}/27 + 4(c^3 + c^{3p})/27$. This implies that $\Delta^p = \Delta$ so that $\Delta \in \mathrm{GF}(p)$.

Corollary 3.4 *The polynomial $F(c, X) \in \mathrm{GF}(p^2)[X]$ is reducible over $\mathrm{GF}(p^2)$ if and only if the r_1 from Step 2 of an application of Algorithm 3.1 to $f(X) = F(c, X)$ is a cube in $\mathrm{GF}(p^2)$.*

Proof. Immediate from Lemma 3.3 and Theorem 3.2.

An element $x \in \mathrm{GF}(p^2)$ is a cube if and only if $x^{(p^2-1)/3} = 1$, which is the case if and only if $x^{p(p+1)/3} = x^{(p+1)/3}$. Thus, testing if an element of $\mathrm{GF}(p^2)$ is a cube can be done at the cost of a $(p+1)/3^{\text{th}}$ powering in $\mathrm{GF}(p^2)$ followed by a p^{th} powering (which is free in $\mathrm{GF}(p^2)$, cf. Section 2).

Algorithm 3.5 (Irreducibility test) To decide if $F(c, X) \in \mathrm{GF}(p^2)[X]$ is irreducible over $\mathrm{GF}(p^2)$, perform the following steps.

1. Compute $F(c, X + c/3) = X^3 + f_1X + f_0 \in \mathrm{GF}(p^2)[X]$ with $f_1 = c^p - c^2/3$ and $f_0 = (-27 + 9c^{p+1} - 2c^3)/27$ (cf. $p \neq 3$).
2. If $\Delta = f_0^2 + 4f_1^3/27 \in \mathrm{GF}(p)$ (cf. $p \neq 3$) is a quadratic non-residue in $\mathrm{GF}(p)$ then $F(c, X)$ is reducible (cf. Lemma 3.6) .

3. Otherwise, compute a root $r_1 \in \mathrm{GF}(p^2)$ (cf. Corollary 3.4) of $X_2 + f_0 X - (f_1/3)^3$: $r_1 = (-f_0 + \sqrt{\Delta})/2$ (cf. $p \neq 2$).
4. Compute $y = r_1^{(p+1)/3} \in \mathrm{GF}(p^2)$, then $F(c, X)$ is irreducible $\iff y \neq y^p$.

Lemma 3.6 *The discriminant Δ as in Step 2 of Algorithm 3.5 is a quadratic residue in* $\mathrm{GF}(p)$ *if and only if either $F(c,X)$ is irreducible in* $\mathrm{GF}(p^2)[X]$ *or all roots in* $\mathrm{GF}(p^2)$ *of $F(c,X)$ have order dividing $p+1$.*

Proof. According to Algorithm 3.1 the roots h_0, h_1, h_2 in $\mathrm{GF}(p^6)$ of $F(c,X) \in \mathrm{GF}(p^2)[X]$ can be written as $u+v+y$, $u\alpha + v\alpha^2 + y$, and $u\alpha^2 + v\alpha + y$ with u and v as in Algorithm 3.1, y some element of $\mathrm{GF}(p^2)$, and α as in Section 4. Without loss of generality we have that $h_0 = u+v+y$, $h_1 = u\alpha + v\alpha^2 + y$, and $h_2 = u\alpha^2 + v\alpha + y$. Multiplying the three identities by 1, α^2, and α, respectively, we get

$$h_0 = u + v + y,\ \alpha^2 h_1 = u + v\alpha + y\alpha^2,\ \alpha h_2 = u + v\alpha^2 + y\alpha.$$

Adding these identities and using that $\alpha^2 + \alpha + 1 = 0$ we find that $u = U/3$ where $U = h_0 + \alpha^2 h_1 + \alpha h_2$.

According to Algorithm 3.1 we have that $U^3/27 = u^3 = r_1$ where $r_1 = (-f_0 + \sqrt{\Delta})/2$ and $f_0 = (-27 + 9c^{p+1} - 2c^3)/27$ (cf. Algorithm 3.5). Since $(-27 + 9c^{p+1})/27 \in \mathrm{GF}(p)$ we have that $\sqrt{\Delta} \in \mathrm{GF}(p)$ if and only if $U^3 - c^3 \in \mathrm{GF}(p)$. With $c_3 = Tr(g^3) = c^3 - 3c^{p+1} + 3$ (cf. Corollary 2.3.5.*i* and *ii* in [4]) and $c^{p+1} \in \mathrm{GF}(p)$ this is the case if and only if $U^3 - Tr(g^3) \in \mathrm{GF}(p)$. With $Tr(g^3) = h_0^3 + h_1^3 + h_2^3$ it follows from a straightforward computation that

$$\begin{aligned} U^3 - Tr(g^3) &= 3(h_0^2 h_2 + h_1^2 h_0 + h_2^2 h_1 - 2)\alpha + 3(h_0^2 h_1 + h_1^2 h_2 + h_2^2 h_0 - 2)\alpha^2, \\ &= 3(h_0/h_1 + h_1/h_2 + h_2/h_0 - 2)\alpha + 3(h_0/h_2 + h_1/h_0 + h_2/h_1 - 2)\alpha^2. \end{aligned}$$

where the last identity follows from $h_0 h_1 h_2 = 1$. According to Lemma 2.3.2.*iv* in [4] we have that $F(c, h_j^{-p}) = 0$ for $j = 0,1,2$. Thus either $h_j = h_j^{-p}$ for $j = 0,1,2$ (i.e., all roots have order dividing $p+1$), or $h_0 = h_0^{-p}$, $h_1 = h_2^{-p}$, and $h_2 = h_1^{-p}$, or $h_j = h_{j+1 \bmod 3}^{-p}$ for $j = 0,1,2$. According to Lemma 2.3.2.*vi* in [4], the last case is equivalent with $F(c,X)$ being irreducible in $\mathrm{GF}(p^2)[X]$. We prove that $U^3 - Tr(g^3) \in \mathrm{GF}(p)$ if and only if the first or the last case applies.

