Analogues to the Gong-Harn and XTR Cryptosystems

Kenneth Giuliani[1] and Guang Gong[2]

[1] Dept. of C&O
University of Waterloo
Waterloo, ON, Canada, N2L 3G1
kjgiulia@cacr.math.uwaterloo.ca
[2] Dept. of E&CE
University of Waterloo
Waterloo, ON, Canada, N2L 3G1
ggong@cacr.math.uwaterloo.ca

Abstract. This paper proposes new discrete logarithm-based cryptographic systems which are analogues of the Gong-Harn and XTR cryptosystems. The systems are based upon fifth-order characteristic sequences over $GF(q)$ and are shown to be equivalent to systems based on the multiplicative group $GF(q^5)^*$.

Keywords: Cryptosystem, characteristic sequences, XTR, Gong-Harn, supersingular elliptic curves.

1 Introduction

In 1999, Gong and Harn [2] proposed a cryptographic system based on third-order characteristic sequences over a field $GF(q)$. This system was based upon the difficulty of solving discrete logarithms over $GF(q^3)$, but computations took place in the field $GF(q)$. In 2000, Lenstra and Verheul [3] proposed the XTR cryptosystem which is based on the discrete logarithm problem in $GF(p^6)$ for a prime p, but where computation occurred in the field $GF(p)$. It turns out that XTR may be viewed as a cryptosystem using a special type of third-order characteristic sequence over $GF(p^2)$ so that parameters are optimized for efficiency.

This paper proposes analogues to both the Gong-Harn and XTR cryptosystems using fifth-order characteristic sequences. Section 2 will give a quick review of third- and fifth-order characteristic sequences and section 3 will give a quick review of the Gong-Harn and XTR cryptosystems. In sections 4 and 5, we demonstrate relations for fifth-order characteristic sequences and investigate ternary expansions which are needed in the cryptosystems presented in section 6. In section 7, we prove the equivalence of discrete logarithms for sequences and discrete logarithms over $GF(q^5)$ and deal with other security concerns such as uses of supersingular elliptic curve. Section 8 summarizes and suggests some areas for future work.

2 Characteristic Sequences

Let q is a prime power, $n \geq 3$ an odd positive integer and

$$f(x) = x^n - a_1 x^{n-1} + \cdots + a_{n-1} x - 1$$

a monic irreducible polynomial of degree n over $GF(q)$. A sequence $\mathbf{s} = \{s_k\}$ is said to be an n-th order LFSR sequence over $GF(q)$ with characteristic polynomial $f(x)$ if the elements of $\mathbf{s}$ satisfy

$$s_k = a_1 s_{k-1} - a_2 s_{k-2} + \cdots - a_{n-1} s_{k-n+1} + s_{k-n} \tag{1}$$

for all $k \geq n$.

Let α be a root of f in $GF(q^n)$. Then the n roots of f are $\alpha_i = \alpha^{q^i}$ for $i = 0, \ldots, n-1$. Suppose now $\mathbf{s}$ is given the initial state

$$s_k = \alpha_0^k + \alpha_1^k + \cdots + \alpha_{n-1}^k$$

for $k = 0, \ldots, n-1$. It can be shown [4] that using the recurrence (1) with this initial state will produce the sequence $\mathbf{s}$ such that

$$s_k = \alpha_0^k + \alpha_1^k + \cdots + \alpha_{n-1}^k$$

for all $k \geq 0$. This sequence is called the *n-th order characteristic sequence generated by $f(x)$ over $GF(q)$*.

Let $per(f)$ denote the period of $\mathbf{s}$. In [4], it is shown that the $per(f)$ is the same as the order of $\alpha \in GF(q^n)$. From the constant term of f, we get the relation

$$\alpha_0 \cdots \alpha_{n-1} = 1 \tag{2}$$

But since $\alpha_i = \alpha^{q^i}$, we see that α must have order dividing $1 + q + \cdots + q^{n-1} = \frac{q^n-1}{q-1}$. Hence, so does $per(f)$. If $Q = per(f)$, we shall use the convention $s_{-k} = s_{Q-k}$ for all $k \in \mathbb{Z}$.

Observe that for each $l = 1, \ldots, n-1$,

$$a_l = \sum_{0 \leq i_1 < i_2 < \cdots < i_l \leq n-1} \alpha_{i_1} \alpha_{i_2} \cdots \alpha_{i_l} \tag{3}$$

is simply the l-th symmetric polynomial in the α_i's. Using (2), we can change (3) into

$$a_l = \sum_{0 \leq i_1 < i_2 < \cdots < i_{n-l} \leq n-1} \alpha_{i_1}^{-1} \alpha_{i_2}^{-1} \cdots \alpha_{i_{n-l}}^{-1} \tag{4}$$

which is the $(n-l)$-th symmetric polynomial in the inverses of the α_i's. Now, if f is irreducible over $GF(q)$, then its reciprocal polynomial

$$\hat{f}(x) = x^n - a_{n-1} x^{n-1} + \cdots + a_1 x - 1$$

is also irreducible over $GF(q)$. The characteristic sequence $\hat{\mathbf{s}} = \{\hat{s}_k\}$ generated by $\hat{f}$ satisfies $\hat{s}_k = \alpha_0^{-k} + \cdots \alpha_{n-1}^{-k} = s_{-k}$ for all k.

