Gong-Harn Digital Signature Algorithm

Susana Sin

October 7, 2003

Outline

- Gong-Harn Public-key Cryptosystem Overview
- Third-order characteristic sequence
- Reciprocal sequence
- Algorithms for computing sequence terms
- GH-DSA
- Implementation
- Remarks

GH Public-key Cryptosystem (GH-PKC) Overview

- GH-PKC is based on the 3-rd order LFSR sequence with particular phase.
- GH Diffie-Hellman (GH-DH) key agreement protocol published in proceeding of ChinaCrypto'1998.
- GH-DH / GH-DSA was published in proceeding of the 8th Annual Workshop on Selected Areas in Cryptography, Aug 2001
- Security is based on the difficulty in solving DL problem in GF(q^3) for q = p or $q = p^2$

Third-Order Characteristic Sequence

- An irreducible polynomial f(x) of degree 3 over GF(p), where p is a prime number:

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

- Initial State:

$$s_0 = 3, s_1 = a, s_2 = a^2 - 2b$$

- The sequence $\{s_k\}$ is called a 3rd-order characteristic sequence.
- Period of this sequence is:

$$Period, Q \mid p^2 + p + 1$$

- kth-state of the LFSR is $\underline{s}_k = (s_k, s_{k+1}, s_{k+2})$

Reciprocal Sequence

- Given

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

- The reciprocal polynomial is:

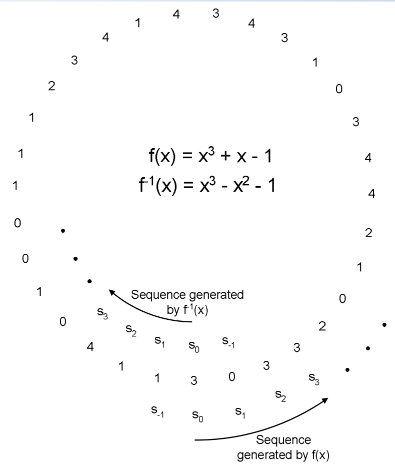
$$f^{-1}(x) = x^3 - bx^2 + ax - 1$$

- Initial State:

$$s_0 = 3, s_1 = b, s_2 = b^2 - 2a$$

- Sequence generated by $f^{-1}(x)$ is also a 3rd-order characteristic sequence

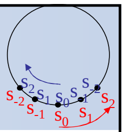
Example, p = 5



Algorithms for Computing Sequence Terms

- Three algorithms for computing sequence term in a 3rd order characteristic sequence:
  - Dual-State Fast Evaluation Algorithm (DSEA)
    - To compute the $\underline{s}_{k-1}$ state and its dual
  - Computation of a Previous Sequence Term
    - To compute $s_{\pm(k-1)}$ using (s_k, s_{k+1}) and it dual
  - Computation of Mixed term
    - To compute $s_{\pm u(k+v)}$

The DSEA Algorithm



- Binary rep of k: $k = \sum_{i=0}^{n} k_i 2^{n-i} = k_0 2^n + k_1 2^{n-1} + \cdots + k_n$
- Let $T_0 = k_0 = 1, T_j = k_j + 2T_{j-1}, T_n = k$ and $t = T_{j-1}, t' = T_j$
- For $k_j = 0$

$$s_{t'+1} = s_t s_{t+1} - a s_{-t} + s_{-(t-1)}$$
$$s_{t'} = s_t^2 - 2 s_{-t}$$
$$s_{t'-1} = s_t s_{t-1} - b s_{-t} + s_{-(t+1)}$$

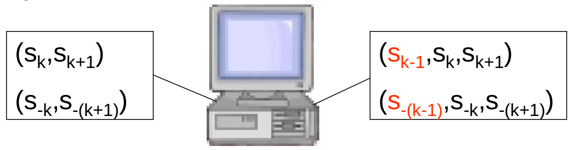
  For $k_j = 1$

$$s_{t'+1} = s_{t+1}^2 - 2 s_{-(t+1)}$$
$$s_{t'} = s_t s_{t+1} - a s_{-t} + s_{-(t-1)}$$
$$s_{t'-1} = s_t^2 - 2 s_{-t}$$

- Interchange a and b and find the ($s_{t'+1}$, $s_{t'}$, $s_{t'-1}$) terms in sequence generated by the reciprocal polynomial which are the ($s_{-(t'+1)}$, $s_{-t'}$, $s_{-(t'-1)}$) terms in the original sequence

Computation of Previous Sequence Term

- The 3rd order characteristic sequence has properties of duality and redundancy.
- Three elements in any state of the sequence are *not* independent.
- If any two consecutive elements are known in the sequence, the third remaining one can be uniquely determined.



