Modern Cryptology: from public key cryptography to homomorphic encryption

2015/12 — Yaoundé, Cameroun

Damien Robert

Équipe LFANT, Inria Bordeaux Sud-Ouest
Institut de Mathématiques de Bordeaux
Équipe MACISA, Laboratoire International de Recherche en Informatique et Mathématiques Appliquées

RSA

- Fermat, Euler: if $x \in (\mathbb{Z}/N\mathbb{Z})^*$ then $x^{\varphi(n)} = 1$.
- RSA: $n = pq$. $\varphi(n) = (p-1)(q-1)$.
- If N is a product of disjoint primes, then for all $x \in \mathbb{Z}/N\mathbb{Z}$, $x^{1+\varphi(n)} = x$.

Proof.

If $N = p$, then Fermat shows this work for all $x \neq 0$, and 0 is trivial to check. If $N = \prod p_i$, by the CRT $\mathbb{Z}/N\mathbb{Z} \simeq \prod \mathbb{Z}/p_i\mathbb{Z}$ as a ring and we are back to the prime case. $\square$

- In RSA, if e is prime to $\varphi(n)$ and d is its inverse, then for all $x \in \mathbb{Z}/N\mathbb{Z}$, $x^{ed} = x$.
- Encryption: $x \mapsto x^e$; Decryption: $y \mapsto y^d$.
- Signature: $x \mapsto x^d$; Verification: $y \mapsto y^e$.

Reductions on RSA

Given the public key (N, e)

- RSADP (Decryption Problem): from $y = x^e$ find x;
- RSAKRP (Key Recovery Problem): find d such that $x^{ed} = x$ for all $x \in \mathbb{Z}/N\mathbb{Z}^*$
- RSAEMP (Exponent Multiple Problem): find k such that $x^k = 1$ for all $x \in \mathbb{Z}/N\mathbb{Z}^*$ (so k is a multiple of $(p-1) \vee (q-1)$);
- RSAOP (Order Problem): find $\varphi(n)$;
- RSAFP (Factorisation Problem): recover p and q.

Theorem

RSAKRP $\Leftrightarrow$ *RSAEMP* $\Leftrightarrow$ *RSAFP* $\Leftrightarrow$ *RSAOP* $\Rightarrow$*RSADP*

Proof.

RSAFP $\Rightarrow$RSAOP $\Rightarrow$RSAKRP $\Rightarrow$RSAEMP. The hard part is to show that RSAEMP $\Rightarrow$RSAFP. The goal is to find $x \neq \pm 1$ such that $x^2 = 1$. Then $x - 1 \wedge n$ gives a prime factor. Write $k = 2^s t$, and look for a random y at $x = y^t$, x^2, x^{2^2}, $\ldots x^{2^j}$ until we find 1, say $x^{2^{j_0+1}} = 1$. Then x^{2^j} is a square root. The bad cases are when $x = y^t = 1$ (but this has probability less than $1/4$) and when $x^{2^{j_0}} = -1$ (but this has probability less than $1/2$). □

Malleability of RSA

- $(m_1 \cdot m_2)^e = m_1^e \cdot m_2^e$ so from several ciphertexts we can generate a lot more;
- As is, RSA is OW-CPA (if factorisation is hard) but malleable.
- Example of CCA2 attack: we know $c = m^e$; we ask to decipher a random $r : m_r = r^d$ and $c/r : m_{c/r} = (c/r)^d$ (c/r looks random). We recover $m = m_r m_{c/r}$.
- We want IND-CCA2 so we need to add padding.
- RSA-OAEP: The padding is $M \oplus G(r) \,||\, r \oplus H(M \oplus G(r))$ where r is random and H and G are two hash functions.

Attacks on RSA

- Best algorithm for factorisation is NFS: $2^{O(n^{1/3})}$;
- Subexponential: Factor 2 in security needs factor 8 in key length.
- Small exponent: if $N > m^e$ finding m is easy. This can happen if the same message is sent to several user with public keys (N_i, e); by the CRT we recover $m^e \mod N = \prod N_i$.
- If e has a small order in $(\mathbb{Z}/\varphi(N)\mathbb{Z})^*$ iterating the encryption yields the decryption.
- If d is small, for instance let $p < q < 2p$, and suppose that $d < n^{1/4}/3$. Write $ed - 1 = k\varphi(n)$; then for n big enough

$$|\frac{e}{n} - \frac{k}{d}| < \frac{1}{2d^2}.$$

  k/d can then be recovered from the continued fraction of e/n which is computed using Euclide's algorithm.

Squares in finite fields

- Let $p > 2$ be a prime. $(\mathbb{Z}/p\mathbb{Z}^*, \times)$ is a cyclic group of order $p-1$;
- There are $(p-1)/2$ squares and $(p-1)/2$ non squares;
- If $x \in \mathbb{Z}/p\mathbb{Z}^*$ then x is a square if and only if $x^{\frac{p-1}{2}} = 1$ (by Fermat $x^{p-1} = 1$ for all $x \in \mathbb{Z}/p\mathbb{Z}^*$);
- Legendre symbol:

$$\left(\frac{x}{p}\right) = \begin{cases} 1 & x \text{ is a square} \\ -1 & x \text{ is not a square} \\ 0 & x = 0 \bmod p; \end{cases}$$

