Random Oracle, Trapdoor Permutation and More Public-Key Schemes

Yu Zhang

HIT/CST/NIS

Cryptography, Spring, 2012

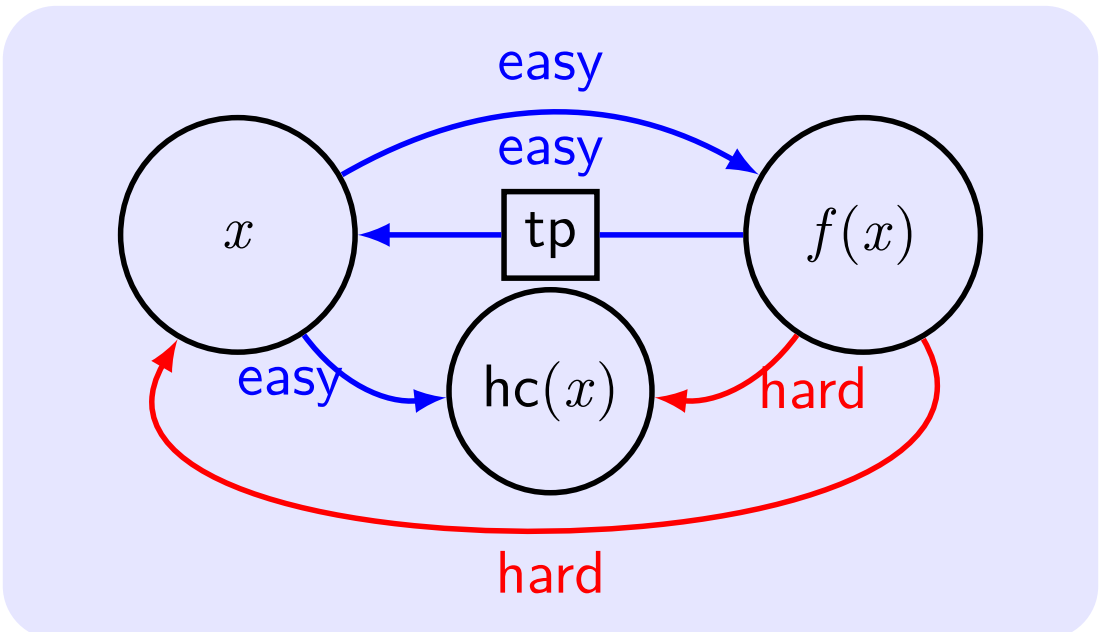
Outline

Content

Overview

Trapdoor function: is easy to compute, yet difficult to find its inverse without special info., the "trapdoor". (One Way Function with the "trapdoor")

A public-key encryption scheme can be constructed from any trapdoor permutation. ("*Theory and Applications of Trapdoor Functions*", [Yao, 1982])



Definition of Families of Trapdoor Permutations

A tuple of polynomial-time algorithms $\Pi = (\mathsf{Gen}, \mathsf{Samp}, f, \mathsf{Inv})$ is a **family of trapdoor permutations (TDP)** if:

- **parameter generation** algorithm Gen, on input 1^n, outputs (I, td) with $|I| \geq n$. (I, td) defines a set $\mathcal{D}_I = \mathcal{D}_{\mathsf{td}}$.
- Gen_I outputs only I. $(\mathsf{Gen}_I, \mathsf{Samp}, f)$ is OWP.
- deterministic **inverting algorithm** Inv. $\forall (I, \mathsf{td})$ and $\forall x \in \mathcal{D}_I$,

$$\mathsf{Inv}_{\mathsf{td}}(f_I(x)) = x.$$

Deterministic polynomial-time algorithm hc is a **hard-core predicate** of Π if $\forall$ PPT $\mathcal{A}$, $\exists$ negl such that

$$\Pr[\mathcal{A}(I, f_I(x)) = \mathsf{hc}_I(x)] \leq \frac{1}{2} + \mathsf{negl}(n).$$

