Lecture Notes

# Advanced cryptography: Pairing-based cryptography

Michael Nüsken

b-it

(Bonn-Aachen International Center
for Information Technology)

Winter 2012/13

# 1. Basics

Given $\overset{+}{G_1}, \overset{+}{G_2}$ and $\overset{\cdot}{G_3}$ finite commutative groups.
Then

$$e: G_1 \times G_2 \longrightarrow G_3$$

is a (compatible) pairing iff it is

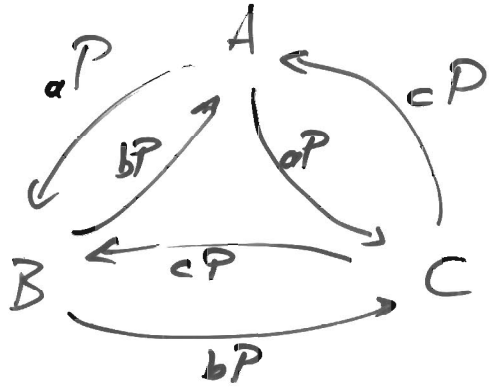
- bilinear, i.e. $e(P+Q, R) = e(P,R) \cdot e(Q,R)$
  $e(P, Q+R) = e(P,Q) \cdot e(P,R)$
  in particular: $e(aP, bQ) = e(P,Q)^{ab}$
  $e(\mathcal{O}, Q) = 1$
  $e(P, \mathcal{O}) = 1.$
- non-degenerate, i.e.
  - if $\forall Q : e(P,Q) = 1$
    then $P = \mathcal{O}$
  - if $\forall P : e(P,Q) = 1$
    then $Q = \mathcal{O}$

Usually $\#G_1$ and $\#G_2$ are prime,
then the pairing is non-degenerate
iff $\exists P \in G_1, Q \in G_2 : e(P,Q) \neq 1.$
In the symmetric setting, i.e. $G_1 = G_2$,
we will require $\exists P \in G_1 : e(P,P) \neq 1.$

- efficiently computable wrt. input size!

For the rest of the basics assume $G_1 = G_2$ of prime size.

## 1.1 Three party key exchange (Joux 2000)




Now each party can compute

$$\boxed{e(P,P)^{abc}} = e(bP, cP)^a$$
$$= e(aP, bP)^c$$
$$= e(cP, aP)^b$$

The attacker only knows $P, aP, bP, cP$ and wants to know $e(P,P)^{abc}$.

Bilinear DH Problem.

(where $a, b, c \in_R \mathbb{Z}_e$ uniformly, $e = \#G_1 = \#G_2$)

We assume that BDH problem is difficult, i.e. it is $(t,\varepsilon)$-difficult for $t \in poly(k)$, $\varepsilon^{-1} \in poly(k)$, ← security param.

i.e. $\forall \mathcal{A}$ algorithm: run time$(\mathcal{A}) \le t$

$$prob(\mathcal{A}(P, aP, bP, cP) = e(P,P)^{abc}) \le \varepsilon.$$

## BDH Assumption

The BDH problem is difficult.

Ie. $\forall t, \varepsilon$ : $t \in poly(k)$, $\varepsilon^{-1} \in poly(k)$
the BDH is $(t, \varepsilon)$-difficult

Ie. $\forall \mathcal{A}$ algorithm $\overset{\infty}{\forall} k$ $(\exists K \forall k > K)$ :

runtime $(\mathcal{A}) \leq t(k) \Rightarrow$

$$prob(\mathcal{A}(P, aP, bP, cP) = e(P,P)^{abc}) \leq \varepsilon(k).$$

## 1.2 Key distribution (Sakai, Ohgishi & Kasahara 2000)

Assume we additionally have a hash function $H_1 : \{0,1\}^* \longrightarrow G_1$.

We further that there is an extra party:
the trusted authority TA.
(sometime called PKG)

It chooses during setup a master secret $s \in \mathbb{Z}_e$.

A obtains $S_A = s\, H_1(ID_A)$ from TA.
and computes $K_A := e(S_A, H_1(ID_B))$

B obtains $S_B = s\, H_1(ID_B)$ from TA
and computes $K_B = e(S_B, H_1(ID_A))$.

Clearly: $K_A = e(s\, H_1(ID_A), H_1(ID_B)) = e(H_1(ID_A), H_1(ID_B))^s$

$$\vdots = e\left(H_1(ID_B), H_1(ID_A)\right)^s = K_B$$

So Actually, $H_1(ID_A) = \alpha \cdot P$, $H_1(ID_B) = \beta \cdot P$.

$$e(H_1(ID_A), H_1(ID_B)) = e(\alpha P, \beta P)$$
$$= e(P, P)^{\alpha\beta}$$
$$= e(\beta P, \alpha P) = e(H_1(ID_B), H_1(ID_A)).$$

This is why that situation is usually called symmetric setting.

So this is an

ID based
Non Interactive
Key distribution scheme.

## 1.3 Short signatures (Boneh, Lynn & Shacham 2001/03)

Symmetric case, $H_1 : \{0,1\}^* \to G_1$ hash fn.

User A selects $a \in_R \mathbb{Z}_q$ at random and publishes $A = aP$.

Sign a message $m$: $\sigma = a \cdot H_1(m)$.

Verify $(m, \sigma)$: check that $(P, aP, H_1(m), \sigma)$ is a DH tuple, as follows:

$$e(\sigma, P) = e(H_1(m), aP)$$

Notice: in every situation with a pairing the decisional DH problem (DDH) is easy!

i.e. given $(P, aP, bP, cP)$ decide whether $c = ab$.

## 1.4 ID based encryption (Boneh & Franklin 2003)

Symmetric case, $H_1 : \{0,1\} \to G_1$, $H_2 : G_3 \to \{0,1\}^n$ hash fns. (message length)

TA (PKG): $s \in_R \mathbb{Z}_e$, public: $S := sP$.

B gets $S_B = s \cdot H_1(ID_B)$ from TA.

A prepares $t \in_R \mathbb{Z}_e$.

$T := tP$,

$V := M \oplus H_2(\underbrace{e(t \cdot H_1(ID_B), S)})$

and sends $C = (T, V)$

B computes: $M' = V \oplus H_2(\underbrace{e(S_B, T)})$.

Verifying correctness:

$$e(t \cdot H_1(ID_B), S) = e(H_1(ID_B), P)^{ts}$$
$$= e(s \cdot H_1(ID_B), tP)$$
$$= e(S_B, T).$$

## 1.5 first discussion of security

Need: a security definition for each type of primitive. Usually, this means that we need to specify which additional possibilities and restrictions the attacker has. And of course we need to specify its goal.

Want: reduce the ~~non~~-existence of such an attacker to solving a (mathematical) problem (like BDH problem).

## 1.2 Elliptic curves

See also the course on the subject from winter 2009/10.

An elliptic curve $E$ over a finite field $\mathbb{F}_q$ is given by an equation (Weierstraß equation)

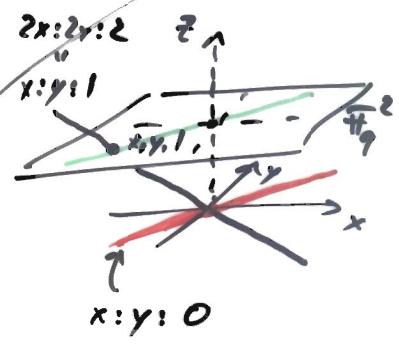
$$y^2 = x^3 + ax + b$$

(or any other cubic equation in $x, y$) ~~provided~~ under the condition that the resulting curve in $\mathbb{P}^2\mathbb{F}_q$ is smooth.

~~Lemma Let $y^2 =$~~

I.e. $E = \{(x,y) \in \mathbb{F}_q^2 \mid y^2 = x^3 + ax + b\} \cup \{O\}$

where $O = 0:1:0 \in \mathbb{P}^2\mathbb{F}_q$ (the point at infinity in $y$-direction).




$$\mathbb{P}^2\mathbb{F}_q = \underset{x:y:1}{\mathbb{F}_q^2} \cup \underset{x:1:0}{\mathbb{F}_q} \cup \{\underset{1:0:0}{*}\}$$

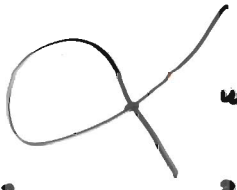
$$= \underset{x:y:1}{\mathbb{F}_q^2} \cup \underset{x:1:z}{\mathbb{F}_q^2} \cup \underset{1:y:z}{\mathbb{F}_q^2}$$

Now, $E$ is smooth

iff $4a^3 + 27b^2 \neq 0$

iff $x^3 + ax + b$ has no multiple zero.

In picture (over $\mathbb{R}$):



not smooth

$y^2 = x(x-1)^2$

$y^2 = x^3$ not smooth

**Theorem**

If char $\mathbb{F}_q \neq 2,3$ then every elliptic curve is isomorphic to one in the above Weierstraß form.

If char $\mathbb{F}_q = 3$, i.e. $q$ is 3-power, then we may always obtain the form

$$y^2 = x^3 + ax^2 + bx + c$$

but it may be that we cannot achieve $a=0$.

If char $\mathbb{F}_q = 2$, i.e. $q$ is 2-power, then we may always obtain

$$y^2 + y = x^3 + ax + b \qquad ?$$

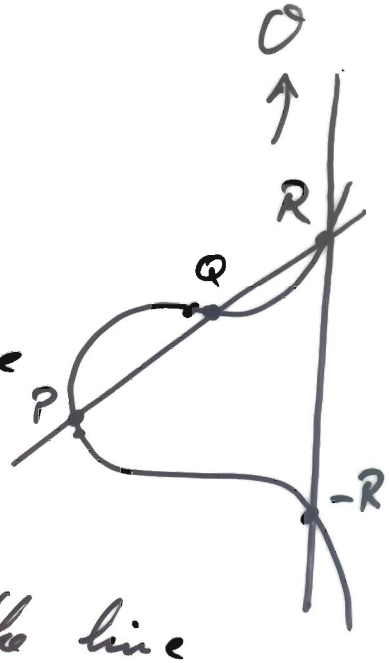
or

$$y^2 + xy = x^3 + ax + b$$

or ...

**Theorem**

$E$ is a group with the addition given by "lines":




If $P, Q, R \in E$ on the same line then $P + Q + R = O$, i.e.

$$P + Q := -R$$

where $-R$ is the third point on the line through $R$ and $O$, i.e. the vertical line through $R$.

The tricky part is to proof that this definition gives an associative operation.

The addition can be transformed into a short algorithm

$1[P] + 1[Q] + 1[R] - 3[O]$

over $\mathbb{F}_q$ involving at most $\approx 20$ operations

Notation:

$E = E$ over $\tilde{\mathbb{F}}_q$

$E(\mathbb{F}_q) = E$ over $\mathbb{F}_q$, only $\mathbb{F}_q$-rational points

Clearly: $E(\mathbb{F}_q)$ is finite.

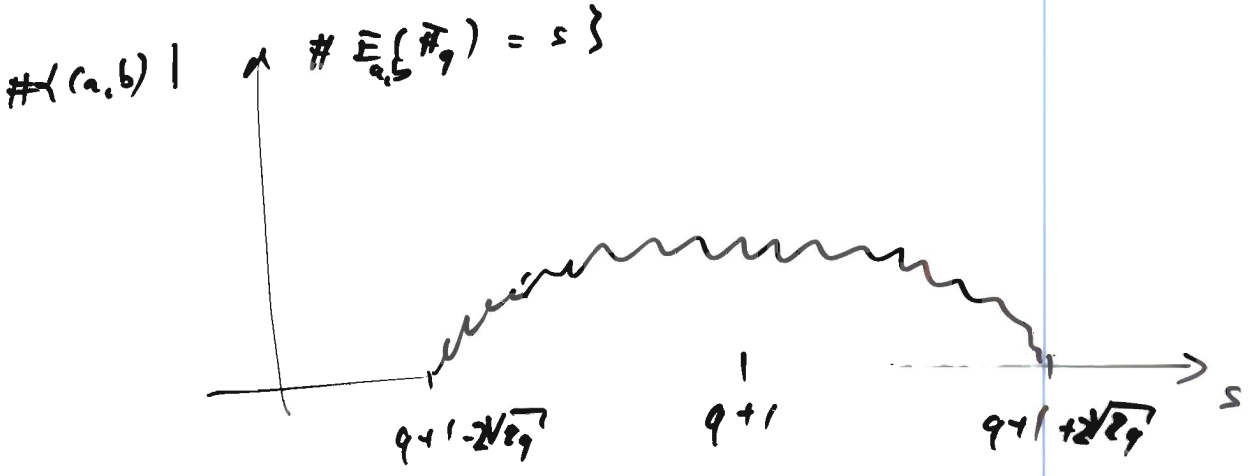
Obvious: $1 \leq \# E(\mathbb{F}_q) \leq q^2+1$

$\leq 2q+1$

<u>Theorem (Hasse)</u>

Write $\# E(\mathbb{F}_q) = q+1-t$.

Then $|t| \leq 2\sqrt{q}$. ∎




Of course, we can now easily compute

$34 \cdot P$ for $P \in E$.

by double-and-add: $34 = \underline{100010}_2$

$P$, $2P$, $4P$, $8P$, $16P$, $17P$, $34P$.

10 110 1000 10000 10001

double double double $+P$ double

Thus we just need $\lfloor \log_2 34 \rfloor + 1$ doublings and at most that many additions.

How to represent these guys in the computer?

ac 24.10.13 (4)

$\mathbb{F}_q$ : $\mathbb{F}_p[x]/(f) \to$ list of coefficients in $\mathbb{F}_p$

$\deg f$

$\mathbb{F}_p = \mathbb{Z}_p$ : $\lfloor \log_2 p \rfloor + 1$ bits for an integer $a \in \mathbb{Z}$, $0 \le a < p$.

$O(\deg f \cdot \log_2 p)$ bits

$\overset{\shortparallel}{O_{\mathbb{C}_1}(\log_2 q)}$ since $q = p^{\deg f}$.