Let $w = h_0/h_1 + h_1/h_2 + h_2/h_0$ and $z = h_0/h_2 + h_1/h_0 + h_2/h_1$. If the first or the last case applies, then $w^p = z$ so that $(U^3 - Tr(g^3))^p = U^3 - Tr(g^3)$, and thus $U^3 - Tr(g^3) \in \mathrm{GF}(p)$. If the second case applies, then $w^p = w$ and $z^p = z$ so that $w, z \in \mathrm{GF}(p)$. Now, if additionally $U^3 - Tr(g^3) \in \mathrm{GF}(p)$ then $w = z$ so that the polynomial $X^3 - wX^2 + zX - 1 = X^3 - wX^2 + wX - 1$ has 1 as a root. As this polynomial has root-set $\{h_0/h_1, h_1/h_2, h_2/h_0\}$, it follows that $h_1 = h_2$, or one of h_1, h_2 is equal to h_0. As the order of h_0 divides $p+1$ by assumption, it follows in each case that the same is true for h_1 and h_2. That is, the first case applies (and we are in the situation that both the first and second case applies).

Theorem 3.7 *Finding the trace of a generator of the XTR group can be done in an expected number $\frac{q}{q-1}(7.2\log_2(p) + 8\log_2((p^2-p+1)/q))$ plus a small constant number of multiplications in* $\mathrm{GF}(p)$.

Proof. The correctness of Algorithm 3.5 follows from Corollary 3.4 and Lemma 3.6. Because Δ is a quadratic residue in $\mathrm{GF}(p)$ if $F(c,X)$ is irreducible (cf. Appendix A) Step 3 of Algorithm 3.5 takes a $((p+1)/4)^{\mathrm{th}}$ powering in $\mathrm{GF}(p)$ (cf. $p \equiv 3 \bmod 4$). Assuming that a squaring in $\mathrm{GF}(p)$ takes 80% of the time of a multiplication (cf. [2]), Step 3 of Algorithm 3.5 can be expected to require $1.3\log_2(p)$ multiplications in $\mathrm{GF}(p)$. Step 4 of Algorithm 3.5 takes an expected $\log_2(p)$ squarings and $0.5\log_2(p)$ multiplications in $\mathrm{GF}(p^2)$, for an expected total of $3.5\log_2(p)$ multiplications in $\mathrm{GF}(p)$ (cf. Lemma 2.1.1 in [4]). Thus the total expected cost of Steps 3 and 4 of Algorithm 3.5 is $4.8\log_2(p)$ multiplications in $\mathrm{GF}(p)$. According to Lemma 3.2.1 in [4] the probability that $F(c,X)$ is irreducible for a random $c \in \mathrm{GF}(p^2)$ is about one third. Furthermore, it can be proved along the lines of the proof of the same lemma that for a random c the Δ as in Step 2 of Algorithm 3.5 is a quadratic non-residue with probability 1/2. The theorem now follows with Section 2 and the fact that the cost of the Jacobi sum test to test the quadratic residuosity of Δ is bounded by a small constant number of multiplications in $\mathrm{GF}(p)$.

Remark 3.8 It follows that a proper $Tr(g)$ can be found more than 50% faster than described in [4]. Theorem 3.7 is however just a side result of a more important consequence of Scipione del Ferro's method, namely the key size reduction method presented in Section 5. Before we can present that method we need some other results that also lead to yet another, even faster, way to find $Tr(g)$.

4 Improved parameter selection if $p \not\equiv 8 \bmod 9$

In this section we prove that if $p \not\equiv 8 \bmod 9$ (but $p \equiv 2 \bmod 3$), then an irreducible $F(c,X) \in \mathrm{GF}(p^2)[X]$ can be written down directly. This follows from a general argument shown to us by H.W. Lenstra, Jr., that applies even to the characteristic zero case. We present a simplified description that applies just to non-zero characteristics.

So far we have considered $p \equiv 2 \bmod 3$, because this implies that the polynomial $(X^3-1)/(X-1) = X^2+X+1 \in \mathrm{GF}(p)[X]$ is irreducible over $\mathrm{GF}(p)$ and $\{\alpha,\alpha^2\}$ with $\alpha^2+\alpha+1=0$ forms an optimal normal basis for $\mathrm{GF}(p^2)$ over $\mathrm{GF}(p)$. As shown in [4] this leads to a very efficient and convenient representation of $\mathrm{GF}(p^2)$ in which p^{th} powering is free. Here we restrict the choice of p to $p \equiv 2 \bmod 9$ or $p \equiv 5 \bmod 9$, i.e., $p \equiv 2 \bmod 3$ but $p \not\equiv 8 \bmod 9$. For these p the polynomial $(Z^9-1)/(Z^3-1) = Z^6+Z^3+1 \in \mathrm{GF}(p)[Z]$ is irreducible over $\mathrm{GF}(p)$, as follows from the well known result that the t^{th} cyclotomic polynomial $\phi_t(Z)$ is irreducible over $\mathrm{GF}(p)$ if $\mathrm{GF}(t)^*$ is cyclic and generated by $p \bmod t$. The multiplicative group $\mathrm{GF}(t)^*$ is cyclic if and only if either $t=2,4$, or t is a power of an odd prime, or t is twice a power of an odd prime, or t is four times the power of an odd prime that is 2 mod 3. Applying this to $t=9$ and $p \equiv 2,5 \bmod 9$ it follows that $\phi_9(Z) = Z^6+Z^3+1 \in \mathrm{GF}(p)[Z]$ is irreducible over $\mathrm{GF}(p)$.

