Cybersecurity Institute
Univ. Grenoble Alpes

INSTITUT FOURIER

Sharing a secret key using supersingular isogenies

Vincent Espitalier, Cyril Hugounenq

12th Atelier PARI GP

18 January, 2019

Cybersecurity Institute
Univ. Grenoble Alpes

INSTITUT FOURIER

Secure key exchange

Diffie-Hellman [DH76]

Given a group $(G, \cdot)$ and a generator of this group g Alice and Bob compute a shared secret proceeding as follows :

- Alice (resp. Bob) computes $h_a = g^a$ ($h_b = g^b$) from her (resp. his) secret exponent a (b) and sends it ;

Secure key exchange

Diffie-Hellman [DH76]

Given a group $(G, \cdot)$ and a generator of this group g Alice and Bob compute a shared secret proceeding as follows :

- Alice (resp. Bob) computes $h_a = g^a$ ($h_b = g^b$) from her (resp. his) secret exponent a (b) and sends it ;
- Alice recepts h_b from Bob (resp. Bob recepts h_a from Alice) and compute $k = {h_b}^a$ (resp. $k = {h_a}^b$).

Secure key exchange

Diffie-Hellman [DH76]

Given a group $(G, \cdot)$ and a generator of this group g Alice and Bob compute a shared secret proceeding as follows :

- Alice (resp. Bob) computes $h_a = g^a$ ($h_b = g^b$) from her (resp. his) secret exponent a (b) and sends it ;
- Alice recepts h_b from Bob (resp. Bob recepts h_a from Alice) and compute $k = {h_b}^a$ (resp. $k = {h_a}^b$).
- They share a common secret key $k = g^{ab} = g^{a^b} = g^{b^a}$

Secure key exchange

Diffie-Hellman [DH76]

Given a group $(G, \cdot)$ and a generator of this group g Alice and Bob compute a shared secret proceeding as follows :

- Alice (resp. Bob) computes $h_a = g^a$ ($h_b = g^b$) from her (resp. his) secret exponent a (b) and sends it ;
- Alice recepts h_b from Bob (resp. Bob recepts h_a from Alice) and compute $k = {h_b}^a$ (resp. $k = {h_a}^b$).
- They share a common secret key $k = g^{ab} = g^{a^b} = g^{b^a}$

Now we will do the same with supersingular isogeny following [LDJ14, JAC+17].

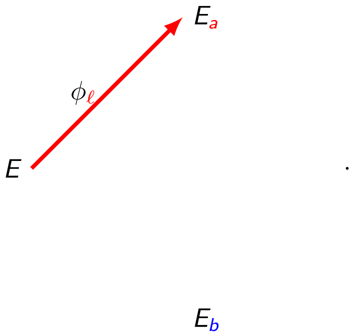
Supersingular Isogeny Diffie Hellman [LDJ14, JAC+17]

Given a supersingular curve E defined over a finite field $\mathbb{F}_q$ and two bases : $\langle P_\ell, Q_\ell \rangle \simeq \mathbb{Z}/\ell^{e_\ell}\mathbb{Z} \times \mathbb{Z}/\ell^{e_\ell}\mathbb{Z}$, $\langle P_m, Q_m \rangle \simeq \mathbb{Z}/m^{e_m}\mathbb{Z} \times \mathbb{Z}/m^{e_m}\mathbb{Z}$.
Alice and Bob compute a shared secret proceeding as follows :

Supersingular Isogeny Diffie Hellman [LDJ14, JAC+17]

Given a supersingular curve E defined over a finite field $\mathbb{F}_q$ and two bases : $\langle P_\ell, Q_\ell \rangle \simeq \mathbb{Z}/\ell^{e_\ell}\mathbb{Z} \times \mathbb{Z}/\ell^{e_\ell}\mathbb{Z}$, $\langle P_m, Q_m \rangle \simeq \mathbb{Z}/m^{e_m}\mathbb{Z} \times \mathbb{Z}/m^{e_m}\mathbb{Z}$.
Alice and Bob compute a shared secret proceeding as follows :

