Isogeny Secrets can be Traded

David Urbanik

University of Waterloo, Waterloo ON, Canada,
dburbani@uwaterloo.ca

Abstract. We consider a situation in which two mutually distrusting parties, each possessing a secret piece of information, wish to exchange these secrets while communicating over a secure channel, in effect "trading" them. Each is afraid of counterparty risk: Alice fears that as soon as she sends her secret to Bob he will cease communication without sending his secret in return, and likewise for the reverse case. In the situation where Alice and Bob's secrets are protected by isogenies, we propose a system in which Alice and Bob may fairly exchange their secrets without counterparty risk, and without a trusted third party. We then discuss potential applications.

Keywords: isogeny, secret, trading, escrow

1 Introduction

Suppose Alice and Bob are each in possession of a piece of secret data. The data could be encrypted information, such as encrypted digital goods; a private key securing a digital token, such as a cryptocurrency; or a secret code or key phrase for an escrow system, such as the one we will describe in Section 3. Alice and Bob are in possession of a secure digital communication channel and wish to exchange their secret data, in effect performing a "trade". The problem is that neither party can assume the other is trustworthy: if Alice sends her secret data to Bob, he may simply cease communication and not hold up his end of the bargain. Alice and Bob could enlist a trusted third party to help carry out this transaction, but this simply changes the problem to one of finding a sufficiently trustworthy intermediary. We consider the question of whether it is possible for Alice and Bob to carry out this exchange without a trustworthy third party.

There is a naïve argument which suggests this isn't possible. Any such protocol, one argues, produces a series of messages between Alice and Bob exchanged over the secure digital channel. Since the messages arrive in a definite order, there is some first message from which Alice (say) may recover Bob's secret. If Alice receives this message before sending her secret to Bob, she can simply neglect to complete her half of the protocol. If Alice divulges the information of her secret to Bob before he reveals his secret to her, then Bob can behave similarly. Since the messages are ordered, and the first party to divulge their secret has no guarantee the other will do likewise, a protocol which guarantees that neither party misbehaves cannot exist.

The issue with the naïve argument is with the claim that there is some first message from which Alice has sufficient information to recover Bob's secret. In fact, in a typical setting with a public-private key pair, each party always has enough information to determine the secret (the private key) from publicly available information (the public key) but simply lacks the computational means to be able to do so. Therefore, Alice and Bob's exchange of information can be viewed as a process through which Alice and Bob provide each other with additional information which assists the other party in obtaining the secret information efficiently from the data available to them. The idea is then to develop a protocol where Alice and Bob can take turns providing information to the other party which lowers the computational difficulty of obtaining their respective secrets from the revealed data, in effect exchanging the secrets “bit by bit”. If either abandons the protocol at any point, both parties will be left with an equivalent (up to a small constant factor) “amount of information” about the other party’s secret, or alternatively, will have their secrets secured from the other party by problems of equal difficulty.

The above protocol is still subject to counterparty risk in the following way: if the data Alice sends Bob to assist his recovery of her secret is faked, then Bob cannot detect this forgery until Alice and Bob have completed enough rounds of the protocol so that they would both be able to recover each other’s secret had both parties been behaving honestly. But in such a situation, an honest Bob would lose his secret to a devious Alice, while the devious Alice would have revealed nothing but faked data. Alice and Bob therefore need a mechanism to not just exchange partial information about their secrets, but also to prove that the revealed partial information is accurate without divulging their secrets further. We show in the next section that such a secret exchange can be achieved with isogeny-based techniques.

2 Implementation Using Isogenies

In the interest of making this paper accessible to a wider audience, we review the basics of isogeny-based cryptography. We assume basic knowledge of elliptic curves, abelian groups, and the elliptic curve group law.

