Key Exchange ~~Cryptosystem~~ Using Elliptic Curves

Public Elements: $E_q(a,b)$, q = prime elliptic curve
a, b = coefficient

G pt on the elliptic curve with a large order (n)

User A: private n_A , $n_A < n$
public $P_A = n_A \times G$.

User B: private n_B $n_B < n$
public $P_B = n_B \times G$.

$K = n_A \times P_B$ (calculated by Alice)
$K = n_B \times P_A$ (calculated by Bob)
$= \boxed{n_A \times n_B \times G}$ $\underbrace{(g^a)^b \text{ or } (g^b)^a}$

Diffie-Hellman Key Exchange

How to send a message

Alice wants to send P_m to Bob.

Alice chooses a random k and sends

$$\text{Ciphertext} = \{kG, P_m + kP_B\}$$

Bob receives this (he knows n_B):

$$\underbrace{P_m + kP_B}_{\text{received}} - n_B \underbrace{(kG)}_{\text{received}} = P_m + \underline{k \cdot n_B \cdot G} \underline{- n_B \cdot k \cdot G}$$

$$= P_m$$

Hash functions

Message Authentication

Digital Signatures

E_K

$M \| (H(m))$ ⟶

Eve → m'

↳ $H(m') == H(m)$

→ Many to ~~One~~ One function

it's possible that $f(x) = f(y)$ but $x \neq y$.

① Output is some fixed number of bits

② Input is arbitrarily long

③ Fast to calculate

To make a GOOD hash function

Given an output value h, it's computationally infeasible to calculate some x such that $f(x) = h$.

Ideally, for any h, ~~p that~~ probability that $f(x) = h$ is $\frac{1}{2^b}$, b = # of bits of output.

---

for any given x, it should be computationally infeasible to find $y \neq x$ such that $f(y) = f(x)$

and $f(x) = f(y)$.

→ MORE Strict requirement!

The function passes all pseudo random tests.

1		11		
2			19	
3				
4	4	6	25	
5		6	15	
6	2	10, 11, 12	26	
7		5	(7)	29
8	1	6	16	
9		14	21	30
10				
11		10, 11	(18)	27
12	7	9		

$$\frac{365}{365} \times \frac{364}{365} \times \frac{363}{365} \times \frac{362}{365} \times \cdots \frac{365-k+1}{365}$$