Public-key Encryption Schemes from TDPs

Construction 1

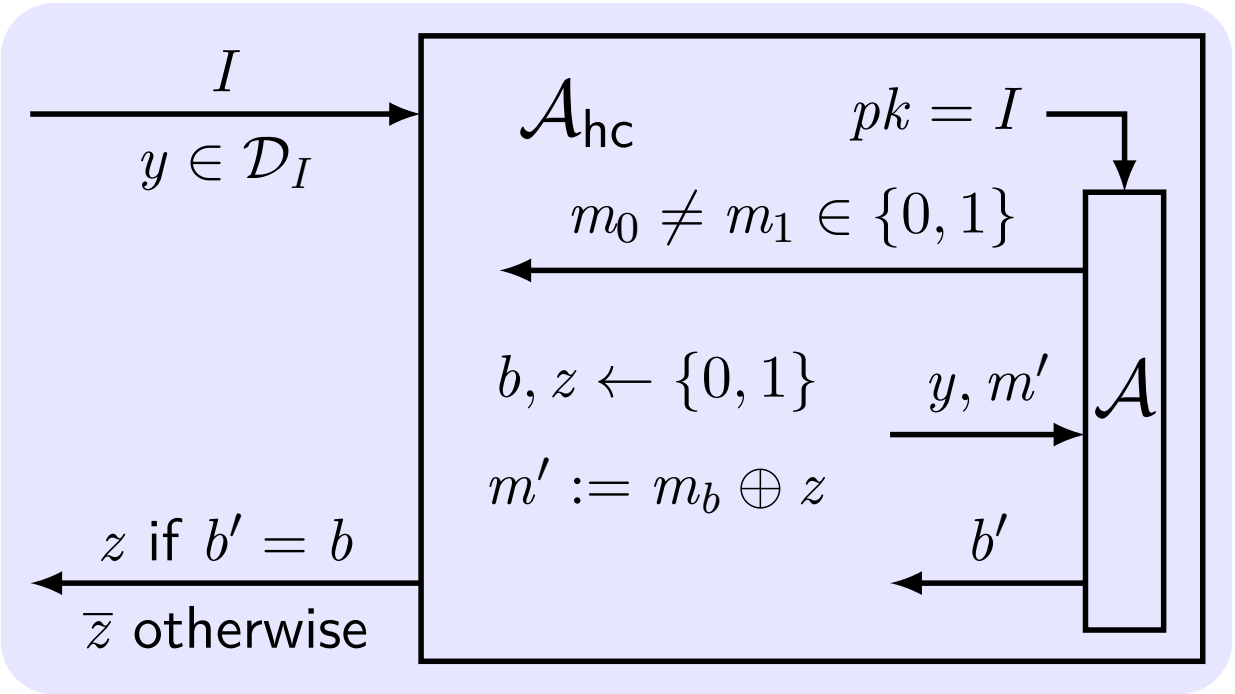
- Gen: $(I, \mathsf{td}) \leftarrow \widehat{Gen}$ *output* ***public key*** I *and* ***private key*** td.
- Enc: *on input* I *and* $m \in \{0,1\}$, *choose a random* $x \leftarrow \mathcal{D}_I$ *and output* $\langle f_I(x), \mathsf{hc}_I(x) \oplus m \rangle$.
- Dec: *on input* td *and* $\langle y, m' \rangle$, *compute* $x := f_I^{-1}(y)$ *and output* $\mathsf{hc}_I(x) \oplus m'$.

Theorem 2

If $\widehat{\Pi} = (\widehat{Gen}, f)$ *is TDP, and* hc *is HCP for* $\widehat{\Pi}$, *then Construction* Π *is CPA-secure.*

Proof

Idea: $\mathsf{hc}_I(x)$ is pseudorandom. Reduce $\mathcal{A}_{\mathsf{hc}}$ for hc to $\mathcal{A}$ for Π.



$$\Pr[\mathcal{A}_{\mathsf{hc}}(I, f_I(x)) = \mathsf{hc}_I(x)] =$$

$$\frac{1}{2} \cdot (\Pr[b' = b | z = \mathsf{hc}_I(x)] + \Pr[b' \neq b | z \neq \mathsf{hc}_I(x)]).$$

Proof (Cont.)