Computation of Previous Sequence Term

- Given (s_k, s_{k+1}) and its dual
- Let

$$delta = s_{k+1}s_{-(k+1)} - s_1 s_{-1}$$

delta cannot be zero!

- Then $s_{\pm(k-1)}$ can be computed by:

$$\boxed{\begin{aligned} s_{k-1} &= \frac{es_{-(k+1)} - s_{-1}D(e)}{delta} \\ s_{-(k-1)} &= \frac{D(e)s_{k+1} - s_1 e}{delta} \end{aligned}} \quad \boxed{\begin{aligned} &D(s_k) = s_{-k} \\ &e = -s_{-1}D(c_1) + c_2 \\ &c_1 = s_1 s_{k+1} - s_{-1}s_k \\ &c_2 = s_k^{\ 2} - 3s_{-k} + (b^2 - a)s_{-(k+1)} \end{aligned}}$$

Computation of Previous Sequence Term

- Consider delta:

$$delta = s_{k+1}s_{-(k+1)} - s_1 s_{-1}$$

- When delta equals zero,

$$s_{k+1}s_{-(k+1)} = ab$$

- Programs written in c++ and Maple are used to compute the value of delta for all sequence terms generated by all irreducible polynomial over GF(p) for prime p between 5 and 127.

Computation of Previous Sequence Term

- It is found that for any prime p, $\begin{cases} s_p = a = s_{-(p+1)} \\ s_{p+1} = b = s_{-p} \end{cases}$
- ie, if k equals to either p-1 or p, the product $s_{k+1}s_{-(k+1)}$ equals to a*b for any prime p. This restriction is added to the signing key selection.
- If either a or b is zero, k will be more likely yield a zero delta. Both a and b should be chosen as non-zero integers.
- As p increases, the percentage of delta that equals zero decreases.
- With the additional restrictions on a and b and the additional rules on k, it is reasonable to reduce bandwidth usage as a way of minimizing cost.

Computation of Mixed Term $s_{\pm u(k+v)}$ with Known $\underline{s}_{k-1}$ State

- This algorithm is used to compute mixed terms $s_{\pm u(k+v)}$ with known (s_{k-1}, s_k, s_{k+1}) and its dual.
- The procedure is as follows:
  1. Compute the sequence terms $s_{\pm(k+v)}$
  2. Construct another irreducible polynomial as:
     $$g(x) = x^3 - s_{(k+v)}x^2 + s_{-(k+v)}x - 1$$
  3. Compute the $\pm u$th term of sequence generate by g(x)

Computation of $s_{\pm(k+v)}$

- $s_{\pm(k+v)}$ terms cannot be computed using the DSEA algorithm
- Instead, use a general result for LFSR sequence:

Transitional Matrix = $A = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & -b \\ 0 & 1 & a \end{bmatrix}$, State Matrix = $M_n = \begin{bmatrix} s_{n-2} & s_{n-1} & s_n \\ s_{n-1} & s_n & s_{n+1} \\ s_n & s_{n+1} & s_{n+2} \end{bmatrix}$

$$\underline{s}_k \cdot A = (s_k, s_{k+2}, s_{k+2}) \cdot \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & -b \\ 0 & 1 & a \end{bmatrix} = (s_{k+1}, s_{k+3}, s_k - bs_{k+1} + as_{k+2}) = \underline{s}_{k+1}$$

Computation of $s_{\pm(k+v)}$

- Two properties:
  1. $\underline{s}_v = \underline{s}_0 \cdot A^v = \underline{s}_1 \cdot A^{v-1} = \underline{s}_2 \cdot A^{v-2} = \cdots = \underline{s}_{v-1} \cdot A^1$
  2. $M_v = M_0 \cdot A^v \Rightarrow A^v = M_0^{-1} \cdot M_v$ if $\det(M_0) \neq 0$
- If $\det(M_0) \neq 0$, then

$$\underline{s}_{k+v} = \underline{s}_k \cdot A^v = \underline{s}_k \cdot \left(M_0^{-1} \cdot M_v\right) = \underline{s}_k \cdot \left(\begin{bmatrix} s_{-2} & s_{-1} & s_0 \\ s_{-1} & s_0 & s_1 \\ s_0 & s_1 & s_2 \end{bmatrix}^{-1} \begin{bmatrix} s_{v-2} & s_{v-1} & s_v \\ s_{v-1} & s_v & s_{v+1} \\ s_v & s_{v+1} & s_{v+2} \end{bmatrix} \right)$$

