Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 2 (June), pp. 258-265

An Authenticated Key Agreement Protocol Using Isogenies Between Elliptic Curves

D. He, J. Chen, J. Hu

Debiao He, Jianhua Chen, Jin Hu
Wuhan University
School of Mathematics and Statistics
Wuhan, Hubei, 430072, China
E-mail: hedebiao@163.com, chenjh-ecc@163.com, hujin-ecc@163.com

Abstract: All the current public-key cryptosystems will become insecure when size of a quantum register is sufficient. An authenticated key agreement protocol, which is against the attack of quantum computer, is proposed. The proposed protocol can provide the security properties known session key security, forward security, resistance to key-compromise impersonation attack and to unknown key-share attack, key control. We also prove its security in a widely accepted model.
Keywords: public-key cryptosystem; quantum computer; isogeny; elliptic curve; key agreement protocol.

1 Introduction

Key agreement is one of the fundamental cryptographic primitive after encryption and digital signature. Such protocols allow two or more parties to exchange information among themselves over an adversarially controlled insecure network and agree upon a common session key, which may be used for later secure communication among the parties. Thus, secure key agreement protocols serve as basic building block for constructing secure, complex, higher-level protocols. The first modern key agreement protocol is the Diffie-Hellman protocol given by Diffie and Hellman in the seminal paper [1] in 1976. Its security is based on the difficulty of solving discrete logarithm problems. As the first practical key agreement protocol without authentication, it is not secure to the man-in-the-middle attack. After that, lots of protocols have been published and some of them use certificates generated by thetrusted third parties (public key infrastructures, PKIs) to prevent attacks such as the man-in-the-middle attack. Most of the systems based on PKI are complex and expensive for the cost of the authentication, refresh andrevocation of certificates. Security of the known key agreement protocols 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 problems can be solved in polynomial time using Shor's algorithm for a quantum computer [2]. Thus, most of the current public-key cryptosystems will become insecure when size of a quantum register is sufficient. Development of key agreement protocols, which would be strong against a quantum computer, is necessary.

The rest of our paper is organized as follows. Section 2 describes theoretical background and a public-key encryption technique. Section 3 analyses the complicity of the problem of searching for an isogeny between elliptic curves. We give the proposed key agreement protocol in Cection4, and analyse the security of the proposed protocol in Section 5. We conclude this paper in Section 6.

2 Elliptic Curves over F_p and Isogeny Star

Let E be a elliptic curve, defined on the finite fields F_p, and it's equation is

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

Then the map

$$\pi : (x, y) \to (x^p, y^p) \tag{2}$$

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

$$\pi^2 - T\pi + p = 0 \tag{3}$$

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

$$|T| < 2\sqrt{p}). \tag{4}$$

So the discriminant D_π of the Frobenius equation (3) satisfies

$$D_\pi = T^2 - 4p < 0 \tag{5}$$

Theorem 1 Elliptic curves are isogenous over F_p if and only if they have equal number of points.

Proof. See[3].

Theorem 2 Let an elliptic curve $E(F_p)$ have the Frobenius discriminant D_π and $(\frac{D_\pi}{l})$ be a Kronecker symbol for some l-degree isogeny. If $(\frac{D_\pi}{l}) = -1$, then there are no l-degree isogenies; if $(\frac{D_\pi}{l}) = 1$, then two l-degree isogenies exist; if $(\frac{D_\pi}{l}) = 0$, then 2 or $l + 1$ l-degree isogenies exist and l is called Elkies prime number.

Proof. See[3].

Let $U = E_i(F_p)$ 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 $\sharp U = h_{D_\pi}$, where h_{D_π} is the degree of Hilbert polynomial[3].

Let l is 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 $\sharp U$ 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 $(\frac{D_\pi}{l_i}) = 1$. 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 $(\frac{D_\pi}{l_i}) = 1$, 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 [4], 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 $F = \pi_i$, 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 [5, 6, 7], we don't give the detail here.

3 Complexity of Isogeny Search

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

- **Brute-force**: Complexity of these attacks is estimated at isogeny computations.
- **Meet-in-the-middle**: Complexity of the attack is estimated at isogeny computations.
- **Method described in [8]**: Complexity of the attack is estimated at isogeny computations.

The reason of the problem of searching for an isogeny between elliptic curves can against the attack of quantum computer is following[3].