Let E be an elliptic curve, and suppose that G is a finite subset of points of E which form a subgroup of E under the group addition. It is then possible to construct another elliptic curve, which we label E/G, and a surjective map $\phi : E \to E/G$ which is both a group homomorphism and an algebraic map; that is, if E and E/G are given by Weierstrass equations in the coordinates (x, y) and (x', y') respectively, then the map ϕ takes the form

$$(x', y') = \phi(x, y) = \left(\frac{\psi_{x'}(x, y)}{\eta_{x'}(x, y)}, \frac{\psi_{y'}(x, y)}{\eta_{y'}(x, y)} \right)$$

for polynomials $\psi_{x'}, \eta_{x'}, \psi_{y'}$, and $\eta_{y'}$ in x and y. The kernel of the map ϕ is exactly the group G, and the *degree* of the map ϕ, which is the same as the degree of the rational functions which define it, is the size $|G|$ of the subgroup G. The map ϕ is called an isogeny,[1] and is the main object of interest in isogeny-based cryptography.

It is known how to compute an appropriate elliptic curve E/G and map ϕ given the subgroup G. For the case where G is cyclic, which is the primary case of interest, there are explicit formulas given by Vélu[5] which accomplish this. It is then a fact that the pair $(E/G, \phi)$ satisfy a kind of “first isomorphism” property: if (E', ϕ') is another pair such that $\phi' : E \to E'$ is an isogeny with kernel G, then there is an isomorphism $\alpha : E' \to E/G$ of elliptic curves such that $\phi = \alpha \circ \phi'$. This property plays a crucial role in isogeny-based protocols, since they often require the parties involved to agree on isomorphic curves by constructing two different isogenies with the same kernels.

[1] More precisely, this is a *separable* isogeny, but we won’t be concerned with the inseparable case here.

Given two elliptic curves E and E', it is typically very difficult to construct an isogeny $\phi : E \to E'$ between them. Moreover, if some additional information such as the degree of the isogeny ϕ is specified, then one can often ensure heuristically that there is at most one appropriate isogeny ϕ. Thus, isogenies naturally lend themselves to the role of public-private key pairs in cryptography: the public key is the pair (E, E'), possibly with some additional information depending on the protocol of interest, and the private key is the isogeny ϕ (or information that allows one to compute it). The person possessing the private key can then use it to either evaluate the isogeny or compute other isogenies determined by ϕ, and from this a number of cryptographic primitives can be constructed (c.f. [3][6][2]).

However, even if we know the subgroup G, it is not always easy to compute the map $\phi : E \to E/G$. The issue is that in order to make finding ϕ (which is the same as finding G) difficult for the attacker, the degree of ϕ has to be exponentially sized so there is an exponential search space (i.e., exponentially many possible kernels G). But if the degree of ϕ is exponentially large, then so are the rational functions which define it, which can make evaluating and working with ϕ computationally difficult.

The problem may be solved in the following manner. Choose a prime $p = n \pm 1$, where n is a positive composite integer with many small factors; say that, in particular, $\ell^e | n$ where ℓ is a small prime and e is an exponent such that ℓ^e is exponential in size. Denote by $\mathbb{F}_{p^2}$ a finite field of order p^2. Then it is possible find a family of elliptic curves, the *supersingular* elliptic curves, such that every isomorphism class of supersingular curve has a representative where the $\mathbb{F}_{p^2}$-points satisfy $E(\mathbb{F}_{p^2}) \cong (\mathbb{Z}/n\mathbb{Z}) \times (\mathbb{Z}/n\mathbb{Z})$. In particular, this means that there are exponentially many subgroups of order ℓ^e defined over the field $\mathbb{F}_{p^2}$. Then if G is such a cyclic subgroup generated by P, we have a chain of subgroups

$$\langle \mathcal{O}_E \rangle \subset \langle [\ell^{e-1}]P \rangle \subset \langle [\ell^{e-2}]P \rangle \subset \cdots \subset \langle [\ell]P \rangle \subset \langle P \rangle = G,$$

where $\mathcal{O}_E$ is the point at infinity on E. What this then means is that we may construct the map $\phi : E \to E/G$ of degree ℓ^e as a series of isogenies $\phi_i : E_{i-1} \to E_i$ of degree ℓ, where we set the curve $E_0 = E$, the curve $E_e = E/G$, and the kernel of ϕ_i to be the image of the group $\langle [\ell^{e-i}]P \rangle$ under the map $\phi_{i-1} \circ \cdots \circ \phi_1$. Since each map ϕ_i has small degree ℓ, the overall isogeny ϕ can be efficiently represented in this way.

