Cryptanalysis on ECC-based Algorithms

NT219.P22.ANTT

NGUYEN NGOC TU

Our Speakers

Phan Nguyễn Việt Bắc

23520087

Trần Gia Bảo

23520139

Châu Hoàng Phúc

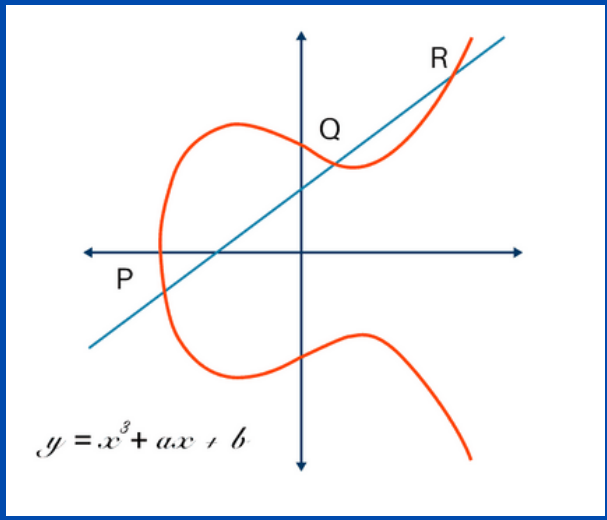
23521191

Table Of Content

ECC quick review

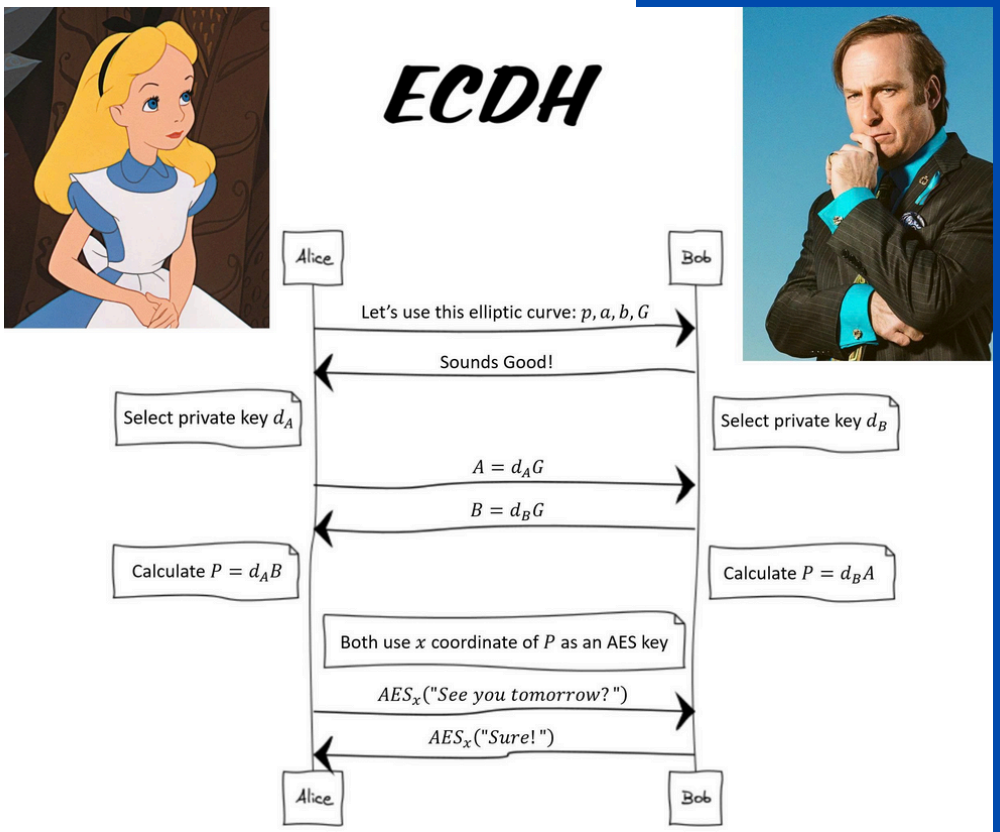
An elliptic curve in **Weierstrass form**