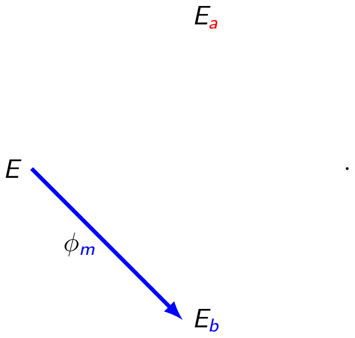
- *Alice* computes an isogeny $\phi_\ell : E \mapsto E/\langle P_\ell + aQ_\ell \rangle$ from her secret exponent a and sends $\phi_\ell(P_m), \phi_\ell(Q_m), E_a = E/\langle P_\ell + aQ_\ell \rangle$;



Supersingular Isogeny Diffie Hellman [LDJ14, JAC+17]

Given a supersingular curve E defined over a finite field $\mathbb{F}_q$ and two bases : $\langle P_\ell, Q_\ell \rangle \simeq \mathbb{Z}/\ell^{e_\ell}\mathbb{Z} \times \mathbb{Z}/\ell^{e_\ell}\mathbb{Z}$, $\langle P_m, Q_m \rangle \simeq \mathbb{Z}/m^{e_m}\mathbb{Z} \times \mathbb{Z}/m^{e_m}\mathbb{Z}$. Alice and Bob compute a shared secret proceeding as follows :

- *Bob* computes an isogeny $\phi_m : E \mapsto E/\langle P_m + bQ_m \rangle$ from his secret exponent b and sends $\phi_m(P_\ell), \phi_m(Q_\ell), E_b = E/\langle P_m + bQ_m \rangle$;

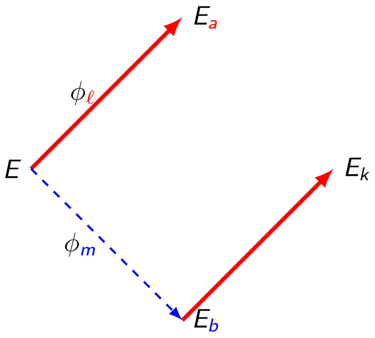


Supersingular Isogeny Diffie Hellman [LDJ14, JAC+17]

Given a supersingular curve E defined over a finite field $\mathbb{F}_q$ and two bases : $\langle P_\ell, Q_\ell \rangle \simeq \mathbb{Z}/\ell^{e_\ell}\mathbb{Z} \times \mathbb{Z}/\ell^{e_\ell}\mathbb{Z}$, $\langle P_m, Q_m \rangle \simeq \mathbb{Z}/m^{e_m}\mathbb{Z} \times \mathbb{Z}/m^{e_m}\mathbb{Z}$. Alice and Bob compute a shared secret proceeding as follows :

- *Alice* computes an isogeny $\phi_\ell : E \mapsto E/\langle P_\ell + aQ_\ell \rangle$ from her secret exponent a and sends $\phi_\ell(P_m), \phi_\ell(Q_m), E_a = E/\langle P_\ell + aQ_\ell \rangle$;

  Alice recepts $E_b, \phi_m(P_\ell), \phi_m(Q_\ell)$ from Bob and computes $k = j(E_b/\langle \phi_m(P_\ell) + a\phi_m(Q_\ell) \rangle)$

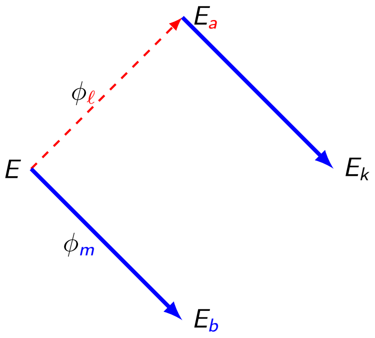


Supersingular Isogeny Diffie Hellman [LDJ14, JAC+17]