$E$ : either it's $\mathcal{O} \to$ special.

or its a pair $(x,y) \in \mathbb{F}_q^2$.

But subject to $y^2 = x^3 + ax + b$.

So actually we just need to write down $x$ and a single bit to select whether we want $+y$ or $-y$.

Thus we need $O_{\mathbb{C}_1}(\log_2 q)$ bits for each curve point.

# 1.2 Summary EC

$E : \quad y^2 = x^3 + ax + b \quad , \quad \mathcal{O}$.

$+$ : operation given by lines.

<u>Thm</u> $E$ is a commutative group. $\triangle$

Proof A : (1) use formulas.

⌜ ⚠ Use $y^2 = x^3 + ax + b$ to reduce

$$((P+Q)+S)_x = \frac{f(x)}{g(x)} + y\frac{h(x)}{g(x)}.$$

Then compare with

$$(P+(Q+S))_x.$$

on the set $\{(P,Q,S) \in E \mid P \neq \pm Q, P+Q \neq \pm S, P \neq \pm(Q+S), P, Q, S \neq \mathcal{O}, Q \neq \pm S\}$.

So the eq. must hold on the closure of this set which is dense in $E$. )

(2) Geometrically.

(3) Use isomorphy with a certain divisor class group.

## 1.3 Pairings

**Torsion**

Given an elliptic curve $E$ over some finite field $\mathbb{F}_q$.
Given a natural number $\ell$ [later mostly prime].
Then

$$E[\ell] := \{ P \in E \mid \ell \cdot P = \mathcal{O} \}.$$

**Example**

$$E[2] = \{ P \in E \mid 2P = \mathcal{O} \}$$
$$\Updownarrow$$
$$P = -P$$

If $P = (x, y)$ then $-P = (x, -y)$.
So $P = -P$ is equiv. to $y = 0$.
In other words:

$$E[2] = \{ \mathcal{O}, (x, 0) \mid x^3 + ax + b = 0 \}.$$

This is always a four element group
unless $q$ is a two-power.

What group is $E[2]$?
There only 2 candidates: $\mathbb{Z}_4^+$ or $\mathbb{Z}_2^+ \times \mathbb{Z}_2^+$.

[Fundamental theorem on finite commutative groups ($\mathbb{Z}$-modules):
Each finite com. group $G$ is isomorphic to $\mathbb{Z}_{k_1} \times \dots \times \mathbb{Z}_{k_r}$ for some $r, k_i \in \mathbb{N}$.]

Since $E[2]$ cannot have an order-4 point, we infer

$$E[2] \cong \mathbb{Z}_2 \times \mathbb{Z}_2.$$

Most of the time we are interested in $E(\mathbb{F}_q)[2] = E[2] \cap E(\mathbb{F}_q)$.

This may be $\{0\}$, $\cong \mathbb{Z}_2$, $\cong \mathbb{Z}_2 \times \mathbb{Z}_2$.

<u>Fact</u> If char $\mathbb{F}_q = 2$ then $E[2] \cong \{0\}$, (super-singular!)
or $E[2] \cong \mathbb{Z}_2$.

<u>Example</u>

$$E[3] = \{ P \in E \mid 3P = O \}$$

$$3P = O \iff 2P = -P.$$

$$2P = (m^2 - 2x, \dots), \quad m = \frac{3x^2+a}{2y}$$

$2P = P$ iff $m^2 - 2x = x$ & $y_{2P} = -y$.

iff $(3x^2+a)^2 - 12x\,(x^3+ax+b) = 0$, & ...

$$= 3x^4 + 6ax^2 + 12bx - a^2$$

Unless char $\mathbb{F}_q = 3$ this is a degree 4 polynomial.
Thus we find 8 points $P \neq O$ with $2P = -P$.
Since $3x^4+6ax^2+12bx-a^2$ is coprime to $x^3+ax+b$ those four x-values lead to $y \neq 0$. &
And it's seperable so the four x-values are different.

Thus $\# E[3] = 9$
and so $E[3] \cong \mathbb{Z}_3 \times \mathbb{Z}_3$.

---

$f = (x-a)^k g \rightsquigarrow f' = (x-a)^{k-1} \{ \dots \}$

**Theorem**

Let $E$ be an ell. curve over a finite field $k$ of characteristic $p$, $\ell$ a prime.

Then

$$E[\ell] \cong \begin{cases} \mathbb{Z}_\ell \times \mathbb{Z}_\ell & \text{if } p \neq \ell, \\ \underset{\{0\}}{\underset{\shortparallel}{0}} \text{ or } \mathbb{Z}_\ell & \text{if } p = \ell. \end{cases}$$

$E$ ← super-singular

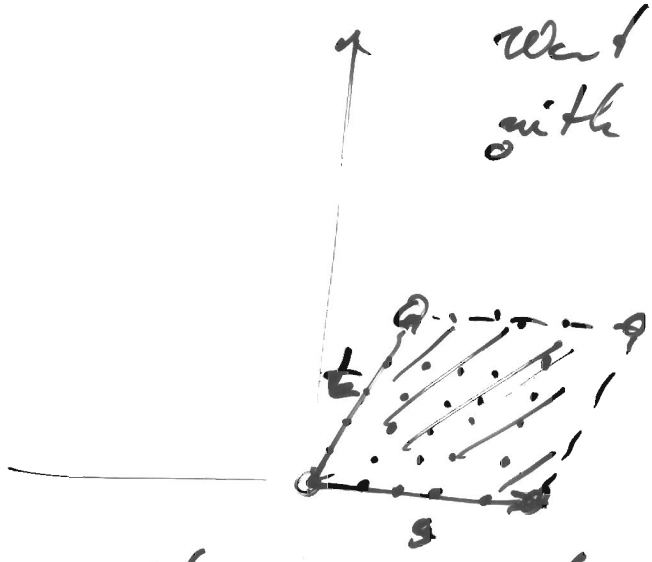
Note: $E[\ell_1 \ell_2] \cong E[\ell_1] \times E[\ell_2]$ if $\ell_1, \ell_2$ coprime.

---

**Excursion**

$$\left.\begin{aligned} e^{x+iy} &= e^x(\cos y + i \sin y) \\ e^z &= \sum_{k \geq 0} \frac{z^k}{k!} \end{aligned}\right\} 2\pi i\text{-periodic}$$

Want $f$ which is $(\omega, \tau)$-periodic with $\omega, \tau \in \mathbb{C}$ $\mathbb{R}$-linearly indep.




Essential solutions: Weierstraß $\wp$-function.

It fulfills

$$4\wp^3 = \wp^3 + g_2 \wp + g_3$$

for some $g_2, g_3 \in \mathbb{C}$

This provides an isomorphy:

$$\mathbb{C}/\Gamma_{\omega,\tau} \longrightarrow E$$

$$z \longmapsto (\wp(z), \wp'(z))$$

ac
30.10.18
5

**Excursion** div. polynomials:

One can define recursively polynomials
$\psi_n(x,y)$, $\varphi_n(x)$, $\omega_n(x,y)$ $\in \mathbb{F}_q[x,y]$
such that

$$n \cdot P = \left( \frac{\varphi_n(x)}{\psi_n^2(x)}, \frac{\omega_n(x,y)}{\psi_n^3(x,y)} \right)$$

Sic!

---

31.10.18
1

Something about the Weil pairing.

**Theorem:** Let $E$ an ell. curve over some field $k$, $n$ coprime to char $k$, Then the Weil pairing

$$e_n : E[n] \times E[n] \longrightarrow \mu_n \in \tilde{\mathbb{F}}_q$$

$$\mu_n = \{ z \in \tilde{\mathbb{F}}_q \mid z^n = 1 \}.$$

satisfying the following properties exist:

(i) $e_n$ bilinear

(ii) $e_n$ non-degenerate

(iii) $e_n$ antisymmetric, i.e. $e_n(T,T) = 1$.
In particular: $e_n(T,S) = e_n(S,T)^{-1}$.

(iv) $e_n$ compatible with the Galois action:
for $\sigma \in \mathrm{Gal}(\tilde{k} \mid k)$ we have

$$e_n(\sigma S, \sigma T) = \sigma(e_n(S,T))$$

(v) For every endomorphism $\alpha$ of $E$ (as curve & group) we have

$$e_n(\alpha(S), \alpha(T)) = e_n(S,T)^{\deg \alpha}$$

The Tate pairing is similar but not antisymmetric and we have

$$e_n(S,T) = \frac{\langle T,S\rangle_n}{\langle S,T\rangle_n}$$

where $\langle\cdot,\cdot\rangle_n$ denotes the Tate pairing.

The Tate pairing is actually defined as a map

$$\langle\cdot,\cdot\rangle_\ell : E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_{q^k})/\ell E(\mathbb{F}_{q^k}) \longrightarrow \mathbb{F}_{q^k}^\times/(\mathbb{F}_{q^k}^\times)^\ell$$

$$\|$$

and modify that

$$\tau_p : E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_{q^k})/\ell E(\mathbb{F}_{q^k}) \xrightarrow{\omega^{\frac{q^k-1}{\ell}}} \mathbb{F}_{q^k}^\times$$

$$(P,Q) \longmapsto \langle P,Q\rangle_\ell^{\frac{q^k-1}{\ell}}$$

This is defined as follows:

Let $P \in G_1 := E(\mathbb{F}_q)[\ell]$. $\quad Q+\ell T$

$Q \in G_{2+} := E(\mathbb{F}_{q^k})$

we define

$$\tau_p(P,Q) := \left(\frac{f_P(Q_1)}{f_P(Q_2)}\right)^{\frac{q^k-1}{\ell}}$$

where $Q_1 - Q_2 = Q$ (say, $Q_1 = Q+S$, $Q_2 = S$.)

and $f_\ell$ is a fn. on $E$ with values in $k \cup \{\infty\} = \mathbb{P}^1 k$

with an $\ell$-fold zero at $P+R$

and an $\ell$-fold pole at $O+R = R$.

where $\#\{O, R, P+R, Q_1, Q_2\} = 4+1$.

Tricky: the result does not depend on $R, S, T$. fo.

<u>Task</u>

Find a function (algebraic) $f_\ell : E \to \mathbb{P}^1 k$

such that $\operatorname{div} f_\ell = \ell\,[P+R] - \ell\,[R]$

and compute

$$\frac{f_\ell(Q_1)}{f_\ell(Q_2)}$$

We solve a more general

<u>Task(j)</u>

Let $P, Q \in E$ (possibly subject to further conditions)

and assume $\operatorname{div} f_j = D_j := j\,[P+R] - j\,[R] - [jP] + [0]$

Compute

$$\frac{f_j(Q_1)}{f_j(Q_2)}$$

where $Q_1 - Q_2 = Q$

We will do that recursively until we reach $j = \ell$.
At that point

$$\operatorname{div} f_\ell = D_\ell = \ell\,[P+R] - \ell\,[R] - [\ell P] + [0].$$

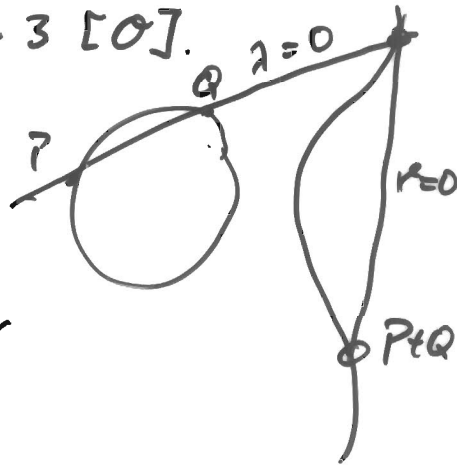
Since our $P \in E[\ell]$ we have $\ell P = 0$ and thus

$$\operatorname{div} f_\ell = D_\ell = \ell\,[P+R] - \ell\,[R] \underbrace{- [0] + [0]}_{0}$$

and so we arrive at the goal of the basic task.

Given any line $\lambda = 0$ through points $P, Q$ its divisor is

$$\operatorname{div} \lambda = [P] + [Q] + [-(P+Q)] - 3\,[O].$$




And also we may compute the divisor of the vertical line $\nu$ through $P+Q$:

$$\operatorname{div} \nu = [P+Q] + [-(P+Q)] - 2\,[O].$$

Thus

$$\operatorname{div} \frac{\lambda}{\nu} = \operatorname{div} \lambda - \operatorname{div} \nu$$
$$= [P] + [Q] - [P+Q] - [O].$$

---

Take $f_1 = \dfrac{\nu}{\lambda}$ for the points $P, R$.

$$\operatorname{div} f_1 = 1\cdot[P+R] - 1\cdot[R] - [1\cdot P] + [O]$$

---

Next step: combine $f_j$ and $f_k$ to $f_{j+k}$.

$jP$ $\quad kP \quad (j+k)P$.

Assume we have solved Task($j_1$) and Task($j_2$)
i.e. we have

$$j_1 \cdot P \,, \quad R \,, \quad \cancel{j_2 P + R} \,, \quad \frac{f_{j_2}(Q_1)}{f_{j_2}(Q_2)}$$

for $i=1$ and $i=2$.

where

$$\operatorname{div} f_{j_i} = j_i [P+R] - j_i [R] - [j_i P] + [0].$$

Consider

$$\operatorname{div}(f_{j_1} \cdot f_{j_2}) = (j_1+j_2)[P+R] - (j_1+j_2)[R]$$

$$- \operatorname{div} g = \begin{cases} -[j_1 P] - [j_2 P] + 2[0]. \\ +[(j_1+j_2)P] - [0] \\ -[(j_1+j_2)P] + [0] \end{cases}$$

Just need a function $g$ with the wanted divisor.
But if we take $\lambda$ as the line through $j_1 P$ and $j_2 P$ and $\nu$ the vertical line through $(j_1+j_2)P = j_1 P + j_2 P$
then

$$\operatorname{div} \frac{\lambda}{\nu} = [j_1 P] + [j_2 P] - [(j_1+j_2)P] - [0].$$

Now put

$$f_{j_1+j_2} := f_{j_1} \cdot f_{j_2} \cdot \frac{\lambda}{\mu}.$$

and

$$\frac{f_{j_1+j_2}(Q_1)}{f_{j_1+j_2}(Q_2)} := \frac{f_{j_1}(Q_1)}{f_{j_1}(Q_2)} \cdot \frac{f_{j_2}(Q_1)}{f_{j_2}(Q_2)} \cdot \frac{\lambda/\mu(Q_1)}{\lambda/\mu(Q_2)}.$$

so we obtain that value and also $(j_1+j_2)P$.