This can be generalized for any power α^m. Let

$$\bar{a}_l = \sum_{0 \leq i_1 < i_2 < \cdots < i_l \leq n-1} \alpha_{i_1}^m \alpha_{i_2}^m \cdots \alpha_{i_l}^m$$

be the l-th symmetric polynomial in the terms α_i^m for $i = 0, \ldots n-1$. Then the sequence $\{\bar{s}_k\}$ generated from the polynomial

$$f_m(x) = x^n - \bar{a}_1 x^{n-1} + \cdots + \bar{a}_{n-1} x - 1$$

satisfies

$$\bar{s}_k = \alpha_0^{km} + \cdots + \alpha_{n-1}^{km} = s_{km} \tag{5}$$

In this paper, we shall be interested in third- and fifth-order characteristic sequences. In these cases, the above polynomials and initial states reduce to

$$f(x) = x^3 - ax^2 + bx - 1$$

with initial state $s_0 = 3$, $s_1 = a$, and $s_2 = a^2 - 2b$ in the third-order case and

$$f(x) = x^5 - ax^4 + bx^3 - cx^2 + dx - 1$$

with initial state $s_0 = 5$, $s_1 = a$, $s_2 = a^2 - 2b$, $s_3 = a^3 - 3ab + 3c$, and $s_4 = a^4 - 4a^2b + 2b^2 - 4d + 4ac$.

Properties of third-order sequences were examined in [2].

Lemma 1. *Let $\{s_k\}$ be a third-order characteristic sequence. Then for any integers n and m,*

1. $s_{2n} = s_n^2 - 2s_{-n}$
2. $s_n s_m - s_{n-m} s_{-m} = s_{n+m} - s_{n-2m}$

Substituting $n = k+1$ and $m = k$ and rearranging the terms gives

$$s_{2k+1} = s_{k+1} s_k - a s_{-k} + s_{-k+1} \tag{6}$$

while substituting $n = k-1$ and $m = k$ gives

$$s_{2k-1} = s_{k-1} s_k - b s_{-k} + s_{-k-1} \tag{7}$$

3 The Gong-Harn and XTR Cryptosystems

The Gong-Harn [2] and XTR [3] cryptosystems are both based on third-order characteristic sequences and thus, although discovered independently, are very similar. The essentials of each cryptosystem are presented in this section. For full descriptions, the reader is referred to the original papers.

3.1 The Gong-Harn Cryptosystem

The Gong-Harn cryptosystem [2] was originally proposed using sequences over $GF(p)$ where p is a prime. However, all of its traits carry over when considering sequences over $GF(q)$ where $q = p^r$ is a prime power. Throughout this subsection, we shall consider the third-order characteristic sequence $\{s_k\}$ generated from the polynomial $f(x) = x^3 - ax^2 + bx - 1$ whose period Q is a divisor of $q^2 + q + 1$.

First, we require a method of efficiently calculating any term s_m given m, $0 \leq m < Q$. This can be done using an analogue to "square-and-multiply" (or "double-and-add") with the formulae presented in the previous section. More specifically, we calculate terms three at a time using the first three terms s_0, s_1, and s_2 which we know, along with the "doubling" formulae

$$\begin{aligned} s_{2k-1} &= s_k s_{k-1} - a s_{-k} + s_{-k+1} \\ s_{2k} &= s_k^2 - 2s_{-k} \\ s_{2k+1} &= s_{k+1} s_k - b s_{-k} + s_{-k-1} \end{aligned}$$

and the "doubling-and-adding" formulae

$$\begin{aligned} s_{2k} &= s_k^2 - 2s_{-k} \\ s_{2k+1} &= s_{k+1} s_k - b s_{-k} + s_{-k-1} \\ s_{2k+2} &= s_{k+1}^2 - 2s_{-k-1} \end{aligned}$$

One may notice that to perform these calculations, we require the terms involving $-k$. We can obtain these from the reciprocal sequence. In particular, we start with the terms $s_0 = 3$, $s_{-1} = b$, and $s_{-2} = b^2 - 2a$, and use the doubling formulae

$$\begin{aligned} s_{-2k+1} &= s_{-k} s_{-k+1} - b s_k + s_{k-1} \\ s_{-2k} &= s_{-k}^2 - 2s_k \\ s_{-2k-1} &= s_{-k-1} s_{-k} - a s_k + s_{k+1} \end{aligned}$$

and the "doubling-and-adding" formulae

$$\begin{aligned} s_{-2k} &= s_{-k}^2 - 2s_k \\ s_{-2k-1} &= s_{-k-1} s_{-k} - a s_k + s_{k+1} \\ s_{-2k-2} &= s_{-k-1}^2 - 2s_{k+1} \end{aligned}$$

Note that the terms s_{2k-1} to s_{2k+2} and s_{-2k+1} to s_{-2k-2} can be obtained from s_{k-1}, s_k, s_{k+1}, s_{-k+1}, s_k, and s_{-k-1}. Thus, s_m and s_{-m} can be calculated in $O(\log Q)$ operations in $GF(q)$.

Alice can thus use these sequences as a discrete-logarithm based system by choosing a secret key m such that $0 \leq m < Q$, and then calculating the public key (s_m, s_{-m}). If Bob wishes to perform a Diffie-Hellman key exchange using this public key, he can form the third-order characteristic sequence $\{\hat{s}_k\}$ from the polynomial $x^3 - s_m x^2 + s_{-m} x - 1$ and choosing an e, $0 \leq e < Q$, calculates the terms $\hat{s}_e$ and $\hat{s}_{-e}$. From (5), these terms are actually s_{em} and s_{-em} which Alice can also compute if Bob sends her his public key (s_e, s_{-e}). An ElGamal-type of encryption scheme can similarly be done.

