A NOTE ON KEY CONTROL IN CSIDH

ANTONIO SANSO

Abstract. In this short note we explore a particular behaviour of the CSIDH key exchange that leads to a very special form of (shared) key control via the use of the quadratic twists. This peculiarity contained in CSIDH with regard to quadratic twists was already noted in the original CSIDH work and used in several subsequent papers but we believe spelling out this in the form of an attack might be useful to the wider community.

1. Introduction

CSIDH is an isogeny based post quantum key exchange presented at Asiacrypt 2018 [1] based on an efficient commutative group action. The idea of using group actions based on isogenies finds its origins in the now well known 1997 paper by Couveignes [2]. Almost 10 years later Rostovtsev and Stolbunov rediscovered Couveignes's ideas [3]. A remark contained in the original CSIDH paper already shows a symmetry of the isogeny graph over $\mathbb{F}_p$ with respect to quadratic twisting. In this short note we are going to demonstrate how a malicious party in a key exchange can leverage this CSIDH peculiarity to force the shared secret to a specific value under the attacker's control. This will render CSIDH susceptible to *key control*. As we will discuss in the rest of the note this doesn't represent an issue for standard Diffie-Hellman key exchange but it might cause some trouble if CSIDH is employed in some unusual non-Diffie-Hellman protocols.

2. CSIDH and quadratic twist

CSIDH is an isogeny based post quantum key exchange that was introduced by Castryck, Lange, Martindale, Panny, and Renes [1] in 2018. CSIDH extends the idea of Couveignes [2] and Rostovtsev-Stolbunov [3] (CRS) by restricting the isogeny graph of supersingular elliptic curves and isogenies defined over $\mathbb{F}_p$. By employing supersingular curves all isogenies can be computed using Vélu's formulas [4]. CSIDH key exchange shares similarities with the original Diffie-Hellman construction. For a thorough explanation of the CSIDH protocol we refer to the original CSIDH paper [1]. We report below a short summary:

CSIDH's key exchange. Suppose Alice and Bob have key pairs $([a], E_1)$ and $([b], E_2)$. Upon receiving Bob's public key E_2, Alice verifies that the elliptic curve is in $Ell_p(\mathcal{O})$. She then applies the action of her own secret $[a]$ to E_2 to compute the curve $[a]E_2 = [a][b]E_0$. Bob proceeds analogously with his secret [b] and Alice's public key E_1 to compute the curve $[b]E_1 = [b][a]E_0$ (which is the same than Alice's shared value due to the commutativity of $cl(\mathcal{O})$).

Ethereum Foundation and Ruhr Universität Bochum

An important fact in CSIDH is the symmetry of the isogeny graph over $\mathbb{F}_p$ with respect to quadratic twisting [1, Remark 5]. The quadratic twist of an elliptic curve $E : y^2 = f(x)$ is $E^t : dy^2 = f(x)$ where $d \in \mathbb{F}_p$ is a non quadratic residue in $\mathbb{F}_p$. The quadratic twist can be efficiently computed in the CSIDH setting.

We report here this very important fact proved in [5, Lemma 5]:

Lemma 2.1. *For all $[a] \in \mathrm{cl}(\mathcal{O})$ and all $E \in Ell_p(\mathcal{O})$ we have $[a]^{-1}E = ([a]E^t)^t$*

Remark 1. Lemma 2.1 is specific to the isogeny commutative group action operating on supersingular elliptic curves (like CSIDH). The same doesn't apply to the ordinary curve case as in the Couveignes–Rostovtsev–Stolbunov key exchange [2, 3].

In [5], the above Lemma 2.1 is used as part of an algorithm to solve the key recovery problem in CSIDH when the full endomorphism ring of the target curve is known. In [6] the quadratic twist is used to compress the three-round oblivious transfer scheme to an optimal two rounds. A recent paper [7] defines a new notion of group actions with twists in order to derive a new Password-Authenticated Key Exchange schema. In the following sections we are going to show how Lemma 2.1 can be leveraged in order to gain some control over the shared key obtained as part of a CSIDH key exchange.

