**Lecture 9: Protecting Master Key via Threshold Scheme over F**2m

Ultimately, the keys are stored in the system and the entire system may depend on a single master key. This master key may be exposed, lost or destroyed. Having copies of the same key to more people will increase the vulnerability of the system from betrayal. Master key is the most supreme key in a cryptosystem.

Threshold scheme provides a solution by breaking the master key into *n*=8 shadows of keys to *n* highest board of directors or command controls. The master key can be of valid use by having some *tm*=3 keys or more together. Anything less than m shadow keys will not be a valid master key. Key recovery can always be done with *m* shadow keys. Fewer than *n−m* shadow keys lost will not endanger the system.

These are the very important key in a cryptosystem.

1. Shadow Keys of the Threshold Scheme,
2. The Master key of the Cryptosystem,
3. The Public and Private Key of the Public Key Infrastructure and then
4. The Session Key of the Symmetric Encryption mode.

The higher keys shall be capable of overruling or opening the lower layer keys.

In this topic, we propose to present and deliver a master key generation process. There will be the master key generating ceremony and the supreme master key shall be distributed into shadow keys. This supreme master key shall be the highest key to overrule the cryptosystems such the PKI at the application layer and Key Exchange mechanism at network layer of communication.

Let us list down irreducible polynomials of degree 8 as shown in Table 3 below.

Table 0. Irreducible Polynomials of degree 8

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *i* | *Pi*(*t*) | Binary | Hexa | Decimal |
| 0 | *t*8 + *t*4 + *t*3 + *t*2 + 1 | 100011011 | 11B | 283 |
| 1 | *t*8 + *t*5 + *t*3 + *t*+ 1 | 100101011 | 12B | 299 |
| 2 | *t*8 + *t*6 + *t*4 + *t*3 + *t*2 + *t*+ 1 | 101011111 | 15F | 351 |
| 3 | *t*8 + *t*6 + *t*5 + *t*+ 1 | 101100011 | 163 | 355 |
| 4 | *t*8 + *t*6 + *t*5 + *t*2 + 1 | 101100101 | 165 | 357 |
| 5 | *t*8 + *t*6 + *t*5 + *t*3 + 1 | 101101001 | 169 | 361 |
| 6 | *t*8 + *t*7 + *t*6 + *t* + 1 | 111000011 | 1C3 | 451 |
| 7 | *t*8 + *t*7 + *t*6 + *t*5 + *t*2 + *t* + 1 | 111100111 | 1E7 | 487 |

Threshold Scheme using Newton Polynomial mod irreducible Polynomial 283.

Let take an irreducible polynomial 283. The policy is *m* = 5 of *n* = 8 shadow keys.

Given the master key K =200 as *a*0(t) = *t*7 + *t*6 + *t*3.

Then generate the coefficients *a*1, *a*2, ..., *am*−1. Take*ai* (*t*) from an irreducible polynomial of degree 6 for *i* = 1, ..., *m*−1 as listed in Table 1.

Table 1. Irreducible Polynomials of degree 6 and lower.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *i* | *ai*(*t*) | Binary | Hexa | Decimal |
| 1 | *t*2 + *t* + 1 | 111 | 7 | 7 |
| 2 | *t*3 + *t* + 1 | 1011 | B | 11 |
| 3 | *t*4 + *t* + 1 | 10011 | 13 | 19 |
| 4 | *t*5 + *t*2 + 1 | 100101 | 25 | 37 |
| 5 | *t*5 + *t*4 + *t*2 + *t* + 1 | 110101 | 35 | 53 |
| 6 | *t*5 + *t*4 + *t*3 + *t*2 + 1 | 111101 | 3D | 61 |
| 7 | *t*6 + *t* + 1 | 1000011 | 43 | 67 |
| 8 | *t*6 + *t*5 + *t*2 + *t* + 1 | 1100111 | 67 | 103 |
| 9 | *t*6 + *t*5 + *t*3 + *t*2 + 1 | 1101101 | 6D | 109 |

Table 2. Irreducible Polynomials of degree 7

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *i* | *xi*(*t*) | Binary | Hexa | Decimal |
| 0 | *t*7 + *t* + 1 | 10000011 | 83 | 131 |
| 1 | *t*7 + *t*3 + 1 | 10001001 | 89 | 137 |
| 2 | *t*7 + *t*3 + *t*2 + *t*+ 1 | 10001111 | 8F | 143 |
| 3 | *t*7 + *t*4 + *t*3 + *t*2 + 1 | 10011111 | 9F | 159 |
| 4 | *t*7 + *t*5 + *t*4 + *t*3 + *t*2 + *t*+ 1 | 10111111 | BF | 191 |
| 5 | *t*7 + *t*6 + *t*3 + 1 | 11001001 | C9 | 201 |
| 6 | *t*7 + *t*6 + *t*4 + *t*2 + 1 | 11010101 | D5 | 213 |
| 7 | *t*7 + *t*6 + *t*5 + *t*2 + 1 | 11100001 | E1 | 225 |
| 8 | *t*7 + *t*6 + *t*5 + *t*4 + *t*2 + *t* + 1 | 11110111 | F7 | 247 |

Compute the polynomial A(*x*) = *a*0 + *a*1⋅*x* + *a*2⋅*x*2 +...+ *am*−1 ⋅ *xm*−1

Generate *x*0, *x*1, *x*2, ..., *xn*−1.