It can be shown that solving the DLP or DHP of these sequences is computationally equivalent to solving the DLP or DHP in the subgroup of order Q in $GF(q^3)^*$. The reader is referred to the original paper for a proof.

3.2 The XTR Cryptosystem

The XTR cryptosystem [3] was originally proposed in the context of using the trace represention of finite field elements to represent them efficiently and compactly. However, the representation and formulae given in the original paper are essentially the same as those of third-order characteristic sequences. The major difference between XTR and Gong-Harn is that XTR was specifically presented using the field $GF(p^2)$ and the polynomial $f(x) = x^3 - ax^2 + a^p x - 1$. This was done to optimize the efficiency of calculation and representation.

Let us first note that if we choose $q = p^2$, then a third-order characteristic sequence has order Q dividing $q^2 + q + 1 = p^4 + p^2 + 1 = (p^2 + p + 1)(p^2 - p + 1)$. Note that the subgroup of order $p^2 - p + 1$ within $GF(p^6)$ is not contained in any proper subfield. To avoid an index calculus attack, we prefer choosing sequences corresponding to this subgroup. Thus, Q should divide $p^2 - p + 1$. If α is a root of the polynomial $f(x) = x^3 - ax^2 + bx - 1$, then $a = \alpha + \alpha^{p^2} + \alpha^{p^4}$ and $b = \alpha^{-1} + \alpha^{-p^2} + \alpha^{-p^4}$. But, using the relation $\alpha^{p^2 - p + 1} = 1$, we get

$$b = \alpha^{-1} + \alpha^{-p^2} + \alpha^{-p^4} = \alpha^p + \alpha^{p^3} + \alpha^{p^5} = a^p$$

Let $\{s_k\}$ be the sequence generated from this polynomial. Using this same relation, we see that

$$s_{-k} = \alpha^{-k} + \alpha^{-kp^2} + \alpha^{-kp^4} = (\alpha^k)^p + (\alpha^{kp^2})^p + (\alpha^{kp^4})^p = s_k^p$$

Hence, we can restrict the public key to s_m and need not calculate the negative terms s_{-k} when performing calculations.

Finally, we shall note that, in the original paper, Lenstra and Verheul present a method for choosing p and a representation of $GF(p^2)$ so that all computation takes place over $GF(p)$.

4 Relations for Fifth-Order Characteristic Sequences

In the case of third-order sequences, we obtained relations which described sequence elements in the ranges s_{2k-1} to s_{2k+2} and s_{-2k+1} to s_{-2k-2} in terms of elements in the ranges s_{k-1} to s_{k+1} and s_{-k+1} to s_{-k-1}. This led to an efficient method for calculating any term in the sequence. We would like to obtain similar relations for fifth-order sequences. To that end, let q be a prime power and

$$f(x) = x^5 - ax^4 + bx^3 - cx^2 + dx - 1$$

generate a fifth-order characteristic sequence $\{s_k\}$ over $GF(q)$. Let α be a root of f in $GF(q^5)$. Then α has order Q dividing $q^4+q^3+q^2+q+1$ which is the same as the period of f. Now $\alpha_i = \alpha^{q^i}$ for $i=0,\ldots,4$ are all the roots of f. We have

$$s_k = \alpha_0^k + \alpha_1^k + \alpha_2^k + \alpha_3^k + \alpha_4^k$$

Also, $a = s_1$ and $d = s_{-1}$.

A major difference we must account for is the fact that there is a second sequence demanding attention. Namely, the sequence $\{t_k\}$ where

$$t_k = \sum_{0 \le i_1 < i_2 \le 4} \alpha_{i_1}^k \alpha_{i_2}^k$$

which is the degree 2 symmetric polynomial in the α_i's. Observe that we have $b = t_1$ and $c = t_{-1}$. Notice that the minimal polynomial of α^m is

$$f_m(x) = x^5 - s_m x^4 + t_m x^3 - t_{-m} x^2 + s_{-m} x - 1$$

As we must form this polynomial to do things such as Diffie-Helman key agreements, it is very important for us to be able to efficiently calculate terms in this second sequence as well.

It appears difficult to obtain relations which isolate terms of the form s_{2k+1} and t_{2k+1}, the keys to the doubling-and-adding algorithm. However, a method has been found to obtain "tripling" relations for terms of the form s_{3k+1} and t_{3k+1}. In the next section, we shall present a tripling analogue to double-and-add which allows us to calculate s_m, t_m, t_{-m}, and s_{-m} for any m, $0 \le m < Q$.