Let ζ denote a zero of Z^6+Z^3+1. This ζ enables us to conveniently represent elements of $\mathrm{GF}(p^6)$, either using a basis over $\mathrm{GF}(p)$ or using a basis over $\mathrm{GF}(p^2)$.

For the purposes of the present section we use a basis over $\mathrm{GF}(p)$ and write elements of $\mathrm{GF}(p^6)$ as $\sum_{i=0}^{5} a_i\zeta^i$ for $a_i \in \mathrm{GF}(p)$. In this representation elements of the subfield $\mathrm{GF}(p^2)$ of $\mathrm{GF}(p^6)$ correspond to elements of the form $a_3\zeta^3 + a_0$; this follows from $3p^2 \equiv 3 \bmod 9$ and a counting argument. The element $\sum_{i=0}^{5} a_i\zeta^i$ can be written as $(a_5\zeta^6 + a_2\zeta^3)\zeta^{-1} + (a_4\zeta^6 + a_1\zeta^3)\zeta^{-2} + (a_3\zeta^6 + a_0\zeta^3)\zeta^{-3}$. Since $\zeta^3 = \alpha$ with α as above this implies that $\{\zeta^{-1}, \zeta^{-2}, \zeta^{-3}\}$ forms a basis for $\mathrm{GF}(p^6)$ over $\mathrm{GF}(p^2)$, using the representation of $\mathrm{GF}(p^2)$ as used in [4]. Obviously, the latter basis is equivalent to the basis $\{\zeta^2, \zeta, 1\}$ which we found convenient for implementation purposes. This basis simply leads to squaring and multiplication in $\mathrm{GF}(p^6)$ at the cost of 12 and 18 multiplications in $\mathrm{GF}(p)$, respectively. Note that one can move back and forth between the representations of $\mathrm{GF}(p^6)$ at the cost of a small constant number of additions in $\mathrm{GF}(p)$.

None of the above bases is optimal normal. For the calculations in this section that is not a problem, since they had to be carried out just once. For practical applications of XTR it is not a disadvantage either, because in the key recovery application (cf. Section 5) at most three multiplications in $\mathrm{GF}(p^6)$ have to be carried out per XTR key recovery. Note that if $p \bmod 7$ generates $\mathrm{GF}(7)^*$ the polynomial $(X^7 - 1)/(X - 1)$ is irreducible over $\mathrm{GF}(p)$ and leads to an optimal normal basis for $\mathrm{GF}(p^6)$ over $\mathrm{GF}(p)$ (cf. [3]). We chose not to use this representation because it imposes an additional restriction on p without leading to significant advantages.

Lemma 4.1 *The trace over* $\mathrm{GF}(p^2)$ *of* $\sum_{i=0}^{5} a_i\zeta^i \in \mathrm{GF}(p^6)$ *equals* $3(a_3\zeta^3 + a_0) = 3(a_3\alpha + a_0) = -3a_0\alpha^2 + 3(a_3 - a_0)\alpha \in \mathrm{GF}(p^2)$.

Proof. Because the trace is $\mathrm{GF}(p^2)$-linear it suffices to show that the trace of ζ^i is zero for $i = 1, 2, 4, 5$ and $3\zeta^i$ for $i = 0, 3$. This follows trivially from $\zeta^9 = 1$, $\zeta^6 + \zeta^3 + 1 = 0$, and the fact that the trace of ζ^i equals $\zeta^i + \zeta^{ip^2} + \zeta^{ip^4}$.

Lemma 4.2 *For* $x \in \mathrm{GF}(p^6)$ *the trace over* $\mathrm{GF}(p^2)$ *of* x^p *equals the* p^{th} *power of the trace of* x *over* $\mathrm{GF}(p^2)$.

Proof. The trace over $\mathrm{GF}(p^2)$ of x^p equals $x^p + x^{p^3} + x^{p^5}$ which is the p^{th} power of the trace $x + x^{p^2} + x^{p^4}$ of x over $\mathrm{GF}(p^2)$.

A particularly convenient property of our representation of $\mathrm{GF}(p^6)$ is that it enables us to do several calculations without using the specific value of p. The following result is an example.

Proposition 4.3 *Let* $a \in \mathrm{GF}(p)$, *let* ζ *and* $\alpha = \zeta^3$ *be as above, and let* $Q = (p^6 - 1)/(p^2 - p + 1)$. *Then the trace over* $\mathrm{GF}(p^2)$ *of the element* $(\zeta + a)^Q$ *of* $\mathrm{GF}(p^6)$ *of order dividing* $p^2 - p + 1$ *equals*

$$\frac{-3}{a^6 - a^3 + 1}((a^2 - 1)^3\alpha + a^3(a^3 - 3a + 1)\alpha^2)$$

if $p \equiv 2 \bmod 9$ *and the* p^{th} *power thereof if* $p \equiv 5 \bmod 9$, *where* $a^6 - a^3 + 1 \neq 0$.