---

Cost of this step:
- 1 point addition to get $(j_1+j_2)P \in \mathbb{F}_q$
- $\leq 4$ mult/div to compute $\lambda, \mu \in \mathbb{F}_q$
- $\leq 7$ mult/div to evaluate $\dfrac{\lambda/\mu(Q_1)}{\lambda/\mu(Q_2)} \in \mathbb{F}_{q^k}$
- $\leq 2$ mult to obtain

$$\frac{f_{j_1+j_2}(Q_1)}{f_{j_1+j_2}(Q_2)} \in \mathbb{F}_{q^k}.$$

---

~~second~~ $O(1)$ mult/div in $\mathbb{F}_q$.
equivalent to $O(1)$ point additions in $E(\mathbb{F}_q)$.

---

Total price for solving Task $(\ell)$ = Task.
is ~~mult~~ $O(1)$ scalar mult. $\underset{\overset{\shortparallel}{O}}{\ell \cdot P}$.

# How to setup an appropriate situation?

Basic first try:

- Pick the curve $E$ given by $a, b \in \mathbb{F}_q$ with $4a^3 + 27b^2 \neq 0$
  $$\overset{\shortparallel}{2^2 a^3 + 3^3 b^2}.$$
  $$y^2 = x^3 + ax + b$$
  (in case char $\mathbb{F}_q \neq 2, 3$).

- Compute and factor $\# E(\mathbb{F}_q)$.

  This is an algorithm by Schoof, refined by Elkies and Atkin.

  $\rightarrow$ SEA.

  It's poly-time. (sth. $O(n^8)$ or $O(n^{5+\varepsilon})$ ... ) $n = \log_2 q$.

  ... may be expensive if you're unlucky: give up if you can't pay the price.

- Pick a prime factor $\ell$ of $\# E(\mathbb{F}_q)$.
  Then $E(\mathbb{F}_q)[\ell] \neq \{0\}$.

- Find the smallest $k$, which we call embedding degree, such that $\mu_\ell = \{x \in \overline{\mathbb{F}}_q \mid x^\ell = 1\}$ is contained in $\mathbb{F}_{q^k}$. [In other words: $\mathbb{F}_{q^k} = \mathbb{F}_q(\mu_\ell)$]

  Necessary condition: $\ell \mid q^k - 1$ because
  $$(\mu_\ell, \cdot) < (\mathbb{F}_{q^k}^\times, \cdot)$$

<u>Lagrange</u> • If $H < G$ then $\#H \mid \#G$.

• If $x \in G$ then $x^{\#G} = 1$. $\triangle$

Thus $\ell = \#\mu_\ell \mid \# \mathbb{F}_{q^k}^\times = q^k - 1$

In other words: $q^k \equiv_\ell 1$.

ie. $k = \mathrm{ord}_{\mathbb{Z}_\ell^\times} q \in \{1, \dots \ell-2\}$.

Typically $k = \mathrm{ord}_{\mathbb{Z}_\ell^\times} q$ will be like random;

and $\log_2 \ell \approx \log_2 \underset{\approx}{\#E(\mathbb{F}_q)}$ (provided $\ell$ is as large as desired)

$\log_2 q = n$.

Thus $k$ will be an $n$-bit number which is more or less randomly chosen.

$\Rightarrow$ Operations in $\mathbb{F}_{q^k}$ cost $\mathrm{poly}(\log q^k)$

$= \mathrm{poly}(2^n \log n)$

$\ddot{\frown}$ $\neq \mathrm{poly}(n)$

So we need specially chosen curves where $k$ is small.

- If $E$ supersingular (i.e. $E[p] = 0$, $\uparrow$ char $\mathbb{F}_q$ (i.e. $\# E(\mathbb{F}_p) = 1$)) (?)

  then $k \in \{1, 2, 3, 4, 6\}$,
  in particular $k \leq 6$.

  This is good because computations are then poly $(n)$.

- There are some constructions, most notably Barreto & Naehrig (2005), which yield curves with $k = 12$ by a probabilistic process.

---

Problem: $k$ should not be too small!

Given a pairing $\tau : E(\mathbb{F}_q)[\ell] \times ? \longrightarrow \mu_\ell \subset \mathbb{F}_{q^k}^\times$
the discrete logarithm problem in $E(\mathbb{F}_q)[\ell]$
to the discrete logarithm problem in $\mathbb{F}_{q^k}^\times$:

$$Q = aP \text{ in } E \Longrightarrow \tau(Q, \cdot) = \tau(P, \cdot)^a \text{ in } \mathbb{F}_{q^k}^\times$$

and thus finding $a$ can be done in $\mathbb{F}_{q^k}^\times$.
There are index calculus methods on $\mathbb{F}_{q^k}$,
but no one knows anything better than generic algorithms for a random curve $E$.

It turns out that with the known algorithm we have roughly the following:

| security level | needed size of $q$ for $DL(E)$ | needed size of $q^k$ for $DL(\mathbb{F}_{q^k})$ | quotient (needed $k$) |
|---|---|---|---|
| 80 | 160 | 1024 | 6 |
| 95 | 190 | 2048 | 10 |
| 110 | 220 | 4096 | 18 |

The other way round: with $k=2$ we need

$$\log q \approx \frac{1}{2} \cdot 1024 \quad \text{for 80-bit security,}$$

ie $\qquad\qquad 512$

But with that ratio ell. curves are much slower than other options (eg. $\mathbb{F}_{q^k}^\times$).

ALGORITHM 5. Miller's algorithm.

Input: Points $P, R, Q_1, Q_2 \in E$, the desired index $\ell$.

Output: The value $\frac{f_P(Q_1)}{f_P(Q_2)}$ where $\operatorname{div} f_P = \ell[P+R] - \ell[R] - [\ell P] + [\mathcal{O}]$.

1. Compute $P+R$, the line $\lambda = ax+by+c$ through $P$ and $R$, the vertical line $\nu = x+d$ through $P+R$ and let $g \leftarrow \left( \frac{\frac{ax+by+c}{x+d}\Big|_{(x,y)=Q_1}}{\frac{ax+by+c}{x+d}\Big|_{(x,y)=Q_2}} \right)^{-1}$.
2. Let $f \leftarrow g$, $J \leftarrow P$, $j \leftarrow 1$.
3. Write $\ell = (p_{r-1}, \ldots, p_1, p_0)$ in base 2.
4. For $i = r-2$ down to 0 do 5–15
5. Let $\lambda = ax+by+c$ be the tangent at $J$.
6. $S \leftarrow 2J$.
7. Let $\nu = x+d$ be the vertical line through $S$.
8. Let $f \leftarrow f^2 \cdot \frac{\lambda}{\nu}\Big|_{Q_1} \cdot \frac{\nu}{\lambda}\Big|_{Q_2}$.
9. $J \leftarrow S$, $j \leftarrow 2j$.
10. If $p_i = 1$ then
11. Let $\ell = ax+by+c$ be the line through $J$ and $P$.
12. $S \leftarrow J+P$.
13. Let $v = x+d$ be the vertical line trough $S$.
14. Let $f \leftarrow f \cdot g \cdot \frac{\lambda}{\nu}\Big|_{Q_1} \cdot \frac{\nu}{\lambda}\Big|_{Q_2}$.
15. $J \leftarrow S$, $j \leftarrow j+1$.
16. Return $f$.

# Notes on divisors

$$\mathrm{Div}(E) = \left\{ \sum_{i \le r} n_i [P_i] \;\middle|\; \begin{array}{l} r \in \mathbb{N}, \\ n_i \in \mathbb{Z}, \\ P_i \in E \end{array} \right\}$$

$$\mathrm{Div}^0(E) = \left\{ D = \sum n_i [P_i] \;\middle|\; \underbrace{\sum n_i}_{\deg D} = 0 \right\}$$

Given a nice function

$$f : E \to k \cup \{\infty\}.$$

we may define its divisor:

$$\mathrm{div}\, f = \sum n_i [P_i]$$

where $P_i$ are the zeroes or poles of $f$

and $n_i$ is the multiplicity of $f$ at $P_i$

(Thinking of $x \mapsto (x-a)^k$, its multiplicity is $k$, $k \in \mathbb{Z}$.)

Define

$$\mathrm{Princ}(E) = \left\{ \mathrm{div}\, f \;\middle|\; \begin{array}{l} f : E \to k \cup \{\infty\} \\ \text{nice function} \end{array} \right\}$$

Now we can state the famous divisor class group:

$$\mathrm{Div}^0(E) / \mathrm{Princ}(E) =: \mathcal{C}(E)$$

This makes sense because we have:

ac
7.11.12
7

<u>Thm</u> $\deg(\operatorname{div} f) = 0$.
(provided $E$ is complete).

Actually:

$$\mathcal{C}(\mathbb{P}^1_{\mathbb{F}_q}) = \{0\}.$$

In other words:

<u>Thm</u> If $E = \mathbb{P}^1_{\mathbb{F}_q}$ (or $\mathbb{P}^1_{\mathbb{C}}$) then

$$\deg D = 0 \iff \exists f : D = \operatorname{div} f.$$

And for an elliptic curve? It turns out that

$$\mathcal{C}(E) \cong E.$$

<u>Thm</u>

$$\begin{array}{ccc} \mathcal{C}(E) & \longrightarrow & E \\ D & \longmapsto & \operatorname{sum} D \\ \| & & \| \\ \sum n_i [P_i] & & \sum n_i P_i \end{array}$$

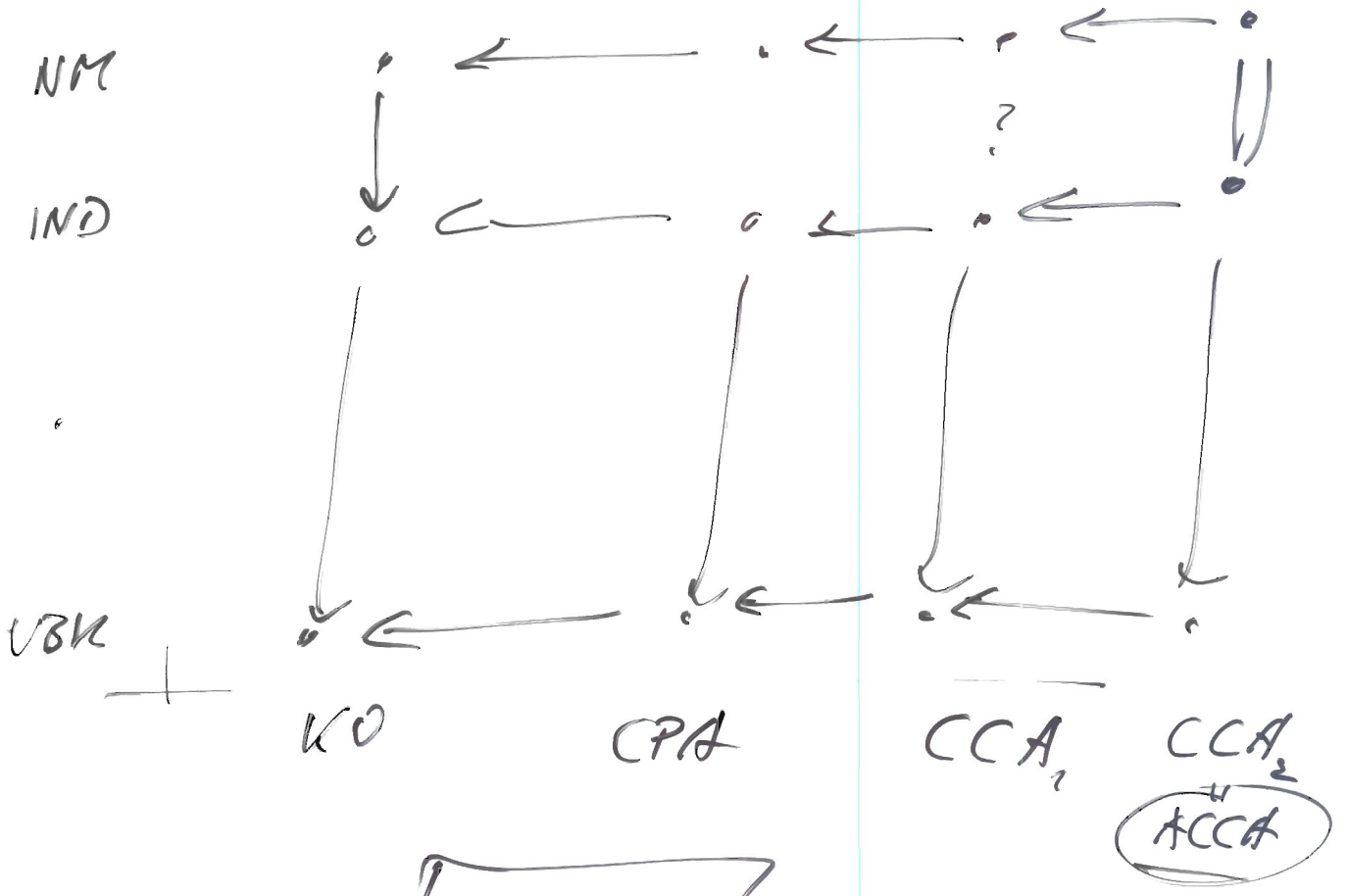
is an isomorphism.

Easy to see: it's surjective!
Take $D = [P] - [O]$ then $\deg D = 0$, $\operatorname{sum} D = P - O = P$.

<u>Thm</u> For $D \in \operatorname{Div}(E)$ we have
$D \in \operatorname{Princ}(E) \iff \deg D = 0$ & $\operatorname{sum} D = O$.