$$\Pr[b' = b | z = \mathsf{hc}_I(x)] = \Pr[\mathsf{PubK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n) = 1] = \varepsilon(n).$$

If $z \neq \mathsf{hc}_I(x)$, $m' = m_b \oplus \overline{\mathsf{hc}}_I(x) = m_{\overline{b}} \oplus \mathsf{hc}_I(x)$,

$$\Pr[b' = b | z \neq \mathsf{hc}_I(x)] = \Pr[\mathsf{PubK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n) = 0] = 1 - \varepsilon(n).$$

$$\Pr[b' \neq b | z \neq \mathsf{hc}_I(x)] = \varepsilon(n).$$

$$\Pr[\mathcal{A}_{\mathsf{hc}}(I, f_I(x)) = \mathsf{hc}_I(x)] = \frac{1}{2} \cdot (\varepsilon(n) + \varepsilon(n)) = \varepsilon(n).$$

Encrypting Longer Messages

Theorem 3

If $\exists$ *TDP* Π, *then* $\exists$ *TDP* $\widehat{\Pi}$ *with a HCP* hc *for* $\widehat{\Pi}$.

Example: If RSA assumption holds then the least-significant bit is hard-core for the RSA family of TDP.

an ℓ-it message $m = m_1 \cdots m_\ell$, the public key I, the ciphertext is

$$\langle f_I(x_1), \mathsf{hc}_I(x_I) \oplus m_1 \rangle, \ldots, \langle f_I(x_\ell), \mathsf{hc}_I(x_\ell) \oplus m_\ell \rangle,$$

with $x_1, \ldots, x_\ell$ chosen independently and *u.r.a* from $\mathcal{D}_I$.

An alternative way: $x_1 \leftarrow \mathcal{D}_I$ and compute $x_{i+1} := f_I(x_i)$ for $i = 1$ to ℓ. the ciphertext is

$$\langle x_{\ell+1}, \mathsf{hc}_I(x_1) \oplus m_1, \ldots, \mathsf{hc}_I(x_\ell) \oplus m_\ell \rangle.$$

Content

Random Oracle Model (ROM) – Overview

- **Random oracle (RO)**: a truly random function H answers every possible query with a random response.
- **Random oracle model (ROM)**: the existence of a public RO H.
- **Standard Model**: the adversary is only limited by time and computational power.
- **Methodology**: for constructing proven security in ROM.
  1. a scheme is designed and proven secure in ROM.
  2. Instantiate H with a hash function $\hat{H}$, such as SHA-1.
- No one seriously claims that a random oracle exists.
- There exists schemes that are proven secure in ROM but are insecure no matter how the random oracle is instantiated.

With ROM, it is easy to achieve proven security, while keeping the efficiency by appropriate instantiation.

ROM in Detail

- **Consistent**: If H ever outputs y for an input x "on-the-fly", then it always outputs the same answer given the same input.
- No one "knows" the entire function H.
- **PRF vs. RO**: PRF emulates RO, and has different usage:
  - PRF: a way of defining what it means for a concrete keyed function to be pseudorandom.
  - RO: as part of the construction of the primitive, and so must be instantiated if we want a concrete realization.

Simple Illustrations of ROM

A RO maps n_1-bit inputs to n_2-bit outputs.

- A RO as a OWF, experiment:
  1. A random function H is chosen.
  2. A random $x \in \{0,1\}^{n_1}$ is chosen, and $y := H(x)$ is evaluated.
  3. $\mathcal{A}$ is given y, and succeeds if it outputs x': $H(x') = y$.
- A RO as a CRHF, experiment:
  1. A random function H is chosen.
  2. $\mathcal{A}$ succeeds if it outputs x, x' with $H(x) = H(x')$ but $x \neq x'$.
- Constructing a PRF from a RO: $n_1 = 2n$, $n_2 = n$.
  $F_k(x) \stackrel{\text{def}}{=} H(k\|x), \quad |k| = |x| = n.$

Is the Random Oracle Methodology Sound?

