A Provably Secure Public Key Encryption Scheme Based on Isogeny Star

Weiwei Han
School of Mathematics and Statistics, Guangdong University of Finance and Economics, China

Abstract: *Public Key Encryption (PKE) scheme based on isogeny star has been proposed to be against the attack of the quantum computer for several years. But, there is no report about provable security PKE scheme based on isogeny star. In this paper, we propose a PKE scheme based on isogeny star and prove the security of the scheme in the random oracle.*

Keywords: *PKE, isogeny, quantum computer, elliptic curve.*

Received December 2, 2011; accepted September 2, 2013; published online March 13, 2014

1. Introduction

Since its invention by Diffie and Hellman [3], Public Key Encryption (PKE) scheme has become one of the essential techniques in providing security services in modern communications. In public-key cryptosystems, a pair of public/private keys is computed by each user. Since, the Public Key (PK) is a tring of random bits, a digital certificate of the PK is required to provide PK authentication. Anyone who wants to send messages to others just needs to obtain their authorized certificates that contain the PK and encrypt the message using the PK.

Security of the traditional PKE scheme is based on two general mathematical problems: Determination of order and structure of a finite Abelian group and discrete logarithm computation in a cyclic group with computable order. Both of the two problems could be solved in polynomial time using Shor's algorithm on a quantum computer [1]. Most of the current pulic-key crypthosystems will become insecure when the size of a quantum register is sufficient. Thus, it is necessary to develop PKE scheme, which could withstand attacks of the quantum computer. Several years ago, researchers demonstrated that it is hard for the quantum compute to solve the computing an isogeny (an algebraic homomorphism) between the elliptic curves problem. Since then, the isogeny between elliptic curves has been widely used to construct PKE scheme [12], key agreement protocol [10] and random number generator [8, 9].

In this paper, we will propose the first probably secure PKE scheme based on isogenies between elliptic curves. The remainder of our paper is organized as follows: Section 2 reviews theoretical background of isogenies between elliptic curves over F_P and security model of the PKE. Section 3 gives the proposed PKE scheme based on isogenies between elliptic curves and gives the security proof of the proposed protocol. Section 4 makes concluding remarks.

2. Preliminaries

In this paper we use an elliptic curve E defined over a finite field F_P, the parameters are described as follows:

- F_P: The finite field.
- E_{init}: An initial elliptic curve, its equation is $y^2 = x^3 + a_{init}x + b_{init}$, $a_{init}, b_{init} \in F_P$.
- S_E: The set of all elliptic curve, which are isogenous with E_{init}.
- E: The isogeny star consists of all elements of S_E.
- d: Number of isogeny degrees being used.
- $L = \{l_i\}$, $1 \le l_i \le d$: a set of Elkies isogeny degrees being used.
- $F = \{\pi_i\}$, $1 \le l_i \le d$: a set of Frobenius eigenvalues, which specify the positive direction for every $l_i \in L$.
- k: A limit for number of steps by one isogeny degree in a root. For any route $\{r_i\}$, numbers of steps are selected in $-k \le r_i \le k$.
- G: The set of all routes, defined over S, satisfy the above conditions.
- H_1, H_2: Two independent random oracles [7].

Let E be an elliptic curve, defined on the finite fields F_P, and its equation is:

$$y^2 = x^3 + ax + b, \quad a, b \in F_P \tag{1}$$

Then, the map:

$$\pi : (x, y) \mathrm{\textcircled{R}} (x^p, y^p) \tag{2}$$

Specifies the Frobenius endomorphism of the curve E. A Frobenius map satisfies its characteristic equation:

$$\pi^2 - T\pi + p = 0 \quad (3)$$

Where $T = p - a - \# E(F_p)$ is the Frobenius trace. Through the Hasse's theorem, we know that:

$$|T| < 2\sqrt{p} \quad (4)$$

So, the discriminant D_π of the Frobenius Equation 3 satisfies:

$$D_\pi = T^2 - 4p < 0 \quad (5)$$

- *Theorem 1* [12]: Elliptic curves are isogenous over F_P if and only if they have equal number of points.
- *Theorem 2* [12]: Let an elliptic curve $E(F_P)$ have the Frobenius discriminant D_π and $\left(\frac{D_\pi}{l}\right)$ be a Kronecker symbol for some l-degree isogeny. If $\left(\frac{D_\pi}{l}\right) = -1$, then there are no l-degree isogenies; if $\left(\frac{D_\pi}{l}\right) = 1$, then two l-degree isogenies exist; if $\left(\frac{D_\pi}{l}\right) = 0$, z then 2 or $l + 1$ l-degree isogenies exist and l is called Elkies prime number.

