Supersingular Isogeny Key Exchange

Paul Galloway

Abstract— **It is estimated that production-ready quantum computers will become a reality within the next 15 to 20 years.[1] If such devices are realized in the near future then many of the currently established public key encryption algorithms (specifically RSA, Diffie-Hellman, Elliptic Curve Diffie-Hellman, and Elliptic Curve DSA) will become insecure and will need to be replaced. It is prudent for us to begin considering such a scenario and to look into possible replacements.**

In this paper we review the post-quantum Key exchange scheme known as Supersingular Isogeny Diffie Hellman (SIDH). This type of key exchange provides forward secrecy and any attack by a quantum system still takes exponential time.[6] Additionally, like the established Elliptic Curve Diffie-Hellman system, SIDH provides similar key sizes and computationally efficient implementations when compared to established schemes.

Along the way we will mention some necessary mathematical constructs that will supplement our basic understanding of Elliptic Curves. Supersingular elliptic curves and isogenies between such curves will be explored. We will show how such constructs allows us to thwart analysis by a quantum system.[5] Finally, we will mention why such a key exchange scheme is superior to other post-quantum systems such as the McEliece system or NTRU.

I. Introduction

In a post-quantum cryptographic landscape many approaches will be insufficient. The widely used RSA system will be obsolete. Any mechanism reliant upon the Discrete Logarithm Problem will be vulnerable. As shown in [2] the Elliptic curve variant of the Discrete Logarithm problem is not immune to quantum analyis. A fundamentally different hard problem will be needed that maintains the need for an exponential time attack, even with a quantum computer. One such system involves Supersingular elliptic curves and isogenies between these curves. Using this scheme has many benefits. Firstly, it relies on many of the same primitives used in common ECC implementations, so existing systems can more easily be upgraded. Secondly, it is not subject to any known patents and thus remains free and open to the research community. What follows is an overview of necessary mathematical concepts and how they build on top of our basic understanding of elliptic curves. We will then get to the actual key exchange algorithm and discuss possible attacks on this system. We will conclude by briefly comparing it to other non-ECC, post-quantum cryptographic schemes.

II. Mathematical Background

Note: Some of the finer details are omitted for simplicity but can be found in [7].

Authors are with the Department of Computer Science, University of California, Santa Barbara, CA 93106. E-mail: {pgalloway}@cs.ucsb.edu

A Supersingular Elliptic Curve

We consider the Elliptic Curve E described by the equation $y^2 + cy = x^3 + ax + b$ and defined over F_q. Let p be the characteristic of F_q . An elliptic curve E defined over F_q is supersingular if p divides t, where t is the trace. If p does not divide t, then E is non-supersingular.[3]

An elliptic curve E/F_q over a finite field of characteristic p is said to be supersingular if $E[p] = 0$ (0 represents the point at infinity). In this case $E[p^n] = 0$ for all n. Otherwise, $E[p^n] = p^n$ for all n, and E is said to be ordinary.[5]

The motiviation for supersingular elliptic curves over ordinary elliptic curves is three fold[7].

1. Ordinary curves allow for a subexpontial quantum attack.
2. Ordinary curves are slow when used in such an isogeny-based context.
3. Isogenies over supersingular curves have already found success in the field of hash functions.

Isogenies

An isogeny is a non trivial, rational algebraic map $\phi : E_1 \rightarrow E_2$ between two elliptic curves such that $\phi(P + Q) = \phi(P) + \phi(Q)$ for all geometric points P, Q in E_1. This map is also a group homomorphism because the group operation (addition/multiplication) is preserved and maps back to the same group of points. The two elliptic curves that form such an isogeny must have the same number of points. There exists a polynomical time algorithm to count these points and therefore constructing an isogeny should take polynomial time on a classical computer.[5]

To identify an isomorphism between curves we use what is called the j-invariant. If two curves possess the same j-invariant, they should be isomorphic. For an elliptic curve given by the equation: $y^2 = x^3 + ax + b$ the j-invariant is given by:

$$j(E) = 1728 * 4a^3/(4a^3 + 27b^2)$$

Basic Examples

If E is an elliptic curve, the multiplication by $[m]$ is an isogeny. If $E : y^2 + cy = x^3 + ax + b$ is an elliptic curve defined over a finite field F_q of characteristic p, the Frobenius $E \rightarrow E(p), (x, y)^p \rightarrow (x^p, y^p)$ is an isogeny.