- **Pro**:
  - enables the design of more efficient schemes.
  - better than no proof at all.
  - only possible weaknesses are due to the hash function.
  - few real-world attacks on "natural" secure schemes in ROM.
- **Con**:
  - No proof that security in ROM implies real-world security.
  - The reduction can not see "queries" in the real world.
  - Not well understand what is a "good" CRHF/PRF like a RO.

- What does security in the ROM guarantee in the real world?
- Are ROM fundamentally different from the standard model?

Content

Security Against CPA

Construction 4

- Gen: $pk = I$, $sk = \mathsf{td}$.
- Enc: $r \leftarrow \{0,1\}^*$, *output* $\langle f_I(r), H(r) \oplus m \rangle$.
- Dec: *input* (c_1, c_2); *compute* $r := f_{\mathsf{td}}^{-1}(c_1)$, *output* $H(r) \oplus c_2$.

Theorem 5

If f is TPD and H is RO, Construction is CPA-secure.

H can not be replaced by PRG, since the partial info on r may be leaked by c_1.

CCA-secure based on Private Key Encryption

Idea: PubK CCA = PrivK CCA + (Secret Key = TPD + RO).

Construction 6

$\Pi' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ *is a private-key encryption scheme.*

- Gen*:* $pk = I$, $sk = \mathsf{td}$.
- Enc*:* $r \leftarrow D_I$ *and compute* $k := H(r)$, *output* $\langle f_I(r), \mathsf{Enc}'_k(m) \rangle$.
- Dec*: input* $\langle c_1, c_2 \rangle$, *compute* $r := f_{\mathsf{td}}^{-1}(c_1)$, $k := H(r)$, *output* $\mathsf{Dec}'_k(c_2)$.

Theorem 7

If f *is TDP,* Π' *is CCA-secure, and* H *is RO, Construction is CCA-secure.*

CCA-secure based on TPD in ROM

Idea: PubK CCA = TDP + 2 RO (one for enc, one for mac).

Construction 8

- Gen: $pk = I$, $sk = \mathsf{td}$.
- Enc: $r \leftarrow D_I$, *output* $\langle c_1 = f_I(r), c_2 = H(r) \oplus m, c_3 = G(c_2 \| m) \rangle$.
- Dec: $r := f_{\mathsf{td}}^{-1}(c_1)$, $m := H(r) \oplus c_2$. *If* $G(c_2 \| m) = c_3$ *output* m, *otherwise* $\perp$.

Theorem 9

If f *is TDP,* G, H *are ROs, Construction is CCA-secure.*

Content

Additional Public-key Schemes

- **Goldwasser-Micali** based on deciding quadratic residuosity problem. (first scheme proven to be CPA-secure)
- **Rabin**: based on the computing square root problem. (security equivalent to the hardness of factoring)
- **Paillier**: based on the decisional composite residuosity problem. (efficient and homomorphic)
- **Elliptic curve**: forms a cyclic group with DH problem (efficient).

Quadratic Residues Modulo a Prime

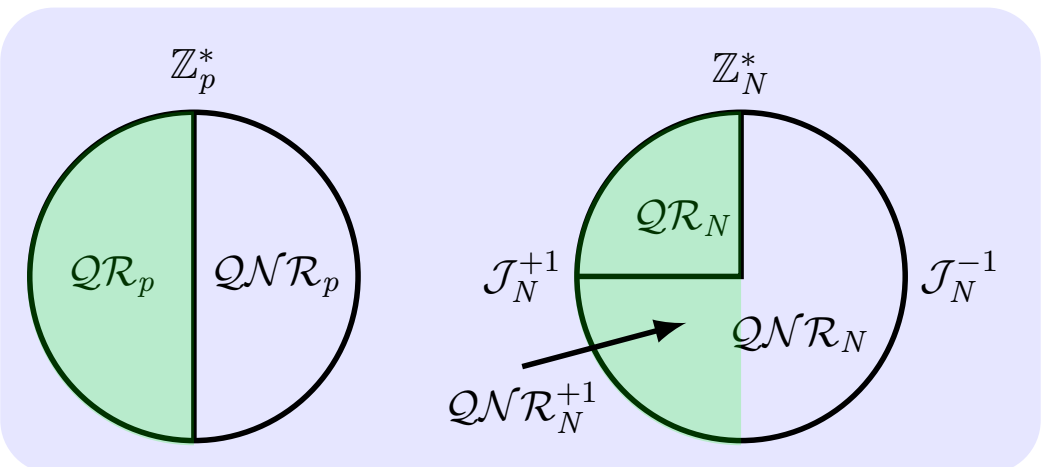
- $y \in \mathbb{G}$ is a **quadratic residue (qr)** if $\exists x \in \mathbb{G}$ with $x^2 = y$. Otherwise, y is a **quadratic non-residue (qnr)**.
- In an abelian group, the set of qr forms a subgroup.
- In $\mathbb{Z}_p^*$, $p > 2$ is prime, every qr has two square roots.
- The set of qr/qnr is $\mathcal{QR}_p/\mathcal{QNR}_p$, $|\mathcal{QR}_p| = |\mathcal{QNR}_p| = \frac{p-1}{2}$.
- $\mathcal{J}_p(x)$ is **Jacobi symbol** of x modulo p:

