A new public key cryptosystem based on polynomials over finite fields $GF(2)$

G.H. Khachatrian
(American University of Armenia)

Abstract

In this paper a new public key system based on polynomials over fields $GF(2)$ is developed. The security of the system is based on the difficulty of finding discrete logarithms over $GF(2^k)$ with sufficiently large k. The presented system has all features of ordinary public key schemes such as public key encryption and digital signatures. The security and implementation aspects of the presented system are also introduced along with comparison with other well known public key systems.

Keywords: discrete logarithm problem, public-key cryptography, digital signatures, polynomials over finite fields

1 Introduction

Public-key cryptography started in 1976 with publication of pioneering work of Diffie and Hellman [1] called DH key exchange and in 1978 with another fundamental work by Rivest, Shamir and Adleman [2], called RSA cryptosystem. DH key exchange is based on the discrete logarithm problem (DLP) and RSA is based on integer factorization problem. RSA provides all features of public key cryptography including public-key encryption and digital signatures. There are public-key encryption and digital signature systems based on the DLP problem such as Digital Signature Standart (DSS) [3] and EIGamal public-key encryption [4]. In this paper we will represent a public key system the security of which is based on DLP but is implemented differently by using relationship between roots of polynomials. We will represent public key encryption as well as digital signature generation and verification operations. The main feature of the new cryptosystem, whose security also is based on DLP, is that its public key encryption is computationally equivalent to EIGamal public-key encryption, public-key decryption and signature generation is computationally equivalent to analogous operations for RSA and EIGamal encryption or DSS, but signature verification is significantly faster than other analogous systems, including even RSA which is considered fastest among existing public key systems.

2 New public key system

In this system binary messages will be regarded as polynomials of degree n over $GF(2)$. A primitive polynomial $g(x)$ of degree n will be considered as the base polynomial of the system and we will denote by α a root of $g(x)$. Let i be any random number less then $2^n - 1$ where $(i, 2^n - 1) = 1$ and let $g_i(x)$ be a primitive polynomial with the root α^i. Let i be the secret parameter of the system and polynomials $g(x)$ and $g_i(x)$ be public polynomials of the system.

a) Public key encryption:

For a given polynomial $g_i(x)$ its root as a polynomial $f(\alpha)$ can be found using an algorithm presented in [5, chapter 4]. The complexity of that algorithm is not more than $O(k^3)$. However for a given $f(\alpha)$ to find i where $\alpha^i = f(\alpha)$ is a DLP. Note that everybody who knows public polynomials $g(x)$ and $g_i(x)$ will be able to calculate for any given polynomial $f(x)$ the values $f(x^i)$ modulo $g(x)$ or modulo $g_i(x)$ and also $f(x^{i^{-1}})$ modulo $g_i(x)$ without the knowledge of i or i^{-1}. The latter value can be calculated simply by solving a linear system of equations based on the equality $f(x^i)(mod\ g_i(x)) = F(x) = f(x^{i^{-1}})$. We have two options to encrypt a given message M and we will call these two options to be "Off-line" and "On-line".

a1) "Off-line" encryption: This kind of encryption can be made in the style similar to [4]. For a randomly generated N with n bits we have

$$x^N = C(x)\ mod\ g(x)$$

and

$$x^N = C_i(x)\ mod\ g_i(x)$$

.

This operation can be made "Off-line" beforehand as in the case with [4].

It is easy to show that

$$C_i(x) = (C(x^{i^{-1}}))^i\ mod\ g_i(x) \qquad (2.1\ a)$$

or

$$C(x) = (C_i(x^i))^{i^{-1}}\ mod\ g(x) \qquad (2.1\ b)$$

Where $i^{-1} \cdot i = 1\ mod\ (2^n - 1)$. Note that $(C_i(x))^{-1}$ or $(C(x))^{-1}$ can also be pre-computed.

