CatíoCrypto - Introducción a isogenias

Valerie Gilchrist

septiembre 2025

En lo que sigue damos una introducción breve al tema de criptografía vasada en isogenias. Varios de los ejemplos y ejercicios incluidos en estas notas necesitan el uso de SageMath.

1. Isogenias

Definición 1. *Una* curva elíptica *E sobre un cuerpo k es un subconjunto de puntos $(x, y) \in k^2$ junto con el punto infinito O donde los puntos satisfacen una relación algebraica*

$$f(x, y) = y^2 + a_1xy + a_3y - x^3 - a_2x^2 - a_4x - a_6,$$

con $a_i \in k, i = 1, \ldots 6$.

Isogenias son objetos geométricos que describen relaciones entre curvas elípticas.

Definición 2. *Sean E_0, E_1 dos curvas elípticas. Una* isogenia *de E_0 a E_1 es un mapa racional*

$$\phi : E_0 \to E_1$$

donde $\phi(O) = (O)$.

Se dice que dos curvas E_0, E_1 son isógenas *si existe una isogenia de E_0 a E_1.*

Toma por ejemplo multiplicación por un entero n.

$$\phi : P \mapsto [n]P$$

Los puntos sobre una curva elíptica forman un grupo y entonces son cerrados bajo multiplicación por un entero. Esto quiere decir que ϕ manda los puntos de una curva a la misma curva,

$$\phi : E \to E.$$

Este tipo de isogenia se llama un *endomorfismo*.

$$E_0 \xrightarrow{\varphi_1} E_1 \xrightarrow{\varphi_2} \cdots \xrightarrow{\varphi_k} E_k$$

Figura 1: Cadena de isogenias.

```
# multiplicacion por un entero
E = EllipticCurve(GF(7), [0,0,0,1,0]) # y^2 = x^3 + x
P = E((0,0))
nP = GF(7).random_element()*P
nP in E
```

Un ejemplo de una isogenia que no sea un endomorfismo puede ser

$$\varphi : E_0 \to E_1,$$

$$(x, y) \mapsto (\frac{x^2+1}{x}, \frac{x^2y - y}{x^2}).$$

En general, es difícil construir isogenias usando expresiones racionales. Por eso es muy común construirlas usando su *kernel*, denotado $\ker(\varphi)$. El kernel de $\varphi : E_0 \to E_1$ es el grupo de puntos sobre E_0 que φ manda a O en E_1. El kernel es suficiente para definir la isogenia usando las formulas de Vélu [9]. En la mayoría de casos, el tamaño del kernel se llama el *grado* de la isogenia, denotado por $\deg(\varphi)$.

En este ejemplo de φ, la isogenia puede ser construida usando un punto P con orden 2. Esto quiere decir que $[2]P = O$. Entonces $\ker(\varphi) = \{O, P\}$ y $\deg(\varphi) = |\ker(\varphi)| = 2$.

```
# isogenia de grado 2
E0 = EllipticCurve(GF(7), [0,0,0,1,0])
P = E0((0,0))
P.order()

phi = EllipticCurveIsogeny(E0, P);
phi
phi.rational_maps()
phi.degree()
```

Aquí, pudimos construir una isogenia muy fácilmente, mediante la selección de un punto P, y definiendo $\ker(\varphi) := \langle P \rangle$. Esta construcción también nos da información sobre la curva E_1 por que $E_1 = E_0/\ker(\varphi)$. Para puntos de bajo orden, este método funciona bien. En el caso de isogenias de grados más altos, a veces es posible construirlas usando varias isogenias más chicas. Por ejemplo, si queremos construir una isogenia de grado ℓ^k, donde ℓ es un primo pequeño, lo podemos hacer con k isogenias de grado ℓ cada uno. En la Figura 1, la composición de las isogenias $\varphi_1 \circ \varphi_2 \circ \cdots \circ \varphi_k$ permite construir una isogenia de E_0 a E_k de grado $\prod_{i=1}^{k} \deg(\varphi_i)$.