$$\mathcal{J}_p(x) \stackrel{\text{def}}{=} \begin{cases} +1 & \text{if } x \text{ is a qr} \\ -1 & \text{if } x \text{ is not a qr} \end{cases}$$

- $\mathcal{J}_p(x) = x^{\frac{p-1}{2}} \bmod p$.
- $\mathcal{J}_p(xy) = \mathcal{J}_p(x) \cdot \mathcal{J}_p(y)$.

Quadratic Residues Modulo a Composite

- $N = pq$, p, q distinct primes, in Chinese Remainder Theorem: $x \in \mathbb{Z}_N^*$ with $x \leftrightarrow (x_p, x_q) = ([x \bmod p], [x \bmod q])$.
- x is a qr mod $N \iff x_p/x_q$ are qr mod p/q.
- x is a qr mod $N \iff \mathcal{J}_p(x) = \mathcal{J}_q(x) = +1$.
- Qr x has 4 roots: $(\pm x_p, \pm x_q)$, so $\frac{|\mathcal{QR}_N|}{|Z_N^*|} = \frac{|\mathcal{QR}_p||\mathcal{QR}_q|}{|Z_N^*|} = \frac{1}{4}$.
- $\mathcal{J}_N(x) \stackrel{\text{def}}{=} \mathcal{J}_p(x) \cdot \mathcal{J}_q(x)$. $\mathcal{J}_N(xy) = \mathcal{J}_N(x) \cdot \mathcal{J}_N(y)$.
- $\mathcal{QNR}_N^{+1}(x) \stackrel{\text{def}}{=} \{x | x \text{ is qnr, but } \mathcal{J}_N(x) = +1\}$.



Goldwasser-Micali Scheme

- **Deciding quadratic residuosity (DQR)** of x, where x is randomly chosen from $\mathcal{J}_N^{+1}$ ($\mathcal{QR}_N$ and $\mathcal{QNR}_N^{+1}$).
- For DQR, no solution is better than factoring N.

Construction 10

- Gen: (N, p, q), $z \leftarrow \mathcal{QNR}_N^{+1}$. $pk = \langle N, z \rangle$ *and* $sk = \langle p, q \rangle$.
- Enc: $m \in \{0, 1\}$, $x \leftarrow \mathbb{Z}_N^*$, *output* $c := [z^m \cdot x^2 \bmod N]$.
- Dec: *If* c *is a qr, output* 0*; otherwise* 1.

Goldwasser-Micali scheme is CPA-secure if DQR problem is hard.