Let $U=\{E_1(F_P)\}$ be a set of elliptic curves with equal number of points, so that each element of U is uniquely determined by a j-invariant of an elliptic curve. According to the Theorem 1 and the Equation 4, we can consider U as a category, and the set of isogenies between elements of U as a set of morphisms of this category. We can compute $U = h_{D_\pi}$, where h_{D_π} is the degree of Hilbert polynomial [12].

Let l be Elkies prime number, we can get that there are two isogenous elliptic curves for any elliptic curves of U, from Theorem 2. It is practically determined that, when $\#U$ is prime, all the elements of U form a single isogeny cycle.

Let $l_1 \neq l$ be one more prime isogeny degree with the property that $\left(\frac{D_\pi}{l}\right) = -1$ In this case, l_i-degree isogenies form a cycle over U as well. Then, we can put the l- and l_i-degree isogeny cycles over each other. Same can be done for other isogeny degrees of such kind.

- *Definition 1*: A graph, consisted of prime number of elliptic curves, connected by isogenies of degrees satisfying $\left(\frac{D_\pi}{l}\right) = -1$, is an isogeny star.

If an isogeny star is wide enough, we can use it for crypto algorithm constructing. For that purpose, it is necessary to specify a direction on a cycle and route of isogeny stat. The method for direction determination on an isogeny cycle is mentioned in [12], we don't give the detail here. It uses impact of Frobenius endomorphism on an isogeny kernel. The definition of isogeny stat is following.

Let S be an isogeny star, $L=\{l_i\}$-a set of Elkies isogeny degrees being used and $F = \{\pi_i\}$ - a set of Frobenius eigenvalues, which specify positive direction for every $l_i \in L$.

- *Definition 2*: A set $R=\{r_i\}$, where r_i is number of steps by the l_i -isogeny in the direction π_i, is a route on the isogeny star.

We can define composition [3] of routes $A=\{a_i\}$ and $B=\{b_i\}$ as $AB=\{a_i+b_i\}$. Routes are commutative: $AB=BA$.

The computation of iosgeny between elliptic curve can be done using the method in [2, 4, 6, 11], we don't give the detail here.

The following several techniques can be used for isogeny search [3]:

1. *Brute-force*: Complexity of these attacks is estimated at $O(n)$ isogeny computations.
2. *Meet-in-the-middle*: Complexity of the attack is estimated at $O(\sqrt{n})$ isogeny computations.
3. *Method described in [6]*: Complexity of the attack is estimated at. $O(\sqrt[4]{p})$. Isogeny computations.

The reason of the problem of searching for an isogeny between elliptic curves can against the attack of quantum computer is following. In order to, compute the isogenies between elliptic, we must solve the equation:

$$\phi_l(X, j)=0 \quad (6)$$

And the process of computing the isogeny cycle is following: $E1 \rightarrow \varphi_{l1}(X, j_{E1})=0 \rightarrow E2 \rightarrow \varphi_{l2}(X, j_{E2})=0 \rightarrow j_{E3}$.

To compute a chain of q isogenies, one should consecutively solve these q equations, because of the equation parameter j-invariant changes with every step. So, one can't parallelize and the problem is against the attack of quantum computer.

Then, we conclude that the complexity of searching for an isogeny between elliptic curves is $O(\sqrt{n}) \approx O(\sqrt[4]{p})$, and the problem can be against the attack of quantum computer. Then, the Computational Diffie-Hellman Problem (CDHP) over the isogeny star can be described as follows. CDHP For $R_1(E_{init})$ and $R_2(E_{init})$ the initial elliptic curve, it is difficult to compute $R_1(R2(E_{init}))$. A PKE scheme consists of four algorithms: Setup, KeyGen, Encrypt and Decrypt.