Hasta ahora hemos visto cómo construir isogenias de grados con factores pequeños. Pero este método no nos da control sobre la curva final, E_k. En general, dadas las curvas E_0, E_k, es extremamente difícil construir una isogenia $\phi : E_0 \to E_k$. Esta es la razón por la cual se puede construir criptografía

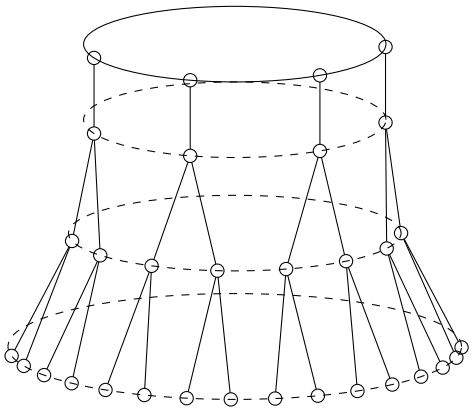

Figura 2: Volcán de 2-isogenias.

usando isogenias. La idea principal es que alguien puede seleccionar una isogenia $\phi : E_0 \to E_k$ como su llave privada, y publicar la curvas E_0 y E_k.

1.1. Grafos de isogenias

Considera un grafo en que los vértices son curvas elípticas y las aristas son isogenias de un grado particular, ℓ. Eso quiere decir que dos vértices están conectados si y sólo si las curvas correspondiendo a los vértices son isógenas de grado ℓ.

Ejercicio 1.1. *Según el teorema de Tate, dos curvas son isógenas si y solo si tienen el mismo número de puntos.*

Usa Sage para verificar el teorema de Tate.

a. Selecciona una curva elíptica.

b. Construye una isogenia.

c. Cuenta el número de puntos sobre el dominio y codominio de la isogenia. Asegurate que sea sobre el mismo cuerpo finito.

Considera el grafo de isogenias sobre $\mathbb{F}_{p^2}$. Una de dos cosas puede ocurrir. La primera opción es que hayan uno o varios subconjuntos no conectados. En este caso, cada subconjunto tiene un ciclo en que cada vértice es la raíz de un árbol. El ejemplo de esta estructura en la figura 2 demuestra visualmente por qué se llama un *volcán*. Las curvas en este grafo se llaman *ordinarias*.

La segunda opción es mucho menos común. En este caso el grafo es finito, conectado, y $(\ell + 1)$-regular. Las curvas en este grafo se llaman *supersingulares*. Este grafo tiene propriedades de mezcla muy buenas. Por eso es muy común usar curvas supersingulares en protocolos criptográficas.

```sage
# para verificar si una curva es supersingular
E = EllipticCurve(GF(7), [0,0,0,1,0]) # y^2 = x^3 + x
E.is_supersingular()
E.is_ordinary()
```

Otra razón por la cual curvas supersingulares son usadas mucho en criptografía es porque el número de puntos sobre $\mathbb{F}_p$ en estas curvas es fácil de calcular: $|E(\mathbb{F}_p)| = p+1$. Este control sobre el número de puntos puede ser atractivo para asegurar que puntos de bajos ordenes existen sobre $\mathbb{F}_p$.

```sage
# cuantos puntos hay sobre E en GF(p)
p = random_prime(2^5)
p
E = EllipticCurve(GF(p), [0,0,0,1,0])
E.is_supersingular()
E.cardinality() == p+1
```

1.2. SIDH

Uno de los protocolos de llave publica más usado hoy en día es el protocolo *Diffie-Hellman*. Este protocolo fue inventado en el año 1976, y sigue siendo usado para aplicaciones como TLS, Bitcoin, WhatsApp, Signal, Facebook Messenger, y Skype.

Protocolo 1 (Diffie-Hellman)**.** *Selecciona un grupo conmutativo G, y un generador del grupo, g.*

Alicia y Beto comienzan por escoger enteros secretos $a, b \in \mathbb{Z}$. Luego calculan y publican sus llaves públicas g^a, g^b respectivamente. Con estas llaves públicas, y sus enteros secretos, los dos pueden calcular el objeto secreto $(g^a)^b = (g^b)^a$.

Uno de los primeros protocolos que se volvieron populares usando isogenias fue *Supersingular Isogeny Diffie-Hellman* (SIDH) [4]. La idea básica de este esquema era la de usar isogenias para construir un protocolo similar al Diffie-Hellman. Dado que en este escenario, la composición de isogenias conmuta, Alicia y Beto pueden escoger un punto cada uno de una curva publica $A, B \in E_0(\mathbb{F}_q)$. Estos puntos pueden entonces ser usados para construir isogenias φ_A, φ_B donde $\ker(\varphi_A) = \langle A \rangle$ y $\ker(\varphi_B) = \langle B \rangle$. El codominio de estas isogenias sirven como llaves públicas, mientras que las isogenias se usan como llaves secretas. Después de compartir las llaves públicas, Alicia y Beto utilizan sus llaves secretas para calcular la curva final. La Figura 3 describe esta idea.