Proof. If $a^6 - a^3 + 1 = 0$, then $b = a^3$ is a zero in $\mathrm{GF}(p)$ of the sixth cyclotomic polynomial $X^2 - X + 1$. It follows that $b^6 = 1$. With $b^{p-1} = 1$ and $\gcd(p-1, 6) = 2$ we find that $b^2 = 1$ so that $b = \pm 1$. But neither $+1$ nor -1 is a zero of $X^2 - X + 1$, and we conclude that $a^6 - a^3 + 1 \neq 0$.

From $Q = (p^6 - 1)/(p^2 - p + 1) = p^4 + p^3 - p - 1$ it follows that

$$(\zeta + a)^Q = \frac{(\zeta + a)^{p^4}(\zeta + a)^{p^3}}{(\zeta + a)^p(\zeta + a)} = \frac{(\zeta^{p^4} + a)(\zeta^{p^3} + a)}{(\zeta^p + a)(\zeta + a)}.$$

With $\zeta^9 = 1$ this reduces to

$$\frac{(\zeta^7 + a)(\zeta^8 + a)}{(\zeta^2 + a)(\zeta + a)}$$

if $p \equiv 2 \bmod 9$ and to

$$\frac{(\zeta^4 + a)(\zeta^8 + a)}{(\zeta^5 + a)(\zeta + a)}$$

if $p \equiv 5 \bmod 9$. If $p \equiv 5 \bmod 9$ the p^{th} power of the former expression equals the latter, so that if $p \equiv 5 \bmod 9$ the trace of $(\zeta + a)^Q$ equals the p^{th} power of the trace of $(\zeta + a)^Q$ when $p \equiv 2 \bmod 9$ (cf. Lemma 4.2). For the computation of the trace of $(\zeta + a)^Q$ when $p \equiv 2 \bmod 9$ one easily verifies that

$$\frac{a^6 - a^3 + 1}{\zeta + a} = (a^3 - \zeta^3 - 1)(\zeta^2 - a\zeta + a^2)$$

and that

$$\frac{a^6 - a^3 + 1}{\zeta^2 + a} = -a\zeta^5 + (a^3 - 1)\zeta^4 + a^2\zeta^3 - a^4\zeta^2 - \zeta + a^5.$$

With $\zeta^6 + \zeta^3 + 1 = 0$ the trace of

$$\frac{(\zeta^7 + a)(\zeta^8 + a)}{(\zeta^2 + a)(\zeta + a)}$$

then follows from a straightforward computation and Lemma 4.1.

Corollary 4.4 *If $a \neq 0, \pm 1$ then*

$$\frac{-3}{a^6 - a^3 + 1}((a^2 - 1)^3\alpha + a^3(a^3 - 3a + 1)\alpha^2) \in \mathrm{GF}(p^2)$$

is the trace over $\mathrm{GF}(p^2)$ of an element of $\mathrm{GF}(p^6)$ of order dividing $p^2 - p + 1$ and > 3.

Proof. If $p \equiv 2 \bmod 9$ it follows from Proposition 4.3 that there is an $x \in \mathrm{GF}(p^6)^*$ of order dividing $p^2 - p + 1$ with the required trace over $\mathrm{GF}(p^2)$. If $p \equiv 5 \bmod 9$ it follows in the same way, after taking conjugates over $\mathrm{GF}(p)$ and using Lemma 4.2. If the order of x is at most 3, i.e., 1 or 3, then x is either equal to 1, α, or α^2, since $p \equiv 2 \bmod 3$. Thus, the trace of x is equal to 3, 3α, or $3\alpha^2$. For the first possibility, $x = 1$, a trace value of 3 leads to two simultaneous polynomial equations $(a^2-1)^3 - (a^6 - a^3 + 1) = 0$ and $a^3(a^3 - 3a + 1) - (a^6 - a^3 + 1) = 0$; since these polynomials are relatively prime, x cannot be equal to 1. For the other two possibilities, $x = \alpha$ or $x = \alpha^2$, the corresponding trace values lead to $a = 0$ or $a = \pm 1$, respectively, which are excluded by assumption.

It follows from Corollary 4.4 with $a = 2$ and $a = 1/2$ that $(-27\alpha - 24\alpha^2)/19$ and $(27\alpha + 3\alpha^2)/19$, respectively, are trace values of elements of $\mathrm{GF}(p^6)^*$ of order dividing $p^2 - p + 1$ and > 3. This leads to the following algorithm to find $Tr(g)$.

Algorithm 4.5 (Computation of $Tr(g)$)

1. Let $c = (27\alpha + 3\alpha^2)/19 \in \mathrm{GF}(p^2)$ and compute $c_{(p^2-p+1)/q}$ (cf. Section 2).
2. If $c_{(p^2-p+1)/q} \neq 3$, then let $Tr(g) = c_{(p^2-p+1)/q}$ and return success.
3. Otherwise, if $c_{(p^2-p+1)/q} = 3$, then replace c by $(-27\alpha - 24\alpha^2)/19 \in \mathrm{GF}(p^2)$ and recompute $c_{(p^2-p+1)/q}$.
4. If $c_{(p^2-p+1)/q} \neq 3$, then let $Tr(g) = c_{(p^2-p+1)/q}$ and return success.
5. Otherwise, if $c_{(p^2-p+1)/q} = 3$, then return failure.