- $\left(\frac{x}{p}\right) = x^{\frac{p-1}{2}}$ (mod p);
- Multiplicativity: $\left(\frac{xy}{p}\right) = \left(\frac{x}{p}\right)\left(\frac{x}{q}\right)$;
- Quadratic reciprocity: p, q primes > 2:

$$\left(\frac{p}{q}\right)\left(\frac{q}{p}\right) = (-1)^{\frac{p-1}{2}\frac{q-1}{2}}.$$

Jacobi symbol

- Jacobi symbol: if n is odd, define the Jacobi symbol by extending the Legendre symbol multiplicatively on the bottom argument:

$$\left(\frac{x}{n_1 n_2}\right) = \left(\frac{x}{n_1}\right)\left(\frac{x}{n_2}\right);$$

- Extension of quadratic reciprocity:

$$\left(\frac{m}{n}\right) = (-1)^{\frac{m-1}{2}\frac{n-1}{2}}\left(\frac{n}{m}\right) \quad (m \text{ and } n \text{ odd and coprime})$$

with the extra relations $\left(\frac{-1}{n}\right) = (-1)^{\frac{n-1}{2}}$, $\left(\frac{2}{n}\right) = (-1)^{\frac{n^2-1}{8}}$;

$\Rightarrow$ The Jacobi symbol can be computed in polynomial time;

- Primality test: if $\left(\frac{x}{n}\right) \neq x^{\frac{n-1}{2}}$ then n is not prime (and if n is not prime at least half the x coprime to n will be witnesses).

Digression: Miller-Rabin

Miller-Rabin primality test

- If n is prime and $n-1 = d2^t$, then for all a prime to n either
- $a^d = 1 \mod n$
- or $a^{d2^u} = -1 \mod n$ (for $0 \leqslant u \leqslant t-1$)
- for any odd composite n, at least $3/4$ of the bases a are witnesses for the compositeness of n.

Heads or tails

- Let $n = pq$ be an RSA number, by the CRT $(\mathbb{Z}/n\mathbb{Z}^*, \times) = (\mathbb{Z}/p\mathbb{Z}^* \times \mathbb{Z}/q\mathbb{Z}^*, \times)$;
- $\left(\frac{x}{n}\right) = \left(\frac{x}{p}\right)\left(\frac{x}{q}\right)$ so if x is prime to n, $\left(\frac{x}{n}\right) = 1$ when x is a square modulo n (=square modulo p and square modulo q) or when x is neither a square modulo p and q;
- Computing $\left(\frac{x}{n}\right)$: polynomial time;
- Deciding if x is a real square (and computing the square root) or false square: factorisation of n
- $x \mapsto x^2$ is a one way trapdoor function!

Heads or tails:

- Bob choose $n = pq$ and sends x such that $\left(\frac{x}{n}\right) = 1$;
- Alice answers “real square” or “false square”;
- Bob sends p and q so Alice can verify if she was right or not.

Zero Knowledge identification

- Secret key of Alice: $p, q, s \bmod n = pq$;
- Public key of Alice: $n = pq, \; r = s^2$;

Zero Knowledge identification:

- Alice chooses a random $u \bmod n$, computes $z = u^2$ and sends $t = zr = u^2 s^2$ to Bob;
- Bob either chooses
  - To check z: he asks u to Alice and checks that $z = u^2$;
  - To check t: he asks us to Alice and checks that $t = (us)^2$.
- A liar will either produce a false u or a false t and has $1/2$ chances to be catched, Bob will ask for several rounds (30);
- To always give the correct answer mean that Alice knows the secret s or is very lucky (probability $1/2^{30}$).

Fermat

- We want to get a factor of a composite number n (see primality tests);
- If $n = x^2 - y^2$ then $n = (x-y)(x+y)$;
- More generally if $x^2 = y^2 \mod n$ then $x - y \wedge n$ may be a non trivial factor (Exercice: if $n = pq$ what is the probability to get a non trivial factor?)

Smooth numbers

- n is B-smooth if n can be written as a product of integer $\leqslant B$;
- Canfield-Erdös-Pomerance: The probability that a number $x \leqslant n$ is B-smooth is
$$u^{-u(1+o(1))}$$
where $u = \frac{\log n}{\log B}$ and when $\log n^{\varepsilon} < u < \log n^{1-\varepsilon}$.
- Subexponential functions: $L_x(\alpha,\beta) = \exp(\beta \log^{\alpha} x \log\log^{1-\alpha} x)$;
- The probability for a number of size $L_x(\alpha,\beta)$ to be $L_x(\gamma,\delta)$-smooth is $L_x(\alpha-\gamma, -\beta(\alpha-\gamma)/\mu + o(1))$.
- Example: a number of size $n = L_n(1)$ is $L_n(1/2)$ smooth with probability $L_n(1/2)$;

Linear and Quadratic Sieves

- Dixon Linear Sieve: Generate squares modulo n: $y = x^2 \mod n$ where y is B-smooth with $B = L_n(1/2) \Rightarrow$ time $L_n(1/2)$ to find them;
- Collect enough relations to use linear algebra so that a suitable product of y is a square;
- Pomerance Quadratic Sieve: let $m = \lceil n^{1/2} \rceil$. Generate the y by $(m+a)^2 = (m^2 - n) + a^2 + 2am \mod n$. The y are of size $\sqrt{n}$ rather than n so the probability to be B-smooth is much higher;
- A detailed complexity analysis give a complexity of $L_n(1/2, \sqrt{2})$ ($B = L_n(1/2, 1/\sqrt{2})$) for the linear sieve and $L_n(1/2, 1)$ ($B = L_n(1/2, 1/2)$) for the quadratic field.