El problema es que no es claro como usar una isogenia $\varphi_A : E_0 \to E_A$ para construir una isogenia "φ_A" : $E_B \to E_{AB}$. El punto A es sobre E_0, no E_B. Sin embargo, la isogenia que nos gustaría calcular tiene kernel generado por $\varphi_B(A)$. ¿Pero cómo podría Alicia calcular este punto sin saber φ_B?

La solución propuesta en [4] fue la de usar bases del subgrupo de torsión. Sean P_A, Q_A don puntos tales que $E_0[2^k] = \langle P_A, Q_A \rangle$. Entonces el punto secreto A se puede escribir como $A = P_A + aQ_A$ donde a es un entero. Similarmente, si P_B, Q_B generan $E_0[3^e] = \langle P_B, Q_B \rangle$, entonces se puede escribir $B = P_B + bQ_B$

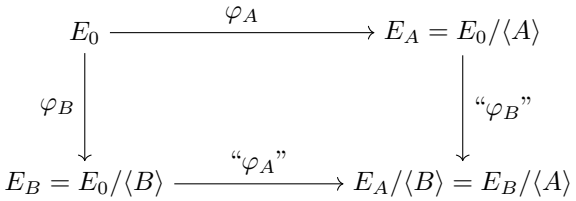

Figura 3: Un cuadrado de SIDH.

donde b es un entero. Así que se puede publicar los bases $\{P_A, Q_A\}$ y $\{P_B, Q_B\}$, y las llaves secretas de Alicia y Beto se vuelven a y b respectivamente.

Ahora, la isogenia que queríamos calcular "φ_A" tiene kernel generado por

$$\varphi_B(A) = \varphi_B(P_A + aQ_A) = \varphi_B(P_A) + a \cdot \varphi_B(Q_A).$$

Asi que si Beto publica los puntos $\varphi_B(P_A), \varphi_B(Q_A)$, Alicia puede calcular "φ_A" usando su entero secreto a. De manera similar, Beto puede calcular "φ_B", y entonces los dos calculan la curva E_{AB}.

En 2022, una serie de ataques [1,5,7] fueron publicados demostrando contundentemente cómo usar la información de la acción de torsión para reconstruir la llave secreta. Desde ese momento, SIDH ya no es seguro para propósitos criptográficos, pero sigue siendo un buen protocolo con el cual aprender sobre isogenias.

Ejercicio 1.2. *Escribe un código en Sage para calcular un cuadrado simplificado de SIDH.*

a. *Escoge un primo p en que $E(\mathbb{F}_p) : y^2 = x^3 + x$ es supersingular, y donde existen puntos de ordenes 4 y 9.*

b. *Selecciona una base sobre $E(\mathbb{F}_{p^2})$ de orden 4 y uno de orden 9 usando la función* `E.torsion_basis(n)`.

c. *Escoge enteros secretos para Alicia y Beto a, b.*

d. *Calcula las isogenias φ_A, φ_B y las curvas E_A, E_B.*

e. *Calcula las isogenias "φ_A", "φ_B" y verifica que los codominios sean iguales.*

Ejercicio 1.3. *¿Por qué es necesario hacer el protocolo sobre $\mathbb{F}_{p^2}$ en vez de hacerlo sobre $\mathbb{F}_p$?*

a. *En la curva E del ejercicio precedente, ¿Cuántos puntos hay sobre $\mathbb{F}_p$ con orden 4?; ¿Cuántos con orden 9?*

b. *¿Cuántos puntos hay sobre $\mathbb{F}_{p^2}$ con orden 4? ¿Cuántos con orden 9?*

2. Acción de grupo

Las acciones de grupos nos darán otra manera de estudiar isogenias.