Given a supersingular curve E defined over a finite field $\mathbb{F}_q$ and two bases : $\langle P_\ell, Q_\ell \rangle \simeq \mathbb{Z}/\ell^{e_\ell}\mathbb{Z} \times \mathbb{Z}/\ell^{e_\ell}\mathbb{Z}$, $\langle P_m, Q_m \rangle \simeq \mathbb{Z}/m^{e_m}\mathbb{Z} \times \mathbb{Z}/m^{e_m}\mathbb{Z}$. Alice and Bob compute a shared secret proceeding as follows :

- *Bob* computes an isogeny $\phi_m : E \mapsto E/\langle P_m + bQ_m \rangle$ from his secret exponent b and sends $\phi_m(P_\ell), \phi_m(Q_\ell), E_b = E/\langle P_m + bQ_m \rangle$;

  Bob recepts $E_a, \phi_\ell(P_m), \phi_\ell(Q_m)$ from Alice and computes $k = j(E_a/\langle \phi_\ell(P_m) + b\phi_\ell(Q_m) \rangle))$.

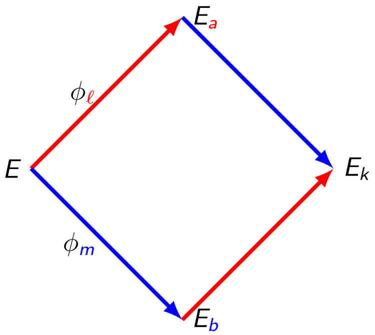


Supersingular Isogeny Diffie Hellman [LDJ14, JAC+17]

Given a supersingular curve E defined over a finite field $\mathbb{F}_q$ and two bases : $\langle P_\ell, Q_\ell \rangle \simeq \mathbb{Z}/\ell^{e_\ell}\mathbb{Z} \times \mathbb{Z}/\ell^{e_\ell}\mathbb{Z}$, $\langle P_m, Q_m \rangle \simeq \mathbb{Z}/m^{e_m}\mathbb{Z} \times \mathbb{Z}/m^{e_m}\mathbb{Z}$. Alice and Bob compute a shared secret proceeding as follows :

- *Bob* computes an isogeny $\phi_m : E \mapsto E/\langle P_m + bQ_m \rangle$ from his secret exponent b and sends $\phi_m(P_\ell), \phi_m(Q_\ell), E_b = E/\langle P_m + bQ_m \rangle$;

  Bob recepts $E_a, \phi_\ell(P_m), \phi_\ell(Q_m)$ from Alice and computes $k = j(E_a/\langle \phi_\ell(P_m) + b\phi_\ell(Q_m) \rangle))$.
- They share a common secret key $k = j(E/\langle P_m + bQ_m \rangle, \langle P_\ell + aQ_\ell \rangle) = j(E_k)$



Computation of the public parameters

We follow the example of SikeP503 of [JAC+17]

```
? e_2=250; e_3=159; p=2^e_2*3^e_3-1;
? isprime(p)
% = 1
```

We first define the finite field $\mathbb{F}_{p^2}$ as follows

```
? P=(X^2+1)*Mod(1,p);
? a=ffgen(P,'a);
```

Now we define a curve on $\mathbb{F}_{p^2}$ that we know is supersingular :

```
? E=ellinit([0,1],a);
```

We can check that this curve has the good structure :

```
? E.cyc
% = [1317584315690711738083925 [...] 961154237431808,
 1317584315690711738083925 [...] 961154237431808]
```

Here we just saw the structure of the group of points of E. We now facotrize this result for some clarity :

```
? factor(%[1])
% =
[2 250]

[3 159]
```

Now we will determine a basis of $E[2^{250}] \simeq \mathbb{Z}/2^{250}\mathbb{Z} \times \mathbb{Z}/2^{250}\mathbb{Z}$