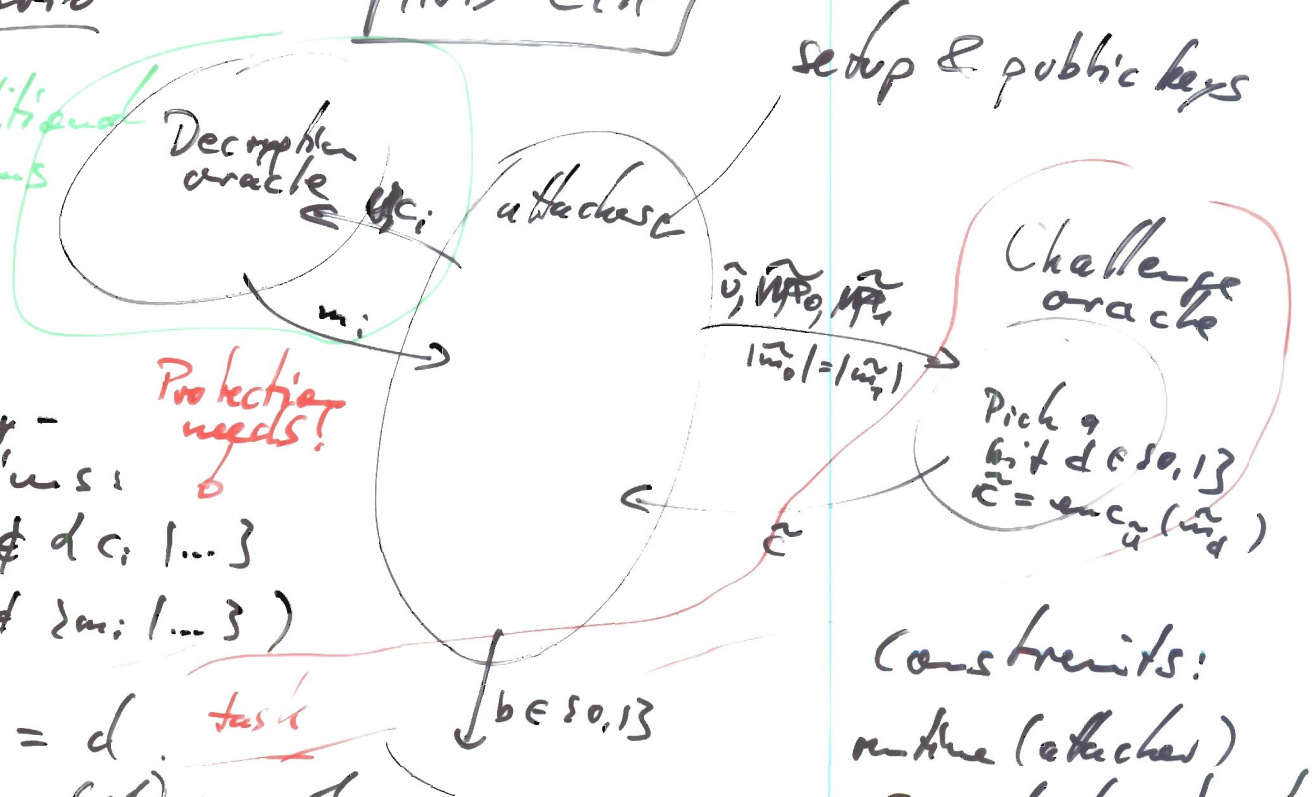
# 1.5 Security notions

let's consider ~~signatures~~ encryption schemes




## Scenario

**IND-CCA**




Protection needs!

Winning-conditions:

$\tilde{c} \notin \{c_i | \dots\}$

$(\tilde{m}_d \notin \{m_i | \dots\})$

& $b = d$ task

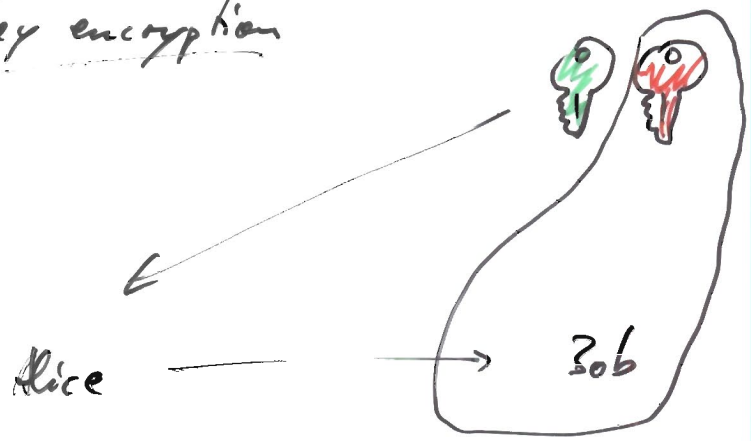
$succ(\mathcal{A}) \geq \frac{1}{poly}$.

Constraints:

runtime (attacker) $\in$ poly / prob poly.

# ID-based versus PKI systems

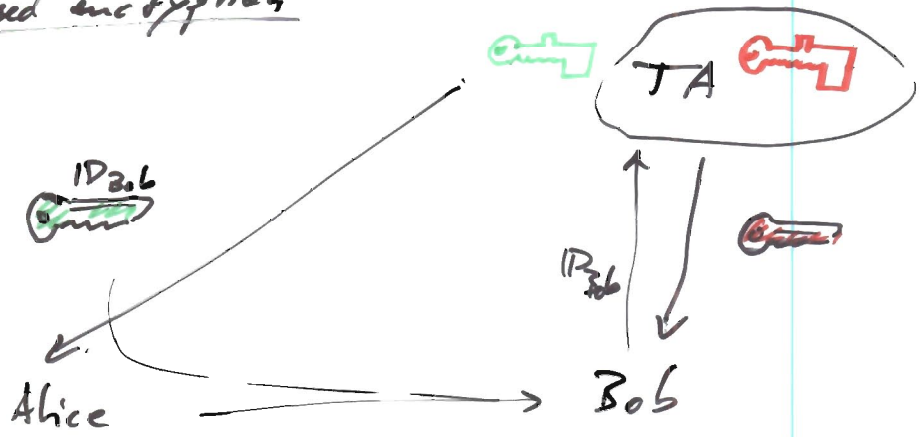
## public key encryption




Protocols:
- setup
- key generation
- encryption
- decryption

This assumes that Alice has an authentic copy of Bob's public key.

## ID-based encryption




Protocols:
- setup [TA!]
- retrieve secret key (includes TA checks)
- encryption
- decryption

In PKE: how to grant the authenticity of the public key? → PKI

In ID-based encr: authenticity of the public key is granted by construction. :)

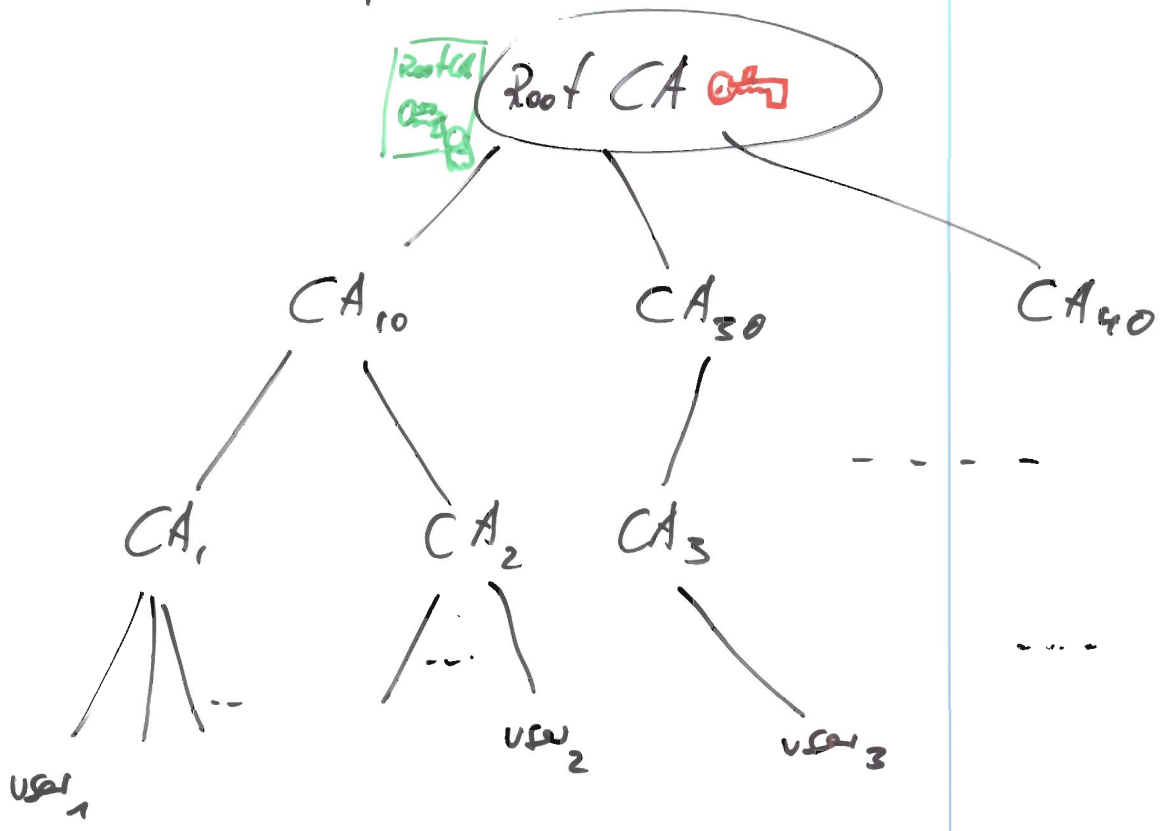
# PKI (public key identification)

Aim: authenticate public key ↔ user (identity) relations.

Solution: certificates, i.e. a document containing

- user identity information
- public key of the user's key pair
- signer information: authority
- signature

Problem: How to verify the authority's signature?
Need public key to do so...




Needed protocols:

- global setup (schemes, X.509, ...) including the RootCA keys in a self-signed certificate
- register identity ↔ (public) key.
- get a certificate for another's identity.
- revoke a key (or a certificate)
- get a certificate revocation list (CRL)

To link the benefits of the PKI with the encryption scheme, the latter needs an additional protocol

- verify certificate chain of recipient's public key including the check of the up-to-date CRLs.

<u>D-based</u> does not need that...

however: revocation is a problem.

Solution proposed by Boneh & Franklin:

use identifier involving validity periods.

# Comparison

- Architecture / Work load.

- PKI: many more protocols → more complicated

Sender load:
- in PKC the sender needs to verify the certificate chain and encrypt
- in ID-PKC the sender just needs to encrypt

Recipient load
- in PKC the recipient needs to generate its keypair and to certify it, and to decrypt.
- in ID-PKC the recipient needs to retrieve the corresponding secret key and to decrypt.

CA/TA load
- in PKI they have to check identities and generate certificates and distribute CRL.
- in ID-PKC they have to check identifiers and generate secret keys and deliver them.

Basic difference: recipient ⟺ TA need a confidential channel!

Use of identifiers may be different. Usually, PKIs deal with long term identities, whereas ID-PKC may use short lived identifiers.

PKI can be easily used in hierarchical system.

14.11.12
ac
5

- key management
  - generation of public key
    - in PKI this is done by the recipient together with the private key (or optionally by the CA!)
    - in ID-PKC that's for free.

Difference here:
- in PKI the public key is random and can thus easily be replaced by another one for the same identity,
- in ID-PKC the public key is given by the identifier and thus difficult to replace.

- ~~generation of private~~

consequence:
- in PKI two users's almost certainly have different public keys!
- in ID-PKC an identifier may refer to two different users.

=> Make sure the identifiers are unique somehow.

- generation of private keys
  - in PKI this is done by the recipient.
  - in ID-PKC this is done by the TA!

=> key-escrow! The TA knows all secret keys for the past and future.

Compromise of a CA only breaks future uses!

14.11.12
ac
6

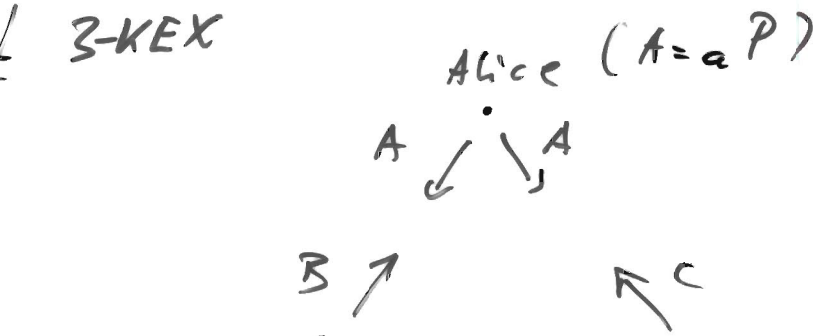
- revocation
  - in PKI need CRLs and management including good ~~watches~~ clocks at all users (~~recipients~~).
  - in ID-PKC with time-~~bla~~enhanced identifiers revocation would be automatic provided the users have correct clocks.

## Rights management

- in PKI: rights may be ~~part~~ issued using certificates
- in ID-PKC: rights may be embedded into identifiers

# §2 Three party key exchange and a few extra remarks

**Protocol 3-KEX**




Then

$$e(A,B)^c = e(B,C)^a = e(C,A)^b \neq 1$$

provided we use e non-trivial. Take like pairing, $abc \neq 0$.

MOV reduction (MOV = Menezes, Okamoto & Vanstone)

$$E[e] \longrightarrow \mu_e \subset \mathbb{F}_{q^k}^\times$$

$$X \longmapsto e(X, P')$$

some gen ✓ with $e(P,P') \neq 1$.

If $X = xP$ then $e(X,P') = e(P,P')^x$

So to determine $x$ we can solve instead:

$$DL_{E[e]} \leq_{ppt} DL_{\mu_e}$$

The classical DH protocol is secure iff CDH is difficult.

CDH: Given $(P, aP, bP)$ find $abP$.

The security notion applied here is

task: find shared key $abP$.
means: nothing but public messages.