The probability of failure of Algorithm 4.5 may be expected to be q^{-2}, i.e., negligibly small. If this is a matter of concern, Algorithm 4.5 can trivially be extended and include more 'hard-wired' choices for c (corresponding to $a \neq 0, \pm 1, 2, 1/2$). In the very unlikely event that Algorithm 4.5 fails, which so far has not happened in our test implementation, a different q and p can be selected. On average one may expect that Algorithm 4.5 finds the trace of a generator of the XTR group in about $8\log_2((p^2 - p + 1)/q)$ plus a small constant number of multiplications in $\mathrm{GF}(p)$. This is almost twice as fast as the method based on Algorithm 3.5 (cf. Theorem 3.7), but Algorithm 4.5 applies only to the case $p \not\equiv 8 \bmod 9$.

5 Key size reduction

In this section we show that $Tr(g^{k+1})$ and $Tr(g^{k-1})$ can be derived from $Tr(g)$ and $Tr(g^k)$, assuming the (unknown) private key k is properly chosen. Throughout this section let $c = Tr(g)$ and $c_n = Tr(g^n)$ for $n \in \mathbf{Z}$. We first show that c_{k-1} (or c_{k+1}) follows directly from c, c_k and c_{k+1} (or c_{k-1}) using surprisingly simple formulas.

Theorem 5.1

1. *If $k \neq p, 1-p \bmod q$ then $c^p c_{k-1} - c c_k \neq 0$ and*

$$c_{k+1} = \frac{c_k^p(c^2 - 3c^p) - c_{k-1}^p(c^{2p} - 3c) - c_{k-1}^2 c + c_k^2(c^p - c^2) + c_k c_{k-1} c^{p+1}}{c^p c_{k-1} - c c_k}.$$

2. If $k \neq -p, p-1 \bmod q$ then $cc_{k+1} - c^p c_k \neq 0$ and

$$c_{k-1} = \frac{c_k^p(c^{2p} - 3c) - c_{k+1}^p(c^2 - 3c^p) - c_{k+1}^2 c^p + c_k^2(c - c^{2p}) + c_k c_{k+1} c^{p+1}}{cc_{k+1} - c^p c_k}.$$

Proof. From Corollary 2.3.5.*ii* in [4] it follows that $c^p c_{k-1} - cc_k = Tr(g^{k-2}) - Tr(g^{k+1})$. Thus $c^p c_{k-1} - cc_k$ can only be zero if $Tr(g^{k-2}) = Tr(g^{k+1})$, which implies that g^{k-2} and g^{k+1} are conjugates. Thus, either $k-2 \equiv k+1 \bmod (p^2 - p+1)$, or $k-2 \equiv p^2(k+1) \bmod (p^2-p+1)$, $k-2 \equiv p^4(k+1) \bmod (p^2-p+1)$. The first equation has no solution, the second one leads to $k \equiv p \bmod (p^2 - p + 1)$, and the third one to $k \equiv 1 - p \bmod (p^2 - p + 1)$. Since $k \neq p, 1-p \bmod q$ and q divides $p^2 - p + 1$ we find that $c^p c_{k-1} - cc_k$ is non-zero.

The polynomial $F(c, X)$ is the characteristic polynomial of the matrix $A = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & -c^p \\ 0 & 1 & c \end{pmatrix}$ (cf. Definition 2.4.1 in [4]). That is, the roots g, g^{p-1}, and g^{-p} of $F(c, X)$ are the eigenvalues of A. Thus g^k, $g^{k(p-1)}$, and g^{-kp} are the eigenvalues of the matrix A^k, so that the polynomial $F(c_k, X)$ with roots g^k, $g^{k(p-1)}$, and g^{-kp} is the characteristic polynomial of A^k. From Lemma 2.4.6 in [4] we have that

$$A^k = \begin{pmatrix} Tr(g^2)^p & c^p & 3 \\ c^p & 3 & c \\ 3 & c & Tr(g^2) \end{pmatrix}^{-1} \begin{pmatrix} Tr(g^{k-2}) & c_{k-1} & c_k \\ c_{k-1} & c_k & c_{k+1} \\ c_k & c_{k+1} & Tr(g^{k+2}) \end{pmatrix}.$$

Computing the characteristic polynomial of A^k using this expression, combined with the fact that $Tr(g^{k-2}) = c_{k+1} - cc_k + c^p c_{k-1}$, $Tr(g^{k+2}) = cc_{k+1} - c^p c_k + c_{k-1}$ and $Tr(g^2) = c^2 - 2c^p$ (cf. Corollary 2.3.5.*ii* and *i* in [4]), one obtains a polynomial $\lambda^3 - c_k \lambda^2 + f_1 \lambda + f_0$ with

$$Df_1 = (c^{2p} - 3c)c_{k+1}^2 + (3c^p c_k - 9c_{k-1} + 2c^2 c_k + c^{p+1} c_{k-1} - c^{2p+1} c_k) c_{k+1}$$

$$-3c^p c_{k-1}^2 + 9c_k^2 + c^{3p} c_k^2 + c^3 c_k^2 + c^2 c_{k-1}^2 + 3cc_k c_{k-1} - c^{p+2} c_k c_{k-1} + 2c^{2p} c_k c_{k-1} - 7c^{p+1} c_k^2.$$

Here $D = c^{2p+2} + 18c^{p+1} - 4(c^{3p} + c^3) - 27 \in \mathrm{GF}(p)$ as in Lemma 2.4.4 of [4] and $D \neq 0$ (cf. Lemma 2.4.5 in [4]). Since also $f_1 = c_k^p$ we find that

$$c_{k+1}^2 = (c^{2p} - 3c)^{-1}((-3c^p c_k + 9c_{k-1} - 2c^2 c_k - c^{p+1} c_{k-1} + c^{2p+1} c_k) c_{k+1} - Dc_k^p$$

$$+3c^p c_{k-1}^2 - 9c_k^2 - c^{3p} c_k^2 - c^3 c_k^2 - c^2 c_{k-1}^2 - 3cc_k c_{k-1} + c^{p+2} c_k c_{k-1} - 2c^{2p} c_k c_{k-1} + 7c^{p+1} c_k^2).$$

Note that $c^{2p} - 3c = c^p Tr(g^{-1}) - cTr(1)$, which is non-zero based on the same argument why $c^p c_{k-1} - cc_k$ is non-zero.