Cybersecurity Institute
Univ. Grenoble Alpes

INSTITUT FOURIER

We look for a candidate point by incrementing the possible abscissae :

```
? x_0=Mod(1,p); issquare(x_0^3+x_0)
% = 1
```

The first one is indeed the abscissa of a point of the elliptic curve, we compute then the ordinate and the order of this point.

```
? issquare(x_0^3+x_0,&y_0);
//or y_0=ellordinate(E,x_0)[2];
? ellisoncurve(E,[x_0,y_0])
% = 1
? P=[x_0,y_0]; factor(ellorder(E,P))
%10 =
[2 247]

[3 156]
```

We see that this point is not of order high enough, thus we search for another point.

```
? x_0=Mod(14,p); issquare(x_0^3+x_0,&y_0)
= 1
```

To obtain directly a point of order 2^{250} we multiply this point directly by the cofactor of the order of the group of points of E : $3^{159} = 3^{e_3}$.

```
? P_2=ellmul(E0,[x_0,y_0],3^(e_3));
? factor(ellorder(E0,P_2))
% =
[2 250]
```

This point P_2 has good order.

Now we search for a second point to form a basis of $E[2^{250}]$.

```
? x_0=a+4; issquare(x_0^3+x_0)
% = 1
? issquare(x_0^3+x_0,&y_0); \
Q_2=ellmul(E,[x_0,y_0],3^(e_3)); factor(ellorder(E,Q_2))
% =
[2 250]
```

We have found a second point Q_2 of order 2^{250}, we have then to test if it forms a basis of $E[2^{250}]$ with the Weil pairing :

```
? t= ellweilpairing(E,P_2,Q_2,2^(e_2))
% = 414[...]58013*a + 1966154683551[...]38443
? t^{2^(e_2-1)}
% = 1317584315690711738083 9[...]1154237431806
```

Since the Weil pairing is of maximal order the points P_2, Q_2 do form a basis of $E[2^{250}]$.

Cybersecurity Institute
Univ. Grenoble Alpes

INSTITUT FOURIER

Now we can compute the secret isogeny ϕ_2 generated from the secret sk_2,

Cybersecurity Institute
Univ. Grenoble Alpes

INSTITUT FOURIER

Now we can compute the secret isogeny ϕ_2 generated from the secret sk_2, but since it is an isogeny of great order we won't compute it directly. We will compute iteratively the 250 2-isogeny as follows.

```
? sk2=random(2^{e_2}); R_2=ellmul(E,Q_2,sk2);
? S_2=elladd(E,P_2,R_2); S_int=ellmul(E,S_2,2^(e_2-1));
? iso=ellisogeny(E,S_int)
 = [[0, 0, 0, 13175843156[...]1154237431803, 0],
  [x^3 + x, y*x^3 +
  131758431569071173808392[...]019593961154237431806*y*x,
   x]]
```

A more clever and optimized method is proposed in [LDJ14, JAC+17]. The first output of ellisogeny is the codomain curve and the second output is the "mapping" itself.

```
? Eb=ellinit(iso[1]);
```

Once we have defined a basis of $E[3^{159}]$:

```
? Q_3x=0x1e7d6 [...] d7e39b6997f70023e0a23b4b3787ef08f;
? Q_3y=0x2ec0a [...] c8ad47064f05c06dc5d4aae61ccceff1f26;
? Q_3=[Q_3x,Q_3y];
? P_3x1=0x21b7 [...] a67dd7ed98b9793685fa2e22d6d89d66a4e;
? P_3x2=0x2f37 [...]9ceb53821d3e8012f7f391f57364f402909;
? P_3y1=0x78f8 [...] efee6010cdf34a7de9f9e239b103e7b3eee;
? P_3y2=0x37f3 [...]61d04f9f3a8317f7916e016f2733b828ac0;
? P_3x=P_3x1+a*P_3x2;  P_3y=P_3y1+a*P_3y2;P_3=[P_3x,P_3y]
```