DDH: Given $(P, aP, bP, cP)$ decide $c \overset{?}{=} ab$.

Of course: $\quad DDH \leq CDH$.

However: <u>if an efficient pairing exists</u> then DDH is easy!

Just compute $\quad e(aP, bP) \overset{?}{=} e(P, cP)$.

To be precise: consider a pairing

$$e: G_1 \times G_1 \longrightarrow G_3$$

with $\#G_1 = \ell$ prime, $\#G_3 = \ell$ also.

Then $\quad DDH_{G_1}$ is easy.

Of course, 3-KEX is broken if $DL_{G_1}$ is easy

and 3-KEX is broken if $DL_{G_3}$ is easy

$\ulcorner$ $e(P, cP) = e(P,P)^c \overset{DL_{G_3}}{\rightsquigarrow} c$ and thus $e(A,B)^c$. $\lrcorner$

Similarly: $\quad CDH_{G_3} \leq DL_{G_3}$

Like for DH for 3-KEX we consider

task: find shared key

means: give public messages.

BDH: Given $(P, aP, bP, cP)$ find $e(P,P)^{abc}$.

DBDH: Given $(P, aP, bP, cP, e(P,P)^d)$ decide $d = abc$.

Obvious:

$$\text{DBDH} \leq \text{BDH}$$

$$\text{BDH} \leq \text{CDH}$$

$\ulcorner$ $\text{CDH}(P, aP, bP) = abP$
and then $e(abP, cP) = e(P,P)^{abc}$. $\lrcorner$

Excursion

Assume a particularly simple situation

$$e: G_1 \times G_1 \longrightarrow G_3$$

with $\# G_1 = \# G_3 = \ell$ prime.

(With Tate: $G_1 = E(\mathbb{F}_q)[\ell]$, $G_3 = \mu_\ell < \mathbb{F}_{q^k}^\times$.)

Suppose ~~there exists~~ we can find $g \in G_3 \setminus \{1\}$, a point $R \in G_1$ and a group homomorphism $\varphi: G_3 \longrightarrow G_1$ efficient such that $e(\varphi(g), R) = g$ efficiently.

Then $\text{DDH}_{G_3}$ is easy! (based solely on the $\exists$ of $\varphi$)

Just take $(g, g^a, g^b, g^c)$ via $\varphi$ to $G_1$:

$$e(\varphi(g^a), \varphi(g^b)) \stackrel{?}{=} e(\varphi(g), \varphi(g^c))$$

Clearly, $e(\varphi(g), \varphi(g)) = g^{\lambda}$
with $\lambda \neq 1$.

and $e(\varphi(g^a), \varphi(g^b)) = g^{\lambda ab}$.

If $\lambda = 1$ (not necessary!) then this solves $CDH_{G_3}$.

By assumption $\ell$ is prime, and $\ell \mid q^k - 1$.

Thus
$$\lambda^{q^k - 1} \equiv_\ell 1$$
or
$$\lambda^{q^k - 3} \equiv_\ell \lambda^{-2}.$$

Furthers
$$e(\varphi(g^{\lambda^i}), \varphi(g^{\lambda^j})) = g^{\lambda^{i+j+1}}$$

Put
$$\Lambda := g^{\lambda^{q^k-3}} = g^{\lambda^{-2}}.$$

Now:
$$e(\varphi(g^{\lambda ab}), \varphi(\Lambda)) = g^{\lambda \cdot \lambda ab \cdot \lambda^{-2}}$$
$$= g^{ab}.$$

Thus we can solve $CDH_{G_3}$!

---

If $\varphi$ can be computed and found efficiently then $CDH_{G_3}$ is easy.

---

The consequence is unlikely and so ~~the~~ ~~[illegible]~~ $\varphi$ probably cannot be found or computed efficiently

## Fixed Take Inversion (FTI)

Given $G_1, G_3, e: G_1 \times G_1 \to G_3$ efficient

find $R \in G_1$, $\varphi: G_3 \to G_1$ morph., eff.

such that

$\forall g: \underset{\substack{\uparrow \\ \text{Take}}}{e}(R, \varphi(g)) = g$.

## Generalised Take Inversion

Given ...

and $g \in G_3$

find $S, T \in G_1$ such that

$$e(S, T) = g.$$

## $CDH_{G_1} \leq FTI$

Given $(P, aP, bP)$.

Task: find $abP$.

Let $A := e(R, aP)$.

Claim: $\varphi(A) = aP$.

$\ulcorner$

$e(R, \varphi(A)) = A = e(R, aP) \neq 1 \to R \neq 0$.

$\hookrightarrow \forall g: e(gR, \varphi(A)) = e(gR, aP) \to e(gR, \varphi(A) - aP) = 1$.

$\hookrightarrow \varphi(A) = aP$. $\lrcorner$

Now $A = e(R, P)^a$

Thus solve $CDH_{G_3}(e(R,P), e(R,P)^a, e(R,P)^b) = e(R,P)^{ab}$.

$$
\begin{tikzcd}
 & & \mathrm{CDH}_{\mathbb{G}_1} \arrow[rr] \arrow[dr] & & \mathrm{DL}_{\mathbb{G}_1} \arrow[d] & & \\
\mathrm{DBDH} \arrow[r] \arrow[dr] & \mathrm{BDH} \arrow[ur] \arrow[dr] & & \mathrm{GTI} \arrow[r] & \mathrm{FTI} \arrow[r] & \mathrm{DL}_{\mathbb{G}_3} \arrow[r, leftrightarrow] & \mathrm{DL}_{\mathbb{G}_1} \text{ or } \mathrm{GTI} \\
 & \mathrm{DDH}_{\mathbb{G}_3} \arrow[r] & \mathrm{CDH}_{\mathbb{G}_3} \arrow[ur] & & & &
\end{tikzcd}
$$

**Fig. 1.** Relations between complexity assumptions in pairing cryptography.

Now: $\varphi(e(R,P)^{ab}) = abP$.
That is we solved $CDH_{G_1}$.

$\square$

---

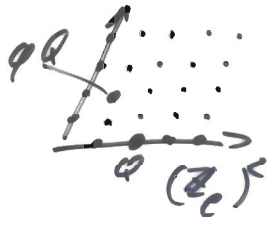
21.11.12

## Introduction

Sometimes one may turn the Weil pairing $e$ into one which has $\hat{e}(P,P) \neq 1$.
To do so use an endomorphism

$$\varphi : E \circlearrowleft$$

which „flips" the $\ell$-torsion.




Then

$$\hat{e}(P,Q) = e(P, \varphi Q).$$

But: such $\varphi$ only exists
$\llcorner$ for supersingular curves.

# §3 Key distribution

Smart 2001:

Authenticated key exchange (AK):

setup ...

key generation center (KGC)

- publishes $P_{KGC} = s \cdot P$ ← KGC's secret; $P$ ← fixed order-$\ell$ point
- after verifying identifiers issues user secret key $S_{ID} = s \cdot H(ID)$. $H$ ← a globally fixed hash function

key exchange between Alice & Bob

| A | | B |
|---|---|---|
| Pick $a \in_R \mathbb{Z}_\ell$. Set $T_A = aP$ | $\xrightarrow{T_A}$ | Pick $b \in_R \mathbb{Z}_\ell$. Set $T_B = bP$ |
| | $\xleftarrow{T_B}$ | |
| $k_A = e(aH(ID_B), P_{KGC}) \cdot e(S_A, T_B)$ | | $k_B = e(bH(ID_A), P_{KGC}) \cdot e(S_B, T_A)$ |
| $K = kdf(k_A)$ | | $K = kdf(k_B)$ ← key derivation fn. |

Correct?

$$k_A = e(aH(ID_B), sP) \cdot e(sH(ID_A), bP)$$
$$= e(asH(ID_B) + bsH(ID_A), P) = k_B.$$

## Known key security

Knowledge of past session keys $k_A$ & $k_B$ does not help...

## Forward security

Compromise-$S_A$: "no" problem for confidentiality of past sessions.

But the attacker can now impersonate $A$.

Given $(P, S_A = sH(ID_A), T_A = aP, T_B = bP, P_{KGC} = sP)$ $H(ID_A), H(ID_B)$,

find

$$k = e(a\,H(ID_B) + b\,H(ID_A), sP).$$

The Forward Security problem is

given $(P, H(ID_A) = h_A P, H(ID_B) = h_B P,$
$T_A = aP, T_B = bP, P_{KGC} = sP,$
$S_A = s h_A P)$

find

$$e(P,P)^{a h_B s} \cdot e(P,P)^{b h_A s}.$$

Claim: this is equivalent to BDH!

Pf ... done ...

Compromise s:

Everything is lost:

$$k = e(sH(ID_B), \underbrace{aP}_{T_A}) \cdot e(sH(ID_A), \underbrace{bP}_{T_B})$$

(Key escrow) This of course also means that the KGC can read all communication.

⚠ KGC is a single point of failure!

Key control

No party can control the outcome of the session key.

(To find b given $T_A$ and k means to solve a $DL_{G_3}$.)

Easy (as usual) to add key confirmation by sending suitable MACs.

Ex: Why do we call this kex "authenticated"?

Dupont & Enge (2002)

Scenario: Groups with pairing, not necessarily symmetric.

Basis:

GBDH: Given $(P, Q, aP, bQ, cP, cQ)$, $P \in G_1$, $Q \in G_2$

find $e(P,Q)^{abc}$.

Problem: $\mathcal{A}$ $(t, \varepsilon)$-solve GBDH

if $\forall k$: time $(\mathcal{A})(k) \le t(k)$

& $\text{prob}(\mathcal{A}(P, Q, aP, bQ, cP, cQ) = e(P,Q)^{abc}) \ge \varepsilon(k)$

Key exchange

Setup(k): Choose $G_1^{(k)}, G_2^{(k)}, G_3^{(k)}, e^{(k)}: G_1 \times G_2 \to G_3$ b.l., non-trivial

$\# G_i^{(k)} = \ell(k)$ prime,

$H_i^{(k)}: \{0,1\}^* \to G_i \setminus \{\mathcal{O}\}$

Master-key generation: Pick $s \in \mathbb{Z}_\ell^\times$.

Private-key-distribution: A $\xrightarrow{H_1, \ldots}$ PKG

$\xleftarrow{(S_A^1, S_A^2)}$ $(S_A^1, S_A^2) = (sH_1(ID_A), sH_2(ID_A))$

Key exchange:

| A | B |
|---|---|
| $[e(S_A^1, H_2(ID_B)), e(H_1(ID_B), S_A^2)]$ | $[e(H_1(ID_A), S_B^2), e(S_B^1, H_2(ID_A))]$ |

Both obtain (correct keys):

$$[e(H_1(ID_A), H_2(ID_B))^s, e(H_1(ID_B), H_2(ID_A))^s]$$

# Attack scenario / security notion

Setup: challenger "publishes" the setup info

Extraction query oracle:

$\mathcal{A}$ sends $ID_i$

and receives $(s\,H_1(ID_i),\ s\,H_2(ID_i))$.

Task: Produce $(ID_A, ID_B, \alpha, \beta)$

such that $(\alpha, \beta)$ is the shared key of $ID_A$ and $ID_B$

and $ID_A, ID_B \notin \{ID_i \mid \dots\}$.

i.e. ① $\alpha = e(H_1(ID_A), H_2(ID_B))^s$,

② $\beta = e(H_1(ID_B), H_2(ID_A))^s$.

We let

$$\mathrm{succ}_{\mathcal{A}} := \mathrm{prob}(①) + \mathrm{prob}(②)$$

$$\gtrsim \mathrm{prob}(① \vee ②)$$

**Prop**

GBDH $(t,\varepsilon)$-solvable

$\Rightarrow$ Protocol can be $(t+\delta, \varepsilon)$-broken.

where $\delta$ = time for one extraction query and one hash computation of $H_1$, $H_2$ each.

**Pf**

1. Extract two pairs $(P, sP) \in G_1^2$, $(Q, sQ) \in G_2^2$ (that's one [or two] extraction queries).
2. Pick two identities $ID_A$, $ID_B$ at random.
3. let $R = H_1(ID_A)$, $T = H_2(ID_B)$
4. Call GBDH solution with input

$$(P, Q, R, T, sP, sQ)$$

and obtain (if successful)

$$e(R, T)^s.$$

5. Return this as $\alpha$.

This breaks the protocol in the given time $t+\delta$ with success $\geq \varepsilon$. $\square$

**Thm**

Model $H_1, H_2$ as random oracles.
Assume that $\mathcal{A}$ $(t, \varepsilon)$-solves the attack scenario.
Assume that $\mathcal{A}$ makes at most $q_E$ extraction queries.

**Then**

there is an algo $\mathcal{B}$ that $\left(t', \underbrace{\frac{\varepsilon}{2\exp(2)\,(1+q_E)^2}}_{=:\varepsilon'}\right)$-solves GBDH

where

$$t' = K \cdot t \cdot (t_1 + t_2 + \log_2 q_E) + t_3,$$

$t_1$ = time (scalar mult in $G_1, G_2$, or $G_3$)
$t_2$ = time (random bit)
$t_3$ = time (EEA with inputs from $\mathbb{N}_{<\ell}$)
$\log_2 q_E$ = time (finding an entry in a sorted list of $q_E$ elements)
$K > 0$ small.

Clearly, $t \in \mathrm{polylg}(\ell)$ in the asymptotic view;

$$\left.\begin{array}{l} t \geq \log \ell, \\ t_1, (t_2,) t_3 \in \mathrm{poly}(\log \ell) \\ q_E \leq t \end{array}\right\} t' \in \mathrm{polylog}(\ell)$$

It would be desirable to have $\varepsilon'$ larger than given here. Best: $\varepsilon' \approx \varepsilon$.

Proof

$\mathcal{B}$ gets a challenge $(P, Q, aP, bQ, cP, cQ)$ for the GBDH and wants to compute $e(P,Q)^{abc}$.