In order to computer the isogenies between elliptic, we must solve the equation

$$\phi(X, j) = 0, \tag{6}$$

and the process of computing the isogeny cycle is following

$$E_1 \to \phi_{l1}(X, j_{E1}) = 0 \to j_{E2} \to E_2 \to \phi_{l2}(X, j_{E2}) = 0 \to j_{E3} \to ... \tag{7}$$

.

To compute a chain of q isogenies, one should consecutively solve these q equations, because of the equation parameter ($j-$invariant) is changed 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.

From the above discussion, the decisional Diffie-Hellman assumption can be easily extent to the isogenies through the property of isogenies between the elliptic curves.

Definition 3. The decisional Diffie-Hellman assumption over isogeny star (*DDHA-IS*): *DDHA-IS* is that it is difficult to distinguish the following real Diffie-Hellman distribution

$$\Gamma_{real} = \{R_1(E_{init}), R_2(E_{init}), R_1R_2(E_{init}) | R_1, R_2 \in G\} \tag{8}$$

and random Diffie-Hellman distribution

$$\Gamma_{rand} = \{R_1(E_{init}), R_2(E_{init}), R_3(E_{init}) | R_1, R_2, R_3 \in G\} \tag{9}$$

.

More formally, if we define the advantage function $Adv_G^{DDH-IS}(A)$ as

$$Adv_G^{DDH-IS}(A) = |Pr[A(X) = 1 | X \in \Gamma_{real}] - Pr[A(Y) = 1 | Y \in \Gamma_{rand}]|, \tag{10}$$

we say that the *DDHA-IS* holds in set G if $Adv_G^{DDH-IS}(A)$ is negligible for any probabilistic polynomial time adversary A.

4 Key Agreement Protocol Based on Isogeny

In this section we describe the proposed key agreement protocol which is specified by the key generation and the protocol description.

4.1 Key Generation

In this paper we use an elliptic curve E defined over a finite field F_p, the parameters is following.

1)F_p: the finite field;

2)E_{init}: an initial elliptic curve, its equation is $y^2 = x^3 + a_{init}x^3 + b_{init}, a_{init}, b_{init} \in F_p$Łt'

3)d:number of isogeny degrees being used;

4)$L = l_i, 1 \leq i \leq d$: a set of Elkies isogeny degrees being used;

5)$F = \pi_i, 1 \leq i \leq d$: a set of Frobenius eigenvalues, which specify the positive direction for every $l_i \in L$;

6)k:a limit for number of steps by one isogeny degree in a root. For any root $\{r_i\}$, numbers of steps are selected in $-k \leq r_i \leq k$;

7)H:SHA-1;

8)Select random routes R_{privA} and R_{privB}. The value R_{privA} is a secret key of the user A, and R_{privB} is the secret key of the user B;

9)Compute the curves $E_{pubA} = R_{privA}(E_{init})$ and $E_{pubB} = R_{privB}(E_{init})$, which are the public key of a user A and B, respectively;

4.2 Our Key Agreement Protocol

Let E be the elliptic curve, defined on the finite field , with the equation (1), and let A_E and B_E be its parameter and j be its $j-$invariant. The proposed protocol is following.

1) A generate random route R_A and computes $E_A = R_A(E_{init}), e_A = H(A_{E_A}, B_{E_A})$. At last, A sends $M_1 = \{A_{E_A}, B_{E_A}, e_a\}$ to B.

2) Upon receiving M_1, B checks whether e_A equals $H(A_{E_A}, B_{E_A})$. If not, B stops the session. Otherwise, B generate random route R_B and computes $E_B = R_B(E_{init})$, $E'_B = R_B(E_A)$, $E''_B = R_{privB}(E_A)$, $e_B = H(A_{E_B}, B_{E_B}, A_{E''_B}), B_{E''_B})$. At last, B sends $M_2 = \{A_{E_B}, B_{E_B}\}$ to A.

3) Upon receiving M_2, A computes $E'_A = R_A(E_B)$, $E''_A = R_A(E_{pubB})$, $E'''_A = R_A(E_{pubB})$ and checks whether the equation $e_B = H(A_{E_B}, B_{E_B}, A_{E''_A}), B_{E''_A})$ holds or not. If it does not hold, then A terminates the execution. Otherwise, A computes $e'_A = H(A_{E'''_A}), B_{E'''_A})$ the session key $sk_{AB} = H(A_{E'_A}) \oplus B_{E'_A})$. At last, A sends $M_3 = e'_A$ to B.

