MULTI-PARTY KEY EXCHANGES

JONATHAN ELLISTON AND LORENZO WOLCZKO

1. Summary

This paper will focus on different types of key exchanges involving more than one party. This is useful during multi-party information exchanges to avoid holding many individual keys for each participant in the exchange. One way to accomplish these multi-way key exchanges is by generalizing Diffie-Hellman to more parties. However, in practice, this method isn't used for these key exchanges as it requires multiple rounds of information exchange. Therefore, we will be focusing on better ways of accomplishing this exchange (and we will find elliptic curves cropping up quite frequently). For example, we can find an analogue for the Diffie-Hellman key exchange (and, by extension, ElGamal) in elliptic curves. This will be by using elliptic curve addition instead of modular exponentiation [Kob87] [AA13]. Lastly, there is a way, using pairings of elliptic curves, to exchange a private key between 3 people over an insecure channel with only one round of information exchange [Jou00].

2. Generalizing Diffie-Hellman to an n-way Key Exchange

Generalizing Diffie-Hellman to make it a multiparty key exchange is straight forward. However, it requires multiple rounds of information exchange. In fact, where n is the number of parties involved, it takes $n-1$ rounds of communication to perform this exchange. Before we go into multi-way Diffie-Hellman, let us refresh our memory of two-way Diffie-Hellman.

(1) Alice and Bob agree on a prime p and g, a primitive root of p.
(2) Alice and Bob choose random integers a and b, respectively.
(3) Alice computes $A = g^a \pmod{p}$, and shares A. Bob computes $B = g^a \pmod{p}$, and shares B.
(4) Finally, Alice computes that the key equals $B^a \pmod{p}$ and Bob computes the same key with $A^b \pmod{p}$.

Now, suppose n parties wish to construct a shared key.

(1) The parties agree on a prime p and a primitive root g.
(2) Party i chooses a random $a_i \in \mathbb{F}_p$ for all i.
(3) Party i computes and sends $g^{a_i} \pmod{p}$ to party $i+1 \pmod{n}$ for all i.
(4) Then, party i computes and sends $g^{a_i \cdot a_{i-1}} \pmod{p}$ to person $i+1 \pmod{n}$ for all i.
(5) Continuing this, they will get a shared key: $g^{a_1 \cdot a_2 \cdot \dots \cdot a_n} \pmod{p}$.

3. Elliptic Curve Diffie-Hellman [AA13]

We can use Diffie-Hellman over any finite Abelian group, but it turns out elliptic curves over $\mathbb{F}_p$ work well. This is mainly because the index calculus attack doesn't work. Here are the steps to create a shared key using Elliptic Curve Diffie-Hellman:

Date: December 8, 2019.

(1) Alice and Bob agree on a prime p, an elliptic curve E over $\mathbb{F}_p$, and a point P.
(2) Alice and Bob compute their private keys, a and b, respectively.
(3) Alice shares $A = P \cdot a$ and Bob shares $B = P \cdot b$, their respective public keys.
(4) Alice and Bob both compute $P \cdot ab$, the shared key.

Just like normal Diffie-Hellman, this also works with more than 2 people, although is requires multiple rounds of information exchange. Something interesting to note is that in some cases, Elliptic Curve Diffie-Hellman is easy to break. It was shown in [Jou00] that the discrete logarithm problem can be solved in polynomial time over elliptic curves over $\mathbb{F}_p$ with exactly p points. Some other elliptic curves over $\mathbb{F}_p$ also reduce to an extension of $\mathbb{F}_p$, making them vulnerable to attack using the Menezes, Okamoto, Vanstone (MOV) reduction and the Frey, Rück (FR) reduction [Jou00], of which the FR reduction is faster. It turns out that these reductions are closely related to a three-way key exchange.

4. The Weil Pairing

Before we talk about the Weil pairing on elliptic curves, we need to talk about divisors and function fields.

Definition 4.1. [Jou00] The *function field* $K(E)$ of an elliptic curve E over a field F is the set of rational maps $g(X, Y)$ which return an element of F, and with X and Y satisfying the equation of E.

Definition 4.2. [Jou00] If a point D on a curve E can be expressed as a sum

$$D = a_1 \cdot P_1 + a_2 \cdot P_2 + \ldots + a_n \cdot P_n$$

then we call D a *divisor*. Furthermore, if $a_1 + a_2 + \ldots + a_n = 0$, then D is a divisor of *degree 0*.

Definition 4.3. [Jou00] Note that because any function $f \in K(E)$ has the same number of zeroes as poles (including multiplicity), we can add the zeroes and subtract the poles to get a divisor of degree zero. We call this divisor $div(f)$ the *principle divisor* of f.

In fact, to test if a divisor D of a function f is principal by evaluating it directly over E. The result is the point at infinity if and only if D is the principal divisor of f. With that out of the way, we can now define the Weil pairing, the basis of our three-way key exchange.