$$y^2 = x^3 + Ax + B, \quad \text{where } 4A^3 + 27B^2 \neq 0$$



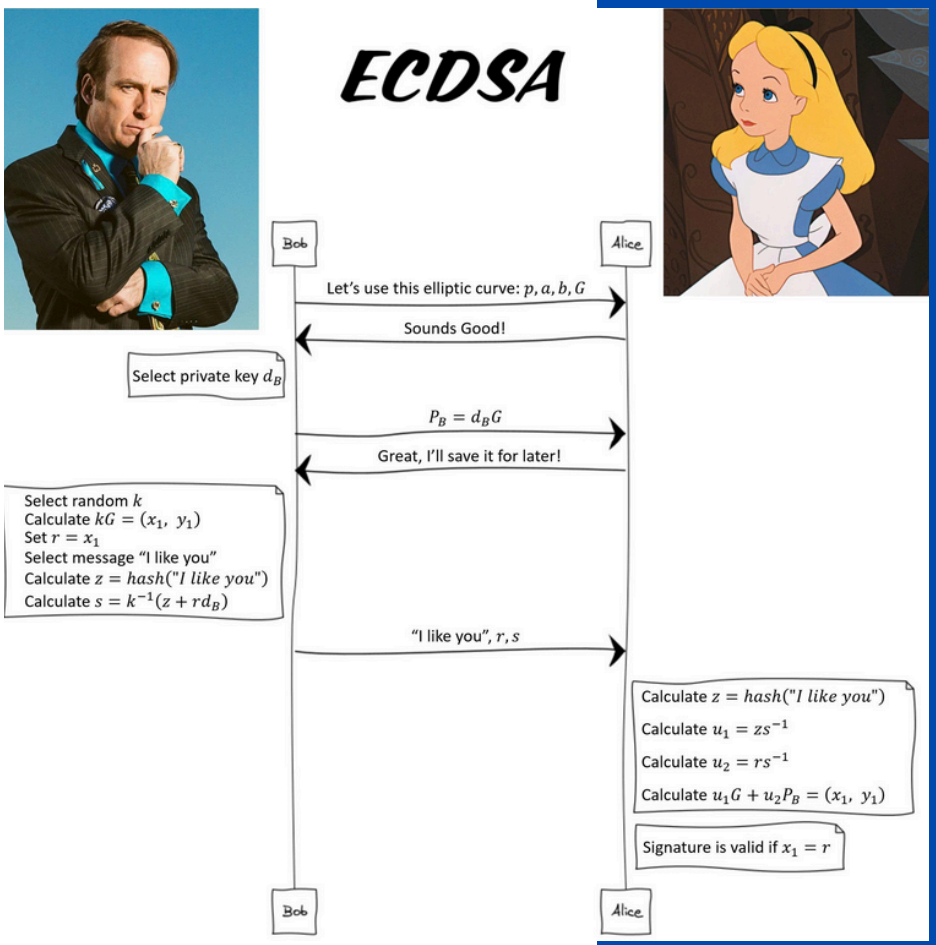
The security foundation of ECC lies in the computational difficulty of the **Elliptic Curve Discrete Logarithm Problem (ECDLP).**

ECDH



ECDH is a variant of the Diffie-Hellman key exchange protocol, allowing two parties to establish a shared secret key over an insecure channel.
The security of ECDH relies on the hardness of the Elliptic Curve Discrete Logarithm Problem (ECDLP).

ECC quick review

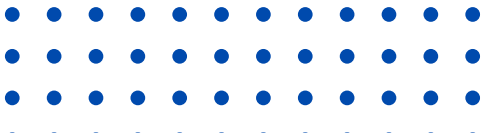


ECDSA is a widely used digital signature algorithm that verifies the authenticity and integrity of messages.
It's a variant of the Digital Signature Algorithm (DSA) that employs elliptic curve operations.

Baby-step Giant-step

The system uses elliptic curves of small orders n

Order of P (n)	$\sqrt{n}$ (BSGS steps)	Memory and computation required
2^{80}	2^{40}	Attackable with supercomputers
2^{160}	2^{80}	Too large for current practical attacks
2^{256} (modern ECC standard)	2^{128}	**Infeasible to attack** using BSGS



Baby-step Giant-step

$$Q = xP$$

Giả sử bạn viết x dưới dạng:

$$x = im + j \Rightarrow Q = xP = imP + jP$$

Chuyển vế:

$$Q - imP = jP$$

For each

Baby-step $j \in [0, m-1]$,

compute:

$$B_j = jP$$

Compute:

Giant-step $S = mP$

This is the "giant step" increment.