4) Upon receiving M_3, B computes $E'''_B = R_B(E_{pubA})$ and checks whether the equation $e'_A = H(A_{E'''_B}), B_{E'''_B})$ holds or not. If it does not hold, then B terminates the execution. Otherwise, computes the session key $sk_{BA} = H(A_{E'_B}) \oplus B_{E'_B})$.

As a result, A and A achieve the same shared secret key:

$$E'_A = R_A(E_B) = R_A(R_B(E_{init})) = R_B(R_A(E_{init})) = R_B(E_A) \tag{11}$$

$$sk_{AB} = H(A_{E'_A}) \oplus B_{E_A}) = H(A_{E'_A}) \oplus B_{E'_A}) = sk_{AB} \tag{12}$$

and authenticate each other.

5 Security analysis

5.1 Security model

In this work we shall use a modified Bellare-Rogaway key exchange model [9, 10] to analyse the protocol security. In the model, each party involved in a session is treated as an oracle, and an adversary can access the oracle by issuing some specified queries (defined later). An oracle $\Pi_{i,j}^s$ denotes the $s-$th instance of party i involved with a partner party j in a session.

The security of a protocol is defined by a game with two phases. In the first phase, an adversary E is allowed to issue the following queries in any order.

1)$Send(\Pi_{i,j}^s, m)$. Upon receiving the message m, oracle $\Pi_{i,j}^s$ executes the protocol and responds with an outgoing message m or a decision to indicate accepting or rejecting the session. If the oracle $\Pi_{i,j}^s$ does not exist, it will be created as initiator if $m = \lambda$, or as a responder otherwise.

2)$Reveal(\Pi_{i,j}^s)$. If the oracle has not accepted, it returns $\perp$; otherwise, it reveals the session key.

3)$Corrupt(i)$. The party responds with its private key.

Once the adversary decides that the first phase is over, it starts the second phase by choosing a fresh oracle and issuing a query, where the fresh oracle and query are defined as follows.

Definition 4 (fresh oracle) An oracle $\Pi_{i,j}^s$ is fresh if (1)$\Pi_{i,j}^s$ has accepted; (2)$\Pi_{i,j}^s$ is unopened (not been issued the query); (3) party $j \neq i$ is not corrupted (not been issued the $Corrupt$ query); (4) there is no opened oracle $\Pi_{j,i}^t$, which has had a matching conversation to . The above fresh oracle definition is particularly defined to cover the key-compromise impersonation resilience property since it implies that the user could have been issued a query.

4)$Test(\Pi_{i,j}^s)$.Oracle $\Pi_{i,j}^s$ which is fresh, as a challenger, randomly chooses $b \in \{0, 1\}$ and responds with the session key, if $b = 0$, or a random sample from the distribution of the session key otherwise.

After this point the adversary can continue querying the oracles except that it cannot reveal the test oracle $\Pi_{i,j}^s$ or its partner $\Pi_{j,i}^t$ (if it exists), and it cannot corrupt party j. Finally, the adversary outputs a guess b'' for b. If $b' = b$, we say that the adversary wins. The adversary's advantage is defined as

$$Adv^E(k) = |2Pr[b' = b] - 1|. \tag{13}$$

We use the session ID which can be the concatenation of the messages in a session to define matching conversations, i.e., two oracles $\Pi_{i,j}^s$ and $\Pi_{j,i}^t$ have matching conversations to each other if they have the same session ID.

A secure authenticated key (AK) agreement protocol is defined as follows.

Definition 5 Protocol Π is a secure AK if:

1). In the presence of a benign adversary, which faithfully conveys messages, on $\Pi_{i,j}^s$ and $\Pi_{j,i}^t$, both oracles always accept holding the same session key, and this key is distributed uniformly on $\{0, 1\}^k$;

2). For any polynomial time adversary E, $Adv^E(k)$ is negligible.

5.2 Security analysis

Using the above security definitions, we have the following Theorem 1.

Theorem 3. In the random oracle, if DDHA-IS is hard, our proposed protocol is a secure AK protocol.

Proof: The first two conditions follow immediately from the description of our proposed protocol and the assumption that H is random oracle.

Let's turn to the second condition. We use the method proposed by Pan et al.[10] to analyze the security. Consider there exists an adversary E and $Adv^E(k)$ is non-negligible. Suppose its

running time is t. We will use E to construct an algorithm F which can solve DDHA-IS. Let l, N and q_H be the number of sessions related to E, the number of entitys and the queries' numbers of $H(\cdot)$ made by E.