Take *xi*from a small irreducible polynomial as listed in Table 2.

Algorithm 1: An efficient mode to evaluate a polynomial

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

An efficient mode to evaluate a polynomial

A(*x*) = *a*0 + *a*1⋅*x* + *a*2⋅*x*2 +...+ *am*−1 ⋅ *xm*−1

given by coefficient *a*0, *a*1, *a*2, …., *am*−1,

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

function *y* = Evaluate(*x*; *a*0, *a*1, *a*2, …., *am*−1)

*y* = 0

for *i* from *m*−1 down to 0,

*y* = *ai* + *x*⋅y (mod P)

end

return *y.*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Compute *y*0, *y*1, *y*2, ..., *yn*−1 by *yi* =*A*(*xi*) via an Algorithm 1.

Shadow keys are (*x*0, *y*0), (*x*1, *y*1), ..., (*xn*−1, *yn*−1) to interpolate an original polynomial.

Having *m*=5 shadow key points, an original polynomial can be regenerated

and evaluated at *x*=0 so that *A*(0)= *a*0 = K.

*x*0

*x*1

*x*2

*x*3

*x*4

*y*0

*y*1

*y*2

*y*3

*x*4

*y*01= (*y*0−*y*1)(*x*0−*x*1)−1

*y*12= (*y*1−*y*2)(*x*1−*x*2)−1

*y*23= (*y*2−*y*3)(*x*2−*x*3)−1

*y*34= (*y*3−*y*4)(*x*3−*x*4)−1

*y*02= (*y*01−*y*12)(*x*0−*x*2)−1

*y*13= (*y*12−*y*23)(*x*1−*x*3)−1

*y*34= (*y*23−*y*34)(*x*3−*x*4)−1

*y*03= (*y*02−*y*13)(*x*0−*x*3)−1

*y*14= (*y*13−*y*34)(*x*1−*x*4)−1

*y*04= (*y*03−*y*14)(*x*0−*x*4)−1

P(*x*) = *y*0

+ *y*01⋅(*x−* *x*0)

+ *y*02⋅(*x−* *x*0)⋅(*x−* *x*1)

+ *y*03⋅(*x−* *x*0)⋅(*x−* *x*1)⋅(*x−* *x*2)

+ *y*04⋅(*x−* *x*0)⋅(*x−* *x*1)⋅(*x−* *x*2)⋅(*x−* *x*3)

If we have 2 points, we can generate a line, a linear polynomial P1(*x*) = b0 + b1 ⋅ *x*

If we have 3 points, we can generate a polynomial P2(*x*) = b0 + b1 ⋅ *x* + b2 ⋅ *x*2 of degree 2.

If we have 4 points, we can generate a polynomial P3(*x*) = b0 + b1 ⋅ *x* + b2 ⋅ *x*2 + b3 ⋅ *x*3 of degree 3.

Finally, if we have 5 points, we can generate

a polynomial P4(*x*) = b0 + b1 ⋅ *x* + b2 ⋅ *x*2 + b3 ⋅ *x*3 + b4 ⋅ *x*4 of degree 4.

The policy is *m*= 5 out of *n* = 8 shadows.

Step 0: Given the secret master key K = 200.

Step 1: Pick an irreducible polynomial P = 283 slightly longer K.

Step 2: Take *a*0 = K, generate random coefficients *a*1, *a*2, *a*3, …., *at*-1,

{ *a*0, *a*1, *a*2, *a*3,… , *a m−*1}= {200, 7, 11, 19, 37}

Step 3: We generate the shadow key set

{ *x*0, *x*1, *x*2, …., *xn−*1}= {131, 137, 143, 159, 191, 201, 213, 225}

Step 4: Decompose K into the *w* shadows by taking *yi* = P4(*xi*) mod p, for every *i* = 1, …., *w*.

*yi* = P4(*xi*) = *a*0 + *a*1⋅*x* + *a*2⋅*x*2 + *a*3⋅*x*3 + *a*4 ⋅*x*4

= 200 + 7⋅*x* + 11⋅*x*2 + 19⋅*x*3 + 37⋅*x*4

We shall have *n* = 8 shadow keys as listed in Table 1.