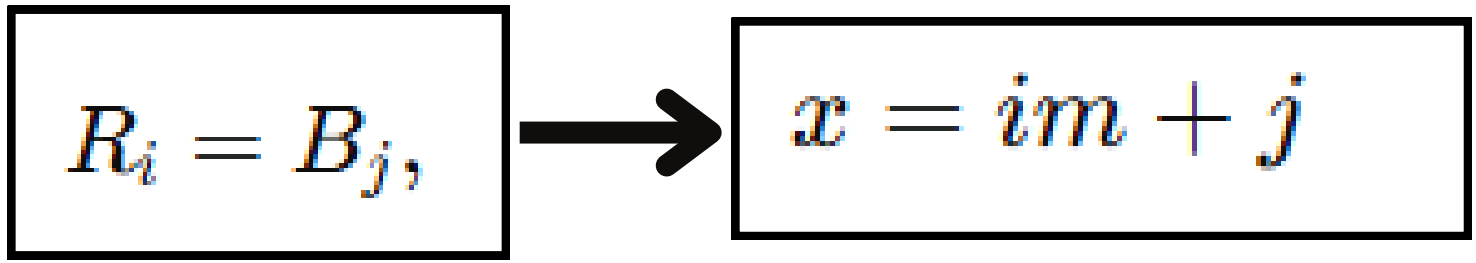
Then, for each

$$i \in [0, m-1],$$

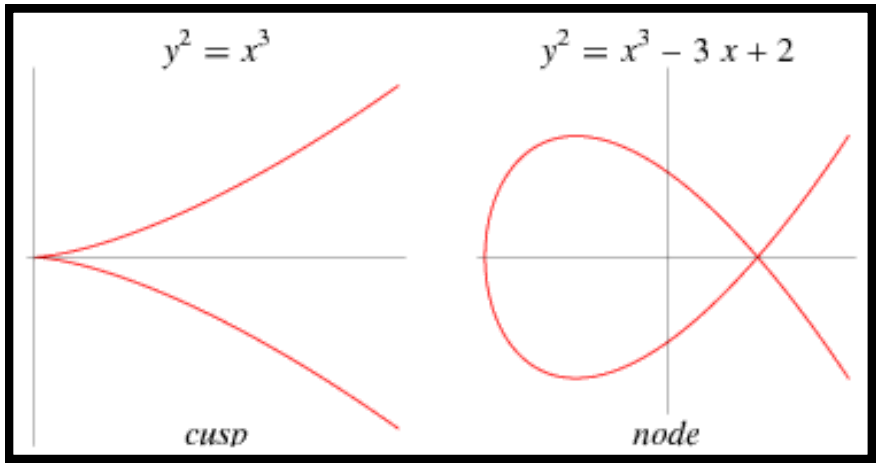
compute:

$$R_i = Q - iS = Q - i(mP)$$

Baby-step Giant-step



Singular Curve



If the discriminant is zero ($\Delta=0$), then **the curve is singular** and **the DLP is easy**.