An isogeny $f : E_1 \rightarrow E_2$ transports the DLP problem from E_1 to E_2. This can be used to attack the Discrete Logarithm Problem on E_1 if there is a weak curve on its isogeny class (and an efficient way to compute an isogeny to it).

If E_1 and E_2 are two elliptic curves given by Weierstrass equations, a morphism of curve $f : E_1 \to E_2$ is of the form $f(x, y) = (R_1(x, y), R_2(x, y))$ where R_1 and R_2 are rational functions, whose degree in y is less than 2 (using the equation of the curve E_1). If f is an isogeny, $f(-P) = -f(P)$. If $char(k) > 3$ we can assume that E_1 and E_2 are given by reduced Weierstrass forms, this mean that R_1 depends only on x, and R_2 is y time a rational function depending only on x.

The set of isogenies of a supersingular elliptic curve, together with the composition operation form a non-abelian group and the added security of SIDH over non-supersingular isogeny approaches is dependent upon the resultant non-abelian structure. But the first layer of difficulty in attacking SIDH does rely upon the following abelian problem.

The Hidden Shift Problem

Like any cryptographic scheme SIDH relies on a fundamentally "hard" problem. In this case the hard problem is the Abelian Hidden Shift Problem[7]. This problem is stated as follows:

-Let A be a finite abelian group.
-Let S be a finite set.
-Let f: $A \to S$ and g: $A \to S$ be two injective functions.
-A b exists in A such that, for all x in A. $f(x) = g(xb)$
-The value that an attacker desires is b (the shift).

This can be easily re-formulated for elliptic curves.
Omitting some extra mathematical details we can say the following:

-We have two isogenous curves E and E'.
-We say that $f_0(a) = a * E$ and $f_1(a) = a * E'$.
-Then $b * E = E'$.
-And $f_1(a) = a * E' = a * b * E = f_0(ab)$.
-Solving the hidden shift problem on f_0, f_1 should yield b.

This relies on computing the group action for elliptic curves and the time and space complexity of finding this is $exp((lnN)^{1/2})$.

Velus formula

Velus formula plays a key role in the construction of isogenies in SIDH. For simplicity, assume the characteristic of K is not equal to 2 or 3. Let $E : y^2 + cy = x^3 + ax + b$ be an elliptic curve in short Weierstrass form, with l odd. Let F be a subgroup of E of order l. In [36], Velu showed how to explicitly find the rational function form of a normalized isogeny $\Phi : E \to E_0$ with kernel F. The formula is presented here. With $F' = F - \{\infty\}$, $P = (x_P, y_P) \in F'$, and sums taken over $Q \in F'$.

$$\Phi(P) = x_P + \sum(x_{(P+Q)} - x_Q), y_P + \sum(y_{(P+Q)} - y_Q)$$

III. Supersingular Isogeny Diffie Hellman Method

Description of the algorithm

The basic setup for SIDH is as follows (all parameters here are public and agreed upon by both participants):

1. A prime of the form $p := (w_A^{e_A}) * (w_B^{e_B}) * f \pm 1$.
(w's are small primes, e's are unrestricted exponents)
2. A supersingular elliptic curve E over F_{p^2}.
3. Fixed elliptic points P_A, Q_A, P_B, Q_B on E.
4. The order of P_A and Q_A is $(w_A)^{e_A}$.
The order of P_B and Q_B is $(w_B)^{e_B}$.