One can check that the isogeny preserves the order of points of $E[3^{159}]$

```
? factor(ellorder(Eb,ellisogenyapply(iso[2],P_3)))
% =
[3 159]

? factor(ellorder(Eb,ellisogenyapply(iso[2],Q_3)))
% =
[3 159]
```

We generalize those computations of isogenies with the following similar functions :

- *isogen*$_\ell$
- *isoex*$_\ell$

They are viewable in `definition_fonctions.gp` (gp2). They only consist in the following algorithm :

isogen$_\ell$

Entrée: Bases of $E[\ell^{e_\ell}] = \langle P_\ell, Q_\ell \rangle$ and $E[m^{e_m}] = \langle P_m, Q_m \rangle$ and a secret exponent sk_ℓ

Sortie: The image of the basis $E[m^{e_m}]$ by the isogeny specified by the secret exponent sk_ℓ and $E[\ell^{e_\ell}]$

Compute $S_\ell = P_\ell + [sk_\ell]Q_\ell$

for $i = 1$ to e_ℓ **do**

  Compute $\phi : E \mapsto E/\langle S_\ell \rangle = E_b$

  Compute $(P_m, Q_m, S_\ell, E) \leftarrow (\phi(P_m), \phi(Q_m), [\ell^{e_\ell - i - 1}]\phi(S_\ell)), E_b)$

end for

return (P_m, Q_m, E)

Cybersecurity Institute
Univ. Grenoble Alpes

INSTITUT FOURIER

Computing a common secret key

Thus using those functions and the points we have computed we can share a common secret key :

```
? R_2 = ellsub(E, P_2, Q_2); //not useful here
? R_3 = ellsub(E, P_3, Q_2); //not useful here
? sk3=random(3^{e_3});
? pk2 = isogen_ell(E, P_2, Q_2, sk2, P_3,
 Q_3, R_3, e_2, 2);
? pk3 = isogen_ell(E, P_3, Q_3, sk3, P_2,
Q_2, R_2, e_3, 3);
```

```
? j_ech1 = isoex_ell( pk2[4], sk3, pk2[1],
  pk2[2], pk2[3], e_3, 3)
% = 95211232169147321 37[...]0222351872676569 04887*a +
  6307916662[...]24346811520916299511477286097663421965
? j_ech2 = isoex_ell( pk3[4], sk2, pk3[1],
pk3[2], pk3[3], e_2, 2)
% = 95211232169147321 37[...]0222351872676569 04887*a +
6307916662[...]24346811520916299511477286097663421965
? j_ech1==j_ech2
% = 1
```

Cybersecurity Institute
Univ. Grenoble Alpes

INSTITUT FOURIER

```
? j_ech1 = isoex_ell( pk2[4], sk3, pk2[1],
 pk2[2], pk2[3], e_3, 3)
% = 95211232169147321 37[...]022235187267656904887*a +
 6307916662[...]24346811520916299511477286097663421965
? j_ech2 = isoex_ell( pk3[4], sk2, pk3[1],
pk3[2], pk3[3], e_2, 2)
% = 9521123216914732137[...]022235187267656904887*a +
6307916662[...]24346811520916299511477286097663421965
? j_ech1==j_ech2
% = 1
```

In the end we get the same *j*-invariant.

Thanks for your attention.

Whitfield Diffie and Martin E. Hellman.
New directions in cryptography.
IEEE Trans. Information Theory, 22(6) :644–654, 1976.

David Jao, Reza Azarderakhsh, Matthew Campagna, Craig Costello, Amir Jalali, Brian Koziel, Brian LaMacchia, Patrick Longa, Michael Naehrig, Joost Renes, et al.
Supersingular isogeny key encapsulation november 30, 2017.
2017.

De Feo Luca, Jao David, and Plût Jérôme.
Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies.
J. Mathematical Cryptology, 8(3) :209–247, 2014.