Using the protocols described in [3] or the signature scheme described in [6], an isogeny $\phi : E \to E/G$ can secure anything ranging from the secret key for a symmetric cipher (using an SIDH-based key agreement), encrypted data (using SIDH encryption), or anything protected by signatures (such as a digital wallet). To implement the trading mechanism described in the previous section, therefore, it suffices to show how one can reveal ϕ "one step at a time" in a way where each step is verifiable by a non-interactive proof. The obvious answer is to reveal the maps ϕ_1, ϕ_2, etc. in order. Note that if $\ell = 2$, then this quite literally reveals one bit of information, since there are exactly two choices for the isogeny ϕ_i for $i \geq 2$. By the obvious symmetry of the problem, it suffices to explain how we may reveal $\phi_1 : E \to E_1$ and prove that there exists an isogeny $\phi' : E_1 \to E/G$ of degree ℓ^{e-1}.

The idea is simply to use the zero-knowledge protocol described in the original paper of Jao and De Feo. Jao and De Feo give an identification scheme where to demonstrate knowledge of an isogeny $\eta : E \to E/G$ of degree $\ell_1^{e_1}$ one computes commuting diagrams of the form

$$\begin{tikzcd} E \arrow[r, "\eta"] \arrow[d, "\psi"'] & E/G \arrow[d, "\psi'"] \\ E/H \arrow[r, "\eta'"] & E/\langle G, H\rangle \end{tikzcd},$$

where H is a subgroup of order $\ell_2^{e_2}$ (ℓ_2 a small prime relatively prime to ℓ_1, and e_2 a large exponent), ψ and ψ' are isogenies of degree $\ell_2^{e_2}$, and η' is an isogeny of degree $\ell_1^{e_1}$ related to η. The prover then commits the curves E/H and $E/\langle G, H\rangle$, and alternately reveals either the maps ψ and ψ' or the map η' depending on the bit sent by the challenger. Applying a Fiat-Shamir transform, one can turn this interactive proof into a non-interactive one. In particular, if we take $\eta = \phi'$, we may demonstrate the correctness of the revealed map ϕ_1 by proving that the isogeny ϕ' satisfying $\phi = \phi' \circ \phi_1$ exists. For more details, see the paper of Jao and De Feo, and also the discussion of applying the Fiat-Shamir transform to this protocol in the paper [6].

3 Blockchain-based Escrow

One natural issue that arises with exchanging digital goods using such a system is deciding when precisely two parties can be assured that the goods are secured by an isogeny. For instance, if the digital goods take the form of encrypted data, the party receiving the key to decrypt the data must somehow be assured in advance that the decrypted data is the in fact the digital good they are interested in. One way this can happen is if the digital goods themselves are part of some larger system that routinely uses cryptographic techniques to secure those goods; a natural candidate for this sort of phenomenon is blockchains, where digital currencies are secured using cryptographic signatures.

Unfortunately, all proposed isogeny-based signature schemes to date have various efficiency drawbacks, which makes their widespread use unlikely (especially in blockchain-based systems, which already have their own efficiency concerns). For this reason, we sketch a blockchain-based escrow system which applies these techniques.

Suppose that Alice and Bob wish to perform an exchange, and that Alice wishes to pay Bob using a digital currency implemented via a blockchain. We suppose that Alice and Bob have well-established identities, and so are able to both authenticate each other's messages, sign messages, and also to commit[2] messages to the blockchain in an authenticated manner.

The blockchain also supports an escrow feature, which works as follows:

(1) To put money in escrow, Alice commits a message consisting of:
   (i) the quantity Q of digital tokens she intends to commit,
   (ii) the intended recipient's wallet (i.e., Bob's wallet address),
   (iii) a "return" address (i.e., Alice's wallet address),
   (iv) an expiry date,
   (v) a public key K_A corresponding to some tradable private secret P_A,
   (vi) and an integer k.
(2) When Alice commits her message, Q digital tokens are removed from Alice's wallet and enter escrow.