Repeating the same argument for the matrix A^{k-1} and its characteristic polynomial $F(c_{k-1}, X)$ (and using Corollary 2.3.5.*ii* of [4] to express $Tr(g^{k-3})$ in terms of c, c_k, c_{k+1}, and c_{k-1}) we obtain another expression for c_{k+1}^2:

$$c_{k+1}^2 = (c^2 - 3c^p)^{-1}(2c^3 c_k - 3cc_{k-1} - c^{p+2} c_{k-1} + 9c_k + 4c^{2p} c_{k-1} - 7c^{p+1} c_k) c_{k+1}$$

$$-Dc_{k-1}^p - c^{2p} c_k^2 - c^4 c_k^2 + 4c^{p+1} c_{k-1}^2 + 6c^p c_k c_{k-1} - 6cc_k^2 + 4c^{p+2} c_k^2$$

$$-c^3c_{k-1}^2 + c^2c_kc_{k-1} - 4c^{2p+1}c_kc_{k-1} - 9c_{k-1}^2 + c^{p+3}c_kc_{k-1}).$$

Here $c^2 - 3c^p$ is non-zero because its conjugate $c^{2p} - 3c$ over $\mathrm{GF}(p)$ is non-zero. Subtraction of the two expressions for c_{k+1}^2 followed by multiplication by $c^{2p} - 3c$ and $c^2 - 3c^p$ and division by D, leads to the formula for c_{k+1}.

For a proof of the second formula, we apply the first one replacing c_k, c_{k+1} and c_{k-1} by $d_{-k} = Tr(g^{-k})$, $d_{-k+1} = Tr(g^{-k+1})$, and $d_{-k-1} = Tr(g^{-k-1})$, respectively. The proof then follows by observing that $c_{k-1}^p = Tr(g^{-k+1}) = d_{-k+1}$, $c_k^p = Tr(g^{-k}) = d_{-k}$ and $c_{k+1}^p = Tr(g^{-k-1}) = d_{-k-1}$ (since $c_n^p = c_{-n}$, cf. Section 2) and by taking the conjugate over $\mathrm{GF}(p)$.

Because p^{th} powering is free in $\mathrm{GF}(p^2)$, computation of the formulas in Theorem 5.1 takes only a small constant number of operations in $\mathrm{GF}(p)$, where the following algorithm can be used for the division.

Algorithm 5.2 (Inversion in $\mathrm{GF}(p^2)$) Let $x = x_1\alpha + x_2\alpha^2 \in \mathrm{GF}(p^2)$. Compute $t = (x_1x_2 + (x_1 - x_2)^2)^{-1} \in \mathrm{GF}(p)$, then $1/x = t(x_2\alpha + x_1\alpha^2) \in \mathrm{GF}(p^2)$.

Theorem 5.1 shows that including both c_{k-1} and c_{k+1} in the XTR public key is never necessary, and that c_{k+1} (or c_{k-1}) suffices (assuming of course that c and c_k are part of the public key). Actually, even c_{k+1} (or c_{k-1}) does in principle not have to be included, because the recipient can determine it by finding the roots of $F(c, X)$ and $F(c_k, X)$, leading to 3 possible representations c_{k+1} (c_{k-1}). Thus, two bits in the public key would suffice to indicate which of the three representations is the correct one, but this would come at the cost of a considerable computation for the recipient of the key.

We now show that if $p \not\equiv 8 \bmod 9$ then the results from Sections 3 and 4 can be used to formulate a fast method to compute c_{k+1} given c and c_k (where, of course, k is unknown) that does not require any additional bits in the public key. The method to compute c_{k-1} given c and c_k is very similar and follows easily from the method for c_{k+1}. Roughly speaking the method works as suggested above, namely by computing explicit representations of g and g^k in $\mathrm{GF}(p^6) = \mathrm{GF}(p)[X]/(X^6 + X^3 + 1)$ (cf. Section 4) based on their representations c and c_k, respectively, so that the value of c_{k+1} follows as the trace over $\mathrm{GF}(p^2)$ of $g * g^k \in \mathrm{GF}(p^6)$.

More precisely, the owner of the private key k computes $c_k = Tr(g^k)$ given $c = Tr(g)$ and k. The same c_k is obtained for $kp^2 \bmod q$ and $kp^4 \bmod q$ since g^k, g^{kp^2}, and g^{kp^4} are conjugates over $\mathrm{GF}(p^2)$ and thus have the same trace over $\mathrm{GF}(p^2)$, namely c_k. As a side result of the computation of c_k, the owner of the private key obtains $c_{k+1} = Tr(g^{k+1})$ (cf. Section 2). However, the value c_{k+1} thus obtained is in general not the same as the value that would be obtained for $kp^2 \bmod q$ or $kp^4 \bmod q$, because $Tr(g^{k+1})$, $Tr(g^{kp^2+1})$, and $Tr(g^{kp^4+1})$ are not the same unless $k = 0 \bmod q$, despite the fact that $Tr(g^k)$, $Tr(g^{kp^2})$, and $Tr(g^{kp^4})$ are the same. This is because g^{k+1}, g^{kp^2+1}, and g^{kp^4+1} are not conjugates over $\mathrm{GF}(p^2)$ unless $k = 0 \bmod q$, despite the fact that g^k, g^{kp^2}, and g^{kp^4} are conjugates over $\mathrm{GF}(p^2)$. It follows that for any pair (c, c_k) there are three possible different values for c_{k+1}: one that corresponds to the proper secret value k, and two that correspond to the 'wrong' values $kp^2 \bmod q$ and $kp^4 \bmod q$