General Number field sieve

- Invented by Pollard and Lenstra;
- Generate smooth numbers in two number fields to get relations (see commutative diagram);
- Linear algebra on the relations to get two squares;
- Use sieves (lattice sieving or line sieving) to generate the smooth numbers;
- In practice very complex (obstructions from the class group and the group of unity, taking square roots in number fields)...
- Heuristic Complexity $L_n(1/3,(64/9)^{1/3})$;
- See for example CADO-NFS for an open-source implementation.

Discrete Logarithm

Definition (DLP)

Let $G = \langle g \rangle$ be a cyclic group of prime order. Let $x \in \mathbb{N}$ and $h = g^x$. The discrete logarithm $\log_g(h)$ is x.

- Exponentiation: $O(\log p)$. DLP: $\widetilde{O}(\sqrt{p})$ (in a generic group). So we can use the DLP for public key cryptography.

$\Rightarrow$ We want to find secure groups with efficient addition law and compact representation.

Discrete logarithm problem

Given a cyclic group $G = \langle g \rangle$.

- Exponentiation $x \mapsto h = g^x$ (via fast exponentiation algorithm); DLP $h = g^x \mapsto x$.
- Shanks: the DLP in G can be done in time $n = \sqrt{\#G}$ via the Baby Steps, Giant Steps algorithm (time/memory tradeoff). Let $c = \sqrt{N}$ and write $x = y + cz$, $y, z \leqslant c$. Compute the intersection of $\{1, g, \ldots, g^c\}$ and $\{hg^{-c}, hg^{-2c}, \ldots, hg^{-cc}\}$ to find $g^z = hg^{-cy}$.
- Pollard: take a random path of $s_i = g^{u_i} h^{v_i}$ (typically find a a suitable function and compute $s_{i+1} = f(s_i)$) until a collision is found: $s_i = s_j$. Then $h = g^{\frac{u_i - u_j}{v_i - v_j}}$. Birthday paradox: a collision is found in time $\sqrt{n}$.
- Pohlig-Helman: the DLP inside G can be reduced to the DLP inside subroups of side $p_i \mid n$.
  - First reduction: CRT. $\mathbb{Z}/N\mathbb{Z} = \prod \mathbb{Z}/p_i^{e_i}\mathbb{Z}$, so to recover x we need to recover $x_i = x \mod p_i^{e_i}$; via $h_i = g_i^{x_i}$ where $h_i = h^{N/p_i^{e_i}}$, $g_i = g^{N/p_i^{e_i}}$.
  - Second reduction: Hensel lift. Write $x_i = x_0 + x_1 p$; and solve $h_i^{p^{e_i - 1}} = g_i^{p^{e_i-1} x_0}$ to recover x_0; write $x_i - x_0 = p(x_1 + p x_2)$ and find x_1 and so on.

Security of the DLP

Theorem

On a generic group, the complexity of the DLP is of complexity the square root of its largest prime divisor.

- But effective groups are not generic!
- $G = (\mathbb{Z}/N\mathbb{Z}, +)$, the DLP is trivial (Euclide algorithm);
- $G = (\mathbb{Z}/p\mathbb{Z})^*$, same methods and subexponential complexity as for factorisation: $2^{O(n^{1/3})}$;
- $G = \mathbb{F}_{2^n}^*$, quasi polynomial algorithm: $n^{\log n}$;
- Generic ordinary elliptic curve over $\mathbb{F}_p$: the generic algorithm is the best available;

$\Rightarrow$ To get 128 bits of security find an elliptic curve $E/\mathbb{F}_p$ where p has 256 bits and $E(\mathbb{F}_p)$ is prime (or almost prime).

Diffie-Helman Key Exchange

- How to share a secret key across a non confidential channel?

⇒ Encrypt it via an asymmetric scheme;

- Or use the Diffie-Helman Key Exchange algorithm (predates asymmetric cryptography).

- Alice sends g^a to Bob
- Bob sends g^b to Alice
- The secret key is g^{ab}.
- Diffie-Helman Problem: Eve has to recover g^{ab} from only g, g^a and g^b.
- DLP ⇒DHP

El Gamal encryption

- Public key: $(g, p = g^a)$, Private key: a;
- Encryption: $m \mapsto (g^k, s = p^k.m)$ (k random);
- Decryption: $m = s/(g^k)^a$.
- Warning: Never reuse k.

DSA (Signature)

- Public key: $(g, p = g^a)$, Private key: a;
- $\Phi : G \to \mathbb{Z}/n\mathbb{Z}$;
- Signature: $m \mapsto (u = \Phi(g^k), v = (m + a\Phi(g^k))/k) \in (\mathbb{Z}/n\mathbb{Z})^2$;
- Verification: $u = \Phi(g^{mv^{-1}} p^{uv^{-1}})$.

Zero Knowledge

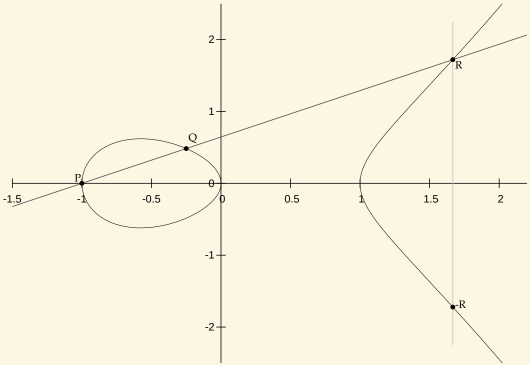
- Alice publish $(g, p = g^a)$, her secret is a.
- Alice choose a random x and sends $q = g^x$;
- Either Bob asks for x and checks that $q = g^x$;
- Either Bob asks for $a + x$ and checks that $q \cdot p = g^{a+x}$.