[2] We use "commit" as a synonym for "broadcast" here.

(3) If at any time before the expiry date the private key P_A is committed to the blockchain (as can be verified using K_A), then the Q digital tokens leave escrow and are transferred to Bob's wallet and the transaction is completed. Note that there is no requirement that the private key remain secret after this point; it is simply a temporary tool used to facilitate the transaction.

(4) Prior to either the expiry date or the completion of the transaction, Alice and Bob may both "dispute" the transaction. If Bob files a dispute against Alice, then he submits a signed message from Alice in which Alice asserts that Bob honestly carried out the secret exchange protocol up until stage ℓ. Alice may due likewise. Note that Bob and Alice will obtain such signed statements from each other in the course of the modified exchange protocol, described below.

(5) If the expiry date is reached, then either the transaction has been disputed or it hasn't. If it hasn't been disputed, then the money is returned to Alice. If it has been disputed, then one of two things can happen:

  (i) If the maximum value of ℓ over all signed statements submitted by Alice and Bob is greater than or equal to k, then the expiry date is invalidated and the money either remains in escrow until P_A is committed (in which case it is released to Bob), or remains in escrow indefinitely.

  (ii) If the maximum value of ℓ over all signed statements submitted by Alice and Bob is less than k, then the money is returned to Alice and the transaction is terminated.

Now suppose that Alice and Bob wish to perform an exchange in which Bob's secret is a private key P_B corresponding to a public key K_B, and Alice is willing to pay him quantity Q in digital currency for his secret. Alice then generates a private-public key pair (P_A, K_A) where K_A will be the key Alice will commit her escrow message with. Alice and Bob agree to carry out the exchange in n stages, using a series of messages $M_A^1, M_B^1, M_A^2, M_B^2, \ldots, M_A^n, M_B^n$, where the message M_A^i will be the i'th message Alice sends to Bob, and the message M_B^i will be the i'th message that Bob sends to Alice. The message M_A^i is of the form $H_A^i|N_A^i|V_A^i$, where

(i) H_A^i is the i'th "hint" that Alice gives Bob about her private key (so the isogeny ϕ_i in the isogeny protocol described in Section 2),

(ii) N_A^i is a non-interactive zero-knowledge proof that H_A^i is correct,

(iii) and V_A^i is a signed statement from Alice that Bob has correctly carried out the protocol up to at least stage $i-1$.

Bob's messages M_B^i are analogous.

Alice and Bob then agree on an integer k, where k is the smallest integer such that the expected cost to Alice of brute-forcing the key P_B (respectively, the cost to Bob of brute-forcing the key P_A) using the hints $H_B^1, \ldots, H_B^k$ (respectively, the hints $H_A^1, \ldots, H_A^k$) is less than the value of Bob's secret to Alice (respectively, Alice's secret to Bob). Since Alice and Bob are using keys of the same type, and Alice and Bob's goods should be of approximately equal value, they should be able to agree on such an integer. Alice and Bob agree on an expiry date, which is chosen so that the average time required to brute force either P_A or P_B using k hints is significantly longer than the difference between the expiry date and the current time. Alice then makes her escrow commitment, and Alice and Bob exchange the messages $M_A^1, M_B^1, M_A^2, M_B^2, \ldots, M_A^n, M_B^n$ in order. If the protocol is completed successfully, then both know each other's secrets, and Bob recovers quantity Q of digital currency by committing P_A to the escrow system. If at any point Alice (respectively Bob) fails to carry out her (respectively his) obligations under the protocol, either by submitting an invalid message or becoming unresponsive, then Alice (respectively Bob) may file a dispute using the last signed statement V_B^j (respectively V_A^j) received. The following outcomes of this protocol are possible:

(1) The protocol completes successfully, in which case Alice and Bob have performed their exchange to the satisfaction of both parties.