Lemma 2. *Let n and m be integers. Then*

1. $s_{2n} = s_n^2 - 2t_n$
2. $t_{2n} = t_n^2 + 2s_{-n} - 2s_n t_{-n}$
3. $s_{3n} = s_n^3 - 3s_n t_n + 3t_{-n}$
4. $t_{3n} = t_k^3 - 3s_{-k}t_k - 3s_k t_k t_{-k} + 3s_k^2 s_{-k} + 3t_{-k}^2 - 3s_k$
5. $s_{n+2m} = s_{n+m}s_m - s_n t_m + s_{n-m}t_{-m} - s_{n-2m}s_{-m} + s_{n-3m}$
6. $t_n t_m - s_{-m}t_{n-m} + 3t_{n+m} = s_n s_m s_{n+m} - s_{n-2m}s_{n-m} + s_{2n-3m} - s_{n+2m}s_n - s_{2n+m}s_m + s_{n+m}^2$
7. $s_n = as_{n-1} - bs_{n-2} + cs_{n-3} - ds_{n-4} + s_{n-5}$

Proof.

$$s_n^2 = (\sum_{i=0}^{4} \alpha_i^n)^2 = \sum_{i=0} \alpha_i^{2n} + \sum_{0 \le i_1 < i_2 \le 4} 2\alpha_{i_1}\alpha_{i_2} = s_{2n} + 2t_n$$

This proves 1. The remaining assertions can similarly be proven. □

n	m	Relation
$k-1$	$k+1$	$s_{3k+1} = s_{2k}s_{k+1} - s_{k-1}t_{k+1} + s_{-2}t_{-k-1} - s_{-k-3}s_{-k-1} + s_{2(-k-2)}$
$k+1$	$k-1$	$s_{3k-1} = s_{2k}s_{k-1} - s_{k+1}t_{k-1} + s_{2}t_{-k+1} - s_{-k+3}s_{-k+1} + s_{2(-k+2)}$
$k-2$	k	$s_{3k-2} = s_{2(k-1)}s_{k} - s_{k-2}t_{k} + s_{-2}t_{-k} - s_{-k-2}s_{-k} + s_{2(-k-1)}$
$k+2$	k	$s_{3k+2} = s_{2(k+1)}s_{k} - s_{k+2}t_{k} + s_{2}t_{-k} - s_{-k+2}s_{-k} + s_{2(-k+1)}$

Table 1. Preliminary relations obtained from lemma 2.5.

4.1 Relation for the Sequence $\{s_k\}$

We can obtain relations by considering n and m as functions of k in lemma 2.5. Table 1 gives a list of values for n and m and the relations obtained from them.

Lemma 2.1 can be used to eliminate all of the doubling terms in Table 1 while lemma 2.7 can be used to eliminate s_{-k-3} and s_{-k+3}. Using this and lemma 2.3 gives us the needed relations as displayed in Table 2.

We can also replace k by $-k$ to obtain relations for the negative terms. Hence we determine s_{3k-3} to s_{3k+3} and s_{-3k+3} to s_{-3k-3} from the terms s_{k-2} to s_{k+2}, s_{-k+2} to s_{-k-2}, t_{k-2} to t_{k+2}, and t_{-k+2} to t_{-k-2}.

$$
\begin{aligned}
s_{3k+3} &= s_{k+1}^3 - 3s_{k+1}t_{k+1} + 3t_{-k-1}\\
s_{3k+2} &= s_{k+1}^2 s_k - 2t_{k+1}s_k - s_{-k+2}s_{-k} - s_{k+2}t_k + s_2t_{-k} + s_{-k+1}^2 - 2t_{-k+1}\\
s_{3k+1} &= s_k^2 s_{k+1} - 2t_k s_{k+1} - ds_{-k-2}s_{-k-1} + cs_{-k-1}^2 - bs_{-k}s_{-k-1} + as_{-k+1}s_{-k-1}\\
&\quad - s_{-k+2}s_{-k-1} + s_{-2}t_{-k-1} - s_{k-1}t_{k+1} + s_{-k-2}^2 - 2t_{-k-2}\\
s_{3k} &= s_k^3 - 3s_kt_k + 3t_{-k}\\
s_{3k-1} &= s_k^2 s_{k-1} - 2t_k s_{k-1} - as_{-k+2}s_{-k+1} + bs_{-k+1}^2 - cs_{-k}s_{-k+1} + ds_{-k-1}s_{-k+1}\\
&\quad - s_{-k-2}s_{-k+1} + s_2t_{-k+1} - s_{k+1}t_{k-1} + s_{-k+2}^2 - 2t_{-k+2}\\
s_{3k-2} &= s_{k-1}^2 s_k - 2t_{k-1}s_k - s_{-k-2}s_{-k} - s_{k-2}t_k + s_{-2}t_{-k} + s_{-k-1}^2 - 2t_{-k-1}\\
s_{3k-3} &= s_{k-1}^3 - 3s_{k-1}t_{k-1} + 3t_{-k+1}
\end{aligned}
$$

Table 2. Relations for the sequence $\{s_k\}$

4.2 Relations for the Sequence $\{t_k\}$

We can obtain relations by considering n and m as functions of k in lemma 2.6. Table 3 gives a list of values for n and m and the relations obtained from them.