$$4A^3 + 27B^2 \equiv 0 \pmod{p}$$

There are 2 type of Singular Curve
So there are 2 differents way to attack

Node

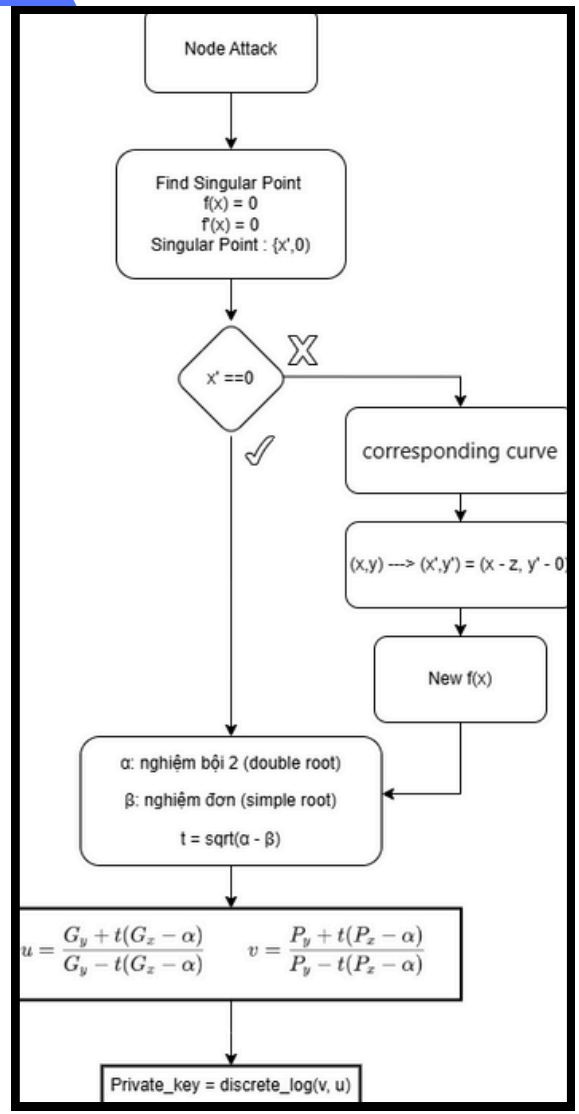
F(x), **Δ=0 and A <> 0, B = 0**

$$y^2 = (x-\alpha)^2(x-\beta)$$

$$t = \sqrt{\alpha - \beta}$$

$$u = \frac{G_y + t(G_x - \alpha)}{G_y - t(G_x - \alpha)}, \quad v = \frac{P_y + t(P_x - \alpha)}{P_y - t(P_x - \alpha)}$$