We have an attacker $\mathcal{A}$ getting

- the setup (assuming same distribution as in "reality")
- an extraction ~~query~~ oracle computing
$$ID \longmapsto (sH_1(ID), sH_2(ID)).$$
- two hash oracles computing
$$H_i : \{0,1\}^* \to G_i$$

Since we are in the ROM the hash functions are unspecified and assumed to be random.
For the hash functions $H_1$ we will always return a multiple of $P$ or $aP$.
Difficulty for $\mathcal{B}$ is that it cannot predict which extraction queries it has to answer. (Here we loose the factor $O(q_E^2)$!)
$\to$ $\mathcal{B}$ has to guess when $\mathcal{A}$ asks for the hash values of the final identities $ID_A, ID_B$.

- $H_1$ ~~query~~ oracle

Input: $ID \in \{0,1\}^*$

Output: $R \in G_1$

0. If $ID$ is on the list, return $R$ from the list.
1. Pick $h \in_R \mathbb{Z}_e^*$.
2. Pick a random bit $u \in_R \{0,1\}$ with $\text{prob}(u=0) = \delta$ ← To be adjusted later.
3. Set $R = h\, a^u P$

   i.e. if $u = 0$ : $R = h \cdot P$
   if $u = 1$ : $R = h \cdot a P$
4. Store $(ID, R, h, u)$ in some sorted list.
5. Return $R$.

- $H_2$ ~~query~~ oracle similar but using $G_2$, $Q$, $bQ$.

Extraction ~~query~~ oracle

Input: $ID \in \{0,1\}^*$

Output: $(S^1_{ID}, S^2_{ID}) \in G_1 \times G_2$

1. Call $H_1(ID)$ and $H_2(ID)$.
   Pick $(ID, R, h, u)$ from the $H_1$ list
   and $(ID, S, \hat{h}, \hat{u})$ from the $H_2$ list.
2. If $u = 1$ or $\hat{u} = 1$ abort (and FAIL)
3. // remaining case: $R = hP$, $S = \hat{h}Q$.
3. Compute $(h \cdot cP, \hat{h} \cdot cQ)$ and return this.

Now run. If $B$ does not abort early then:
At some ~~H~~ decides it's done and returns

$$(ID_A, ID_B, \alpha, \beta) \in \{0,1\}^* \times \{0,1\}^* \times G_3 \times G_3$$

to $B$.

~~Finally,~~

~~B checks the $H_1$-list for $ID_A$:~~

~~$(ID_A, h_A P, h_A, v)$~~

~~and the $H_2$-list for $ID_B$:~~

~~$(ID_B, h_B Q, h_B, \hat{v})$.~~

~~If $v=0$ or $\hat{v}=0$ then abort and FAIL.~~

~~Otherwise~~

Finally, B does the following:

1. Pick $t \in_R \{0,1\}$ at random.

In case $t=0$: Find $(ID_A, h_A P, h_A, v)$ on the $H_1$-list
$(ID_B, \tilde{h}_B Q, \tilde{h}_B, \hat{u})$ on the $H_2$-list

In case $t=1$: Find $(ID_B, h_B P, \tilde{h}_B, u)$ on the $H_1$-list
$(ID_A, \tilde{h}_A Q, \tilde{h}_A, \hat{v})$ on the $H_2$-list

3. If $v=0$ or $\hat{v}=0$ then abort and FAIL

4. Compute
$$\gamma = \begin{cases} \alpha^{1/(h_A h_B)} & \text{if } t=0 \\ \beta^{1/(h_A h_B)} & \text{if } t=1 \end{cases}$$

5. Answer $\gamma$.

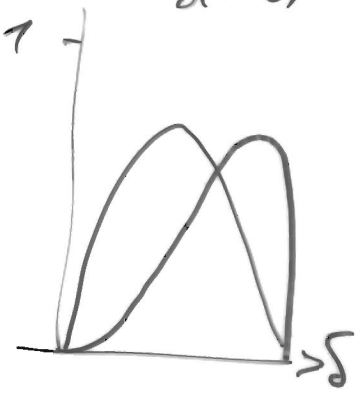
This is the correct answer if $\alpha$ was correct in case $t=0$ or if $\beta$ was correct in case $t=1$:

<u>Case $t=0$</u>

$$\alpha^{1/h_A h_B} \overset{\alpha \text{ correct}}{=} e(\underbrace{H_1(ID_A)}, H_2(ID_B))^{c/h_A h_B}$$
$$= e(h_A a P, h_B b Q)^{c/h_A h_B} = e(P,Q)^{abc}. \checkmark$$

Whenever $\mathcal{B}$ reaches that point it has a $\frac{1}{2}\varepsilon$ chance to get a correct $\alpha$ or $\beta$ and then is successful.

$\delta(1-\delta)$




$\left(1+\frac{x}{n}\right)^n \to \exp(x)$

prob ( $\mathcal{B}$ reaches that point )

$$= \delta^{2q_E} \cdot (1-\delta)^2 .$$

Taking $\delta = 1 - \frac{1}{1+q_E}$

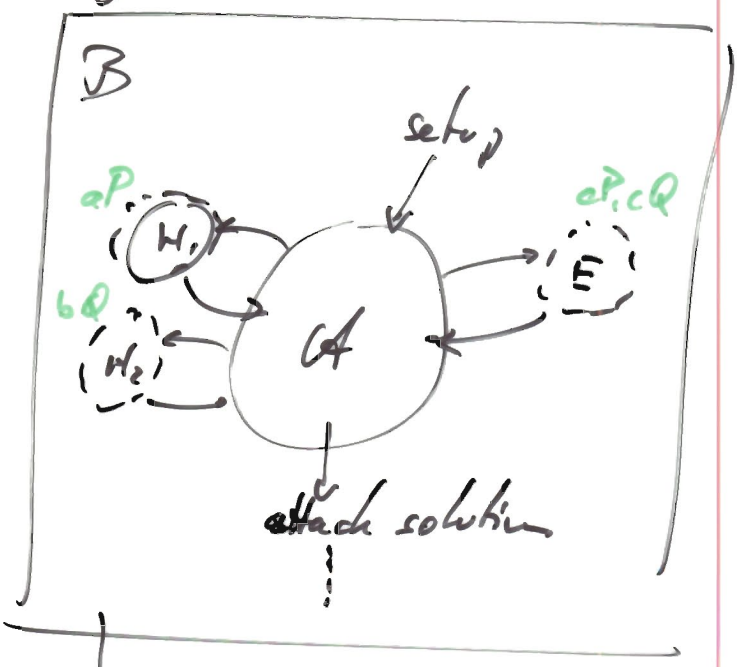
maximizes this probability.

The chance that $\mathcal{B}$ is successful is

$$\geq \frac{\varepsilon}{2} \cdot \underbrace{\left(1-\frac{1}{1+q_E}\right)^{2q_E}}_{\approx \exp(-2)} \cdot \frac{1}{(1+q_E)^2} \geq \frac{\varepsilon}{2\exp(2)(1+q_E)^2}$$

The run time of $\mathcal{B}$ is as claimed... $\square$

GBDH challenge




GBDH answer

# Sakai, Ohgishi & Kasahara (2000)

Scenario: Groups with symmetric pairing.
[not necessary but simpler]

Setup: Choose $G_1, G_2, G_T$, $e: G_1 \times G_2 \to G_T$ bil, non-trivial,

$\# G_i = \ell$ prime, $G_2 = G_1$,

$H_i : \{0,1\}^* \to G_i \setminus \{0\}$, $H_2 = H_1$,

$H_T : G_T \to \{0,1\}^k$

Master key gen: Pick $s \in \mathbb{Z}_\ell^*$.

Private key distrib: $A \xleftarrow[S_A]{\vdots} PKG$

$S_A = S_{ID_A} = (\underbrace{sH_1(ID_A)}_{S_A^1}, \underbrace{sH_2(ID_A)}_{S_A^2})$

Key exchange (non-interactive):

| A | B |
|---|---|
| $SK_{A,B} := H_T(e(S_A^1, H_2(ID_B)))$ | $H_T(e(H_1(ID_A), S_B^2))$ |

# Attack model

INDSK (Indistinguishability ~~for~~ of Shared Key)

Attacker $\mathcal{A}$:

- Setup: $\mathcal{A}$ obtains all setup info.
- Extract oracle: Input: $ID$
  Output: $S_{ID}$
- Reveal oracle: Input: $ID_1, ID_2$
  Output: $SK_{ID_1, ID_2}$
- Challenge oracle: Input: $ID_A, ID_B$
  Output: $CH$ = either $SK_{ID_A, ID_B}$ $\{d=1$
  or random $\{d=0$
- $H_i$ oracle.

× Task: decide whether $CH = SK_{ID_A, ID_B}$ : $d'$

× Restrictions: attacker looses if he has called Extract($ID_A$), Extract($ID_B$), Reveal($ID_A, ID_B$) or Reveal($ID_B, ID_A$).

$$adv_{\mathcal{A}}^{INDSK} := \left| prob(d' = d) - \frac{1}{2} \right|$$

$\mathcal{A}$ $(t, \varepsilon)$-attacker
if run time $(\mathcal{A}) \leq t$,
$adv_{\mathcal{A}}^{INDSK} \geq \varepsilon$.

<u>Prop</u> 3DH broken $\Rightarrow$ $\exists$ an attacker of this form. (Ex)

**Thm**

Assume $\mathcal{A}$ is a $(t,\varepsilon)$-attacker, and makes $q_i$ queries to the $H_i$-oracle, $i \in \{1,2,T\}$, $q_2 = q_1$.

Then there is an algorithm $\mathcal{B}$ that solves BDH with probability at least $\varepsilon' = \dfrac{2\varepsilon}{q_1 q_2 q_T}$

in time $O(t)$.

**Pf. sketch**

**Algorithm $\mathcal{B}$** Input: $P, aP, bP, cP$.

- Prepare setup for $\mathcal{A}$ dictated by its input.
- $H_1$ oracle (which is also the $H_2$ oracle in the symmetric case):

  Input: $ID_i$

  [Fix indices $I, J \in \mathbb{N}_{\leq q_1}$] ←

  0. If $ID_i$ was already there, give same answer.
  1. If $i = I$ return $aP$
  2. if $i = J$ return $bP$
  3. return $x_i P$ with $x_i \in_R \mathbb{Z}_e^\times$.

  keep track of all this.

- $H_T$ oracle:

  just random values, but make a list i/o.

- Extract ~~oracle~~ oracle

  Input: $ID$

  Output: $S_{ID}$

  1. Find $ID = ID_i$ in the list of I/O to $H_T$. (if necessary call $H_1(ID)$ first)
  3. Return $x_i \cdot cP$ (if possible).
  2. If $i = I$ or $i = J$ then ABORT

- Reveal oracle
  - Call Extract oracle for one of the identities and compute the shared key from the answer...

After $\mathcal{A}$ terminates, pick $t \in \mathbb{N}_{<q_T}$.
and return the $t$-th input of $H_T$.

prob ( $B$ does not abort $\wedge$ $\mathcal{A}$ successful )
$= \text{prob}\left( \{I, J\} = \{ i \mid ID_i = ID_A \vee ID_i = ID_B \} \right)$
$= \binom{q_1}{2}^{-1} \sim i q_1^{-2}$

prob ( $B$ guesses $t$ correctly ) $= \frac{1}{q_T}$.

Thus prob ( $B$ answers the BDH challenge correctly ) $= \frac{\varepsilon}{\binom{q_1}{2} \cdot q_T} \sim \frac{2\varepsilon}{q_1^2 q_T}$

$\square$

This works because the attacker must call $H_T$ on $e(aP, bP)^c$ to answer his challenge successfully. This is only possible since we are in the ROM and assume that $H_T$ is completely black-box to $\mathcal{A}$.

Ex: Repeat in the asym. setting

Ex: Repeat that for Dupont&Enge assuming DBDH.

2010 OPEN PROBLEM: Find a ID-based key exchange secure in the standard model.

2011 solution available!

2 other applications and generalizations.

Recall

DH

A: $T_A = aP$ B: $T_B = bP$

$SK = a \cdot T_B = abP$ $SK = b \cdot T_A = baP$

ElGamal

A $\xleftarrow{P_B}$ B: $P_B = bP$

$a \in_R \mathbb{Z}_e$

$T_A = aP$

$S = a \cdot T_B$

$C = M \oplus S$ $\xrightarrow{(T_A, C)}$ $S = b \cdot T_A$, $M' = C \oplus S$.

---

Actually, Boneh & Franklin (2001)

$\cong$ SOK (2000) + the above idea.

→ Shamir (1984) proposed IBE

→ SOK came without security proof, published in Japanese

⚠ Boneh & Franklin was the main trigger for the research in pairing-based crypto.

# Identity-based encryption

~~cannot~~ An IBE scheme always consists of four algorithms/protocols:

**Setup**
Input: security parameter $1^k$
Output: master public key mpk, ← parameters
master secret key msk.

**Extract**
Input: mpk, msk and identity ID
Output: a private key $d_{ID}$

**Encrypt**
Input: mpk, ID, message m
Output: ciphertext c

**Decrypt**
Input: mpk, $d_{ID}$, ciphertext c
Output: Either message m' or FAIL.

Correctness: obvious.

# Boneh-Franklin IBE

## Setup

Give sec-param $1^k$

generate groups $G_1, G_T$ of order $\ell$ (prime)

a pairing $e: G_1 \times G_1 \to G_T$

where $|\ell| = k$

Select hash functions $H_1: \{0,1\}^* \to G_1$,

$H^T: G_T \to \{0,1\}^n$, where

$n$ (related to $k$) is the message length.

Pick a generator $P \in G_1$.

Select $s \in_R \mathbb{Z}_\ell$ and set $MPK = sP$

Return $mpk = (G_1, G_T, e, \ell, P, MPK, H_1, H^T)$,

$msk = s$.

## Extract

Given an identity $ID$.

Return $d_{ID} = s H_1(ID)$.

## Encrypt

Given $mpk$, $ID$, $m$.

Choose $r \in_R \mathbb{Z}_\ell^\times$.

Compute $U = rP$,

$$V = m \oplus H^T(e(H_1(ID), MPK)^r)$$

## Decrypt

Given $mpk, ID, d_{ID}, c = (U, V)$.

$$m' = V \oplus H^T(e(d_{ID}, U))$$

Correctness: ... ✓

Security?