Any method to recover c_{k+1} from (c, c_k) will have to resolve this ambiguity. To do this without requiring additional bits in the public key we do the following. The owner of the private key computes not only $Tr(g^{k+1})$, but $Tr(g^{kp^2+1})$ and $Tr(g^{kp^4+1})$ as well. Next he selects the secret key k as k, $kp^2 \bmod q$, or $kp^4 \bmod q$ depending on which of the three values $Tr(g^{k+1})$, $Tr(g^{kp^2+1})$, $Tr(g^{kp^4+1})$ is the 'smallest' (or 'largest')[3]. It follows that c_{k+1} is the 'smallest' possibility given the pair (c, c_k). Obviously this way of changing an initially selected private key value k does not have a negative impact on the security.

How this method enables the recipient of the pair (c, c_k) to compute the proper c_{k+1} without knowing k is described in Algorithm 5.6 below. We first describe how the owner of the private key computes $Tr(g^{k+1})$, $Tr(g^{kp^2+1})$, and $Tr(g^{kp^4+1})$. A conceptually straightforward method would be for the owner of the private key to compute c_m three times, once for $m = k$ itself, once for $m = kp^2 \bmod q$, and once for $m = kp^4 \bmod q$, and to pick the k corresponding to the smallest c_{m+1} (the three c_m's are the same, as noted above). A more complicated but faster method is as follows. Suppose that (c_{k-1}, c_k, c_{k+1}) and $(c_{-p-1}, c_{-p}, c_{-p+1})$ have been computed, at the cost of $16\log_2(q)$ multiplications in GF(p) (cf. Section 2). The values $c_{k\pm2}$ can then easily be obtained and $c_2 = c^2 - 2c^p$ (cf. [4]). To compute $Tr(g^{kp^2+1})$ we observe that $Tr(g^{kp^2+1}) = Tr(g^{kp^2-p^3}) = Tr(g^{(k-p)p^2}) = Tr(g^{k-p})$. We then use Lemmas 2.4.2 and 2.4.5 from [4] and find that

$$\begin{pmatrix} Tr(g^{k-p-1}) \\ Tr(g^{k-p}) \\ Tr(g^{k-p+1}) \end{pmatrix}^T = \begin{pmatrix} c_{-p-1} \\ c_{-p} \\ c_{-p+1} \end{pmatrix}^T \begin{pmatrix} c_2^p & c^p & 3 \\ c^p & 3 & c \\ 3 & c & c_2 \end{pmatrix}^{-1} \begin{pmatrix} c_{k-2} & c_{k-1} & c_k \\ c_{k-1} & c_k & c_{k+1} \\ c_k & c_{k+1} & c_{k+2} \end{pmatrix},$$

so that $Tr(g^{kp^2+1})$ follows after a small constant number of multiplications in GF(p). A similar matrix identity involving (c_{p-1}, c_p, c_{p+1}) (obtained using $c_{-n} = c_n^p$, cf. Section 2) is used to compute $Tr(g^{kp^2-1}) = Tr(g^{k+p})$. Given $(Tr(g^{kp^2-1}), Tr(g^{kp^2}), Tr(g^{kp^2+1}))$ (with $Tr(g^{kp^2}) = c_k$) and $(c_{-p-1}, c_{-p}, c_{-p+1})$, the same method is then used to compute $Tr(g^{kp^4+1})$.

The corresponding method to compute the 'smallest' c_{k+1} given just (c, c_k) but without knowing the secret k relies on Algorithm 3.1, Scipione del Ferro's method. We need two auxiliary algorithms, the correctness of which follows by inspection (cf. [Lemma 2.1.1] in [4]).

Algorithm 5.3 (Exponentiation in GF(p^2)) Let $x \in \mathrm{GF}(p^2)$ and let e be an integer. To compute $x^e \in \mathrm{GF}(p^2)$ do the following.

[3] For $x \in \mathrm{GF}(p)$ let $\pi_0(x) \in \{0, 1, \ldots, p-1\}$ be the image of x under the 'natural' bijection between GF(p) and $\{0, 1, \ldots, p-1\}$. For $x = x_1\alpha + x_2\alpha^2 \in \mathrm{GF}(p^2)$, using the representation of elements of GF(p^2) from [4], let $\pi(x) = \pi_0(x_1) + p * \pi_0(x_2)$. We use the ordering on GF(p^2) induced by π.

1. Compute $e_0, e_1 \in \{0, 1, \ldots, p-1\}$ such that $e_0 + e_1 p = e \bmod (p^2 - 1)$ and let $e_i = \sum_j e_{ij} 2^j$, with $e_{ij} \in \{0, 1\}$ for $i = 0, 1$ and $j \geq 0$, be the binary representations of e_0 and e_1.
2. Let n be the largest index such that $e_{in} \neq 0$ for $i = 0$ or 1.
3. Compute $x' = x * x^p \in \mathrm{GF}(p)$.
4. Let $y = 1$ in $\mathrm{GF}(p^2)$. For $j = n, n-1, \ldots, 0$ in succession do the following:
   - if $e_{0j} = 1$ and $e_{ij} = 1$, then replace y by $y * x'$;
   - if $e_{0j} = 1$ and $e_{1j} = 0$, then replace y by $y * x$;
   - if $e_{0j} = 0$ and $e_{1j} = 1$, then replace y by $y * x^p$;
   - if $j > 0$, then replace y by y^2.
5. Return $y = x^e \in \mathrm{GF}(p^2)$.