where

$$s_{v-2} = s_{v+1} - as_v + bs_{v-1}$$
$$s_{v+2} = as_{v+1} - bs_v + s_{v-1}$$

Computation of $s_{\pm(k+v)}$

- Procedure for finding $s_{(k+v)}$ term using $\underline{s}_{k-1}$ state
  1. Construct Matrix M_0 and compute M_0^{-1} $\longleftarrow$ Can be done offline
  2. Compute (s_{v-1}, s_v, s_{v+1}) terms generated by f(x) using DSEA algorithm
  3. Compute s_{v-2} and s_{v+2} terms and construct Matrix M_v
  4. Compute $s_{(k+v)}$ term by:
     $$\underline{s}_{k-1+v} = \underline{s}_{k-1} \cdot A^v = \underline{s}_{k-1} \cdot \left(M_0^{-1} \cdot M_v\right)$$

     In particular, the s_{k+v} term is equal to $\underline{s}_{k-1}$ multiplied by the middle column of $\left(M_0^{-1} \cdot M_v\right)$

Computation of $s_{\pm(k+v)}$

- Consider M_0:

$$M_0 = \begin{bmatrix} s_{-2} & s_{-1} & s_0 \\ s_{-1} & s_0 & s_1 \\ s_0 & s_1 & s_2 \end{bmatrix} = \begin{bmatrix} b^2-2a & b & 3 \\ b & 3 & a \\ 3 & a & a^2-2b \end{bmatrix}$$

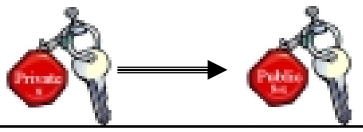
- Determinant of M_0 is:

$$\det(M_0) = (b^2-2a)[3(a^2-2b)-a^2] - b[b(a^2-2b)-3a] + 3[ab-3\cdot 3] \neq 0$$

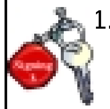
- It depends only on a and b. By choosing a and b correspondingly, $\det(M_0)$ can be guaranteed to be non-zero.

GH-DSA

- ElGamal-like signature algorithm
- System Public Parameters:
  Irreducible polynomial $f(x) = x^3 - ax^2 + bx - 1$ over GF(p), where p is a prime number, with period Q
- Alice:
  Private Key: Choose x, with $0 < x < Q$ and $\gcd(x, Q) = 1$
  Public Key: The $s_{\pm x}$ terms generate by f(x)



GH-DSA: Signing Process



- Signing Process:
  1. Randomly choose k, with $0 < k < Q$ and $\gcd(k,Q) = 1$. Use DSEA algorithm to compute (s_{k-1}, s_k, s_{k+1}) and its dual such that $\gcd(s_k, Q) = 1$. $r = s_k$
  2. Compute h=h(m), where h() is a hash function
  3. Solve for t in the signing equation: $h \equiv xr + kt \bmod Q$ $\Rightarrow t \equiv k^{-1}(h - rx) \bmod Q$
- (r, t) is a digital signature of the message m. Alice sends Bob (m, r, t) together with (s_k, s_{k+1}) and its dual.

GH-DSA: Verification Process

- Signing Equation $h \equiv xr + kt \bmod Q$ ← x and k are unknown to Bob
- But, Bob can verify Sequence Terms
- Verification Process:
  Compute $s_{\pm(k-1)}$ using (s_k, s_{k+1}) and its dual
  If $\gcd(t, Q) = 1 \Rightarrow$ Case 1, else $\Rightarrow$ Case 2
- Case 1: $s_{\pm x} = s_{\pm r^{-1}(h-kt)} = s_{\mp r^{-1}t(-ht^{-1}+k)} = s_{\pm u(k+v)}$
  1. Compute $u = -r^{-1}t \bmod Q$, $v = -ht^{-1} \bmod Q$
  2. Compute mixed terms $s_{\pm u(k+v)}$
  3. Verify sequence terms $s_{\pm x} = s_{\pm u(k+v)}$