Definición 3 (Acción de grupo (group action))**.** *Sea G un grupo, y X un conjunto. Un mapa*

$$\star : G \times X \to X$$

es un acción de grupo si :

1. *existe un elemento $e \in G$ donde $e \star x = x$ para todos los $x \in X$;*
2. *para cualesquiera $g, h \in G$, $g \star (h \star x) = gh \star x$.*

En el Diffie-Hellman original, se calcula g^x para un elemento de grupo $g \in G$, y un entero x. Esto es un ejemplo de un acción de grupo donde X también es un grupo. Aquí, el grupo de enteros está actuando sobre el grupo G a través de exponenciación. Pero Diffie-Hellman puede ser descrito de forma más general usando acciones de grupos.

Protocolo 2 (Diffie-Hellman con acción de grupo)**.** *Selecciona un acción de grupo $(G, X, \star)$. Selecciona un elemento $x \in X$. Sean $(\star, G, X, x)$ los parámetros públicos.*

Ahora Alicia y Beto seleccionan sus elementos secretos del grupo $g_A, g_B \in G$. Cada uno calcula y publica $g_A \star x, g_B \star x$ respectivamente. Ahora Beto puede calcular $g_B \star (g_A \star x)$ y Alicia puede calcular $g_A \star (g_B \star x)$. Si G es conmutativo, entonces Alicia y Beto pueden llegar a calcular el mismo objeto $(g_B \cdot g_A) \star x = (g_A \cdot g_B) \star x$.

Desafortunadamente, los acciones de grupo donde X también es un grupo no son seguros contra ataques cuánticos. Por eso hace falta estudiar acciones de grupo en que X no es un grupo. Un protocolo que usa una de estas acciones es *Commutative Supersingular Isogeny Diffie-Hellman* (CSIDH) [2], que estudiáramos en mas detalle. Pero primero veremos como definir una acción de grupo con isogenias.

2.1. CSIDH

Tomando X como el conjunto de curvas elípticas supersingulares sobre $\mathbb{F}_p$ que tengan grupo de endomorfismos isomórfico a un orden imaginario cuadrático particular,

$$\mathcal{E}\ell\ell_p(\mathcal{O}) := \{E \text{ supersingular } : \mathrm{End}_{\mathbb{F}_p}(E) \cong \mathcal{O}\}$$

y G el grupo de clases de ideales de $\mathcal{O}$,

$$\mathrm{cl}(\mathcal{O})$$

podemos construir una acción de grupo

$$\star : \mathrm{cl}(\mathcal{O}) \times \mathcal{E}\ell\ell_p(\mathcal{O}) \to \mathcal{E}\ell\ell_p(\mathcal{O}).$$

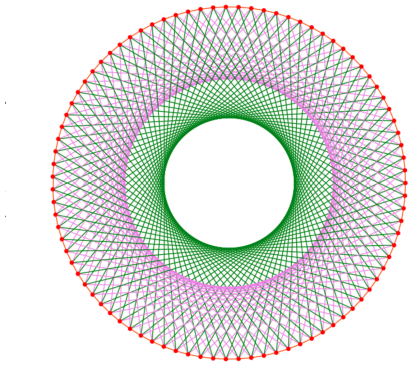

Figura 4: Grafo de isogenias sobre $\mathbb{F}_p$, con $p = 4 \cdot 5 \cdot 13 \cdot 61 - 1$. Las lineas anaranjadas corresponden a isogenias de grado 5, las verdes a isogenias de grado 13, y las moradas a isogenias de grado 61. El imagen fue hecha por Francisco Rodríguez-Henríquez [8].

Esta acción nos da una nueva manera de tomar pasos en el grafo de isogenias. Por ejemplo, empezando desde una curva $E \in \mathcal{E}\ell\ell_p(\mathcal{O})$, podemos seleccionar un elemento $\mathfrak{a} \in \mathrm{cl}(\mathcal{O})$, y calcular

$$E' = \mathfrak{a} \star E.$$

Este paso corresponde a una isogenia de grado $n(\mathfrak{a})$, donde $n(\cdot)$ se refiere a la norma del elemento. Como ya vimos, calcular isogenias de alto grado no es fácil. Así que queremos dividir $\mathfrak{a}$ en pasos mas chicos. Sea $\mathfrak{l}_1, \ldots \mathfrak{l}_k$ un conjunto de elementos (primos) de $\mathrm{cl}(\mathcal{O})$ de bajo norma y que generen un gran parte de $\mathrm{cl}(\mathcal{O})$. Entonces, en vez seleccionar $\mathfrak{a} \in \mathrm{cl}(\mathcal{O})$, podemos seleccionar un vector $[e_1, \ldots e_k] \in \mathbb{Z}_m^k$ y calcular

$$E' = \prod_{i=1}^{k} \mathfrak{l}_i^{e_i} \star E,$$

donde m es seleccionado según el tamaño de $\mathrm{cl}(\mathcal{O})$.