Computing Square Roots mod a Prime

Algorithm 1: computing square root of a prime

input : Prime p; quadratic residue $a \in \mathbb{Z}_p^*$

output: A square root of a

1 **case** $p = 3 \bmod 4$: **return** $[a^{\frac{p+1}{4}} \bmod p]$

2 **case** $p = 1 \bmod 4$: let b be a qnr modulo p

3 **compute** l and m odd with $2^\ell \cdot m = \frac{p-1}{2}$

4 $r := 2^\ell$, $r' := 0$

5 **for** $i = \ell$ **to** 1 **do**

6 $\quad r := r/2$, $r' := r'/2$ `/* maintain` $a^r \cdot b^{r'} = 1 \bmod p$ `*/`

7 $\quad$ **if** $a^r \cdot b^{r'} = -1 \bmod p$ **then** $r' := r' + 2^\ell \cdot m$

`/* now` $r = m$, r' `is even, and` $a^r \cdot b^{r'} = 1 \bmod p$ `*/`

8 **return** $\left[a^{\frac{r+1}{2}} \cdot b^{\frac{r'}{2}} \bmod p\right]$

Rabin Scheme

- **Computing square roots (CSR)** of qr mod N is **proven to be hard** if factoring N is hard.
- $N = pq$ is a **Blum integer** if $p \neq q$ and $p \equiv q \equiv 3 \bmod 4$.
- $\mathcal{QR}$ for Blum integer can form TDP.

Construction 11

- Gen: *Blum integer* $N = pq$, $pk = N$ *and* $sk = \langle p, q \rangle$.
- Enc: $m \in \{0,1\}$, $x \leftarrow \mathcal{QR}_N$, *output* $c := \langle [x^2 \bmod N], \mathsf{lsb}(x) \oplus m \rangle$.
- Dec: *Input* $\langle c, c' \rangle$. $x = c^{1/2}$, *output* $\mathsf{lsb}(x) \oplus c'$.

Rabin scheme is CPA-secure if factoring problem is hard.

Paillier Scheme

- $\mathbb{Z}_N \times \mathbb{Z}_N^* \simeq \mathbb{Z}_{N^2}^*$ with $f(a,b) = [(1+N)^a \cdot b^N \bmod N^2]$.
- $\mathrm{Res}(N^2)$ is the set of Nth residue mod N^2: $\{(0,b) | b \in \mathbb{Z}_N^*\}$.
- **Decisional composite residuosity (DCR)** problem is to distinguish a random element of $\mathbb{Z}_{N^2}^*$ from one of $\mathrm{Res}(N^2)$.

Construction 12

- Gen: (N,p,q), $pk = N$ *and* $sk = \langle N, \phi(N) \rangle$.
- Enc: $m \in \mathbb{Z}_N$, $r \leftarrow \mathbb{Z}_N^*$, *output* $c := [(1+N)^m \cdot r^N \bmod N^2]$.
- Dec: *output* $\left[\frac{[c^{\phi(N)} \bmod N^2] - 1}{N} \cdot \phi(N)^{-1} \bmod N \right]$.

$c^{\phi(N)} \bmod N^2 \leftrightarrow (m,r)^{\phi(N)} = (m \cdot \phi(N), r^{\phi(N)})$.

Paillier scheme is CPA-secure if DCR problem is hard.

Homomorphic Encryption

- **Homomorphic Encryption** with $\circ$: $\mathsf{Dec}_{sk}(c_1 \circ c_2) = m_1 \circ m_2$.
- Elgamal encryption is homomorphic with $\times$:
  $\langle g^{y_1}, h^{y_1} \cdot m_1 \rangle \cdot \langle g^{y_2}, h^{y_2} \cdot m_2 \rangle = \langle g^{y_1+y_2}, h^{y_1+y_2} \cdot m_1 m_2 \rangle$
- Paillier scheme is homomorphic with $+$:
  $\mathsf{Enc}_N(m_1) \cdot \mathsf{Enc}_N(m_2) = \mathsf{Enc}_N([m_1 + m_2 \bmod N])$.
- **Application**: voting without learning any individual votes.

$$c_i := [(1+N)^{v_i} \cdot r^N \bmod N^2], v_i \in \{0,1\}$$

$$c^* := [\Pi_i c_i \bmod N^2], v^* = \Sigma_i v_i$$

- First **Fully** homomorphic with $\times$ and $+$ by Craig Gentry in 2009.