Using the common elliptic curve E above, the parties A and B each create an isogeny based off of this curve. A random point is chosen in what will be the kernel of their isogeny. The two points chosen by each party (their P and Q) will span this kernel. By using two points, this guarantees that the two parties will not create two isogenies that commute. Each party produces a random point in ther kernel of their isogeny that is a random linear combination of their two points P and Q. These random points are R_A and R_B in the steps below.

Key Exchange [10]
Public: prime number p, supersingular elliptic curve E/F_{p^2}, fixed elliptic points P_A, Q_A, P_B, Q_B on E
Alice: $R_A = m_A P_A + n_A Q_A$
$\phi_A : E \to E/\langle R_A \rangle$ Send $E/\langle R_A \rangle, \phi_A(P_B), \phi_A(Q_B)$ to Bob
Bob: $R_B = m_B P_B + n_B Q_B$
$\phi_B : E \to E/\langle R_B \rangle$ Send $E/\langle R_B \rangle, \phi_B(P_A), \phi_B(Q_A)$ to Alice

1: A generates two random integers $m_A, n_A < (w_A)^(e_A)$
2: A generates $R_A := m_A * (P_A) + n_A * (Q_A)$
3: A uses the point R_A to create an isogeny mapping $\Phi_A := E \to E_A$ and curve E_A isogenous to E
4: A applies ϕ to P_B and Q_B to form two points on $E_A := \phi_A(P_B)$ and $\phi_A(Q_B)$
*Repeat steps 1-4 but with A and B subcripts swapped

5a: A sends to $BE_A, \phi_A(P_B)$ and $\phi_A(Q_B)$
5b: B sends to $AE_B, \phi_B(P_A)$ and $\phi_B(Q_A)$
6a: A has $m_A, n_A, \phi_B(P_A)$, and $\phi_B(Q_A)$ and forms $S_{BA} := m_A(\phi_{B(P_A)}) + n_A(\phi_{B(Q_A)})$.
7a: A uses S_{BA} to create an isogeny mapping ψ_{BA}
8a: A uses ψ_{BA} to create an elliptic curve E_{BA} which is isogenous to E.
9a: A computes $K :=$ j-invariant(j_{BA}) of the curve E_{BA}
6b: B has $m_B, n_B, \phi_{A(P_B)}$, and $\phi_{A(Q_B)}$ and forms $S_{AB} := m_B(\phi_{A(P_B)}) + n_B(\phi_{A(Q_B)})$.
7b: B uses S_{AB} to create an isogeny mapping ψ_{AB}
8b: B uses ψ_{AB} to create an elliptic curve E_{AB} which is isogenous to E.
9b: B computes $K :=$j-invariant(j_{AB}) of the curve E_{AB}

The curves E_{AB} and E_{BA} will be guaranteed to both have the same j-invariant. A function of K is used as the shared key.

IV. Hardness and possible attacks

There is more than one possible attack against any isogeny-based cryptographic scheme. We summarize them in this section and discuss the efficiency of such attacks. Each of these attacks relies on computing the group action. To compute this group action and find $b * E$ we can do it either directly or indirectly.

Direct Approach

The direct approach works with b directly. In this case we try factoring b. This should take subexponetial time. However this requires computing a classical modular polynomial of level l which grows very large as l increases.

Indirect Approach

There is also an indirect way to compute $b * E$ much faster. Using index calculus we can find a factorization of $[b]$:

$$[b] = [p_1^{e_1} p_2^{e_2} p_3^{e_3} ... p_t^{e_t}]$$

Evaluating this should take subexponential time when evaluating the above expression one (small) prime at a time.

Supersingular Case

The previous two approaches apply to ordinary elliptic curves. Because the outcome of these approaches is a subexponential attack on the isogeny-based scheme, we need something better. In [8] we see that the graph of supersingular isogenies is an expander graph which is a Ramanunjan graph (this being the ideal condition for the graph). In order to attack this isogeny-based scheme we must start two random walks on the graph until a collision occurs. Because of the features of these types of graphs the probability of landing on any given node is uniform. A collision should occur after $O(q^{1/4})$ where q is the group order and therefore this is the difficulty of finding the specific isogeny on an isogeny graph. In order to make this problem harder SIDH uses F_{p^2} rather than F_p as shown in [9].