$$k = \log_u v \mod p$$



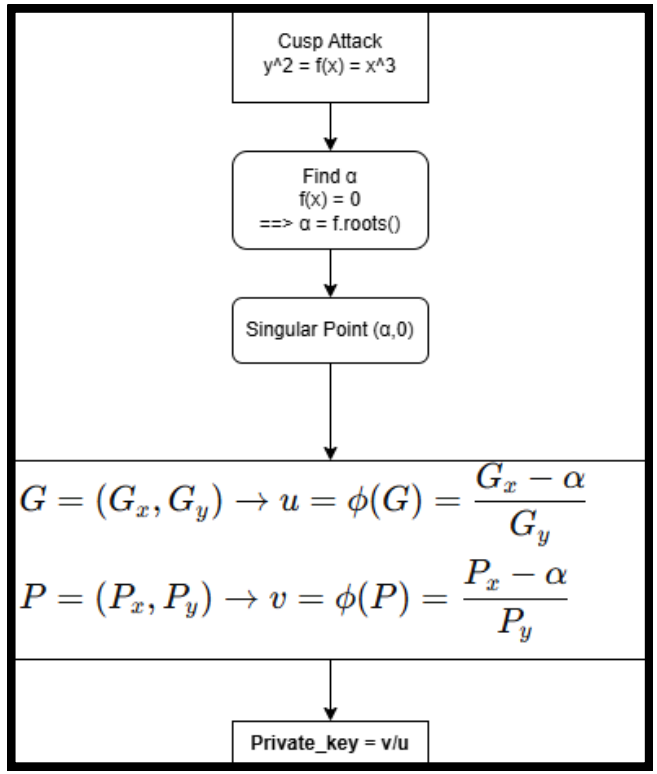
Cusp

F(x), **Δ=0 and A = 0, B = 0**

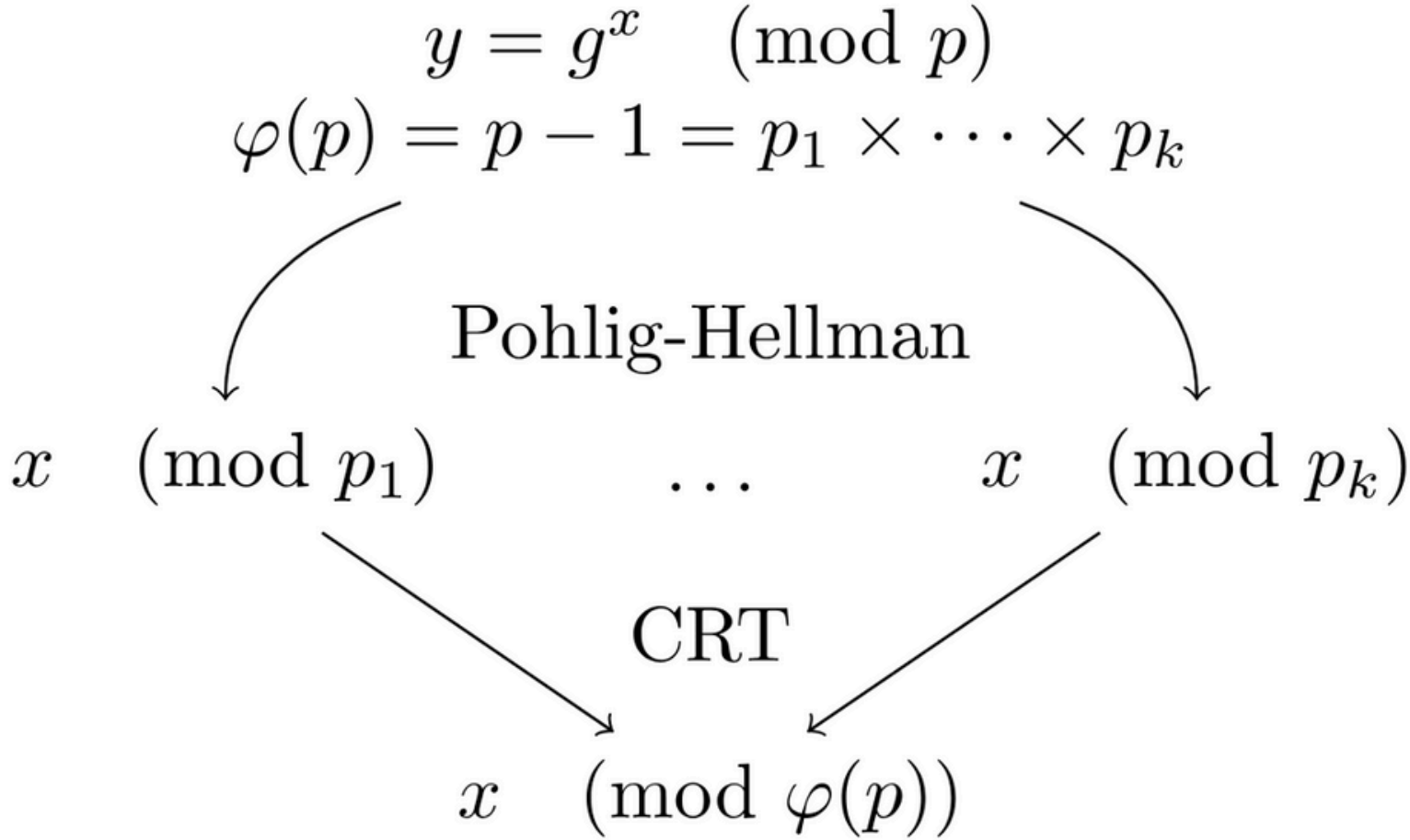
$$y^2 = x^3 \pmod{p}$$

$$y^2 = (x - \alpha)^3$$

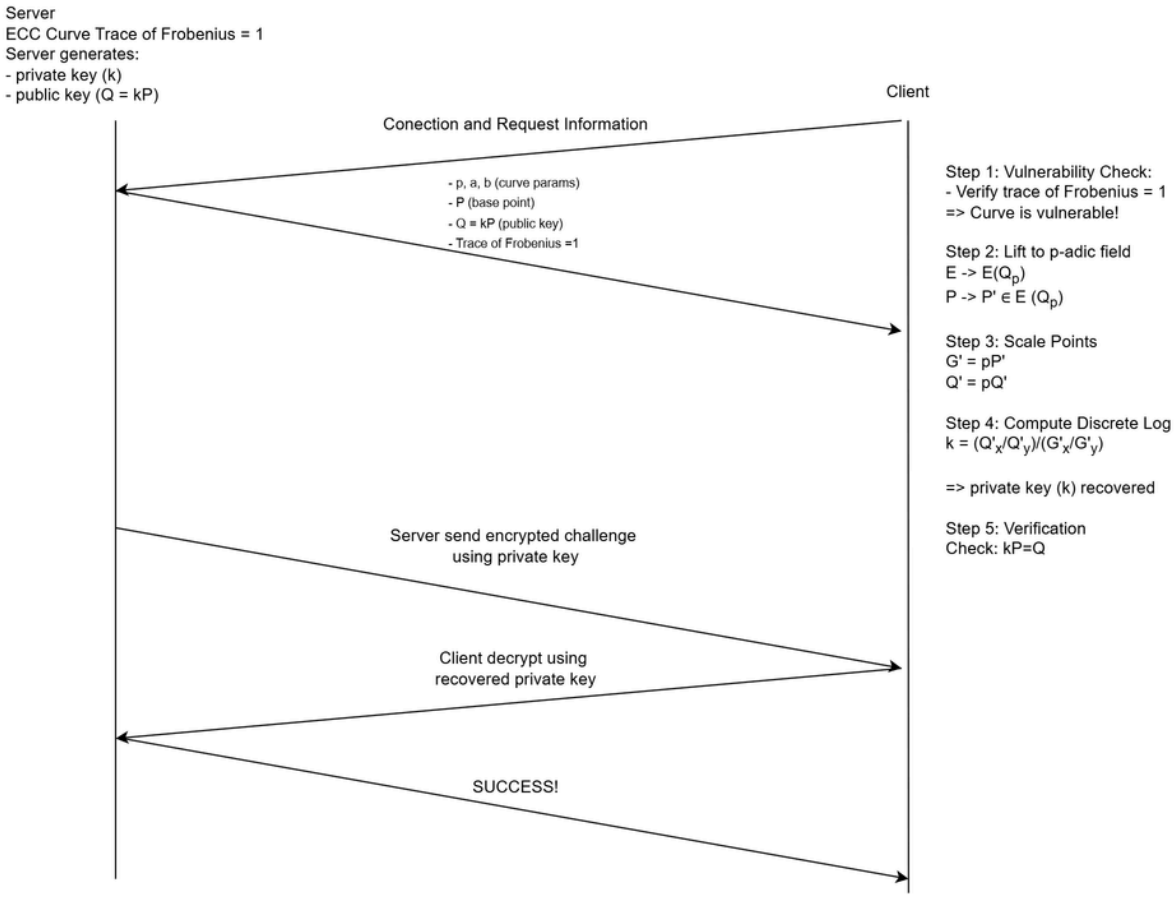
$$\phi(P) = k \cdot \phi(G) \Rightarrow k = \frac{\phi(P)}{\phi(G)} = \frac{x_2 - \alpha}{y_2} \cdot \frac{y_1}{x_1 - \alpha}$$



Pohlig-Hellman



Smart Attack



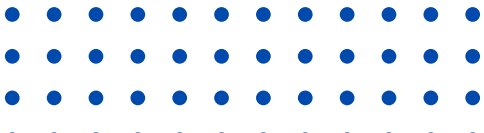
MOV Attack

MOV attack - (1)

- Choose m , such that $E[N] \subset E(F_{q^m})$