Elliptic curves

Definition (char $k \neq 2,3$)

An elliptic curve is a plane curve with equation

$$y^2 = x^3 + ax + b \qquad 4a^3 + 27b^2 \neq 0.$$



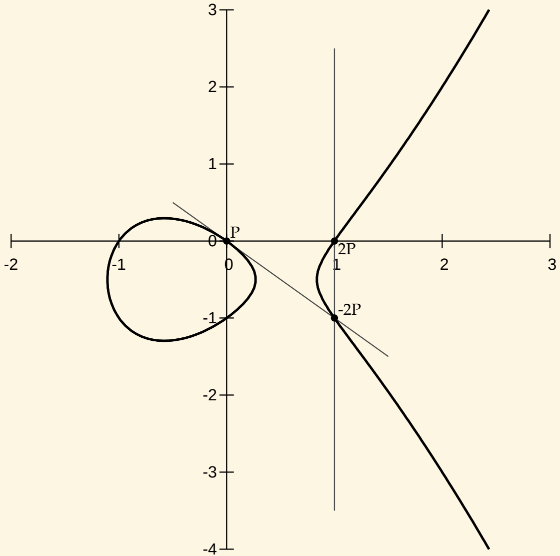
Exponentiation:

$$(\ell, P) \mapsto \ell P$$

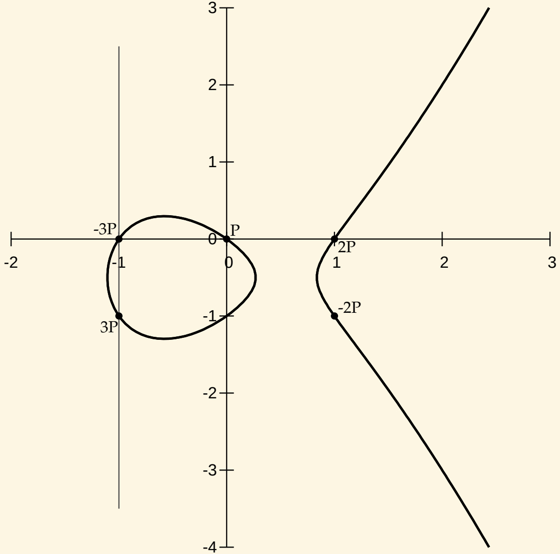
Discrete logarithm:

$$(P, \ell P) \mapsto \ell$$

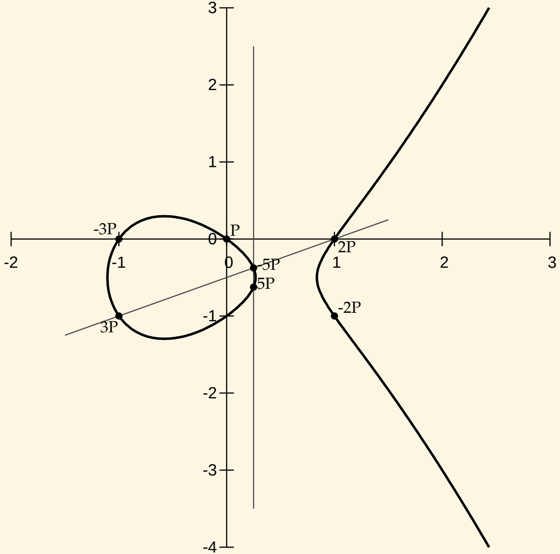
Scalar multiplication on an elliptic curve



Scalar multiplication on an elliptic curve



Scalar multiplication on an elliptic curve



ECC (Elliptic curve cryptography)

Example (NIST-p-256)

- E elliptic curve $y^2 = x^3 - 3x + 41058363725152142129326129780047268409114441015993725554835256314039467401291$ over $\mathbb{F}_{115792089210356248762697446949407573530086143415290314195533631308867097853951}$
- Public key:

$$P = (48439561293906451759052585252797914202762949526041747995844080717082404635286, 36134250956749795798585127919587881956611106672985015071877198253568414405109),$$
$$Q = (76028141830806192577282777898750452406210805147329580134802140726480409897389, 85583728422624684878257214555223946135008937421540868848199576276874939903729)$$

- Private key: ℓ such that $Q = \ell P$.

- Used by the NSA;
- Used in Europeans biometric passports.

ECC vs RSA for 128 bits of security

- ECC (Curve25519) 256 bits:

AAAAC3NzaC1lZDI1NTE5AAAAIMoNrNYhU7CY1Xs6v4Nm1V6oRHs/FEE8P+XaZ0PcxPzz

- RSA 3248 bits:

