How do I multiply two points on an elliptic curve?

Asked 2 years, 3 months ago Modified 2 years, 3 months ago Viewed 1k times

4

Tell me if there is a way to multiply two points on an elliptic curve?

For example, as in `secp256k1`

```
Pcurve = 115792089237316195423570985008687907853269984665640564039457584007908834671663 # The
proven prime
GPoint = (Gx, Gy) # Generator Point
N = 115792089237316195423570985008687907852837564279074904382605163141518161494337 # Number of
points in the field
Acurve = 0; Bcurve = 7 # This defines the curve.

y ^ 2 = x ^ 3 + Acurve * x + Bcurve
```

```
(Gx1, Gy1) * (Gx2, Gy2) = (Gx3, Gy3)
```

Unfortunately I do not know their private key from which they were generated, but I really need to get a common point `(Gx3, Gy3)`

What algorithm is used to multiply two points on the `secp256k1` elliptic curve?

public-key elliptic-curves diffie-hellman keys secret-sharing

Share Improve this question Follow

asked Feb 13, 2021 at 9:43

Derick Swodnick
67 4

1 There is no standard *meaning for*, much less *way to* "multiply two points on an Elliptic Curve" like secp256k1. Given points/public keys, it's possible to add them (formulas for that are in sec1 §2.2.1, with the constants for secp256k1 in sec2 §2.4.1), but the result bears little useful relation to the corresponding private keys. – fgrieu ♦ Feb 13, 2021 at 11:33

@DerickSwodnick Please reconsider your question. Having to do that is normal here. I do it constantly. – Patriot Feb 13, 2021 at 14:04

4 @fgrieu You could define multiplication of points as $aG \cdot bG = (ab)G$, which should have all the expected properties of multiplication. Of course computing this operation is infeasible on secure curves, since this is the computational Diffie-Hellman problem. – CodesInChaos Feb 13, 2021 at 14:15

1 Answer

Sorted by: Highest score (default) ⬍

7

There is no point multiplication operation on secure elliptic curves[*], there is only scalar multiplication apart from the point addition of the group.

Scalar multiplication

Scalar multiplication with a scalar t is adding a point P it self t-times

$$[t]P := \underbrace{P + P + \cdots + P}_{t-times}$$

and it is well defined operation since the group is operation is the addditon of the points. Scalar multiplication is a natural writing of the adding t-times.

If you want to execute a Diffie-Hellman here it is;

Elliptic Curve Diffie-Hellman Key Exchange (ECDH)

ECDH is executed with the parties' public keys. Let say Alice has secret key a and public key $A = [a]G$ and Bob has secret key b with public key $B = [b]G$ with the base point G defined by the standards.

Then both parties can agree on the key as;

- Alice: get Bob's public key B and multiply it with secret key a; $S = [a]B = [ab]G$
- Bob: get Alice's public key A and multiply it with secret key b; $[b]A = [ab]G = S$

So both arrived at the same point S, now they can use the x-corrdinate point S with a Key Derivation Function (KDF) like HKDf to derive a symmetric key.

Ephemeral

Originally, Diffie-Hellman is defined as ephemeral, that is for each key generation, Alice and Bob generates new uniform random points by selecting a uniformly random a and b per run of the protocol. This provides forward secrecy if you delete the keys after decryption.

If you don't generate new uniform random keys, it is called the static DHKE. Therefore you will see around static-static, ephemeral-static, static-ephemeral, and ephemeral-ephemeral usages of the DHKE on around. Hopefully, the static case is no longer available in TLS 1.3.

MITM attack

DHKE is vulnerable to a man-in-the-middle attack especially the ephemeral case since the public points are changing all the time. A signature must be used to prevent this attack.

[*] As noted by CodeinChaos in the comments we can define a multiplication on curves with $[a]G \cdot [b]G = [ab]G$. This, clearly, have commutativity, associativity, identity, and distribution law. The inverse depends on the order of the G. This multiplication, however, is exactly the Computational Diffie-Hellman (CDH) problem. If you solve the Discrete Log (DL) problem on the curve then you can solve the CHD, the reverse is not known in the general case.

Since we required that the DL (or CDH) must be infeasible for security, this multiplication operation is not a feasible operation on the secure curves.

Share Improve this answer Follow

edited Feb 13, 2021 at 15:26

answered Feb 13, 2021 at 10:34

kelalaka

46k 9 106 182