  By Corollary 3.11, $\mu_N \subset F_{q^m}$
- MOV attack (Want to solve $Q = kP$ for k.)
  1. Choose a random point $T \in E(F_{q^m})$
  2. Compute $M = ord(T)$
  3. Let $d = \gcd(M, N)$, and let $T_1 = (M/d)T$
     Then $d = ord(T_1), \quad d \mid N$, so $T_1 \in E[N]$
  4. Compute $\zeta_1 = e_N(P, T_1)$ and $\zeta_2 = e_N(Q, T_1)$
     Then both $\zeta_1, \zeta_2 \in \mu_d \subseteq F^*_{q^m}$
  5. Solve discrete logarithm problem

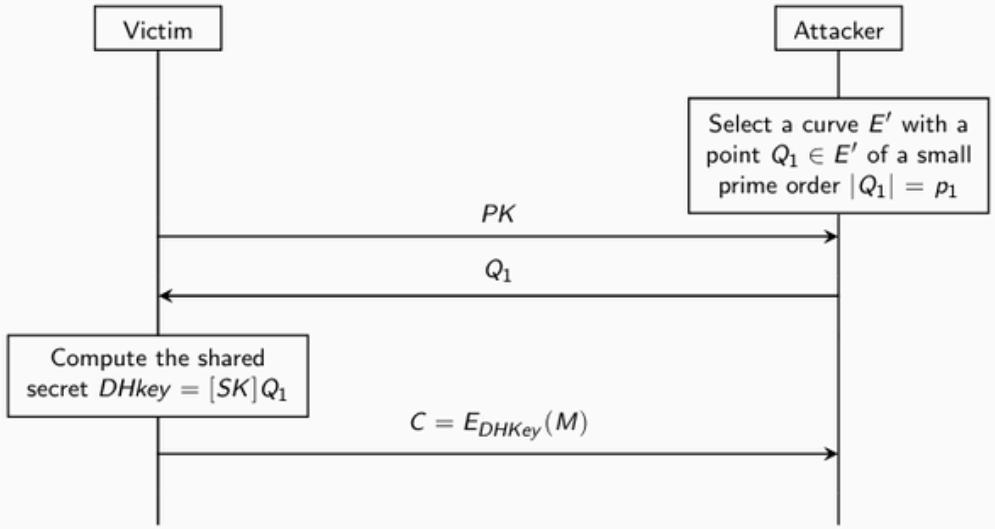
     $$\zeta_2 = \zeta_1^k \quad \text{in } F^*_{q^m}$$