MIIHRgIBAAKCAZcAvlGW+b5L2tmqb5bUJMrfLHgr2jga/Q/8IJ5QJqeSsB7xLVT/
ODN3KNSPxyjaHmDNdDTwgsikZvPYeyZWWFLP0B0vgwDqQugUGHVfg4c73ZolqZk6
1nA45XZGHUPt98p4+ghPag5JyvAVsf1cF/VlttBHbu/noyIAC4F3tHP81nn+lOnB
eilEALbdmvGTTZ5jcRrt4IDT5a4IeI9yTe0aVdTsUJ6990hpKrVzyTOu1eoxp5eV
KQ7aIX6es9Xjnr8widZunM8rqhBW9EMmLqabnXZItPQoV3rUAnwKzDLV7E56viJk
S2xU5+95IctYu/RTTbf3wTxnkDOqxId0MONHyBJsukXgYKxVB1fWhBKZ4tWui1gw
UCIiKTqLml2zJhLn4WovaxrvvTx0082S0xncEfYDXYu4xbRnJn+ZsTTguqufwC1M
U4MYRdWy7uj+H1EmIGul69Fw9NkuCitWI9dFpcDtSP+/1eEN7wc2FlxhDIRwer0F
6I1P4StWn1uQyHzsTLVdcP+rqA1AsvbWBCKL4ravEO2CEQIDAQABAoIBllWt5YoJ
YZzk4RXbkSX/LvmWICfdmkjTKW6F1w+P4TnotCr0WPG0ObDoANJoUcnbSqNGMgCu
01SF8q9+UuDwZx4KBZm0j8IPOPzJ2nYcK5dYDhyMHzDq1LJ4zJfgPQGQ5WWq2BWm
2RHDhADdTth6YZArs/z9hAqtA9gqMPnMPcdQpIvlsHSOn06zBJD8sJQA+kOxG+Y2
GS8NakLcUVlDpNd/Q+QHkv4AW1ge2EF8QvmKtU/9rekOBqWNm2Tapd6RtAhZwPJX
UhD9yiesTF6rjZ1ZcMGXUaN5Rt0zD3D4zowRz2JLtCe4GkiJmtc3waN6hu1IaIqz
boI11evqnbatqnC4rCq8sf21yZqaLUIbwH4lW2G3K8xMJNh3iy8cgHTYneNYa+/d
7xyNWlMO9SKlHsyaPcWv98BdD+At0x/6R6YPYkeR+qXJ9ETGFKW4U6iNbBQXOMbh
kZb1Ry8vfMH8vsYIzh8Edg6aq00ScU57KiDS/Gc8KuqI6vmf2leCdCa487kVCgw6
cGXQ2bLZGYBiMZFfOOlpCQECgcwA5ZUh3/8yS0duNhsDz3sgC2u40HwHUbxuSOUa
a5t4CoUY9iuF7b7qhBEcvdLgIOiXA5xo+r4p0xgbLvDUTsRR1mrDM2+wRcjjwXcW
pFaMFRl2Rr72yLUC7N0WNcoUshrNL4X/1j8T4WLRcannpXcor+/kn1rwdLEbRCC+
zRTAdJlgMPt4kwJeHtE9Mzw2/O3GX3MeLvzvJklzvpCGw20N/2Yqjs++V5hXoHPs
21y6y6/FV097dvFctf7NahS04JsjubfnjOMx89AUNZsCgcwA1DfabCGJSCkmQ+mg
2q9lDPJz6r29wmBtYyT20oZ2kd4QBHrOp0t59yG4bvdRqcZG/Dr5LjuVDWMPyetV
dksK7hVYQz2B7Nzy7W3waPVrhA0N4fqbIFGxih5QiSFG7/oroZ8PdZDcfVRKroh1
/JJ7rIz/ZBQCLRS5t7/G2B0kBDOMMM+02wR60CTmxUhmgvsoDZWRp5KKha5PSvZa
WAu2CN3mXNK72RLF3RFUvuhNYnkOEj5Oau1RaGgpZoB0JTKYI9nffbe8up+DV8MC
gcwA18be28Ti5FXyg+/IGQ3EBHfucCTiTDQqA2Ew/8pTfK+z0kr9yYISsKXUuaSk
+skghkhPcrugW8LgabH4GT/zGu+lH4btyekSBxeCtFqTtpED1WJOWD2ozi7NXSjd
YrhF+VCcMCWA7ekOqSHjkmT4XMO/wPab4VFEKzgLnHzQlcZB3ke7/4/OHnDScIE7
vWVNeRCdYdRggT+wBX+Y6bxp142Smj8uyu1oDmpmR5ZUCnTdqT4O8K/RT0x4jCeC
CUhGv5rVillO7bS4CdkCgctXvnQwCzmwvVrV744TfTuhu8lTwHnqGWaA/LKU3wW9
T/x9ba1uHFXkaWvRba61LIcDGPsYM4hwTYokqYnfbC2rvOWOf6rtnXlP1An3y6lV
ovQfgDeNiFmIyvnviPPEm0JZA+QnburLYwOx4DgwYvyBnpal8WPo8c3L/J4hkwLm

Addition law on the Weierstrass model

$E : y^2 = x^3 + ax + b$ (short Weierstrass form).

- Distinct points P and Q:

$$P + Q = -R = (x_R, -y_R)$$

$$\alpha = \frac{y_Q - y_P}{x_Q - x_P}$$

$$x_R = \alpha^2 - x_P - x_Q \qquad y_R = y_P + \alpha(x_R - x_P)$$

  (If $x_P = x_Q$ then $P = -Q$ and $P + Q = 0_E$).

- If $P = Q$, then α comes from the tangent at P:

$$\alpha = \frac{3x_P^2 + b}{2y_P}$$

$$x_R = \alpha^2 - 2x_P \qquad y_R = y_P + \alpha(x_R - x_P)$$

- Indeed write $l_{P,Q} : y = \alpha x + \beta$ the line between P and Q (or the tangent to E at P when $P = Q$). Then $y_{-R} = \alpha x_{-R} + \beta$ and $y_P = \alpha x_P + \beta$ so $y_{-R} = \alpha(x_R - x_P) + y_P$. Furthermore x_R, x_P, x_Q are the three roots of $x^3 + ax + b - (\alpha x + \beta)^2$ so $x_P + x_Q + x_R = \alpha^2$.