The terms of concern are those with a coefficient of 4 or 5 in front of k. However, we can eliminate these with the supplementary relations in Table 4 obtained from lemma 2.5.

n	m	Relation
$2k$	$k+1$	$3t_{3k+1} = s_{-k-1}t_{k-1} - t_{2k}t_{k+1} + s_{2k}s_{k+1}s_{3k+1} - s_{-2}s_{k-1} + s_{k-3}$ $- s_{4k+2}s_{2k} - s_{5k+1}s_{k+1} + s_{3k+1}^2$
$2k$	$k-1$	$3t_{3k-1} = s_{-k+1}t_{k+1} - t_{2k}t_{k-1} + s_{2k}s_{k-1}s_{3k-1} - s_2 s_{k+1} + s_{k+3}$ $- s_{4k-2}s_{2k} - s_{5k-1}s_{k-1} + s_{3k-1}^2$
$k-2$	$2k$	$3t_{3k-2} = s_{-2k}t_{-k-2} - t_{k-2}t_{2k} + s_{k-2}s_{2k}s_{3k-2} - s_{-3k-2}s_{-k-2}$ $+ s_{4(-k-1)} - s_{5k-2}s_{k-2} - s_{4(k-1)}s_{2k} + s_{3k+2}^2$
$k+2$	$2k$	$3t_{3k+2} = s_{-2k}t_{-k+2} - t_{k+2}t_{2k} + s_{k+2}s_{2k}s_{3k+2} - s_{-3k+2}s_{-k+2}$ $+ s_{4(-k+1)} - s_{5k+2}s_{k+2} - s_{4(k+1)}s_{2k} + s_{3k+2}^2$

Table 3. Preliminary relations obtained from lemma 2.6.

n	m	Relation
$2k-2$	$-k-1$	$s_{5k+1} = s_{4k}s_{k+1} - s_{3k-1}t_{k+1} + s_{2(k-1)}t_{-k-1} - s_{k-3}s_{-k-1} + s_{-4}$
$2k+2$	$-k+1$	$s_{5k-1} = s_{4k}s_{k-1} - s_{3k+1}t_{k-1} + s_{2(k+1)}t_{-k+1} - s_{k+3}s_{-k+1} + s_4$
$3k+2$	k	$s_{5k+2} = s_{4k+2}s_k - s_{3k+2}t_k + s_{2(k+1)}t_{-k} - s_{k+2}s_{-k} + s_2$
$3k-2$	k	$s_{5k-2} = s_{4k-2}s_k - s_{3k-2}t_k + s_{2(k-1)}t_{-k} - s_{k-2}s_{-k} + s_{-2}$
$2k+2$	k	$s_{4k+2} = s_{3k+2}s_k - s_{2k+2}t_k - s_{k+2}t_{-k} - s_2 s_{-k} + s_{-k+2}$
$2k-2$	k	$s_{4k-2} = s_{3k-2}s_k - s_{2k-2}t_k - s_{k-2}t_{-k} - s_{-2}s_{-k} + s_{-k-2}$

Table 4. Supplementary relations obtained from lemma 2.5.

Properties 1 and 2 of lemma 2 can be used to eliminate all of the doubling terms in Tables 3 and 4 while lemma 2.7 can be used to eliminate s_{k-3} and s_{k+3}. Thus after all of the substitutions, we have the needed relations as displayed in Table 5. Note again that replacing k by $-k$ will again give the corresponding negative terms.

$$
\begin{aligned}
t_{3k+3} &= t_{k+1}^3 - 3s_{-k-1}t_{k+1} - 3s_{k+1}t_{k+1}t_{-k-1} + 3s_{k+1}^2 s_{-k-1} + 3t_{-k-1}^2 - 3s_{k+1} \\
t_{3k+2} &= (s_{-2k}t_{-k+2} - t_{k+2}t_{2k} + s_{k+2}s_{2k}s_{3k+2} - s_{-3k+2}s_{-k+2} \\
&\quad + s_{4(-k+1)} - s_{5k+2}s_{k+2} - s_{4(k+1)}s_{2k} + s_{3k+2}^2)/3 \\
t_{3k+1} &= (s_{-k-1}t_{k-1} - t_{2k}t_{k+1} + s_{2k}s_{k+1}s_{3k+1} - s_{-2}s_{k-1} + s_{k-3} \\
&\quad - s_{4k+2}s_{2k} - s_{5k+1}s_{k+1} + s_{3k+1}^2)/3 \\
t_{3k} &= t_k^3 - 3s_{-k}t_k - 3s_k t_k t_{-k} + 3s_k^2 s_{-k} + 3t_{-k}^2 - 3s_k \\
t_{3k-1} &= (s_{-k+1}t_{k+1} - t_{2k}t_{k-1} + s_{2k}s_{k-1}s_{3k-1} - s_2 s_{k+1} + s_{k+3} \\
&\quad - s_{4k-2}s_{2k} - s_{5k-1}s_{k-1} + s_{3k-1}^2)/3 \\
t_{3k-2} &= (s_{-2k}t_{-k-2} - t_{k-2}t_{2k} + s_{k-2}s_{2k}s_{3k-2} - s_{-3k-2}s_{-k-2} \\
&\quad + s_{4(-k-1)} - s_{5k-2}s_{k-2} - s_{4(k-1)}s_{2k} + s_{3k+2}^2)/3 \\
t_{3k-3} &= t_{k-1}^3 - 3s_{-k+1}t_{k-1} - 3s_{k-1}t_{k-1}t_{-k+1} + 3s_{k-1}^2 s_{-k+1} + 3t_{-k+1}^2 - 3s_{k-1}
\end{aligned}
$$

Table 5. Relations for the sequence $\{t_k\}$

This enables us to calculate t terms in the range $3k-3$ to $3k+3$ and $-3k+3$ to $-3k-3$ from the s and t terms in the range $k-2$ to $k+2$ and $-k+2$ to $-k-2$ and the s terms in the range $3k-2$ to $3k+2$ and $-3k+2$ to $-3k-2$.