Table 1. A set of shadow keys (*xi*, *yi*)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *i* | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| *xi* | 131 | 137 | 143 | 159 | 191 | 201 | 213 | 225 |
| *yi* | 216 | 47 | 224 | 97 | 98 | 165 | 44 | 237 |

Step 5: Pick the first 5 shadow keys

|  |  |
| --- | --- |
| *xi* | *yi* |
| 131 | 216 |
|  |  |
| 137 | 47 |
|  |  |
| 143 | 224 |
|  |  |
| 159 | 97 |
|  |  |
| 191 | 98 |

What is 6−1 ≡ 123 (mod 283)?

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *xi* | *ki* |  |  |  |  | |  |  |
| 101 | 128 | 240 | (75-240)/(105-101) | -165⋅193 | 23 |
| 103 | 94 | 75 | (230-75)/(107-103) | 155⋅193 | 103 |
| 105 | 244 | 230 | (60-230)/(109-105) | -170⋅193 | 86 |
| 107 | 190 | 60 |  |  |  |
| 109 | 53 |  |  |  |  |

What is 4−1 ≡ 2−1 ⋅ 2−1 mod 257?

= 129 ⋅ 129 ≡ 193 (mod 257).

Step 6: Generate the Divided Difference Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *i* | *xi* | *f* [*xi*] | *f* [*xi*, *xi*+1] | *f* [*xi*, *xi*+1, *xi*+2] | *f* [*xi*, *..*, *xi*+3] | *f* [*xi*, *..*, *xi*+4] |
| 0 | 101 | 128 | 240 | 23 | 99 | 89 |
| 1 | 103 | 94 | 75 | 103 | 40 |  |
| 2 | 105 | 244 | 230 | 86 |  |  |
| 3 | 107 | 190 | 60 |  |  |  |
| 4 | 109 | 53 |  |  |  |  |

*f* [*x*0, *..*, *x*3] = (103−23)⋅43 = 80⋅43 = 3440 = 99 (mod 257)

*f* [*x*1, *..*, *x*4] = (86−103)⋅43 = −17⋅43 = −731 = 40 (mod 257)

What is 8−1 ≡ 2−1 ⋅ 4−1 = 129⋅193 = 24897 ≡ 225 mod 257?

*f* [*x*0, *..*, *x*4] = (40−99)⋅225 = −59⋅225 ≡ 89 (mod 257)

Step 7: We will write the newton polynomial as

P4(*x*) = *f* [*xi*] + *f* [*xi*, *xi*+1](*x* − *x*0) + *f* [*xi*, *xi*+1, *xi*+2]

P1(*x*) = *f* [*x*0] + *f* [*x*0, *x*1](*x* − *x*0)

P2(*x*) = *f* [*x*0] + *f* [*x*0, *x*1](*x* − *x*0) + *f* [*x*0, *x*1, *x*2](*x* − *x*0)(*x* − *x*1)

P3(*x*) = *f* [*x*0] + *f* [*x*0, *x*1](*x* − *x*0) + *f* [*x*0, *x*1, *x*2](*x* − *x*0)(*x* − *x*1)

+ *f* [*x*0, *…*, *x*3](*x* − *x*0)(*x* − *x*1)(*x* − *x*2)

P4(*x*) = *f* [*x*0] + *f* [*x*0, *x*1](*x* − *x*0) + *f* [*x*0, *x*1, *x*2](*x* − *x*0)(*x* − *x*1)

+ *f* [*x*0, *…*, *x*3](*x* − *x*0)(*x* − *x*1)(*x* − *x*2)

+ *f* [*x*0, *…*, *x*4](*x* − *x*0)(*x* − *x*1)(*x* − *x*2)(*x* − *x*3)

P4(*x*) = 128 + 240(*x* − *x*0) + 23(*x* − *x*0)(*x* − *x*1)

+ 99(*x* − *x*0)(*x* − *x*1)(*x* − *x*2)

+ 89(*x* − *x*0)(*x* − *x*1)(*x* − *x*2)(*x* − *x*3)

P4(*x*) = 128 + 240(*x* − 101) + 23(*x* − 101)(*x* − 103)

+ 99(*x* − 101)(*x* − 103)(*x* − 105)

+ 89(*x* − 101)(*x* − 103)(*x* − 105)(*x* − 107)

Step 8: The master key is recovered when we evaluate the polynomial P4(*x*) at *x* =0.

P4(0) = 128 + 240(0 − 101) + 23(0 − 101)(0 − 103)

+ 99(0 − 101)(0 − 103)(0 − 105)

+ 89(0 − 101)(0 − 103)(0 − 105)(0 − 107)

P4(0) = 128 − 240⋅101 + 23(101)(103)

− 99(101)(103)(105)

+ 89(101)(103)(105)(107)

P4(0) = 128 − 24240 + 239269 − 108139185+ 10402115745

= 10294191717

≡ 177 (mod 257)

Tutorial 9: Newton Threshold Scheme

Take the master key as 100 + (ID mod 100)

Repeat the process above.