- *Setup*: Taking security parameter k as input and returns the system parameters *params*.
- *KeyGen*: Taking the system parameters *params* as input and return public/private key pairs.
- *Encrypt*: It takes as input a message M, *params*, and the PK of the receiver, the sender runs this algorithm to produce a ciphertext C.

- *Decrypt*: Taking as input a ciphertext C, params and the private key, the user runs this algorithm to output a message M or a failure symbol $\perp$.
- Semantically Security: First of all, we review two security definitions [5, 12] semantic security against Chosen Plaintext Attack (IND-ID-CPA) and semantic security against Chosen Ciphertext Attack (IND-ID-CCA).

2.1. Semantic Security Against CPA

The following game is played between a challenger and an adversary.

- *Setup*: The challenger runs the Setup algorithm and generates the public parameters *params* with a security parameter k then it sends *params* to the adversary.
- *Phase 1*: The adversary can make queries $q_1, q_2, \ldots, q_l$, and each query q_i may adaptively depend on the answers to the previous queries $q_1, q_2, \ldots, q_{l-1}$. In this phase, there exists one type of queries: Key generate queries. The challenge runs the algorithm KeyGen and replies with the corresponding private/ PK pair (SK_i, PK_{-i}).
- *Challenge*: The adversary query the challenger another never used PK and two different message M_0 and M_1. The challenger uniformly selects $\sigma \in \{0, 1\}$ at random and encrypts M_σ by computing $C = Encrypt(M_\sigma, PK)$. Then, he returns the ciphertext as the challenge to the adversary.
- *Phase 2*: Repeating the Phase 1 with the only one restriction that the adversary cannot request a private key for *PK*.
- *Guess*: Finally, the adversary submits a guess $\sigma' \in \{0, 1\}$. If $\sigma' = \sigma$, the adversary wins the game; otherwise, he loses the game.

The advantage of the adversary against the PKE in the IND-CPA game is defined as follows and the probability is over the coin flips of the challenger:

$$Adv_{PKE}^{CCA} = \left| Pr[\sigma' - \sigma] - 1/2 \right|$$

- *Definition 3*: (IND-CPA secure). An PKE scheme is secure against an adaptive chosen plaintext attack if Adv_{PKE}^{CPA} is negligible for all Probabilistic Polynomial Time (PPT) adversaries.

2.2. Semantic Security Against CCA

Now, we discuss the IND-ID-CCA game played between a challenger and an adversary.

- *Setup*: The challenger runs the Setup algorithm and generates the public parameters *params* with a security parameter k, then it sends *params* to the adversary.
- *Phase 1*: The adversary can make queries $q_1, q_2, \ldots, q_l$, and each query q_i may adaptively depend on the answers to the previous queries $q_1, q_2, \ldots, q_{l-1}$. In this phase, there exist two types of queries: Key generate queries and ciphertext queries. The challenge runs the algorithm KeyGen and replies with the corresponding private/ PK pair (SK_i, PK_{-i}). And the challenge runs the algorithm Decrypt to decrypt the ciphertext C_i and replies with the corresponding plaintext when he receives a ciphertext query (PK_i, C_i).
- *Challenge*: The adversary query the challenger another never used PK and two different messages M_0 and M_1. The challenger uniformly selects $\sigma \in \{0, 1\}$ at random and encrypts M_σ by computing $C = Encrypt(M_\sigma, PK)$.Then, he returns the ciphertext as the challenge to the adversary.
- *Phase 2*: Repeating the Phase 1 with the only one restriction that the adversary cannot request a private key for the public *PK*.
- *Guess*: Finally, the adversary submits a guess $\sigma' \in \{0, 1\}$. If $\sigma' = \sigma$, the adversary wins the game; Otherwise, he loses the game.