3. Key control in key agreement protocols

The term *key control* indicates the situation where, during a key exchange protocol, a malicious party forces the other party's shared secret to lie in a small key space chosen by the attacker. The first to mention the issue were Mitchel *et al.* in [8]. One known way to achieve this for classic Diffie-Helmann is the *small subgroup confinement attack* that was first defined in a paper by van Oorschot and Wiener [9] and attributed to Vanstone, Anderson and Vaudenay [10] who came up with a description of this notion. In a small subgroup confinement attack, an attacker (either a man-in-the-middle or a malicious client or server) provides a key exchange value y that lies in a subgroup of small order. Small subgroup confinement attacks are possible even when the server does not repeat exponents—the only requirement is that an implementation does not validate that received Diffie-Hellman key exchange values are in the correct subgroup.

We have the following definition of key agreement protocol [11]:

Definition 3.1. A **key agreement protocol** is a key establishment technique whereby a shared secret key is derived by two or more specified parties as a function of information contributed by, or associated with, each of these, such that no party can predetermine the resulting value.

A Diffie-Helmann key exchange can be seen as a special case of the more general key agreement primitive.

Consider a simplified Diffie-Hellman authenticated key exchange protocol (see Figure 1). The assumption here is that Alice's encrypting g^x to Bob's public key $\mathtt{ENC}_B(g^x)$ ensures that only Bob can learn g^x. Then Alice and Bob negotiate a shared secret $k = g^{xy} \bmod p$ using Diffie-Hellman key exchange. Bob's hashing the shared secret $\mathtt{H}(k)$ is used in Bob's handshake response to demonstrate knowledge of the computed shared key.

Now if an attacker has full control over the shared key it might represent an issue as we can see an in Figure 2. Mallory can send a generator g_i of a group of order

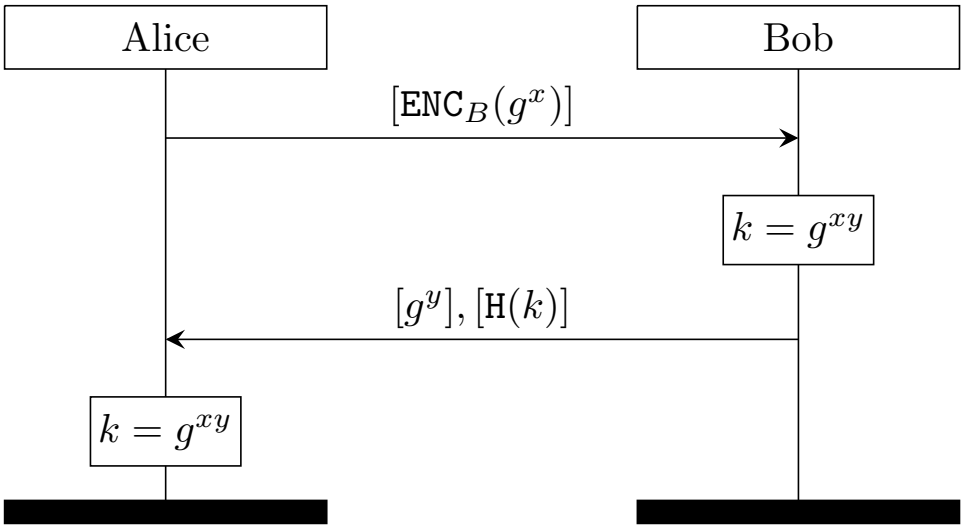

FIGURE 1. Simplified Diffie-Hellman authenticated key exchange protocol.

q_i (q_i is equal to one in the Figure 2's example) as her Diffie-Hellman key exchange value, and the shared secret will be an element of the subgroup of order q_i. Mallory can do the same with Bob. Mallory then has a good chance of guessing Alice's and Bob's shared secret in this invalid group. This was indeed the case for Tor's (older) TAP circuit handshake [12]. Also the Triple Handshakes attack on earlier versions of TLS [13] where Diffie-Hellman outputs were directly used for channel binding hence falling to this trap deserves a mention here.

Bhargavan and Delignat-Lavaud [14] describe "key synchronization" attacks against IKEv2 where a man-in-the-middle connects to both initiator and responder in different connections, uses a small subgroup confinement attack against both, and observes that there is a $1/q_i$ probability of the shared secrets being the same in both connections. Bhargavan and Leurent [15] describe several attacks that use this type of small subgroup confinement attack to obtain a transcript collision and break protocol authentication.