Elliptic Curve Groups

Elliptic curve group: points with "addition" operation.
Any **elliptic curve** is a plane algebraic curve:

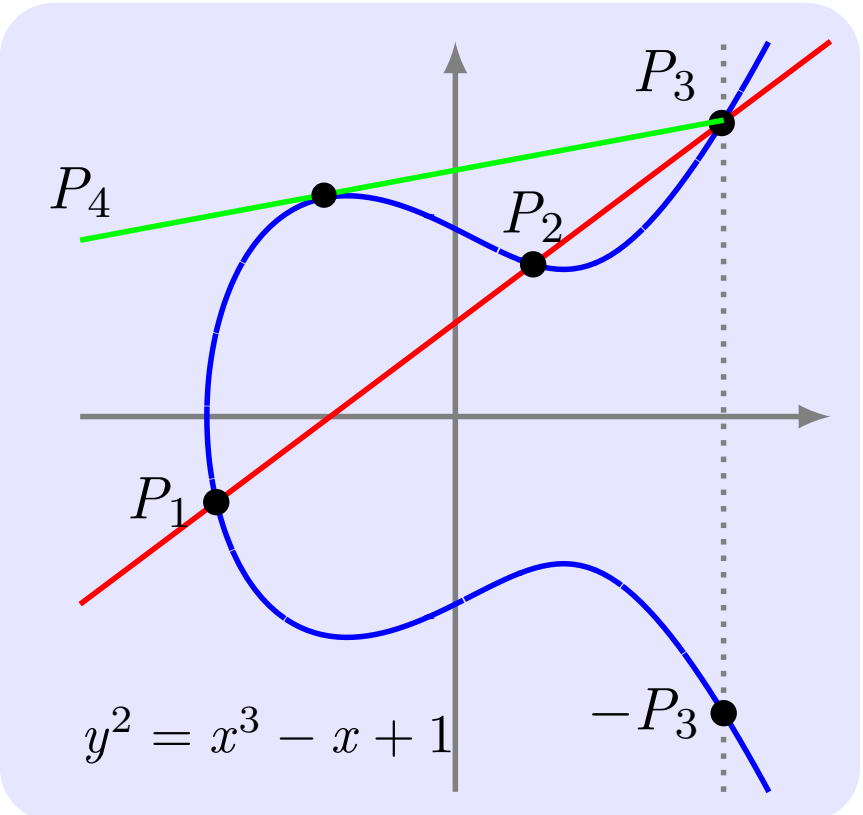
$$y^2 \equiv x^3 + Ax + B \pmod{p}$$

where $A, B \in \mathbb{Z}_p$ are constants with $4A^3 + 27B^2 \not\equiv 0 \pmod{p}$.
$\hat{E}(\mathbb{Z}_p)$ is the set of pairs $(x, y) \in \mathbb{Z}_p \times \mathbb{Z}_p$:

$$\hat{E}(\mathbb{Z}_p) \stackrel{\text{def}}{=} \{(x, y) \mid x, y \in \mathbb{Z}_p \wedge y^2 \equiv x^3 + Ax + B \pmod{p}\}$$

$E(\mathbb{Z}_p) \stackrel{\text{def}}{=} \hat{E}(\mathbb{Z}_p) \cup \{\mathcal{O}\}$, $\mathcal{O}$ is identity, "**point at infinity**".

"Addition" on Points of Elliptic Curves



Every line intersects the curve in 3 points:

- count twice if tangent.
- count $\mathcal{O}$ at the vertical infinity of y-axis.

"**Addition**" on points:

- $P + \mathcal{O} = \mathcal{O} + P = P$.
- If P_1, P_2, P_3 are co-linear, then $P_1 + P_2 + P_3 = \mathcal{O}$.

$-P = (x, -y)$
$P_1 + P_2 = -P_3$
$2P_4 = -P_3$
$dP = P + (d-1)P$

$$sk = (P, d); pk = (P, Q = dP)$$

Key Size Comparison

Key lengths (in bits) with comparable security

Symmetric	RSA/DH	ECC
56	512	112
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	521

Summary

- public key encryption from tpd
- random oracle model vs. standard model
- CPA/CCA in ROM, RSA-FDH
- Coldwasser-Micali, Rabin, Paillier (homomorphic with $+$), elliptic curve.