The advantage of the adversary against the PKE in the IND-CCA game is defined as follows and the probability is over the coin flips of the challenger:

$$Adv_{PKE}^{CCA} = \left| Pr[\sigma' - \sigma] - 1/2 \right|$$

- *Definition* 4: (IND-CCA secure). An PKE scheme is secure against an adaptive chosen ciphertext attack if Adv_{PKE}^{CCA} is negligible for all PPT adversaries

3. Our PKE Protocol and Security Analysis

In this section, this section, we will proposed an IND-CPA secure PKE scheme π based on isogeny star, then we can get an IND-CCA secure PKE scheme π' with the method proposed by Fujisaki *et al*. [5]. The detail of scheme π is described as follows:

- *Setup*: Taking a security parameter k, generate the system parameters $params = \{F_P, E_{init}, d, L, F, H_l\}$.
- *KeyGen*: The user selects a random route R_{priv}, compute the elliptic curve $E_{pub} = R_{priv}(E_{init})$ and set R_{priv} and E_{pub} as the private key and PK separately.
- *Encrypt*: For given plaintext M, the user choose a random route R and compute the ciphertext:

$$C = \left(R(E_{init}), M \oplus H_1 R(E_{pub})) \right)$$

- *Decrypt*: Let $C = (C_1, C_2)$ be a valid ciphertext encrypted using the PK E_{pub}. The user can decrypt ciphertext using the private key R_{priv}: $C_2 \oplus H_1\left(R_{priv}(C_1)\right) = M$.

It is easy to say that the random routes could be generated by the random number generators [5, 13]. Since:

$$R_{priv}(C_1) = R_{priv}(R(E_{init})) = R_{priv}(R(E_{init})) = R(E_{pub})$$

Then, the correctness of our scheme is provided.

- *Theorem 3*: Uppose the hash function H is a random oracle. The scheme π is an IND-CPA secure encryption scheme assuming CDHP is hard. Concretely, suppose there is an IND-CPA adversary A that has advantage ε against π scheme and A makes at most q_{H_1} queries to H_l. Then, there is a simulator B which solves CDHP with advantage at least $\frac{2\varepsilon}{q_{H_1} q_u}$, where q_u is the user in the system.
- *Proof*: We will use Zhang *et al*. [13] method to prove the correctness of the theorem. Let E_{init} be the initial elliptic curve and R_1, R_2 be random route in G. Let $(E_{init}, R_1(E_{init}), R_2(E_{init})) = (E_{init}, E_1, E_2)$ be a CDHP and $E3 = R1R2(E_{init})$ be the solution to this CDHP. Simulator B finds E_3 by interacting with A as follows:
  - *Setup*: Simulator A produces the parameter A. $params = \{F_P, E_{init}, d, L, F, H_l\}$ randomly selects a user $I \in \{1, 2, \cdots, q_u\}$ and set E_1 as its PK. So, the corresponding private key is R_1 (unknown to A). Here, H_1 is a random oracle controlled by B as described below:
    1. *H_1-queries:* For each query issued by the adversary A to the random oracle H_1, simulator B maintains a list B -list. Each entry in the list is a tuple of the form (E_i, h_i). Initially the lisSt is empty. Simulator B replies each query E_i as follows:
       a. If the query E_i already appears on the H_1-list in a tuples (E_i, h_i), then simulator B responds with the recorded hash value h_i.
       b. If the query is a new query to the H_1 oracle, B responds a random string h_i and records the tuple E_i, h_i in list ... -list.
- *Challenge*: Adversary submits two plaintexts M_0, M_1 to simulator B. B selects a random string C_2 and sets ciphertext $C = (E_2, C_2)$. Simulator B sends C as the challenge to adversary A. Observe that, the decryption of C. is:

$$C_2 \oplus H_1(R_1(E_2)) = C_2 \oplus H_1(R_1(R_2(E_{init}))) = C_2 \oplus H_1(E_3)$$

- *Guess*: Adversary A outputs its guess $\sigma' \in \{0, 1\}$. Simulator B chooses a random tuple (E_i, h_i) from the H_1-list and outputs E_i as the solution to the given instance of CDHP.

Let e be the event that adversary A issues a query for $H_1(E_i)$ during the simulation above. Then, we know that r and $|Pr[\sigma' = \sigma] - 1/2| \geq 2\varepsilon$. So, we have $Pr[e] \geq 2\varepsilon$. Therefore, simulator B outputs the correct answer E_3 with probability at least $\frac{2\varepsilon}{q_{H_1} q_u}$.