As we will see in the next paragraph in the case of CSIDH the *key control* the attacker will manage to pull is really special and pretty different from the one over the finite field or elliptic curve cases.

4. Key control on CSIDH

Now we are going to spell out a simple key control attack that affects CSIDH key exchange. This deviates substantially from the classical *small subgroup confinement attack*. The main differences respect the classical attack are:

- The small key space chosen by the attacker where the shared secret lies has always cardinality equal to one (i.e. it is a single element). This allows the attacker to predict exactly the value of the shared secret with probability one.

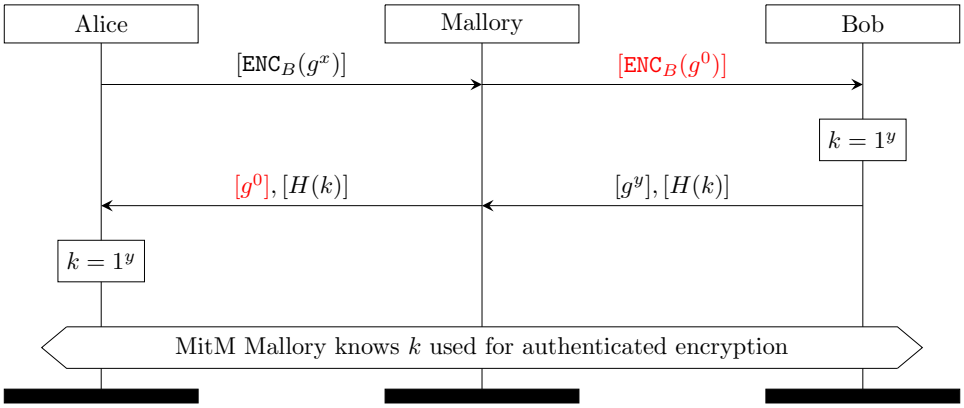

FIGURE 2. A man-in-the-middle can force both parties to agree on the same key.

- The key exchange value provided by the attacker is dynamically generated upon receiving the victim's public key.
- There is no possibility of input validation (with regard to key control) in plain CSIDH. In order to stop the attack a protocol change is needed.

We start by presenting a weak form of the attack where the shared secret derived after the CSIDH handshake will be the starting curve E_0. We will see that in this simple case the validation is possible hence preventing the attack will be possible.

Strawman attack. Upon receiving Alice's public key $E_1 = [a]E_0$, Eve computes its quadratic twist $E_1^t = [a]^{-1}E_0$ and provides it as key exchange value. The shared secret will then be $[a][a]^{-1}E_0 = E_0$ as predicted. This attack can be easily prevented. Indeed it is trivial for Alice to ensure that the received exchange value is not the quadratic twist of her contribution.

Next we present a modification of the previous strawman attack. In this case the attacker will still force the shared secret to be a specific single value under attacker control but the victim would not have the possibility to spot the attack if plain CSIDH is used (we will discuss later in the paper a modification to CSIDH contained in [7] that solves the issue but doubles the cost of the protocol). The attack though is still very simple:

Key control attack. Let assume Eve wants to have the shared secret to be $[b]E_0$. Upon receiving Alice's public key $E_1 = [a]E_0$, Eve computes its quadratic twist $E_1^t = [a]^{-1}E_0$ and then applies the action of her own secret $[b]$ to E_1^t. Then Eve provides $[b]E_1^t = [b][a]^{-1}E_0$ to Alice. The shared curve will then be $[a]([b][a]^{-1}E_0) = [b]E_0$ as expected. As mentioned before, differently than the strawman attack Alice doesn't have any chance to validate Eve's public key for plain CSIDH (this is supported by [6, Lemma 2.1]).

It is evident from the example above that Eve's public key has the intent to cancel out Alice's contribution to the shared key, namely $[a]$.

In order to prevent the attack described above altogether we could modify the standard CSIDH flow in the same way modelled in [7] where a new notion of group actions with twists is defined. In this setting the new shared key is computed as

$$\mathrm{KDF}([a]E_2, [a]E_2^t) = \mathrm{KDF}([ab]E_0, [a/b]E_0) = \mathrm{KDF}([b]E_1, [1/b]E_1)$$

where KDF is a standard key derivation function. The idea here is that Eve will not be able to cancel out both inputs of the KDF function at the same time. The cost increase due the protocol modification is quite significant though. Now both Alice and Bob need to double their work compared to CSIDH because they need to apply the action of their own secret twice.