<u>Attack game</u>

Players: Challenger $\mathcal{C}$,
Attacker $\mathcal{A}$

$\mathcal{A}$ $\qquad\qquad\qquad\qquad$ $\mathcal{C}^{\text{INDID-CCA}}$

Executes Setup

$\xleftarrow{\quad mpk \quad}$

Extract
or Decrypt
oracle calls

$\xrightarrow{\quad ID^*, m_0, m_1 \quad}$

pick a bit $d$
and computes
$c^* = \text{Encrypt}(m_d, ID^*)$

$\xleftarrow{\quad c^* \quad}$

$\xrightarrow{\quad d' \quad}$

If $d = d'$ and
no oracle call
to $ID^*, c^*$ was made
then the $\mathcal{A}$ wins.

$$adv_{\mathcal{A}} = \left| \text{prob}(d' = d) - \frac{1}{2} \right|$$

this assumes that $\mathcal{A}$ makes no
unallowed calls.
One option: whenever a bad oracle
query is done, the attacker's answer
is set to a randomly guessed value.
Alternatively use

$$\left| \text{prob}(d' = d \mid \mathcal{A} \text{ made no bad oracle call}) - \frac{1}{2} \right|$$

This leads INDID-CCA.
If Decrypt calls are forbidden this leads INDID-CPA

**Thm**

If BDH is hard,
the Boneh-Franklin IBE is IND-ID-CPA secure.
(Quantitative version also possible... ) in ROM.

**Sketch**

Proof is similar to the one given with SOK scheme!

Assume $\mathcal{A}$ is a $(t,\varepsilon)$-attacker to IND-ID-CPA game against the IBE scheme.

We construct a "reduction" $\mathcal{B}$ that solves BDH which is equivalent to assuming a BDH challenge generator on the outside

$\mathcal{A}$ $\quad$ $\mathcal{B}$ $\quad$ $\mathcal{C}^{BDH}$

$\xleftarrow{(P, aP, bP, cP)}$

msk = c (unknown to $\mathcal{B}$!),

$\xleftarrow{mpk}$ MPK = cP.

Pick $I \in \{1, \dots, q_1\}$

Simulate $H_1$ returning $aP$ if $i = I$
or $x_i P$ if $i \neq I$

Simulate $H^T$ return $R_j$

Simulate Extract by

ABORT if $i = I$

$x_i \cdot cP$ if $i \neq I$

Simulate Challenge by

$U = bP$,

$V = m_d \oplus R$

where $R$ is a random bitstring

$\xrightarrow{d' \text{ or HALT}}$ Pick an input of $H^T$ $\xrightarrow{f}$

Note:

- Simulation is perfect if no ABORT occurs. The chance that $f = e(P,P)^{abc}$ is $\frac{1}{q_T}$. ($q_T$ = # calls to $H^T$.)

  For that check that we should have taken

$$R = H^T\Big(e\big(\underbrace{H_1(ID^*)}_{\substack{\parallel \\ aP}},\ \underbrace{MPK}_{cP}\big)^{b}\Big) \quad \text{dlog}_P\, U.$$

  if ~~ID~~ $ID^* = ID_I$

$$= H^T\big(e(P,P)^{abc}\big)$$

- Chance of no ABORT: $\frac{1}{q_1}$.

- Thus with advantage $\varepsilon' = \frac{\varepsilon}{q_1 q_T}$ we solve the challenge... $\square$

Note:

$$BDH(\overset{cP}{\overset{\shortparallel}{H(ID_C)}}, \overset{aP}{\overset{\shortparallel}{H(ID_A)}}, \overset{bP}{\overset{\shortparallel}{H(ID_B)}}, \overset{sc\cdot P}{\overset{\shortparallel}{s\cdot H(ID_C)}})$$

$$= e(cP, cP)^{\frac{a}{c}\cdot\frac{b}{c}\cdot s} = e(P,P)^{abs}$$

Actually,

$$BDH(P, aP, bP, cP) = e(aP, bP)^c$$

$$\boxed{BDH(Q, aP, bP, cQ) = e(aP, bP)^c.}$$

$$BDH(Q, a\delta Q, b\delta Q, cQ) = e(a\delta\cdot Q, b\delta Q)^c$$

Note that there are a lot of other conversions:

$$DH(P, aP, bP) = abP$$

$$SQ(P, aP) = a^2P$$

$$SQ \leq DH \leq SQ$$

$$(a+b)^2P - a^2P - b^2P = 2\cdot abP$$

Halfing is easy! provided we can decide whether $abP$ or $T_2 + abP$ is obtained for some $T_2 \in E[2]\setminus\{0\}$.

However, wrt. to ppt it's enough.

Another option: if DDH is easy, use it to decide.

Also $INV(P, aP) = a^{-1}P$ is equivalent (in some sense) to DH ...

Boneh - Franklin with <u>IND ID-CCA</u>

Modify the encryption:

<u>CCAsafe - Encrypt</u>

Given mpk, ID, m.

Compute $H_1(ID)$,

pick $\sigma \in_R \{0,1\}^n$,

set $r = H_5(\sigma, m) \in \mathbb{Z}_q^*$

and return

$$\cancel{C = [\, rP,\ \sigma \oplus H^T(e(H_1(ID), MPK)^r),\ m \oplus H(\sigma)\,]}$$

$$C = [\, rP,\ \sigma \oplus H^T(e(H_1(ID), MPK)^r),\ m \oplus H_6(\sigma)\,]$$

<u>CCAsafe decrypt</u>

Given mpk, ID, $d_{ID}$, $C = [U, V, W]$.

Compute $\sigma' = V \oplus H^T(e(d_{ID}, U))$,

$m' = W \oplus H_6(\sigma')$

and $r' = H_5(\sigma', m')$

and check $U \stackrel{?}{=} r'P$.

If check fails, reject the ciphertext.

Otherwise return $m'$.

# Getting rid of random oracles

→ Boneh & Boyen (2004)
→ Waters (2005)
→ Gentry (2006)

## Boneh & Boyen

**Setup**

$\alpha \in_R \mathbb{Z}_e$

$P_1 := \alpha P$,

$P_2 \in G$,

$U = (U_{ij}) \in_R G^{n \times 2}$

$k \in_R \mathcal{K}$ security parameters: $H_k : \{0,1\}^{*} \to \{0,1\}^n$.

$\text{params} = (P, P_1, P_2, U, k)$,

$\text{master-key} = (\alpha P_2)$.

**Key Gen / Extract**

Input: params, ID, master-key

Output: $d_{ID} \in G^{n+1}$.

$d_0 = \alpha P_2 + \sum U_{i,a_i} d_i$ where $U_{ij} = u_{ij} P$

$a_1 \dots a_n = H_k(ID) \in \{0,1\}^n$.

$r_1, \dots, r_n \in_R \mathbb{Z}_e$

$$d_{ID} = \left( \alpha P_2 + \sum_{i=1}^{n} r_i U_{i,a_i}, r_1 P, \dots, r_n P \right) \in G^{n+1}$$

**Encrypt**

Input: $M \in G_T$, $ID \in \{0,1\}^{n(k)}$, params.

Output: ciphertext $C \in G_T \times G^{n+1}$

Pick $t \in_R \mathbb{Z}_e$; $a_1 \dots a_n = H_k(ID)$

and return $C = \left( e(P_1, P_2)^t \cdot M, tP, t \cdot U_{1a_1}, \dots, tU_{na_n} \right)$

**Decrypt**

Input: $C = (A, B, C_1, \dots, C_n)$, $d_{ID}$

Output: $M' := A \prod_{j=1}^{n} e(C_j, d_j)\, e(B, d_0)^{-1}$

Correct: $M' = M$

Theorem (B&B)

Assume DBDH [distinguish $(P, aP, bP, cP, e(P,P)^{abc})$ and $(P, aP, bP, cP, Q)$]

Assume $\{H_k\}_{k \in \mathcal{K}}$ is a family of hash fns with a certain property w.r.t. $(m, \varepsilon_{PRF}, q)$

Put $\delta = \left(\cancel{1 - \frac{1}{s}}\right) 2^{-sm}$, $\Delta = \delta (1-\delta)^q > \varepsilon_{PRF}$

Then the B&B IBE scheme

is IND-ID-CPA secure

with $\varepsilon_{IBE} \leq \dfrac{2}{\Delta - \varepsilon_{PRF}} \cdot \varepsilon_{DBDH}$.

(Note: $m = \Theta(2 \ln q) \rightsquigarrow \Delta = \Theta(1/q)$ )

and $\varepsilon_{PRF}$ is assumed to be very small.

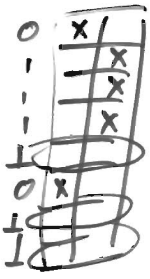
$q$ = # of private key queries.

Experiment 1 : $\mathrm{BDH\text{-}Exp}_{\mathcal{A}}(b, (P, P_1, P_2, P_3, T))$

Let $\mathcal{A}$ be an algorithm,
$b \in \{0,1\}$, $(P, P_1, P_2, P_3, T)$ a DBDH tuple.
We define a game between $\mathcal{A}$ and a simulator $\mathcal{B}$.

Setup

Pick $v = v_1, \dots v_n \in \{\perp, 0, 1\}^n$ with $\#\{\text{non-}\perp \text{ components}\} = m$



Generate $U$ as follows:
Pick $\alpha_{ij} \in_R \mathbb{Z}_q$, $i \in 1 \dots n$, $j \in \{0,1\}$.
and set
$$U_{ij} = \begin{cases} P_2 + \alpha_{ij} P & \text{if } v_i = j \\ \alpha_{ij} P & \text{otherwise.} \end{cases}$$
Pick $k \in \mathcal{K}$. $\rightsquigarrow$ determines $H_k : \{0,1\}^m \to \{0,1\}^n$.
Give $(P, P_1, P_2, U, k)$ as params to $\mathcal{A}$.
The corresponding master-key is $\alpha P_2$ if $P_1 = \alpha P$.
[We, $\mathcal{B}$, do not have $\alpha P_2$ since that is a solution to the DH $(P, P_1, P_2)$. ]

Phase 1

$\mathcal{A}$ issues up to $q$ private key queries.
say for $ID \in \{0,1\}^m$, $a_1 \dots a_n = H_k(ID)$.
If for all $i$ $a_i \neq v_i$ : ABORT.
Otherwise: say $a_i = v_i$.
Pick $r_i \in_R \mathbb{Z}_q$ and let

$$d_0 = -\alpha_{i,v_i} P_1 + \sum_{j=1}^{n} r_j U_{j,a_j}, \quad d_j = \begin{cases} r_j P & \text{if } j \neq i \\ r_i P - P_1 & \text{if } j = i. \end{cases}$$

$U_{ij} = v_{ij} P$

$$d_0 - \sum_{j=1}^{n} v_{j,a_j} d_j = -\alpha_{i,v_i} P_1 + \underbrace{r_i U_{i,a_i}} - v_{i,a_i}(\underbrace{r_i} P - P_1)$$
$$= v_{i,a_i} P_1 - \alpha_{i,a_i} P_1 = \alpha(U_{i,a_i} - \alpha_{i,a_i} P) = \alpha P_2.$$

The simulator $\mathcal{B}$ gives $d_{ID} = (d_0, d_1, \dots, d_n)$ to $\mathcal{A}$.

## Challenge

$\mathcal{A}$ gives an identity $ID^*$ and two message $M_0, M_1 \in G$ to the simulator.

Write $a_1 \dots a_n = H_k(ID^*)$.

If for some $i$ we have $a_i = v_1$ then ABORT

Otherwise:

Give the ciphertext

$$C^* = (M_b \cdot T, P_3, \alpha_{1,a_1} P_3, \dots, \alpha_{n,a_n} P_3).$$

If $T = e(\alpha P_3, P_2)^{\alpha}$ then this is the encryption of $M_b$ for $ID^*$.

## Phase 2

$\mathcal{A}$ issues further private key queries for identities $ID \neq ID^*$ ...

## Guess

Finally, $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$.

The simulator returns $b'$ as the result of the experiment.

**Experiment 2** $PRF\text{-}Exp_{\mathcal{A}}(b, F, k)$.

Let $\mathcal{A}$ be an algorithm, $b \in \{0,1\}$,
$F : \{0,1\}^w \to \{0,1\}$ and $k \in \mathcal{K}$.

[Earlier the authors define the <u>bias map</u>

$$F_{K,H}(x) = [\ \forall i \in \{1,\dots,n\} : H(x)|_i \neq K_i\ ]$$

where $H : \{0,1\}^w \to \{0,1\}^n$, $K \in \{\perp, 0, 1\}^n$, $\#\{\perp \in K\} = n-m$.

**Setup**

Pick $P \in_R G \setminus \{0\}$.
Pick $\alpha \in_R \mathbb{Z}_e$, $P_1 = \alpha P$.
Pick $P_2 \in_R G$.
Pick a matrix $U \in G^{n \times 2}$
Give the params $= (P, P_1, P_2, U, \ell)$ to $\mathcal{A}$,
and keep the private master-key $\alpha P_2$ to itself.

**Phase 1**

$\mathcal{A}$ issues up to $q$ private key queries.
The simulator does the following:
[If $F(ID) = 1$ the simulator ABORTS. ||
produce the secret key $d_{ID}$ as in the scheme
and give it to $\mathcal{A}$.

**Challenge**

$\mathcal{A}$ gives $ID^*, M_0, M_1 \in G$ to simulator.
The simulator does this:
[If $F(ID^*) = 0$ then ABORT ||
encrypts $M_b$ for $ID^*$
and gives the ciphertext to $\mathcal{A}$.

**Phase 2** as Phase 1

**Guess** $\mathcal{A}$ returns a bit $b'$. The simulator returns $1 \oplus b \oplus b'$.

We define r.v.

(1) $BDH\text{-}Exp_{\mathcal{A}}(b, (P, P_1, P_2, P_3, T)) \in \{ABORT, 0, 1\}$

(2) $PRF\text{-}Exp_{\mathcal{A}}(b, F, k) \in \{ABORT, 0, 1\}$.