$\Rightarrow$ Avoid divisions by working with projective coordinates $(X : Y : Z)$:

$$E : Y^2 Z = X^3 + aXZ^2 + bZ^3.$$

Scalar multiplication

- The scalar multiplication $P \mapsto n.P$ is computed via the standard double and add algorithm;
- On average $\log n$ doubling and $1/2 \log n$ additions;
- Standard tricks to speed-up include NAF form, windowing ...
- The multiscalar multiplication $(P, Q) \mapsto n.P + m.Q$ can also be computed via doubling and the addition of P, Q or $P+Q$ according to the bits of n and m;
- On average $\log N$ doubling and $3/4 \log N$ additions where $N = \max(n, m)$;
- GLV idea: if there exists an efficiently computable endomorphism α such that $\alpha(P) = u.P$ where $u \approx \sqrt{n}$, then replace the scalar multiplication $n.P$ by the multiscalar multiplication $n_1 P + n_2 \alpha(P)$;
- One can expect n_1 and n_2 to be half the size of $n \Rightarrow$ from $\log n$ doubling and $1/2 \log n$ additions to $1/2 \log n$ doubling and $3/8 \log n$ additions.

Edwards curves

$E : x^2 + y^2 = 1 + d x^2 y^2$, $d \neq 0, -1$.

- Addition of $P = (x_1, y_1)$ and $Q = (x_2, y_2)$:

$$P + Q = \left(\frac{x_1 y_2 + x_2 y_1}{1 + d x_1 x_2 y_1 y_2}, \frac{y_1 y_2 - x_1 x_2}{1 - d x_1 x_2 y_1 y_2} \right)$$

- When $d = 0$ we get a circle (a curve of genus 0) and we find back the addition law on the circle coming from the sine and cosine laws;
- Neutral element: $(0,1)$; $-(x, y) = (x, y)$; $T = (1, 0)$ has order 4, $2T = (0, 1)$.
- If d is not a square in K, then there are no exceptional points: the denominators are always nonzero ⇒ complete addition laws;

⇒ Very useful to prevent some Side Channel Attacks.

Twisted Edwards curves

- $E : a x^2 + y^2 = 1 + d x^2 y^2$;
- Extensively studied by Bernstein and Lange;
- Addition of $P = (x_1, y_1)$ and $Q = (x_2, y_2)$:

$$P + Q = \left(\frac{x_1 y_2 + x_2 y_1}{1 + d x_1 x_2 y_1 y_2}, \frac{y_1 y_2 - a x_1 x_2}{1 - d x_1 x_2 y_1 y_2} \right)$$

- Neutral element: $(0, 1)$; $-(x, y) = (x, y)$; $T = (0, -1)$ has order 2;
- Complete addition if a is a square and d not a square.

Montgomery

- $E : By^2 = x^3 + Ax^2 + x$;
- Birationally equivalent to twisted Edwards curves;
- The map $E \to \mathbb{A}^1, (x, y) \mapsto (x)$ maps E to the Kummer line $K_E = E/\pm 1$;
- We represent a point $\pm P \in K_E$ by the projective coordinates $(X : Z)$ where $x = X/Z$;
- Differential addition: Given $\pm P_1 = (X_1 : Z_1)$, $\pm P_2 = (X_2 : Z_2)$ and $\pm(P_1 - P_2) = (X_3 : Z_3)$; then one can compute $\pm(P_1 + P_2) = (X_4 : Z_4)$ by

$$X_4 = Z_3 \left((X_1 - Z_1)(X_2 + Z_2) + (X_1 + Z_1)(X_2 - Z_2)\right)^2$$
$$Z_4 = X_3 \left((X_1 - Z_1)(X_2 + Z_2) - (X_1 + Z_1)(X_2 - Z_2)\right)^2$$

Montgomery's scalar multiplication

- The scalar multiplication $\pm P \mapsto \pm n.P$ can be computed through differential additions if we can construct a differential chain;
- If $\pm[n]P = (X_n - Z_n)$, then

$$X_{m+n} = Z_{m-n}\left((X_m - Z_m)(X_n + Z_n) + (X_m + Z_m)(X_n - Z_n)\right)^2$$
$$Z_{m+n} = X_{m-n}\left((X_m - Z_m)(X_n + Z_n) - (X_m + Z_m)(X_n - Z_n)\right)^2$$

- Montgomery's ladder use the chain nP, $(n+1)P$;
- From $nP, (n+1)P$ the next iteration computes $2nP$, $(2n+1)P$ or $(2n+1)P$, $(2n+2)P$ via one doubling and one differential addition.

Side channel resistant scalar multiplication

- Start with $T_0 = 0_E$ and $T_1 = P$. At each step do
  - If $k_i = 1$, $T_0 = T_0 + T_1$, $T_1 = 2T_1$
  - Else $T_1 = T_0 + T_1$, $T_0 = 2T_0$
- Constant time execution, but vulnerable to branch prediction attacks. Remove the branch:

$$T_{1-k_i} = T_0 + T_1, \quad T_{k_i} = 2T_{k_i}$$

- The memory access pattern depend on the secret bit $k_i \Rightarrow$ vulnerable to cache attacks. Use bit masking to mask the memory access pattern:
  - $M = (k_i \dots k_i)_2$ the bitmask
  - $R = T_0 + T_1$, $S = 2\left((\overline{M} \& T_0) \,|\, (M \& T_1)\right)$
  - $T_0 = (\overline{M} \& S) \,|\, (M \& R)$
  - $T_1 = (\overline{M} \& R) \,|\, (M \& S)$

Pairing-based cryptography

Definition

A pairing is a non-degenerate bilinear application $e : G_1 \times G_1 \rightarrow G_2$ between finite abelian groups.