5. Conclusions

In this short note we analyzed the *key control* in the context of the CSIDH setting. This is usually not a problem for plain key exchange nevertheless it might be harmful in more convoluted protocols. In section 4 we mentioned a costly modification to the CSIDH protocol that prevents the key control issue. Another way to have a group action based on isogeny where key control cannot be performed would be to bring back the key exchange in graphs of ordinary isogenies (CRS) and try to speed up its last incarnation [16].

Acknowledgments. We would like to thank Luca De Feo, Steven Galbraith, Gottfried Herold, Simon Masson and Christophe Petit for fruitful discussions and Luca De Feo for pointing out the key control attack's remediation and clarifying the presentation of the note.

References

[1] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: An efficient post-quantum commutative group action. In Thomas Peyrin and Steven D. Galbraith, editors, *Advances in Cryptology – ASIACRYPT 2018*, pages 395–427. Springer International Publishing, 2018.

[2] Jean-Marc Couveignes. Hard homogeneous spaces. Cryptology ePrint Archive, Report 2006/291, 2006.

[3] Alexander Rostovtsev and Anton Stolbunov. Public-key cryptosystem based on isogenies. Cryptology ePrint Archive, Report 2006/145, April 2006.

[4] Jacques Vélu. Isogénies entre courbes elliptiques. *Comptes Rendus de l'Académie des Sciences de Paris*, 273:238–241, 1971.

[5] Wouter Castryck, Lorenz Panny, and Frederik Vercauteren. Rational isogenies from irrational endomorphisms. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020*, pages 523–548, Cham, 2020. Springer International Publishing.

[6] Yi-Fu Lai, Steven D. Galbraith, and Cyprien Delpech de Saint Guilhem. Compact, efficient and uc-secure isogeny-based oblivious transfer. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021*, pages 213–241, Cham, 2021. Springer International Publishing.

[7] Michel Abdalla, Thorsten Eisenhofer, Eike Kiltz, Sabrina Kunzweiler, and Doreen Riepel. Password-authenticated key exchange from group actions. Cryptology ePrint Archive, Paper 2022/770, 2022. `https://eprint.iacr.org/2022/770`.

[8] Chris J. Mitchell, Mj Ward, and Paul Wilson. Key control in key agreement protocols. *Electronics Letters*, 34:980–981, 1998.

[9] Paul C Van Oorschot and Michael J Wiener. On Diffie-Hellman key agreement with short exponents. In *EUROCRYPT*, 1996.

[10] Ross Anderson and Serge Vaudenay. Minding your p's and q's. In *ASIACRYPT*, 1996.

[11] Giuseppe Ateniese, Michael Steiner, and Gene Tsudik. Authenticated group key agreement and friends. In *Proceedings of the 5th ACM Conference on Computer and Communications Security*, CCS '98, pages 17–26, New York, NY, USA, 1998. Association for Computing Machinery.

[12] Roger Dingledine. Tor security advisory: Dh handshake flaw. https://lists.torproject.org/pipermail/tor-announce/2005-August/000009.html.

[13] Karthikeyan Bhargavan, Antoine Delignat Lavaud, Cédric Fournet, Alfredo Pironti, and Pierre Yves Strub. Triple handshakes and cookie cutters: Breaking and fixing authentication over tls. In *2014 IEEE Symposium on Security and Privacy*, pages 98–113, 2014.

[14] Karthikeyan Bhargavan, Antoine Delignat-Lavaud, and Alfredo Pironti. Verified contributive channel bindings for compound authentication. In *Network and Distributed System Security Symposium*, 2015.

[15] Karthikeyan Bhargavan and Gaetan Leurent. Transcript collision attacks: Breaking authentication in TLS, IKE, and SSH. In *Network and Distributed System Security Symposium*, 2016.

[16] Luca De Feo, Jean Kieffer, and Benjamin Smith. Towards practical key exchange from ordinary isogeny graphs. In Thomas Peyrin and Steven D. Galbraith, editors, *Advances in Cryptology – ASIACRYPT 2018*, pages 365–394. Springer International Publishing, 2018.