De esta forma, nos aseguramos que la isogenia correspondiente se pueda construir de isogenias con grados $n(\mathfrak{l}_i)$. En la Figure 4 se ve un ejemplo un grafo de isogenias con tres opciones de $\mathfrak{l}_i$.

Esta idea de computación fue usada en [2] para construir *Commutative Supersingular Isogeny Diffie-Hellman* (CSIDH). Este protocolo es muy similar al Protocolo 2. CSIDH sigue siendo de los protocolos de isogenias más populares hoy en día.

Protocolo 3. *Primero se definen los parámetros públicos : el conjunto $\mathcal{E}\ell\ell_p(\mathcal{O})$, una curva $E_0 \in \mathcal{E}\ell\ell_p(\mathcal{O})$ inicial, los elementos $\mathfrak{l}_1, \ldots \mathfrak{l}_k$, y el entero m.*

Ahora Alicia y Beto escogen sus vectores secretos $(a_1, \ldots a_k), (b_1, \ldots b_k) \in \mathbb{Z}_m^k$.

Alicia publica la curva

$$E_A := \prod_{i=1}^{k} \mathfrak{l}_i^{a_i} \star E_0$$

y Beto la curva

$$E_B := \prod_{i=1}^{k} \mathfrak{l}_i^{b_i} \star E_0.$$

Como la acción es conmutativa, cada uno puede calcular la curva secreta

$$\prod_{i=1}^{k} \mathfrak{l}_i^{b_i} \star E_A = \prod_{i=1}^{k} \mathfrak{l}_i^{a_i} \star E_B =: E_{AB}$$

Este próximo ejemplo simplificado de CSIDH fue escrita por Lorenz Panny [6].

```
p = 38798759
E0 = EllipticCurve(GF(p), [1,0])
factor(p+1)
ls = [3,5,7,11,13,17,19]

def point_of_order(E, l):
    assert l in ls
    while True:
        P = E.random_point()
        P *= (p+1)//l
        if P:
            return P

point_of_order(E0, 7)

def step(E, l):
    K = point_of_order(E, l)
    phi = E.isogeny(K)
    assert phi.degree() == l
    return phi.codomain()
```

Ejercicio 2.1. *Usa el ejemplo precedente para verificar en Sage que la acción de grupo es conmutativo. Toma varios pasos de las opciones ls en ordenes distintos y chequea que el resultado final sea siempre el mismo.*

De manera interesante, la acción puede ser generalizada para un conjunto de curvas más grandes. Para lograrlo, necesitaríamos más información de las curvas. Específicamente, necesitamos que las curvas sean *orientadas*. Sin detallarlo formalmente, una orientación es un mapa descrito como un cuerpo de números imaginario cuadrático que *encaja* en el grupo de endomorfismos de la curva. Este mapa nos informa de cómo restringir el anillo de endomorfismos a un conjunto que sea conmutativo. Estas orientaciones no siempre son fáciles de construir. Por eso es necesario realizar un estudio muy cuidadoso.

Para un estudio más completo del grupo de clases de ideales, consulta el libro de Cox [3]. Para aprender sobre orientaciones, revisa el artículo presentado en este LatinCrypt 2025:

Orient Express: Using Frobenius to Express Oriented Isogenies escrito por Wouter Castryck, Riccardo Invernizzi, Gioella Lorenzon, Jonas Meers, y Frederik Vercauteren.

3. Emparejamientos

Un *emparejamiento* (pairing) es un mapa bilinear $e : G_1 \times G_2 \to G_3$ donde

$$e(x, ry) = e(rx, y) = re(x, y);$$

$$e(xz, y) = e(x, y) \cdot e(z, y);$$

$$e(x, yz) = e(x, y) \cdot e(x, z).$$

Estos mapas son muy útiles en criptografía de curvas elípticas e isogenias. Uno de los más conocidos es el *emparejamiento de Weil* que proyecta puntos de orden n sobre una curva elíptica a raíces n-ésimas de unidad.