Let the message M that needs to be encrypted be represented as a polynomial $M(x)$ of degree n over $GF(2)$. The encryption operation is the following:

$$M \cdot (C(x))^{-1}\ ,\ C_i(x^i) \qquad (2.2\ a)$$

or

$$M \cdot (C_i(x))^{-1}\ ,\ C(x^{i^{-1}}) \qquad (2.2\ b)$$

and the encrypted message is a pair as represented in (2.2 a) or (2.2 b).

b1) Decryption: Decryption is based on the fact that only the "owner" of the system knows i or i^{-1} and having $C_i(x)$ or $C(x)$ he/she can calculate either $(C_i(x^i))^{i^{-1}}$ or $(C(x^{i^{-1}}))^i$ and can get M by multiplying the respective results with the first part of the encrypted message.

a2) "On-line" encryption:

Let M be the message to be encrypted. Then the values

$$x^M = C(x)\ mod\ g(x) \qquad \text{and} \qquad x^M = C_i(x)\ mod\ g_i(x) \qquad (2.3)$$

are calculated. The encrypted message will then be a pair:

$$(M \oplus C(x)), C_i(x^i) \qquad (2.3\ a)$$

or

$$(M \oplus C_i(x)), C(x^{i^{-1}}) \qquad (2.3\ b)$$

b2) Decryption: Decryption is based on the fact that only the "owner" of the system knows i or i^{-1} and having $C_i(x)$ or $C(x)$ he/she can calculate either $(C_i(x^i))^{i^{-1}}$ or $(C(x^{i^{-1}}))^i$ and can get M by XOR-ing the respective results with the first part of the encrypted message.

c) Digital signature:

Signature generation:

1) the signer computes the hash of the message M which is denoted by polynomial $H(M) = h(x)$.

2) the signer computes $h_i(x) = (h(x^{i^{-1}}))^i \; mod \; g_i(x) \qquad (2.4)$
and $h_i(x)$ is the signature of the message.

d) Signature verification:

The signature verification is to show that

$$X^S = h(x) \; mod \; g(x) \qquad (2.5)$$

$$X^S = h_i(x) \; mod \; g_i(x) \qquad (2.6)$$

i.e. $h(x)$ and $h_i(x)$ both have the same index S modulo $g(x)$ and $g_i(x)$ respectively.

By multiplying both sides of the equations (2.5) and (2.6) respectively by $g(x)$ and $g_i(x)$ respectively and adding them together we note that

$$X^S = (h(x) \bullet g_i(x) + h_i(x) \bullet g(x))/(g(x) + g_i(x)) = r \qquad mod \; (g(x) \bullet g_i(x))$$

Furthermore by multiplying equations (2.5) and (2.6) together we note that

$$(X^S - h(x)) \bullet (X^S - h_i(x)) = 0 \qquad mod \; (g(x) \bullet g_i(x))$$

which is equivalent to

$$r^2 + r(h(x) + h_i(x)) + h(x) \bullet h_i(x) = 0 \qquad mod \; (g(x) \bullet g_i(x)) \qquad (2.7)$$

The signature of the message $h_i(x)$ is verified if the equality (2.7) is satisfied.

3 Security of the system, choice of parameters and their generation

The security of the presented system is based on the discrete logarithm problem (DLP) over field $GF(2^k)$. Assuming that α is the root of the base primitive polynomial $g(x)$ the root of the given public primitive polynomial $g_i(x)$ is α^i. It is well known that for a given α^i it is quite easy to construct its minimal polynomial $g_i(x)$ [4]. For given polynomial $g_i(x)$ its root as a polynomial $f(\alpha)$ can be found using the algorithm presented in [6, chapter4]. The complexity of that algorithm is not more than $O(k^3)$. However for a given $f(\alpha)$ to find $\alpha^i = f(\alpha)$ is a DLP. Equivalently to find $C(x)$ or $C_i(x)$ from $(2.2a)$ or $(2.2b)$ - the complexity for decryption for an unauthorized user or to calculate $h_i(x)$ from $h(x)$ in (2.4) which is the complexity for signing a document by unauthorized user, both are DLP. DLP in this case is a very well studied problem and a good survey of this topic can be found in [6-7]. Recent developments in DLP for the fields of small characteristics [8] shows, that according to the current state of the art for DLP it would be prudent to consider the size of the field with prime extension to be equal at least to 2048 for example the field $GF(2^{2053})$.