5 Ternary Expansions and Calculating s_m and t_m

Let us now examine the ternary expansion of a positive integer m. In particular, we are looking for a string of coefficients (which we shall call "trits") $a_n a_{n-1} \cdots a_1 a_0$ with $a_i \in \{-1, 0, 1\}$ such that

$$m = \sum_{i=0}^{n} a_i 3^i$$

For positive m, we shall require $a_n = 1$. Let us first determine the length n of our string. If $m = 3^l$, then m has ternary representation $\overbrace{10\cdots 00}^{l+1}$. Similarly, all numbers from $m = 3^l + 1$ to $m = 3^l + 3^{l-1} + \cdots + 1 = (3^{l+1} - 1)/2$ will have trit-length $n = l + 1$. Note that $l = \lfloor \log_3 m \rfloor$. But if $(3^{l+1} - 1)/2 < m < 3^{l+1}$, we have trit-length $n = l + 2$. In terms of a simple formula, we can thus state that m has trit-length n where

$$n = \lfloor \log_3 m \rfloor + \lceil \frac{2m}{3^{\lfloor \log_3 m \rfloor + 1} - 1} \rceil$$

We can determine the ternary expansion of m by using the following algorithm.

Let $n = \lfloor \log_3 m \rfloor + \lceil \frac{2m}{3^{\lfloor \log_3 m \rfloor + 1} - 1} \rceil$.
for $i = 0, \ldots, n$ do
  $a_i = m \pmod 3$
  if $a_i = 2$ **then** $a_i = -1$
  $m = m - a_i$
  $m = m/3$
end for Output $a_n a_{n-1} \cdots a_1 a_0$.

The ternary expansion of m with $0 < m < Q$ gives us the following algorithm to compute the terms s_m, t_m, t_{-m}, and s_{-m}.

Get the ternary representation $a_n a_{n-1} \cdots a_1 a_0$ of m.
$k = 1$.
Let $s_+ = (s_{-1}, s_0, s_1, s_2, s_3)$.
Let $t_+ = (t_{-1}, t_0, t_1, t_2, t_3)$.
Let $t_+ = (t_1, t_0, t_{-1}, t_{-2}, t_{-3})$.
Let $s_- = (s_1, s_0, s_{-1}, s_{-2}, s_{-3})$.
for $i = n - 1, \ldots, 0$
  $l = 3k + a_i$
  Calculate $s_+ = (s_{l-2}, s_{l-1}, s_l, s_{l+1}, s_{l+2})$
  Calculate $t_+ = (t_{l-2}, t_{-l+1}, t_l, t_{l+1}, t_{l+2})$
  Calculate $t_- = (t_{-l+2}, t_{-l+1}, t_{-l}, t_{-l-1}, t_{-l-2})$
  Calculate $s_- = (s_{-l+2}, s_{-l+1}, s_{-l}, s_{-l-1}, s_{-l-2})$
  $k = l$
end for
Output s_+, t_+, t_-, s_-.

Note that the middle term in each outputted list gives s_m, t_m, t_{-m}, s_{-m} respectively.

6 The Analogue Cryptosystems

We are now ready to state analogues to the Gong-Harn and XTR cryptosystems.

6.1 The Gong-Harn Analogue

Let $\{s_k\}$ be a fifth-order characteristic sequence over $GF(q)$ generated from the polynomial $f(x) = x^5 - ax^4 + bx^3 - cx^2 + dx - 1$ whose period Q is a divisor of $q^4 + q^3 + q^2 + q + 1$. Let $\{t_k\}$ be the secondary sequence. Alice chooses a secret key m, $0 \le m < Q$, and calculates her public key $(s_m, t_m, t_{-m}, s_{-m})$.

If Alice wishes to do a Diffie-Hellman key exchange with Bob, she sends her public key to Bob who calculates his private key e and public key $(s_e, t_e, t_{-e}, s_{-e})$. Bob then forms the polynomial

$$\hat{f}(x) = x^5 - s_m x^4 + t_m x^3 - t_{-m} x^2 + s_{-m} x - 1$$

and calculates the e-th terms $\hat{s}_e = s_{em}$, $\hat{t}_e = t_{em}$, $\hat{t}_{-e} = t_{-em}$, and $\hat{s}_{-e} = s_{-em}$ to get the shared secret $(s_{em}, t_{em}, t_{-em}, s_{-em})$. Bob then sends his public key to Alice who calculates the same shared secret.

6.2 The XTR Analogue

As in the case with third-order characteristic sequences, the XTR analogue is much the same as the Gong-Harn analogue except with parameters chosen to optimize efficiency and security. Let $q = p^2$. Observe then that

$$q^4+q^3+q^2+q+1 = p^8+p^6+p^4+p^2+1 = (p^4+p^3+p^2+p+1)(p^4-p^3+p^2-p+1)$$

Note that the subgroup of order $p^4 - p^3 + p^2 - p + 1$ within $GF(p^{10})$ is not contained in any proper subfield. To avoid an index calculus attack, we prefer choosing sequences corresponding to this subgroup. Thus, Q should divide $p^4 - p^3+p^2-p+1$. If α is a root of the polynomial $f(x) = x^5-ax^4+bx^3-cx^2+dx-1$, then $a = \alpha + \alpha^{p^2} + \alpha^{p^4} + \alpha^{p^6} + \alpha^{p^8}$ and $d = \alpha^{-1} + \alpha^{-p^2} + \alpha^{-p^4} + \alpha^{p^6} + \alpha^{p^8}$. But, we have that

$$\alpha^{p^4-p^3+p^2-p+1} = 1 \Rightarrow \alpha^{p^5+1} = 1$$

so $\alpha^{-1} = \alpha^{p^5}$. Thus, we get

$$d = \alpha^{-1} + \alpha^{-p^2} + \alpha^{-p^4} + \alpha^{p^6} + \alpha^{p^8} = \alpha^p + \alpha^{p^3} + \alpha^{p^5} + \alpha^{p^7} + \alpha^{p^9} = a^p$$