Given (E_1, E_2, E_3), where $E_1 = R_1(E_{init}), E_2 = R_2(E_{init}), E_3 = R_3(E_{init})$. First, F selects randomly i, j, r, s and generates the public/private key pair for every entity. Then, F starts E and answers all queries made by E.

H query: If the input exists, answers by its corresponding value. Otherwise, F picks a random number as the answer to the new query, and adds the input, output pair at the end of the $H-$string;

Corrupt query: Because F knows all entities' private keys, F can answers by the corresponding private key;

Reveal query: If the input is $\Pi^s_{i,j}$ or $\Pi^t_{j,i}$, F selects b' as its output and halts. Suppose the input is $\Pi^{s'}_{i',j}$ or $\Pi^{t'}_{j,i'}$, where $(i', s') \neq (i, t)$ and $(i', t') \neq (i, t)$. Because F knows all entity's private keys and simulates the run of i' and j according to the protocol, F can get random numbers selected by i' and j. So, F knows the session key of $\Pi^{s'}_{i',j}$ or $\Pi^{t'}_{j,i'}$, and answers by this session key.

Send query: If the input is $\Pi^s_{i,j}$ and an empty string, F computes $e_A = H(A_{E_1}, B_{E_1})$ and answers by A_{E_1}, B_{E_1}, e_A. If the input is $\Pi^s_{i,j}$ and a string that is not none, F computes $E'''_A = R_{privA}(E_2)$ and answers by $\{e'_A\}$. If the input is $\Pi^t_{j,i}$ and a string that is not none, computes $E''_B = R_{privB}(E_1)$, $e_B = H(A_{E_2}, B_{E_2},), A''_{E_2}, B''_{E_2}$ and answers by A_{E_2}, B_{E_2}, e_B. Else, answers by random numbers according to the protocol.

Test query: If the input is not $\Pi^s_{i,j}$, F outputs $b' \in \{0,1\}^k$ and halts. Else, if A_{E_3} and B_{E_3} exist in $H-$string, let the corresponding value be r; Else, F selects r randomly and appends $\{A_{E_3}, B_{E_3}, r\}$ to the $H-$string. F answers by r.

When E halts, F outputs E's output b' and halts.

1)Suppose $\Pi^s_{i,j}$, selected by F, is the input of the *Test* oracle, and $\Pi^s_{i,j}$ and $\Pi^t_{j,i}$ have matching conversations. If E_3 is the DH value of E_1, E_2, r is the session key. Else r is a random number. Because E_3 is the DH value of E_1, E_2 with the probability $\frac{1}{2}$ and F answers all queries made by E correctly, the probability of the event that distinguishes r and the session key correctly is equivalent to the probability of the event that F decides whether (E_1, E_2, E_3) is a DH triple.

2) Suppose $\Pi^s_{i,j}$, selected by F, is the input of the *Test* oracle, or $\Pi^s_{i,j}$ and $\Pi^t_{j,i}$ have not matching conversations. F outputs correctly with the probability $\frac{1}{2}$.

Suppose the success probability of E is $\frac{1}{2}+\epsilon$, where ϵ is non-negligible. Because the first event happens with the probability $\frac{1}{l^2}$, the success probability of F in this condition is $\frac{1}{2l^2} + \frac{\epsilon}{l^2}$. The second event happens with the probability $1 - \frac{1}{l^2}$. So the success probability of in this condition is $\frac{1}{l^2} - \frac{1}{2l^2}$. From the above discussion, we know the success probability of F is $\frac{1}{2} + \frac{\epsilon}{l^2}$. Then we can solve the DDHP-IS with non-negligible probability. This is a contradiction. So our proposed protocol satisfies the second condition. i.e., the protocol is a secure AK protocol.

5.3 Other discussion

A number of desirable attributes of key agreement protocols have also been identified [9] and nowadays most protocols are analyzed with such attributes. Here, the following six security properties must be considered for the proposed protocol: a known-key security, perfect forward secrecy, a key-compromise impersonation attack, a unknown key-share security, a key-control security. Regarding the above mentioned definitions, the following theorems are used to analyze the six security properties of the proposed protocol. Our protocol also satisfies the following security notions which are often used to judge the security of key agreement protocols.