(2) Alice fails to live up to her obligations, either by sending an invalid message or becoming unresponsive, in which case Bob files a dispute using the last signed statement V_A^j he received from Alice. There are then two cases:

   (i) If $j < k$, the money will be returned to Alice after the expiry date and brute-forcing Bob's private key will be prohibitively expensive, so neither party will obtain each other's secret.

   (ii) If $j \geq k$ then the expiry date will be nullified, and Bob can brute-force Alice's key and commit it to the escrow system. Alice (if she chooses) can also brute-force Bob's key. Both then obtain each other's digital goods at the mutual cost associated with the brute-force computation. In this case, both Alice and Bob gain equal value in goods and incur equal expected costs.

(3) If Bob fails to live up to his obligations, the analogous scenario occurs.

(4) If one party dishonestly commits a signed statement that misrepresents the stage to which the protocol was carried out properly, the other party can submit a more recent signed statement and proceed as in case (2) or (3).

In all of the above cases Alice and Bob (in their own estimation) either receive goods of equal value and/or incur (approximately) equal costs, and so the protocol is fair.

One might imagine that Bob could try to obtain an advantage from the fact that he goes second in the message-exchange stage of the protocol, and thus if he fails to live up to his obligations at stage ℓ, he will have one additional hint to work with when brute-forcing Alice's key. If Bob wishes to implement such a strategy, Bob must carry out his obligations up until at least stage k, since otherwise if he only does so up until stage $\ell < k$, Alice can submit a signed statement which, when the expiry date arrives, will ensure that the money is returned to her before Bob can brute-force Alice's key, and Bob won't have a signed statement of his own which can ensure the money remains in escrow. But if Bob carries out his obligations to stage k, then Alice will have enough hints such that the cost of brute-forcing Bob's secret will be less than the value of his secret to her, and so will brute-force Bob secret if necessary. Hence Bob will still lose his secret if he takes such an approach, and will incur additional costs in brute-forcing Alice's key, so there is no advantage to Bob over behaving honestly.

The protocol described has the downside that the concrete cost associated with brute-forcing a key can be wildly unpredictable, and that even when assuming each additional hint provides exactly one bit of information, can vary by a factor of two with each additional hint. This means that the protocol is best suited to applications where the value of the goods being exchanged is small and the parties involved can be assumed to be typically reliable; that is, the dispute mechanism may effectively deter scammers, but may be an unreasonable hassle if used on a regular basis. We leave further analysis and improvements to future work.

4 Conclusion

To the author's knowledge, the ideas presented in this paper represent the first proposed mechanism for a fair exchange of digital goods without a trusted third party. Indeed, there is prior work which argues that such protocols can't even exist[4]. Some solutions to similar problems, like Zero-Knowledge Contingent Payments or key escrow, do exist, but these proposals depend on either a third party or have various implementation drawbacks[1]. Obviously, more work is needed to examine to what extent the protocols proposed here still suffer from counterparty risk and what the potential applications are. However, we believe that with the increasing prominence of digital goods and digital currencies in terms of both cryptographic interest and use in the real-world marketplace, such ideas are worthy of further investigation.

References

1. Matteo Campanelli, Rosario Gennaro, Steven Goldfeder, and Luca Nizzardo. Zero-knowledge contingent payments revisited: Attacks and payments for services. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, pages 229–243, New York, NY, USA, 2017. ACM.
2. Denis X. Charles, Kristin E. Lauter, and Eyal Z. Goren. Cryptographic hash functions from expander graphs. *Journal of Cryptology*, 22(1):93–113, Jan 2009.
3. Luca De Feo, David Jao, and Jérôme Plût. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *J. Math. Cryptol.*, 8(3):209–247, 2014.
4. Henning Pagina and Felix C. Gartner. On the impossibility of fair exchange without a trusted third party. Technical report, 1999.
5. Jacques Vélu. Isogénies entre courbes elliptiques. *C. R. Acad. Sci. Paris Sér. A-B*, 273:A238–A241, 1971.
6. Youngho Yoo, Reza Azarderakhsh, Amir Jalali, David Jao, and Vladimir Soukharev. A post-quantum digital signature scheme based on supersingular isogenies. Financial Cryptography 2017 (to appear), 2017.