We can similarly show that $c = b^p$. Hence the generating polynomial is

$$f(x) = x^5 - ax^4 + bx^3 - b^p x^2 + a^p x - 1$$

Moreover, using this same formula, we see that $s_{-k} = s_k^p$ and $t_{-k} = t_k^p$ for all k.

Hence, we can restrict the public key to (s_m, t_m) and need not calculate the negative terms of the sequences when performing calculations.

Finally, we note that much of the arithmetic speedups presented by Lenstra and Verheul [3] apply to this analogue so that it is possible to reduce computation to $GF(p)$.

7 Security

7.1 Computational Complexity

Let us examine the computational complexity of fifth-order characteristic sequences. As before, let $\{s_k\}$ be the fifth-order characteristic sequence over $GF(q)$ generated from

$$f(x) = x^5 - ax^4 + bx^3 - cx^2 + dx - 1$$

whose period Q divides $q^4 + q^3 + q^2 + q + 1$. Let α be a root of f in $GF(q^5)$ and $\{t_k\}$ be the secondary sequence.

We define the Fifth-Order Characteristic Sequence Discrete Logarithm Problem ($5OCS - DLP$) as the problem of determining a discrete logarithm k given $(s_k, t_k, t_{-k}, s_{-k})$. Note that there are at most 5 such logarithms, namely $\{kq^i \mid i = 0, \ldots, 4\}$. We define the Fifth-Order Characteristic Sequence Diffie-Hellman Problem ($5OCS - DHP$) as the problem of computing $(s_{kl}, t_{kl}, t_{-kl}, s_{-kl})$ from $(s_k, t_k, t_{-k}, s_{-k})$ and $(s_l, t_l, t_{-l}, s_{-l})$ and we write

$$5OCS - DH((s_k, t_k, t_{-k}, s_{-k}), (s_l, t_l, t_{-l}, s_{-l})) = (s_{kl}, t_{kl}, t_{-kl}, s_{-kl})$$

We define the Fifth-Order Characteristic Sequence Diffie-Hellman Decision Problem ($5OCS - DHDP$) as the problem of determining whether

$$5OCS - DH((s_k, t_k, t_{-k}, s_{-k}), (s_l, t_l, t_{-l}, s_{-l})) = (w, x, y, z)$$

for $w, x, y, z \in GF(q)$.

We say that a problem $\mathcal{A}$ is (u, v)-equivalent to a problem $\mathcal{B}$ if any instance of problem $\mathcal{A}$ (resp. $\mathcal{B}$) can be solved in at most u (resp. v) calls to an algorithm solving $\mathcal{B}$ (resp. $\mathcal{A}$).

Theorem 1. *1. The $5OCS - DLP$ is $(1,1)$-equivalent to the DL problem in $\langle\alpha\rangle$.*
2. The $5OCS - DHP$ is $(1,2)$-equivalent to the DH problem in $\langle\alpha\rangle$.
3. The $5OCS - DHDP$ is $(5,2)$-equivalent to the DHD problem in $\langle\alpha\rangle$.

Proof. Note that for any $\beta = \alpha^k$, we can easily compute (s_k, t_k, t_{-k}, s_k) which we shall denote as s_β.

To compute $DL(\beta)$, let $k = 5OCS - DL(s_\beta)$, then $DL(\beta)$ is one of kq^i where $1 = 0, \ldots, 4$ which can be easily checked. Conversely, if β is a root of the polynomial formed from s_β, then $DL(\beta)$ is a solution to $5OCS - DL(s_\beta)$. This proves the first assertion.

To compute $DH(\beta, \gamma)$, compute $5OCS{-}DH(s_\beta, s_\gamma)$ and $5OCS{-}DH(s_{\beta\alpha}, s_\gamma)$. Taking roots of these polynomials gives the values $\{DH(\beta,\gamma)^{q^i} \mid i = 0, \ldots, 4\}$ and $\{(DH(\beta,\gamma)\gamma)^{q^i} \mid i = 0, \ldots, 4\}$ from which the unique value $DH(\beta, \gamma)$ can be easily determined. Conversely, if β and γ are roots of the polynomials formed from s_β and s_γ respectively, then $s_{DH(\beta,\gamma)}$ is a solution to $5OCS - DH(s_\beta, s_\gamma)$. This proves the second assertion.

To prove the third assertion, we note that $DH(\beta, \gamma) = \delta$ if and only if $5OCS - DH(s_\beta, s_\gamma) = s_\delta$ and $5OCS - DH(s_{\beta\alpha}, s_\gamma) = s_{\delta\gamma}$. Conversely, $5OCS - DH(s_\beta, s_\gamma) = s_\delta$ if and only if $DH(\beta, \gamma) = \delta^{q^i}$ for some i in the range $0, \ldots, 4$. $\square$

7.2 Supersingular Elliptic Curves

In [5], it was shown that the Weil pairing could be used to embed a special type of supersingular elliptic curve over $GF(p^2)$ into the XTR group. This has implications relative to the difficulty of the Diffie-Hellman problem of both the XTR group and of supersingular elliptic curves.