Known session key security: A protocol is called Known session key security, if an adversary, having obtained some previous session keys, cannot get the session keys of the current run of the key agreement protocol. In our scheme, the agreed session key relies on the one-way hash function and session secrets. The output of hash function is distributed uniformly in $\{0,1\}^k$, thus one session key which is the output of hash function has no relation with the others. Besides, the session key is generated with the session secrets which are computed from the random ephemeral key, thus even one session's session secrets are revealed, the other session secrets will still remain safe.

Perfect forward secrecy: A protocol is called Perfect forward secrecy, if compromise of the three private keys of the participating entities does not affect the security of the previous session keys. Even if an attacker gets the value the secret key R_{privA} and R_{privB} in our scheme, he can't deduce E'_A or E'_B, without the knowledge of the two random numbers R_A and R_B. Therefore, our scheme can provide perfect forward secrecy.

No key-compromise impersonation: The compromise of one entity's static private key does not imply that the private keys of other entities will also be compromised in our protocol. The adversary may impersonate the compromised entity in subsequent protocols, but he cannot impersonate other entities. This property is called no key-compromise impersonation.

First, suppose an attacker C obtains the long-term private key R_{privA} from the compromised user A. In order for the key-compromise impersonation attack to succeed, C must know A's ephemeral keys . In this case, C would also have to extract from 's ephemeral public value R_A, so as to generate the same session key with A. C, however, will face the problem of searching for an isogeny between elliptic curves. Therefore, the proposed protocol is secure against a key-compromise impersonation attack.

No unknown key-share: If the adversary convinces a group of entities that they share some session key with the adversary, while in fact they share the key with another entity, we call the protocol as suffering from unknown key-share attack. To implement such an attack on our protocol, the adversary is required to learn the private key of some entity. Otherwise, the attack hardly works. Hence, we claim that our protocol has the attribute of no unknown key-share.

No key control: No key-control security means that neither entity can't force the session key to a preselected value. From the execution of the proposed protocol, we know that the only possibility of key-control attack may be brought out by the participant of the protocol B. But, for the party B to make the party A generate the session key K_B which is preselected value by B, B should solve the equation $E'_B = R_B(E_A)$.This is the problem of searching for an isogeny between elliptic curves. Therefore, the proposed protocol provides a no key-control security.

6 Conclusion

In this paper, we propose a secure and efficient authenticated key agreement, which works on the isogeny star. We prove that our protocol meets the security attributes under the assumption that the problem of searching for an isogeny between elliptic curves is secure.

7 Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive comments.

Bibliography

[1] W. Diffie and M. Hellman, New directions in cryptography, *IEEE Trans. Info. TH*, vol. 22, pp.644-654, 1976.

[2] Boneh D., Lipton R. Quantum cryptanalysis of hidden linear functions. *Proceedings of the 15th Annual International Cryptology Conference on Advances in Cryptology (LNCS 963)*, 1995:424-437.

[3] Rostovtesv A. and Stolbunov A., Public-key cryptosystem based on isogenies. Cryptology ePrint Archive, Report 2006/145, 2006. http://eprint.iacr.org/.

[4] Couveignes J. M., Dewaghe L., Morain F. *Isogeny cycles and the schoof-elkies-atkin algorithm.* Ecole polytechnique, France, 1996.

[5] Elkies N., Elliptic and modular curves over finite fields and related computational issues, *Proceedings of a Conference in Honor of A.O.L. Atkin*, AMS International Press, 1998, pp.21-76.

[6] Muller V., Ein Algorithmus zur Bestimmung der Punktanzahl elliptisher Kurven uber endlichen Korpern der Charakteristik groser drei, 1995. http://www.informatik.tu-darmstadt.de/ti/forschung/ecc.

[7] F.Morain, E.Schost, Fast Algorithms for Computing Isogenies between Elliptic Curves. http://www.lix.polytechnique.fr/ morain/jcomp.pdf, 2006.

[8] S. Galbraith. Constructing isogenies between elliptic curves over finite fields, *Journal of Computational Mathematics*, vol. 2, pp.118-138, 1999.

[9] S. Blake-Wilson, D. Johnson and A. Menezes, Key Agreement Protocols and Their Security Analysis, *Proceedings of Sixth IMA International Conference on Cryptography and Coding*, Cirencester, UK, 1997, pp. 30-45.

[10] H. Pan, J.-F. Li, Q.-S. Zheng, A Provable-Security Mutual Authenticated Key Agreement Protocol for Mobile Communication, *The 4th International Conference on Wireless Communications*, Networking and Mobile Computing, 2008, pp.1-4.