Example

- If the pairing e can be computed easily, the difficulty of the DLP in G_1 reduces to the difficulty of the DLP in G_2.

$\Rightarrow$ MOV attacks on supersingular elliptic curves.

- Identity-based cryptography [BF03].
- Short signature [BLS04].
- One way tripartite Diffie–Hellman [Jou04].
- Self-blindable credential certificates [Ver01].
- Attribute based cryptography [SW05].
- Broadcast encryption [GPS+06].

Example of applications

Tripartite Diffie–Helman

Alice sends g^a, Bob sends g^b, Charlie sends g^c. The common key is

$$e(g,g)^{abc} = e(g^b, g^c)^a = e(g^c, g^a)^b = e(g^a, g^b)^c \in G_2.$$

Example (Identity-based cryptography)

- Master key: (P, sP), s. $\quad s \in \mathbb{N}, P \in G_1$.
- Derived key: Q, sQ. $\quad Q \in G_1$.
- Encryption, $m \in G_2$: $m' = m \oplus e(Q, sP)^r$, rP. $\quad r \in \mathbb{N}$.
- Decryption: $m = m' \oplus e(sQ, rP)$.

Divisors

- Let C be a projective smooth and geometrically connected curve;
- A divisor D is a formal finite sum of points on C: $D = n_1[P_1] + n_2[P_2] + \cdots n_e[P_e]$. The degree $\deg D = \sum n_i$.
- If $f \in k(C)$ is a rational function, then

$$\operatorname{Div} f = \sum_P \operatorname{ord}_P(f)[P]$$

$((O_C)_P$ the stalk of functions defined around P is a discrete valuation ring since C is smooth and $\operatorname{ord}_P(f)$ is the corresponding valuation of f at P).

Example

If $C = \mathbb{P}^1_k$ then $\operatorname{Div} \frac{\prod (X-\alpha_i^{e_i})}{\prod (X-\beta_i^{f_i})} = \sum e_i[\alpha_i] - \sum f_i[\beta_i] + (\sum \beta_i - \sum \alpha_i)\infty$. In particular $\deg \operatorname{Div} f = 0$ and conversely any degree 0 divisor comes from a rational function.

Linear equivalence class of divisors

- For a general curve, if $f \in k(C)$, $\mathrm{Div}(f)$ is of degree 0 but not any degree 0 divisor D comes from a function f;
- A divisor which comes from a rational function is called a principal divisor. Two divisors D_1 and D_2 are said to be linearly equivalent if they differ by a principal divisor: $D_1 = D_2 + \mathrm{Div}(f)$.
- $\mathrm{Pic}\, C = \mathrm{Div}^0\, C / \text{Principal Divisors}$
- A principal divisor D determines f such that $D = \mathrm{Div}\, f$ up to a multiplicative constant (since the only globally regular functions are the constants).

Divisors on elliptic curves

Theorem

Let $D = \sum n_i [P_i]$ be a divisor of degree 0 on an elliptic curve E. Then D is the divisor of a function $f \in \overline{k}(E)$ (ie D is a principal divisor) if and only if $\sum n_i P_i = 0_E \in E(\overline{k})$ (where the last sum is not formal but comes from the addition on the elliptic curve).
In particular $P \in E(\overline{k}) \to [P] - [0_E] \in \mathrm{Jac}(E)$ is a group isomorphism between the points in E and the linear equivalence classes of divisors;

The Weil pairing on elliptic curves

- Let $E: y^2 = x^3 + ax + b$ be an elliptic curve over a field k (char $k \neq 2, 3$, $4a^3 + 27b^2 \neq 0$.)
- Let $P, Q \in E[\ell]$ be points of ℓ-torsion.
- Let f_P be a function associated to the principal divisor $\ell(P) - \ell(0)$, and f_Q to $\ell(Q) - \ell(0)$. We define:

$$e_{W,\ell}(P,Q) = \frac{f_P((Q)-(0))}{f_Q((P)-(0))}.$$

- The application $e_{W,\ell}: E[\ell] \times E[\ell] \to \mu_\ell(\overline{k})$ is a non degenerate pairing: the Weil pairing.

Definition (Embedding degree)

The embedding degree d is the smallest number such that $\ell \mid q^d - 1$; $\mathbb{F}_{q^d}$ is then the smallest extension containing $\mu_\ell(\overline{k})$.

The Tate pairing on elliptic curves over $\mathbb{F}_q$

Definition

The Tate pairing is a non degenerate bilinear application given by

$$\begin{array}{rcll} e_T : E_0[\ell] \times E(\mathbb{F}_q)/\ell E(\mathbb{F}_q) & \longrightarrow & \mathbb{F}_{q^d}^* / {\mathbb{F}_{q^d}^*}^{\ell} & . \\ (P,Q) & \longmapsto & f_P((Q)-(0)) & \end{array}$$

where

$$E_0[\ell] = \{P \in E[\ell](\mathbb{F}_{q^d}) \mid \pi(P) = [q]P\}.$$

- On $\mathbb{F}_{q^d}$, the Tate pairing is a non degenerate pairing

$$e_T : E[\ell](\mathbb{F}_{q^d}) \times E(\mathbb{F}_{q^d})/\ell E(\mathbb{F}_{q^d}) \rightarrow \mathbb{F}_{q^d}^* / {\mathbb{F}_{q^d}^*}^{\ell} \simeq \mu_\ell;$$

- If $\ell^2 \nmid E(\mathbb{F}_{q^d})$ then $E(\mathbb{F}_{q^d})/\ell E(\mathbb{F}_{q^d}) \simeq E[\ell](\mathbb{F}_{q^d})$;
- We normalise the Tate pairing by going to the power of $(q^d - 1)/\ell$.