We run into a few problems when we try to do the analogous for our case. Any elliptic curve over a field $GF(q)$ must have cardinality within the Hasse interval $[q-2\sqrt{q}+1, q+2\sqrt{q}+1]$. A fifth-order characteristic sequence has cardinality dividing $q^4+q^3+q^2+q+1$ which is strictly greater than the upper limit q^4+2q^2+1 of the Hasse interval for an elliptic curve over $GF(q^4)$. Multiplying this by $q-1$, however, gives q^5-1, and there do exist curves over $GF(q^5)$ with this cardinality. However, it is, in general not easy to find such curves.

On the other hand, let us consider the XTR analogue. The period of the sequence is $p^4-p^3+p^2-p+1$ which is strictly less than the lower limit p^4-2p^2+1 of the Hasse interval for an elliptic curve over $GF(p^4)$. However, multiplying by $p+1$ gives p^5+1 and there does exist a curve over $GF(p^5)$ with this order. In fact, when considering the the case where $p \equiv 2 \pmod 3$ as suggested by [3],

$$Y^2 = X^3 + 1$$

has order p^5+1 over $GF(p^5)$, and a point P of order Q dividing $p^4-p^3+p^2-p+1$ can be embedded into the analogue-XTR group using the Weil pairing on this curve.

8 Conclusions and Discussion

We have presented new cryptosystems based on fifth-order characteristic sequences which are analogues of the Gong-Harn and XTR cryptosystems. We have proven that this cryptosystem is equivalent to the multiplicative group of the finite field $GF(q^5)$.

Logically, the next set of sequences to study would be seventh-order characteristic sequences. This would be equivalent to working in $GF(q^7)^*$ and would require use of a tertiary sequence. Another possible area to explore would be to see if we can elegantly generalize these results for any odd-order characteristic sequence.

References

1. G. Gong and L. Harn, "A new approach on public-key distribution", *ChinaCRYPT '98*, pp 50-55, May, 1998, China.
2. G. Gong and L. Harn, "Public-key cryptosystems based on cubic finite field extensions", *IEEE IT* vol 45, no 7, pp 2601-2605, Nov. 1999.
3. A. K. Lenstra and E. R. Verheul, "The XTR public key system", *Advances in Cryptology, Proceedings of Crypto'2000*, pp. 1-19, Lecture Notes in Computer Science, Vol. 1880, Springer-Verlag, 2000.

4. R. Lidl and H. Niederreiter, *Finite Fields*, Addison-Wesley Publishing Company, Reading, MA, 1983.
5. E. R. Verheul, "Evidence that XTR is More Secure than Supersingular Elliptic Curve Cryptosystems", *Advances in Cryptology, Proceedings of EuroCrypt'2001*, pp. 195-210, Lecture Notes in Computer Science, Vol. 2045, Springer-Verlag, 2001.

A Generating Instances

This appendix shall decribe an efficient method to generate large instances of these cryptosystem presented in this paper, in the event one wishes to implement them. In particular, we shall focus on generating the Gong-Harn analogue over the field $GF(p)$ and the XTR analogue over the field $GF(p^2)$.

First of all, we require a prime $Q > 2^{160}$ which is large enough to prevent general group attacks. We also require a prime p such that Q divides $p^4 + p^3 + p^2 + p + 1$ in the Gong-Harn case and $p^4 - p^3 + p^2 + p + 1$ in the XTR case.

The parameters p and Q can be chosen using an algorithm similar to the one given in [3]. We simply find an $r \in \mathbb{Z}$ such that $Q = r^4 \pm r^3 + r^2 \pm r + 1$ is a prime of at least 160 bits. Next we find $k \in \mathbb{Z}$ such that $p = r + kQ$ is a prime of the correct bit size.

Also in [3], it was remarked that such "nice" p may be undesirable from a security point of view, thus another algorithm was given. The analogue to this algorithm goes as follows. Select a Q-bit prime $Q \equiv 1 \pmod 5$, and find the roots r_1, r_2, r_3, r_4 of the polynomial $X^4 \pm X^3 + X^2 \pm X + 1 \pmod Q$. Finally, find a $k \in \mathbb{Z}$ such that $p = r_i + kQ$ is a prime of the correct bit size for $i = 1, 2, 3,$ or 4.

In the Gong-Harn analogue, security depends on solving logarithms in $GF(p^5)$, so we would like $5 \log p \simeq 1024$ which means that p should be on the order of 205 bits. In the XTR analogue, the equivalence is to logarithms in $GF(p^{10})$, so p should be approximately 103 bits. Note also that in the XTR analogue, we have the additional requirement $p \equiv 2 \pmod 3$ in order to take advantage of speedups.

Once we have our primes p and Q, we need only find an element α of order Q in the field $GF(p^5)^*$ or $GF(p^{10})^*$ as appropriate. We can do this by taking a random element in the field and raising to the power $(p^5 - 1)/Q$ or respectively $(p^{10} - 1)/Q$. If we do not get 1, which occurs with probability $1/Q$, we have our element α. We then calculate the values a, b, c, d and we have our sequence.