Lemma 5.4 *The expected cost of Algorithm 5.3 is* $4 \log_2(p)$ *multiplications in* $\mathrm{GF}(p)$.

Algorithm 5.5 (Cube root in $\mathrm{GF}(p^2)$ if $p \not\equiv 8 \bmod 9$) To compute a cube root in $\mathrm{GF}(p^6)$ of $r \in \mathrm{GF}(p^2)$ perform the following steps.

1. Use Algorithm 5.3 to compute $t = r^{(8p^2-5)/9} \in \mathrm{GF}(p^2)$ if $p \equiv 2 \bmod 9$ or $t = r^{(p^2+2)/9} \in \mathrm{GF}(p^2)$ if $p \equiv 5 \bmod 9$.
2. Compute $s = t^3 \in \mathrm{GF}(p^2)$ and determine $j = 0, 1$ or 2 such that $\alpha^j s = r$.
3. Return a cube root $\zeta^j t \in \mathrm{GF}(p^6)$ of r (the result is in $\mathrm{GF}(p^2)$ if $j = 0$).

Algorithm 5.6 (Key recovery) To compute the 'smallest' c_{k+1} corresponding to (c, c_k), perform the following steps.

1. Use Algorithm 3.1 to compute a root $g \in \mathrm{GF}(p^6) = \mathrm{GF}(p)[X]/(X^6 + X^3 + 1)$ of the polynomial $F(c, X)$, using Algorithm 5.5 to compute a cube root in Step 3. Note that Algorithm 5.2 can be used for the division by u in Step 3, since u is a $\mathrm{GF}(p^2)$-multiple of a power of ζ.
2. Use Algorithm 3.1 to compute the three roots $y_1, y_2, y_3 \in \mathrm{GF}(p^6)$ of $F(c_k, X)$, with $w = \alpha$ in Step 4.
3. For $i = 1, 2, 3$ compute the trace t_i over $\mathrm{GF}(p^2)$ of $g y_i \in \mathrm{GF}(p^6)$ (cf. Lemma 4.1).
4. Let c_{k+1} be the 'smallest' of t_1, t_2, and t_3.

Theorem 5.7 *Algorithm 5.6 can be expected to require* $10.6 \log_2(p)$ *multiplications in* $\mathrm{GF}(p)$.

Proof. The square-root computation in Step 2 of Algorithm 3.1 can be expected to require $1.3 \log_2(p)$ multiplications in $\mathrm{GF}(p)$ (cf. Proof of Theorem 3.7). The application of Algorithm 5.5 in Step 3 of Algorithm 3.1 requires a call to Algorithm 5.3, at an expected cost of $4 \log_2(p)$ multiplications in $\mathrm{GF}(p)$ (cf. Lemma 5.4). Thus, a single call to Algorithm 3.1 can be expected to require $5.3 \log_2(p)$ multiplications in $\mathrm{GF}(p)$, from which the proof follows.

We conclude that $Tr(g^{k-1})$ and $Tr(g^{k+1})$ do not have to be included in the XTR public key data $(p, q, Tr(g), Tr(g^k))$ for digital signature or authentication applications, if

1. the owner of the private key has selected its private exponent k in the proper fashion as explained above, and if
2. the recipient of the public key is willing and able to perform Algorithm 5.6 to compute $Tr(g^{k+1})$ followed by an application of Theorem 5.1 to compute $Tr(g^{k-1})$.

To summarize, there are three options for XTR public keys used for digital signatures or authentication, namely to include one, two, or all three of the values $Tr(g^{k-1})$, $Tr(g^k)$, $Tr(g^{k+1})$. In some applications, e.g. issuance of a certificate by a Certificate Authority, it may be required that the relative correctness of these components can be verified by a third party. A method to do this will be published at a later date (cf. [5]).

Acknowledgment. The method from Section 4 is based on a more general argument from H.W. Lenstra, Jr. We gratefully acknowledge his assistance.

References

1. A.E. Brouwer, R. Pellikaan, E.R. Verheul, *Doing more with fewer bits*, Proceedings Asiacrypt99, LNCS 1716, Springer-Verlag 1999, 321-332.
2. H. Cohen, A. Miyaji, T. Ono, *Efficient elliptic curve exponentiation using mixed coordinates*, Proceedings Asiacrypt'98, LNCS 1514, Springer-Verlag 1998, 51-65.
3. A.K. Lenstra, *Using cyclotomic polynomials to construct efficient discrete logarithm cryptosystems over finite fields*, Proceedings ACISP97, LNCS 1270, Springer-Verlag 1997, 127-138. 1998, 1-10.
4. A.K. Lenstra, E.R. Verheul, *The XTR public key system*, Proceedings Crypto 2000, LNCS 1880, Springer-Verlag 2000, 1-20; available from www.ecstr.com.
5. A.K. Lenstra, E.R. Verheul, *Fast irreducibility testing for XTR*, in preparation.
6. W.K. Nicholson, *Introduction to abstract algebra*, PWS-Kent Publishing Company, Boston, 1993.