4 Implementation aspects of the system

Now let us discuss implementation aspects of the system: An encryption operation for “off-line” mode according to the formulae $(2.2a)$ or $(2.2b)$ is just one multiplication. An encryption operation for “on-line” mode is more complicated. For calculation of $C(x)$ and $C_i(x)$ according to formula (2.3) we will need on average 2048 multiplication and 2 × 2048 squaring operations.

However all squaring operations can be in this case pre-computed, namely meaning that the values α^{2^r} $\{r = 1, 2, \cdots 2048\}$ and $\alpha^{i \cdot 2^r}$ $\{r = 1, 2, \cdots 2048\}$ can be pre-computed which will require 0,6 mbyte of storage. As such the complexity of "on-line" encryption will be equivalent to the one regular exponentiation for the length 2048.

Decryption operation for both "off-line" and "on-line" encryption will require one regular exponentiation as in the case with "on-line" encryption and one multiplication or one XOR operation. Signature generation will basically require, except for calculation of the hash of the message, one regular exponentiation for the length 2048.

This brief analysis shows that encryption operation for both modes for this system has the same complexity as for the EIGamal type encryption. RSA encryption operation in "on-line" mode is faster, but for "off-line" mode the presented system is much faster, provided that for RSA encryption "off-line" mode does not exit. When comparing decryption operations we can conclude that the system presented here has about the same complexity compared with both RSA or EIGamal type decryption since it basically requires one regular exponentiation. The signature generation of this system is equivalent to RSA since both require one exponentiation. The comparison with DSS signature generation is a more subtle matter. DSS requires two exponentiations, one of which can be made "off-line" and it also requires one inverse operation. However DSS exponentiation is made over about six times shorter exponents and thus would be faster.

Finally the signature verification according to formula (2.7) is extremely fast - it will be even much faster than the corresponding RSA signature verification. It should be mentioned that the most costly operation in this case, which is the computation of $(g(x) + g_i(x))^{-1}$ can be pre-computed. As such the signature verification will require according to formula (2.7) just 4 multiplications and one squaring.

5 Conclusion

In this paper a new public-key system which is based on DLP is developed. It is shown, that all public key operations of the presented system can be implemented virtually with the same complexity compared with existing systems, except signature verification which can be implemented significantly faster, even compared with RSA.

References

[1] W. Diffie and M.E. Helman, New Directions in Cryptography, IEEE Transactions on Information Theory, Vol. IT-22, Nov.1976, 644–654.

[2] R. Rivest, A. Shamir, L. Adleman, A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, Communications of the ACM 21 (2), (1978), 120–126.

[3] Digital Signature Standart, Federal Information Processing Standarts Publication 186, May 1994.

[4] Taher EIGamal, A public-key cryptosystem and a signature scheme based on discrete logarithms, IEEE Transactions on Information Theory, Vol. IT-31, n.4, 1985, 469–472, also in CRYPTO 84, 10–18, Springer-Verlag.

[5] Lidl, Niederreiter (1997), Finite Fields (2nd ed.), Cambridge Univ. Press

[6] K. McCurley, The discrete logarithm problem, Proceedings of Symposia in Applies Mathematica, Vol. 42, 1990, 49–74.

[7] A. Odlyzko, Discrete logarithms: The past and the Future; Designs, Codes, and Cryptography, (2000), 129–145.

[8] R.Barblescu, P. Gaudry, A. Joux, E. Thome. "A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristics" Advances in cryptography-Eurocrypt, Lecture Notes in computer Science, Vol. 844, pp 1-8.