El emparejamiento de Weil tiene la propiedad que calculándolo en el mismo punto es trivial, i.e. $e_n(P, P) = 1$ para todos los puntos P.

```
p = 107
n = 9
E = EllipticCurve(GF(p^2), [0,0,0,1,0])
P,Q = E.torsion_basis(n)
P.weil_pairing(Q,n)
```

Ejercicio 3.1. *Verifica en el ejemplo precedente que el resultado del emparejamiento es un raíz n-ésima de la unidad usando Sage.*

También verifica que el emparejamiento es trivial cuando es aplicado al mismo punto.

3.1. Uso con isogenias

El emparejamiento de Weil respeta la acción de isogenias. Sea $\phi : E \to E'$ una isogenia de grado d, y e_n, e'_n emparejamientos de Weil sobre E y E' respectivamente. Para puntos $P, Q \in E[n]$,

$$e'_n(\phi(P), \phi(Q)) = e_n(P, Q)^d.$$

Esta relación es muy importante y demuestra que los emparejamientos pueden ser herramientas útiles en el estudio de isogenias.

Ejercicio 3.2. *Selecciona parámetros E, N, P, Q, ϕ, d y verifica en Sage que $e'_n(\phi(P), \phi(Q)) = e_n(P, Q)^d$.*

Para aprender más sobre el uso de *(*emparejamientos) en criptografía, estudia los artículos presentados en este LatinCrypt 2025:

A Note on the Advanced Use of the Tate emparejamiento escrito por Krijn Reijnders ;

Improved algorithms for ascending isogeny volcanoes, and applications escrito por Steven D. Galbraith, Valerie Gilchrist, y Damien Robert.

```
Firmas con Isogenias:
Tenemos una isogenia y queremos mostrar que la
sabemos sin darla.
Sea phi una isogenia, su dual ~phi es tal que (phi o
~phi) = [deg(phi)]

GPS/csi-fish:
Tenemos el siguiente esquema donde las curvas son
públicas y phi es la
isogenia privada.
E_0 -psi-> E_1
|         /
phi  psi o ~phi
|   /
E_A/

El commitment es psi y el reto puede ser dar psi, o
bien, (psi o ~phi).
Tenemos un soundness de 1/2.

SQISign:
El nuevo esquema es:
E_0 -psi-> E_1
|           |
phi        varphi
|           |
E_A -sigma-> E_2
donde sigma = (varphi o phi o ~phi), phi es la clave
secreta,
psi es el commitment y varphi el challenge.

KLPT se usa acá y es un algoritmo que encuentra una
isogenia entre dos
curvas elípticas dadas, conectando sus órdenes. Se
usa para generar la clave
secreta y su correspondiente clave pública.
```

Referencias

[1] Wouter Castryck and Thomas Decru. An efficient key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023*, volume 14008 of *Lecture Notes in Computer Science*, pages 423–447. Springer, 2023.

[2] 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 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part III*, volume 11274 of *Lecture Notes in Computer Science*, pages 395–427. Springer, 2018.

[3] David A. Cox. *Primes of the form x^2+ny^2: Fermat, Class Field Theory, and Complex Multiplication*. AMS Chelsea Publishing/American Mathematical Society, third edition, 2022.

[4] David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In Bo-Yin Yang, editor, *Post-Quantum Cryptography*, pages 19–34, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[5] Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski. A direct key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023*, volume 14008 of *Lecture Notes in Computer Science*, pages 448–471. Springer, 2023.

[6] Lorenz Panny. Some exercises on isogeny-based cryptography. https://yx7.cc/docs/isog/isog_taipei_exercises.pdf#page=3.08, July 2022. Post-Quantum Crypto Mini-School 2022, Taipei, Taiwan.

[7] Damien Robert. Breaking SIDH in polynomial time. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023*, volume 14008 of *Lecture Notes in Computer Science*, pages 472–503. Springer, 2023.

[8] Francisco Rodríguez-Henríquez. A tale of two isogeny-based protocols: State-of-the-art review of key-exchange and signature isogeny-based protocols, June 2024. EPFL, Lausanne, Switzerland.

[9] Jacques Vélu. Isogénies entre courbes elliptiques. *Comptes rendus hebdomadaires des séances de l'Académie des sciences, Série A*, 273:238–241, 7 1971.