Digital Signature Schemes

- Schnorr
- ECDSA (Elliptic Curve Digital Signature Algorithm)
- Boneh-Lynn-Shacham (BLS)

Goal: Description of generating
- Public & Private keys
- Signing Messages
- Validating Signatures

One-Way Functions

f() easy to compute but hard to calculate $f^{-1}()$

Assumption 1
- f(x) given, no polynomial time function to compute X

Signatures use the additive property of f()

$f(x+y) = f(x) \oplus f(y)$

$f(x-y) = f(x) \ominus f(y)$

$f(cx) = f(\underbrace{x+\ldots+x}_{c\text{ times}}) = \underbrace{f(x) \oplus \ldots \oplus f(x)}_{c\text{ times}} = cx\, f(x)$

Why do the points using modulo q?
isn't that implied by the fact inputs are in the range [1, q-1]

What does it mean for an operator to be defined along an elliptic curve?

Elliptic Curve Cryptography

$f(x) \rightarrow$ to a point on elliptic curve over finite field F_p

[1, q-1]

q & p are large prime numbers

- Points on elliptic curve are denoted in Capital Letter
- Integers in the range [1, q-1] with lowercase letters

relates to conforming to properties of EC

– Means operator maintains EC properties
- Consistency w EC
- Closure → result is also on the curve
- defined algebraically or geometrically

One-way function implementations on elliptic curves

Finite Set which satisfies equation on elliptic curve

$y^2 \equiv x^3 + ax + b \pmod p$ w $0 \le y < p$ and $0 \le x < p$ are x&y

Digital Signatures

- public & private keys are paired to create "unforgeability" of signatures

- each public key has secret key, one can generate a signature (Primitive Sign) and corresponding key (Primitive Verify Sig)

VerifySig → 1 if valid sig
↳ 0 otherwise

Schnorr Signatures

- takes adv of fact $ax+b$, $a \neq 0$ has a single root
- only one single x such that $ax+b=0$ for $a \neq 0$

So is s the signature value for Schnorrs?
Yes

do all signature generations go through that equation?
Yes

- See operator definition regs above

$s \equiv r + h \cdot x \pmod q$, congruence modulo prime q

only satisfied by ints s, r, h, x from $1, \dots, q-1$

also satisfy

$f(s) = f(r) \oplus hx\, f(x)$

- $f(s), f(r), f(x)$ points on elliptic curve

Schnorr Primitive $\text{Sign}(m, x) = \sigma$

- Public key $X = f(x)$
  - pick random number r, called the nonce

    $R = f(r)$ ↖ point on EC

$H(m \mid R \mid X)$ ↖ concatenate the message with the Point and Hash

So there are certain bits for each m, R, X?
what is the function of them to give the amount?

input to another value so bit placement irrelevant

each are represented in 256 bits

Compute $s \equiv r + h \cdot x \pmod q$

Return $\sigma := (R, s)$

Schnorr Primitive $\underline{\text{VerifySig}}(\sigma, m, X) = \{0, 1\}$

Signature is $(R, s) = \sigma$

Compute $S = f(s)$ from s

Concatenate the message m w R and X & compute its hash $h = H(m \mid R \mid X)$

Return $S \overset{?}{=} R \oplus h \times X$

Schnorr Signature

- Composed of 2 parts

  $R = f(r)$ and s
  - s does not reveal the secret key as long as r is not presented

Instead of Verifying $s \equiv r + h \cdot x \pmod q$

$\boxed{\text{Algo verifies whether the equality } f(s) = f(r) \oplus h \times f(x) \text{ holds}}$

ECOSA Signature

In DSA calculates the nonce & public key in a finite field

- Signature is calc in a cyclic subgroup using suitable operator
- Converts curve point R to int r

- public key : 256 bits
- Secret key : $[1, q-1]$, $q \lesssim 2^{256} \rightarrow$ 256 bits

Operator $|R| \rightarrow [1, q-1]$

$r \equiv h \cdots + R \cdot s \cdot x \pmod q$

$f(r) = f(h \cdot s) \oplus (R \cdot s) \times f(x)$

$R = |f(r)|$

ECDSA $\text{Sign}(m, x) = \sigma$

Hash message $h = H(m)$

pick random r, $R = |f(r)|$ why r^{-1}?

compute $s \equiv (h + R \cdot x) r^{-1} \pmod q$

Return $\sigma := (R, s)$

ECDSA $\text{VerifySig}(\sigma, m, X) = \{0, 1\}$

Extract sig $(R, s) = \sigma$, $h = H(m)$, $R = |R|$

Return $s \times R \stackrel{?}{=} f(h) \oplus R \times H$

BLS Signatures

- In Schnorr and ECDSA, use of nonce can create vulnerability
- Often favored for signature aggregation capabilities
- two homomorphic one-way functions
  $f_1()$, $f_2()$ map to seperate elliptic curves

$f_1() : \oplus$ $\quad$ $f_2() : \oplus$

$G_1 = f_1(1)$ $\quad$ $G_2 = f_2(1)$

BLS Sign

Hash msg. hash hash

$h = H(m)$, $H = \text{SWU}(h)$, $S = xH \rightarrow$ sign hash w private key

equation must hold

$e(H, X) = e(H, xG_2) = e(xH, G_2) = e(S, G_2)$

BLS Primitive VerifySig

$(S) = \sigma$, $h = H(m)$, $H = \text{SWU}(h)$

Return $e(S, G_2) \stackrel{?}{=} e(H, X)$