GH-DSA: Verification Process

- Case 2: $s_{\pm kt} = s_{\pm(h-rx)} = s_{\mp r(-hr^{-1}+x)} = s_{\pm u(x+v)}$
  1. Compute u = -r mod Q, v = -hr^{-1} mod Q
  2. Compute mixed terms $s_{\pm u(x+v)}$
  3. Compute $s_{\pm kt}$
  4. Verify sequence terms $s_{\pm kt} = s_{\pm u(k+v)}$

Software Design

- Signing Process:
  1. Input private key x. Compute (s_x, s_{-x}).
  2. Input signing key k. Compute (s_{k-1}, s_k, s_{k+1}) and it dual. If gcd(s_k, Q) ≠ 1, prompt user to choose another k.
  3. If delta = 0, prompt user to pick another k.
  4. Input message m to be signed. Assume h = m.
  5. r = s_k, Compute t.
  6. If t = 0, prompt user to choose another k.
  7. Output (s_x, s_{-x}), (m, r, t), (s_k, s_{k+1}) and its dual

Software Design

- Verification Process:
  1. Input (m, r, t), (s_x, s_{-x}), (s_k, s_{k+1}) and its dual.
  2. Compute $s_{\pm(k-1)}$.
  3. If gcd(t, Q) = 1, then
     i. u = -r^{-1}t mod Q, v = -ht^{-1} mod Q
     ii. Compute $s_{\pm u(k+v)}$
     iii. Verify if $s_{\pm u(k+v)} = s_{\pm x}$
  4. If gcd(t, Q) ≠ 1, then
     i. u = -r mod Q, v = -hr^{-1} mod Q
     ii. Compute $s_{\pm u(x+v)}$
     iii. Compute $s_{\pm kt}$
     iv. Verify if $s_{\pm u(x+v)} = s_{\pm kt}$
  5. Accept signature if sequence terms match

Design Issues

- Storage requirement for recursive iteration in DSEA algorithm
  – Only six sequence terms need to be kept after each iteration.
- Delta cannot be zero
  – k cannot be either p - 1 or p
  – Both a and b should be non-zero integers
- Invertible Matrix M_0
  – This can be guaranteed by choosing a and b correspondingly such that det(M_0) ≠ 0.

Testing

- Toy Case
- Real System

Testing – Toy Case

- An irreducible polynomial f(x) over a small field is used to make it possible to verify all the sequence term computations including the ones in the intermediate steps.
- Irreducible polynomial f(x) over GF(5):

$$f(x) = x^3 + x - 1$$

- Period of this sequence, Q:

$$Q = 5^2 + 5 + 1 = 31$$

Testing – Toy Case

- All sequence terms are compared to the sequence terms generated by the recursive formula:

$$s_{j+3} = as_{j+2} - bs_{j+1} + s_j,\ for\ i = 0,1,2,\ldots$$

Testing – Real System

- Implement GH-PKC over GF(p)
- For 1024-bit security level, a 341-bit p is used.
- Parameters are chosen as:

$$\begin{aligned}
\mathrm{p} &= 25241001428020650913199864753466204394427825281223 81\\
&\quad 640812816384384364195892628818440024729407595209291\\
\mathrm{a} &= 100967846246663453437323616599547897779132286415320 7\\
&\quad 149330490776209148279733077179938397109115148708951\\
\mathrm{b} &= 206216022644184759815024549954227848108708723659854 5\\
&\quad 481740882935002939062370689540637392192938836162683\\
\mathrm{Q} &= 164705219395020291376758884936962412458513495611 1
\end{aligned}$$

Testing – Real System

- NTL library is used in the implementation to handle the large numbers in c++.
- It is not practical to verify all the generated sequence terms as they are too big!
- Instead, two copies of the program are executed at the same time to simulate two users, Alice and Bob, to make sure the generated shared key pair is the same and the message can be successfully signed and verified.

Remarks

- The GH-DH key agreement protocol and the signature generation in the GH-DSA scheme are successfully implemented in c++ and are tested thoroughly.
- The implementation of signature verification is still in progress.
- Several restrictions on system parameters and signing key selection are added to the system to increase system efficiency.
- Implement GH-PKC on constrained devices, such as blackberry, to analyze performance.

References

- Papers on GH-PKC:
  - G. Gong and L. Harn, A new approach for public key distribution, *Proceedings of China-Crypto'98*, May 1998, Chengdu, China
  - G. Gong and L. Harn, The GH public-key cryptosystems, the Proceedings of the 8th Annual Workshop on Selected Areas in Cryptography, Toronto, Aug 16-18, 2001
- NTL Library is available for download at:

  www.shoup.net/ntl