Miller's functions

- We need to compute the functions f_P and f_Q. More generally, we define the Miller's functions:

Definition

Let $\lambda \in \mathbb{N}$ and $X \in E[\ell]$, we define $f_{\lambda,X} \in k(E)$ to be a function thus that:

$$(f_{\lambda,X}) = \lambda(X) - ([\lambda]X) - (\lambda - 1)(0).$$

- We want to compute (for instance) $f_{\ell,P}((Q) - (0))$.

Miller's algorithm

- The key idea in Miller's algorithm is that

$$f_{\lambda+\mu,X} = f_{\lambda,X} f_{\mu,X} \mathfrak{f}_{\lambda,\mu,X}$$

where $\mathfrak{f}_{\lambda,\mu,X}$ is a function associated to the divisor

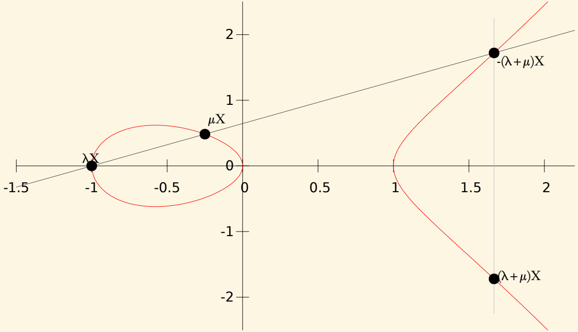
$$([\lambda]X) + ([\mu]X) - ([\lambda+\mu]X) - (0).$$

- We can compute $\mathfrak{f}_{\lambda,\mu,X}$ using the addition law in E: if $[\lambda]X = (x_1, y_1)$ and $[\mu]X = (x_2, y_2)$ and $\alpha = (y_1 - y_2)/(x_1 - x_2)$, we have

$$\mathfrak{f}_{\lambda,\mu,X} = \frac{y - \alpha(x - x_1) - y_1}{x + (x_1 + x_2) - \alpha^2}.$$

Miller's algorithm

$$[\lambda]X = (x_1, y_1) \quad [\mu]X = (x_2, y_2)$$



$$\mathfrak{f}_{\lambda,\mu,X} = \frac{y - \alpha(x - x_1) - y_1}{x + (x_1 + x_2) - \alpha^2}.$$

Miller's algorithm on elliptic curves

Algorithm (Computing the Tate pairing)

Input: $\ell \in \mathbb{N}$, $P = (x_1, y_1) \in E[\ell](\mathbb{F}_q)$, $Q = (x_2, y_2) \in E(\mathbb{F}_{q^d})$.

Output: $e_T(P, Q)$.

1. *Compute the binary decomposition:* $\ell := \sum_{i=0}^{I} b_i 2^i$. *Let* $T = P, f_1 = 1, f_2 = 1$.
2. *For* i *in* $[I..0]$ *compute*
   1. α, *the slope of the tangent of* E *at* T.
   2. $T = 2T$. $T = (x_3, y_3)$.
   3. $f_1 = f_1^2(y_2 - \alpha(x_2 - x_3) - y_3)$, $f_2 = f_2^2(x_2 + (x_1 + x_3) - \alpha^2)$.
   4. *If* $b_i = 1$, *then compute*
      1. α, *the slope of the line going through* P *and* T.
      2. $T = T + Q$. $T = (x_3, y_3)$.
      3. $f_1 = f_1^2(y_2 - \alpha(x_2 - x_3) - y_3)$, $f_2 = f_2(x_2 + (x_1 + x_3) - \alpha^2)$.

Return

$$\left(\frac{f_1}{f_2}\right)^{\frac{q^d-1}{\ell}}.$$

Ring Learning With Errors

- $R = \mathbb{Z}/q\mathbb{Z}[x]/\Phi_{2n}$ where $\Phi_{2n} = x^{2^n} + 1$;
- RLWE assumption: from $(a_i, b_i = a_i s + e_i)$ where s is secret and e_i are small Gaussian error terms, the b_i look random;
- Encryption: fix t a power of two and $m \mapsto P = (as + te + m) - aX$. We have $P(s) = m \mod t$;
- Decryption: $P \mapsto P(s) \mod t$;
- Homomorphic addition: $P_m + P_{m'} = P_{m+m'}$;
- Homomorphic multiplication: $P_m \times P_{m'} = P_{m \times m'}$;
- The homomorphic properties are valid as long as the coefficient of P_m, $P_{m'}$ are small enough (to not overflow q) and in the case of multiplication when $\deg P_m + \deg P_{m'} < 2^n$;
- Optimisations: when $q = 1 \mod 2^{n+1}$, then $x^{2^{n+1}} - 1$ and hence $x^{2^n} + 1$ split totally modulo q;
- Modulus switching to reduce noise;
- Security: based on assumptions about ideal lattices (beware recent attacks on these kinds of lattices).