     This will give $k \pmod d$
  6. Repeat with random points T until the lcm of d's is N .
     This determines $k \pmod N$

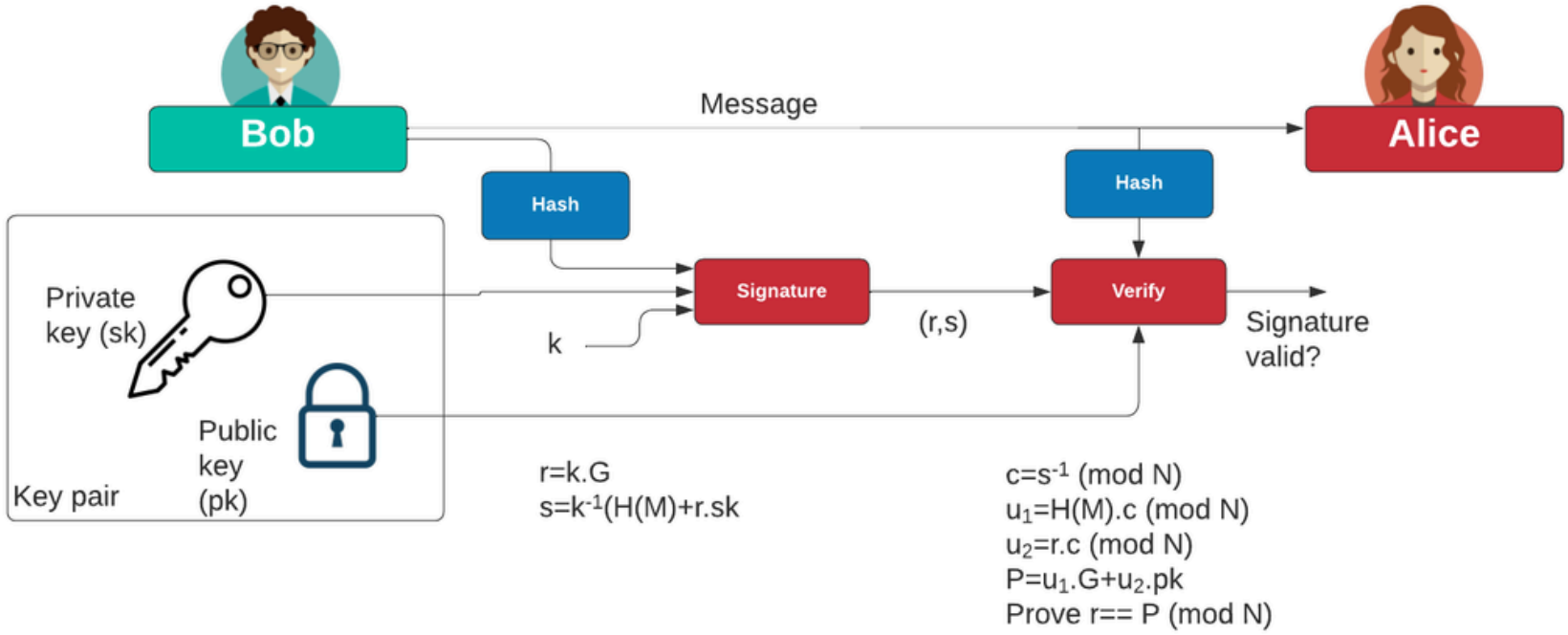


Invalid Curve Attack

- Let SK be the secret key of the victim device and let $PK = [SK]P$ its public key.
- Let E' be a different group defined by the curve equation $y^2 = x^3 + ax + b'$ with the same a and a different b' parameter.