V. Comparison with alternatives

There are several different post-quantum cryptography systems worth noting. Such systems include lattice-based, hash-based, and code-based systems. Two key features of such systems that should be emphasized are key sizes required and whether or not such schemes provide what is called "Forward Secrecy". This feature means that the compromise of one message in one session cannot be used to compromise other messages sent in different sessions in which a different random key is used. The lattice based scheme Ring-LWE supports forward secrecy. Another well-known lattice-based scheme NTRU does not provide this feature. SIDH does provide forward secrecy, but also comes out on top when it comes to key size. With a public, private bit size of 3071 and 3072, respectively, it wins out over its closest competitor NTRU which has 6130 and 6743, respecively.

VI. Conclusions

There are many different proposals for cryptographic schemes that could one day replace existing schemes that rely upon the Discrete Logarithm or Integer Factorization Problems. Supersingular Isogeny Diffie Hellman (SIDH) is a front runner in this competition. Isogenies using supersingular curves already have found success in the construction of secure hash functions. Additionally, SIDH can be built on top of existing ECC primitives and when compared to other post-quantum schemes, SIDHs key sizes clearly set it apart (which is of course consistent with the lower key sizes of ECC schemes in general). Lastly, the fact that SIDH does not fall under any filed patents makes it very attractive when compared to pre-quantum schemes such as RSA. Several implementations of it already exist for both embedded[11] and non-embedded[12] applications For these reasons, SIDH deserves serious consideration for mainstream use in a post-quantum world.

References

[1] J. Utsler, "http://www.ibmsystemsmag.com/mainframe/trends/IBM-Research/" *IBM Systems Magazine*, Sep. 2013.

[2] S. Green, C. Kizilkale, "http://cs.ucsb.edu/ koc/ecc/project/2015Projects" *Attacking the ECDLP with Quantum Computing*, Nov. 2015.

[3] D. Hankerson, A. Menezes, S. Vanstone, "http://cs.ucsb.edu/ koc/ecc/docx/GuideEllipticCurveCryptography.pdf" *Guide to Elliptic Curve Cryptography*, Nov. 2015.

[4] "PATENTSCOPE," *World Intellectual Property Organization*, jun. 2014.

[5] D. Robert, "http://ecc2011.loria.fr/slides/summerschool-robert.pdf" *Isogenies and endomorphism rings of elliptic curves*, Sep. 2011.

[6] D. Jao, L. De Feo, "http://cacr.uwaterloo.ca/techreports/2011/cacr2011-32.pdf" *Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies*, June, 2011.

[7] D. Jao, "http://ecc2011.loria.fr/slides/jao.pdf" *Isogenies in a Quantum World*, Sep. 2011.

[8] D. Jao, "http://www.prism.uvsq.fr/ dfl/talks/yacc-27-09-12.pdf" *Isogeny Graphs in Cryptography*, Sep. 2011.

[9] S. Galbraith, "http://iml.univ-mrs.fr/ati/geocrypt2013/slides/galbraith.pdf" *Isogeny graphs, algorithms and applications*, Jan. 2013.

[10] Wikipedia, "https://en.wikipedia.org/" *Wikipedia - Supersingular Isogeny Key Exchange*, Nov. 2015.

[11] Azarderakhsh, Fishbein, Jao, "http://cacr.uwaterloo.ca/techreports/2014" *Efficient Implementations of A Quantum-Resistant Key-Exchange Protocol on Embedded systems*, June. 2014.

[12] L. De Feo, "https://github.com/defeo/ss-isogeny-software" *Quantum-Resistant Cryptosystems from Supersingular Elliptic Curve Isogenies*, Nov. 2015.