Definition 4.4. [Jou00] The *Weil Pairing* is a bilinear function from the torsion group $E[n]$ to the multiplicative group μ_n of the nth-roots of unity in some extension of $\mathbb{F}_p$. Given P and Q, two n-torsion points on an elliptic curve E, their pairing is defined to be

$$e_n(P, Q) = \frac{f_P(Q)}{f_Q(P)}$$

where f_P and f_Q exist such that $div(f_P) = n \cdot P - n \cdot O$, $div(f_Q) = n(Q) - n(O)$, and O is the point at infinity.

The Weil pairing also has some interesting properties. It is non-degenerate, meaning that

$$e(aP, bQ) = e(P, Q)^{ab}$$

5. Three-Way Elliptic Curve Diffie-Hellman

Alice, Bob, and Charlie are looking to construct a shared key with only one round of information exchange. They all agree on an elliptic curve E and a point P. Just like normal Diffie-Hellman, they choose private keys a, b, and c respectively. Then, Alice computes and sends $P_A = aP$ to everyone else, Bob computes and sends $P_B = bP$ to everyone else, and Charlie computes and sends $P_C = cP$ to everyone else. Now we need a function F such that

$$F(a, P_B, P_C) = F(b, P_A, P_C) = F(c, P_A, P_B)$$

and that $F(a, P_B, P_C)$ is hard to calculate given only P_A, P_B, and P_C. Using the Weil pairing, we can find such an F

$$F_W(x, P, Q) = e_n(P, Q)^x.$$

It is easy to check that this function indeed satisfies the requirements, given that the Weil pairing is non-degenerate. Unfortunately, it is not this easy because that would mean the shared key is always 1. So, instead, we use two points P and Q. Alice, Bob, and Charlie also compute their respective powers of Q, and the shared key is

$$F(a, P_B, Q_C) = F(b, P_A, Q_C) = F(c, P_A, Q_B).$$

6. An Example

Say Alice, Bob, and Charlie agree on the curve $y^2 = x^3 + x$ over the field $\mathbb{F}_{p^2}$ over $x^2 + 1$ where

$$\begin{gathered} p = 4826777781577004353504441085636004703895396072911357 4 \\ 2953085077414483299007817968457323051999107203153032 9 \\ 3733302359127163605069681752367164649238072377341901 1. \end{gathered}$$

Also, we chose p in such a way that

$$q = 593917583375891588584754753148372137203682206097$$

divides $p + 1$. We choose points P and Q of order q:

$$\begin{gathered} P = (44190300200219570605979955052143576952357255515115 68 \\ 685117019181831684209548690762548088439531761686340 19 \\ 275510060661896927080959248158979274985085358232623 71, \\ 260909476808609223955403306134286905254063296164284 70 \\ 738073031338841260885477380307130420220342204765301 86 \\ 516348020375757022366460623538154080107556380111875 1) \\ Q = (41741839015179817915732768381465901446084951835050 84 \\ 364114477814173114302373312329585774568654291610400 89 \\ 806217226455983348248260335272068783343983410685645 620, \\ 859840794383280668295355038064028484251137556880426 14 \\ 534609435398882015068450504353865472815063531531657 21 \\ 001906397291121864181015596430468303363508583810642 5i) \end{gathered}$$

Using $a = 4, b = 7, c = 28$, we get the shared key.

$$\begin{gathered} F(a, P_B, Q_C) = 21704655273258595020185058036714661585432952223857344835 \\ 67773957210551020200586870416066057916675619991969502192 \end{gathered}$$

$$\begin{gathered}6418504583078280015614517038669660149631872711 9i \\ +1854796754535600500024199532873596699011379170363502841 6 \\ 2348376178652213528456277384398902756897609415503827104 8 \\ 944364817877003701614538998745627383212540261 46\end{gathered}$$

7. Conclusion

Whereas normal Diffie-Hellman is the easiest form of the key exchange to understand, it generalizes to more parties poorly. Requiring $n-1$ exchanges of information, it is inefficient. Elliptic Curve Diffie-Hellman is simply an analogue of Modular Exponentiation Diffie-Hellman, with similar drawbacks. However, we can utilize the Weil pairing (or Tate pairing) on elliptic curves to accomplish something similar to Elliptic Curve Diffie-Hellman, but with only one round of information exchange, if three parties are involved.

References

[AA13] Ram Ratan Ahirwal and Manoj Ahke. “Elliptic Curve Diffie-Hellman key exchange algorithm for securing hypertext information on wide area network”. In: *International Journal of Computer Science and Information Technologies* 4.2 (2013), pp. 363–368.

[Jou00] Antoine Joux. “A one round protocol for tripartite Diffie-Hellman”. In: *International Algorithmic Number Theory Symposium*. Springer. 2000, pp. 385–393. URL: `https://link.springer.com/article/10.1007%2Fs00145-004-0312-y`.

[Kob87] Neal Koblitz. “Elliptic curve cryptosystems”. In: *Mathematics of Computation* 48.177 (1987), pp. 203–209. URL: `https://www.ams.org/journals/mcom/1987-48-177/S0025-5718-1987-0866109-5/home.html`.