Not Hashing Message



Nonce Reuse Attack

$$s = k^{-1}(z + r \cdot d_A) \pmod{n}$$

REUSE NONCE K

$$\Rightarrow$$

$$s_1 = k^{-1}(z_1 + r_1 \cdot d_A) \pmod{n}$$

$$s_2 = k^{-1}(z_2 + r_2 \cdot d_A) \pmod{n}$$

Nonce Reuse Attack

By rearranging the equations to solve for k^{-1}:

1. $k^{-1} = s_1^{-1}(z_1 + r_1 \cdot d_A) \pmod{n}$
2. $k^{-1} = s_2^{-1}(z_2 + r_2 \cdot d_A) \pmod{n}$

Since k^{-1} is the same in both cases, we can set them equal:

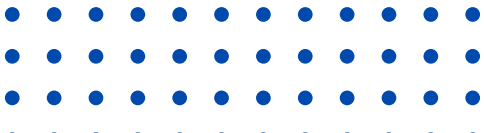
$$s_1^{-1}(z_1 + r_1 \cdot d_A) = s_2^{-1}(z_2 + r_2 \cdot d_A) \pmod{n}$$

This equation can then be rearranged to solve for d_A, the private key:

$$s_2 z_1 + s_2 r_1 d_A = s_1 z_2 + s_1 r_2 d_A \pmod{n}$$
$$s_2 z_1 - s_1 z_2 = s_1 r_2 d_A - s_2 r_1 d_A \pmod{n}$$
$$s_2 z_1 - s_1 z_2 = d_A(s_1 r_2 - s_2 r_1) \pmod{n}$$

Finally, the private key d_A can be found:

$$d_A = (s_2 z_1 - s_1 z_2)(s_1 r_2 - s_2 r_1)^{-1} \pmod{n}$$



Biased Nonces attack

Formulating ECDSA as a hidden number problem

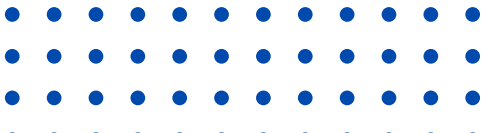
[Howgrave-Graham Smart 2001], [Nguyen Shparlinski 2003]

We have a system of equations in unknowns $k_1, \ldots, k_m, d$:

$$\begin{aligned} k_1 - t_1 d - a_1 &\equiv 0 \bmod n \\ k_2 - t_2 d - a_2 &\equiv 0 \bmod n \\ &\vdots \\ k_m - t_m d - a_m &\equiv 0 \bmod n \end{aligned}$$

We assume the k_i are small.

(Instance of the *hidden number problem* [Boneh Venkatesan 96].)



Biased Nonces attack

Solving the hidden number problem with CVP

Input:

$$k_1 - t_1 d - a_1 \equiv 0 \bmod n$$
$$\vdots$$
$$k_m - t_m d - a_m \equiv 0 \bmod n$$

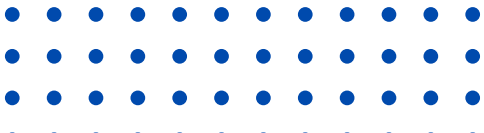
in unknowns $k_1, \ldots, k_m, d$, where $|k_i| < B$.

Construct the lattice basis

$$M = \begin{bmatrix} n & & & \\ & n & & \\ & & \ddots & \\ & & & n \\ t_1 & t_2 & \ldots & t_m \end{bmatrix}$$

Solve CVP with target vector $v_t = (a_1, a_2, \ldots, a_m)$.

$v_k = (k_1, k_2, \ldots, k_m)$ will be the distance.



Biased Nonces attack

Solving the hidden number problem with CVP embedding

Construct the lattice

$$M = \begin{bmatrix} n & & & & & \\ & n & & & & \\ & & \ddots & & & \\ & & & n & & \\ t_1 & t_2 & \dots & t_m & B/n & \\ a_1 & a_2 & \dots & a_m & & B \end{bmatrix}$$

Want vector
$v_k = (k_1, k_2, \dots, k_m, Bd/n, B)$

We have:

- $\dim L = m + 2$ $\qquad$ $\det L = B^2 n^{m-1}$
- Ignoring approximation factors, LLL or BKZ will find a vector

$$|v| \leq (\det L)^{1/\dim L}$$

- We are searching for a vector with length $|v_k| \leq \sqrt{m+2}B$.
- Thus we expect to find v_k when

$$\log B \leq \lfloor \log n(m-1)/m - (\log m)/2 \rfloor$$

If you have any questions, you are welcome to ask

Thank You