Since, our scheme π is an IND-CPA secure encryption scheme, then we can use Fujisaki *et al*. [5] method to construct an IND-CCA secure encryption scheme π'. The detail of the scheme π' is described as follows:

- *Setup*: Taking a security parameter k, generate the system parameters $parms = \{Fp, E_{init}, d, L, F, H_1, H_2\}$.
- *KeyGen*: The user selects random routes R_{priv}, compute the elliptic curve $E_{pub} = R_{priv}(E_{init})$ and set R_{priv} and E_{pub} as the private key and PK separately.
- *Encrypt*: For given plaintext M, the user choose a random route R, a random number r, computes $\alpha = H_2(M \| r), M' = M \| r \| \alpha$ and the ciphertext $C = (R(E_{init}), M' \oplus H_1(R(E_{pub})))$.
- *Decrypt*: Let $C = (C_1, C_2)$ be a valid ciphertext encrypted using the PK E_{pub}. The user with the private key R_{priv} computes $C_2 \oplus H_1(R_{priv}(C_1)) = M' = M \| r \| \alpha$ and checks whether $\alpha = H_2(M \| r)$ holds. If it does, the user output the message M. Otherwise, the user output $\perp$ and stops.
- *Theorem 4*: Suppose the hash function H_1 and H_2 are random oracles. The scheme π' is an IND-CCA secure encryption scheme assuming CDHP is hard.
- *Proof*: Since, our scheme π is an IND-CPA secure encryption scheme, then we can conclude the scheme π' is an IND-CCA secure encryption scheme from the theorem 3 in [5].

4. Conclusions

We review the knowledge of isogeny star and presented a new key agreement protocol based on the isogeny star. The proposed protocol can be against the attack of the quantum computer and can be proved security in the random oracle.

References

[1] Boneh D. and Lipton R., "Quantum Cryptanalysis of Hidden Linear Functions," *in Proceedings of the 15th Annual International Cryptology Conference on Advances in Cryptology*, Santa Barbara, USA, pp. 424-437, 1995.

[2] Couveignes J., Dewaghe L., and Morain F., available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.48.3312, last visited 2009.
[3] Diffie W. and Hellman M., "New Directions in Cryptography," *IEEE Transaction Information Theory*, vol. 22, no. 6, pp. 644-654, 1976.
[4] Elkies N., "Elliptic and Modular Curves Over Finite Fields and Related Computational Issues," *in Proceedings of a Conference in Honor of A.O.L.* Atkin, Chicago, USA, pp. 27-45, 1998.
[5] Fujisaki E. and Okamoto T., "How to Enhance the Security of Public-Key Encryption at Minimum Cost," *in Proceedings of the 2nd International Workshop on Practice and Theory in Public Key*, London, UK, pp. 53-68, 1999.
[6] Galbraith S., "Constructing Isogenies between Elliptic Curves Over Finite Fields," *Journal of Computational Mathematics*, vol. 2, no. 1, pp. 118-138, 1999.
[7] He D. and Chen J., "An Efficient Certificate less Designated Verifier Signature Scheme," *the International Arab Journal of Information Technology*, vol. 10, no. 4, pp. 317-324, 2013.
[8] He D., Chen J., and Hu J., available at: http://eprint.iacr.org/2010/094. pdf, last visited 2010.
[9] He D., Chen J. and Hu J., "A Random Number Generator Based on NTRU Cryptosystem," *Maejo International Journal Science Technol*ogy, vol. 4, no. 3, pp. 428-434, 2010.
[10] He D., Chen J., and Hu J., "An Authenticated Key Agreement Protocol Using Isogenies Between Elliptic Curves," *International Journal of Computers, Communications & Control*, vol. 6, no. 2, pp. 258-265, 2011.
[11] Morain F. and Schost E., available at: http://www.lix.polytechnique.fr/~morain/jcomp.pdf, last visited 2006.
[12] Rostovtsev A. and Stolbunov A., available at: http://eprint.iacr.org/2006/145, last visited 2009.
[13] Zhang R., Liu J., Han Z. and Zheng L., "An IBE Scheme Using ECC Combined Public Key," *Computers and Electrical Engineering*, vol. 36, no. 6, pp. 1046-1054, 2010.

Weiwei Han received her MS degree in mathematics from Zhongshan University, Guangzhou, China, in 2005. She is a teaching staff in Guangdong University of Business Study. Her research interests include information security, statistical learning, machine learning, and data mining.