Define:

- $Z = (P, P_1, P_2, P_3, T)$ unif. random positive DBDH challenge.
  ie. $T = e(P_2, P_3)^{\alpha}$ if $P_1 = \alpha P$.
- $T_b = BDH\text{-}Exp_{\mathcal{A}}(b, Z)$ r.v. for $b \in \{0,1\}$.
- $t_b = prob(T_b = 1 \mid T_b \neq ABORT)$
- $(F_{K,H_k}, k)$ is a r.v. where $k \in \mathcal{K}$, $K \in \{\perp, 0, 1\}^n$ with $m$ def'd places are uniformly chose.
- $\delta = 2^{-m}$, $\Delta = \delta(1-\delta)^q$


where $q$ is the bound on the # of private key queries.

<u>Claim 1</u>

Let $(F, k) = (F_{K,H_k}, k)$, $b \in \{0,1\}$.

then $T_b = PRF\text{-}Exp_{\mathcal{A}}(b, F, k)$.

This says that Exp 2 is precisely the same as Exp 1 when the ABORT are put in the same places and that is done by that bias map $F$.

## Claim 2

For $b \in \{0,1\}$ we have

$$t_b = \text{prob}(\text{CPA-Exp}_{\mathcal{A}}(b) = 1).$$


That is: the attacker does not learn anything from the artificial ABORTs in Experiment 2.

## Claim 3

Let $(F, k) = (F_{K,H_k}, k)$, $b \in \{0,1\}$.

Then

$$\text{prob}(\text{PRF-Exp}_{\mathcal{A}}(b, F, k) = \text{ABORT}) \leq 1 - \Delta + \varepsilon_{PRF}.$$

If the ABORTs were independent then we would expect $= 1 - \Delta$ for this prob. The $\varepsilon_{PRF}$ accounts for the deviation from independent random choice.

## Claim 4

$$|t_0 - t_1| < \frac{2}{\Delta - \varepsilon_{PRF}} \cdot \varepsilon_{BDH}.$$

Putting everything together:

$$\varepsilon_{IBE} = \text{adv}_{\mathcal{A}}^{CPA} = |\text{prob}(\text{CPA-Exp}_{\mathcal{A}}(0) = 1) - \text{prob}(\text{CPA-Exp}_{\mathcal{A}}(1) = 1)|$$

$$= |t_0 - t_1| < \frac{2}{\Delta - \varepsilon_{PRF}} \cdot \varepsilon_{BDH}.$$

$\triangle$

**News** 1425 bit field DL broken (Joux)

↳ Don't use 160 bit EC with embedding degree 6 any more!

## Waters (2005)

**Setup** $G, G_T$ order $\ell$ prime groups, $P \in G$ generator,

$e: G \times G \to G_T$ eff. pairing.

$n$ = bit length for identities,

$H: \{0,1\}^* \longrightarrow \{0,1\}^n$.

$\alpha \in_R \mathbb{Z}_\ell$ secret

$P_1 = \alpha P, \quad P_2 \in_R G$

$U' \in G, \quad (U_i) \in G^n$

Public: $P, P_1, P_2, U', (U_i)$.

Master secret: $\alpha P_2 = DH(P, P_1, P_2)$

**KeyGen** Input: $v \in \{0,1\}^n$ identity

Output: $d_v \in G^2$.

$r \in_R \mathbb{Z}_\ell$.

$$d_v = \left[ \alpha P_2 + r\left(U' + \sum_{v_i=1} U_i\right), \; rP \right]$$

**enc** Input: $M \in G_T$ message, $v \in \{0,1\}^n$ recipient id

Output: $C \in G_T \times G^2$

$t \in_R \mathbb{Z}_\ell$.

$$C = \left[ M \cdot e(P_1, P_2)^t, \; tP, \; t\left(U' + \sum_{v_i=1} U_i\right) \right]$$

**dec** Input: $C = (c_1, c_2, c_3)$ ciphertext, $d_v$ secret key. $(d_1, d_2)$

Output: $M' \in G_T$

$$M' = c_1 \cdot \frac{e(d_2, c_3)}{e(d_1, c_2)}$$

Thm 1

Assume that DBDH is $\left(t + O\left(\varepsilon^{-2}\ln(\varepsilon^{-1})\lambda^{-1}\ln(\lambda^{-1})\right), \frac{\varepsilon}{32(n+1)q}\right)$-secure;

where $\lambda = \frac{1}{8(n+1)q}$.

Then the above scheme is $(t, q, \varepsilon)$-secure

time

# private key queries

(success) advantage

IND-CPA

Still:

- no tightness (loose factor $q$)
+ security in standard model
++ efficient in data.

Three players:

challenger $C$:

produces $(P, A, B, C, T)$

with $T = e(P, P)^{abc}$ if $\gamma = 1$

or $T$ random if $\gamma = 0$

reduction $B$ (which we have to define):

tries to solve $C$'s challenge and produce $\gamma' = \gamma$ using ~~the~~ an attacker $A$ to scheme.

attacker $A$:

may ask arbitrary private key queries for identities $v^{(i)}$

generates a identity $v^*$ and msgs $M_0, M_1$ and obtains the enc of $M_\beta$ for $v^*$

~~and has to decide for~~

and tries to produce $\beta' = \beta$.

# The reduction $\mathcal{B}$

## Setup

Put $m = 4q$, small compared to $\ell$.
even $\boxed{m(n+1)k \ell}$

$k \in_R \mathbb{N}_{<n}$ assuming $n$ to be given.

Then pick $x_i \in_R \mathbb{N}_{<m}$ for $i \in \mathbb{N}_{<n}$.

$x' \in_R \mathbb{N}_{<m}$.

and pick $y_i \in_R \mathbb{Z}_\ell$ for $i \in \mathbb{N}_{<n}$,

$y' \in_R \mathbb{Z}_\ell$.

Put $P = P_1$, $P_1 = A$, $P_2 = B$,

$$U' = (\ell - km + x')\, P_2 + y'\, P.$$

$$U_i = x_i\, P_2 + y_i\, P.$$

Notice: • due to the choice of the $y$-s the $U'$, $U_i$ are random.

• master secret must be $DH(P, A, B)$ which is not available to $\mathcal{B}$.

We define some expressions:

$$F(v) = \underbrace{(\ell - mk)}_{>0} + x' + \sum_{v_i = 1} x_i$$

$$J(v) = y' + \sum_{v_i = 1} y_i$$

Now, $U' + \sum_{v_i = 1} U_i = F(v) \cdot P_2 + J(v) \cdot P$.

$$K(v) = \left[\, x' + \sum_{v_i = 1} x_i \equiv_m 0 \,\right].$$

## Private key query

$\mathcal{A}$ asks for private key of identity $v$.
$\mathcal{B}$ has to produce a valid key now.

If $V_1(v) =$ ~~false~~ true ABORT and pick $\beta' \in_R \{0,1\}$.

Otherwise:

pick $r \in_R \mathbb{Z}_\ell$ and construct

$$d_v = \Big( -\frac{J(v)}{F(v)} P_1 + r\Big(U' + \sum_{v_i=1} U_i\Big),$$

$$-\frac{1}{F(v)} P_1 + rP \Big) =: (D_1, D_2)$$

Noting that $K(v) =$ ~~false~~ false $\Rightarrow$ $F(v) \neq 0$ in $\mathbb{Z}_\ell$.

$x' + \sum x_i \neq_m 0$ $\xrightarrow{\text{provided}}$ $2mn < \ell$

$$\underbrace{(\ell - mk)}_{>0} + \underbrace{(x' + \sum x_i)}_{\geq 0,\ \neq_m 0}$$

Put $\tilde{r} = r - \frac{a}{F(v)}$, $a = \mathrm{dlog}_P A$.

Obviously,

$$D_2 = -\frac{1}{F(v)} aP + rP = \underbrace{\Big(r - \frac{a}{F(v)}\Big)}_{\tilde{r}} P.$$

$$= \tilde{r} P.$$

Check that

$$D_1 \overset{?}{=} aP_2 + \tilde{r}\Big(U' + \sum_{v_i=1} U_i\Big)$$

to verify that $d_v$ is the correct private key.

Now,

$$D_1 = -\frac{J(v)}{F(v)} aP + \overbrace{\frac{a}{F(v)}}^{r-\tilde{r}}\big(F(v) P_2 + J(v) P\big) + \tilde{r}\Big(U' + \sum_{v_i=1} U_i\Big)$$

$$= aP_2 + \tilde{r}\Big(U' + \sum U_i\Big).$$

Challenge

The attacker produces $v^*$, $M_0$, $M_1$ and the reduction $\mathcal{B}$ now has to produce the encryption of $M_0$ or $M_1$.

If $K(v^*) = \text{false}$ or $x' + \sum_{v_i^*=1} x_i \neq km$ ABORT and submit a random answer $\beta'$.

Otherwise: $x' + \sum_{v_i^*=1} x_i = km$ with some probability $\left(\frac{1}{n+1}\right)$

Then $F(v^*) = 0$ in $\mathbb{Z}_e$.

Now $\mathcal{B}$ flips a coin $\beta$ and produces the ciphertext

$$\left( M_\beta \cdot T,\ C,\ J(v^*) \cdot C \right)$$

If we are in the case $y = 1$, i.e. $T = e(P,P)^{abc}$ then this is a encryption of $M_\beta$.

$$M_\beta \cdot T = M_\beta \cdot e(P,P)^{abc} = M_\beta \cdot e(P_1, P_2)^c$$

$$C = c\,P.$$

$$J(v^*) \cdot C = \left(y' + \sum_{v_i=1} y_i\right) \cdot c\,P$$

$$= \underbrace{c \cdot \Big[ \overbrace{\underbrace{\left((l-mk) + x' + \sum_{v_i^*=1} x_i\right)}_{F(v^*)}}^{=0 \text{ in } \mathbb{Z}_e} P_2 + \underbrace{\left(y' + \sum_{v_i^*=1} y_i\right)}_{J(v^*)} P \Big]}$$

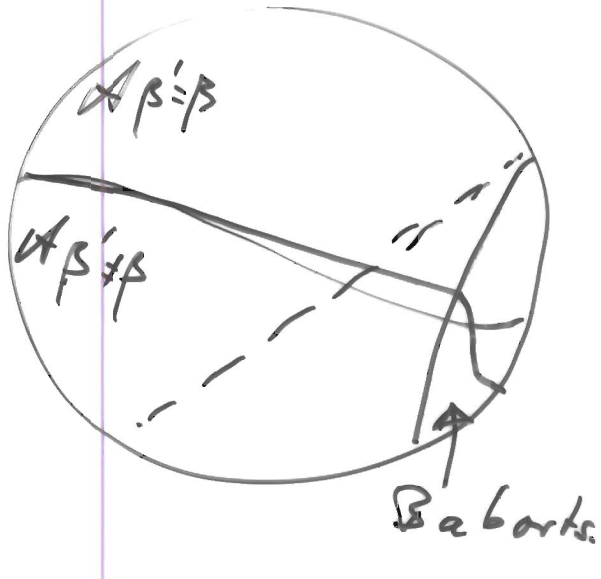
$$= c \cdot \left( U' + \sum_{v_i^*=1} U_i \right)$$

## Guess

Finally, the attacker returns a guess $\beta'$.

B will answer $\gamma' = 1$ if $\beta' = \beta$
and $\gamma' = 0$ otherwise.

Unlikely situation!




Thus B aborts in some more cases!
~~The~~ B will abort with a certain probability here. And only if it doesn't do this return the above $\gamma'$.

Waters says: the many moving that

$$\text{prob}(\ \beta' = \beta \ \mid \ \overline{\text{ABORT}}\ ) = \frac{1}{2} \text{ (or very close)}$$

even if $\text{prob}(\ \beta' = \beta\ )$ is non-negligibly away from $\frac{1}{2}$.

We define a second game to compare with.
It is essentially the 'real' challenge situation for the attacker or attack scenario with one exception:
we add aborts so that both games abort in the same situations.

<u>Setup</u>

Choose uniformly like in the real setup:

Master key $= \alpha P_2 = DH(P, P_1, P_2)$,

Pick $x', x_i, y', y_i$ and compute $U', (U_i)$ as $\mathcal{B}$ does.

<u>Phase 1, 2, Challenge</u>: as in the real game.

<u>Guess</u>:

The simulator receives $\beta'$ from the attacker.

Consider

$$\mathrm{ABORT}(x', (x_i), (v_i^{(i)}), v^*)$$

$$= \neg \Big[ \forall i = 1..q : \underbrace{K(v_i^{(i)}) = \text{false}}_{x' + \sum_{v_j^{(i)}=1} x_i \neq_m 0} \wedge x' + \sum_{v_i^*=1} x_i = k_m \Big]$$

$= \cancel{\mathrm{ABO}}$

If this expression is true we ABORT (~~and answer with a random $\beta'$~~).

With a certain probability (as in the other game) we artificially ABORT.

Otherwise we finish.

Claim

The event ABORT is the same in both simulations.

In particular, prob $(\beta' = \beta)$ are equal in both simulations.

Rus-ac 15.1.13 (4)

$\triangle$

16.1.13 (1)

Since $\mathcal{A}$ only uses $U'_i(U_i)$, but ABORT also depends on the $x'_i(x_i)$ which remain freely choosable if $U'_i(U_i)$ are fixed, we probably have

$$\text{prob}(\text{ABORT} \mid \text{view of } \mathcal{A}) = \text{prob}(\text{ABORT})$$

(or at least close to each other).

This would simplify the following, at least intuitively we would be done.

**Goal**

$$\text{prob}(\overline{\text{ABORT}}) \geq \lambda := \frac{1}{8(n+1)q}.$$

**Proof**

$$\text{prob}\Big(\bigwedge_{i=1}^{q} \big(K(v_i) = \text{false}\big) \wedge \underbrace{x' + \sum_{v_i^*=1} x_i = km}_{x' + \sum_{v_i^{(i)}=1} x_j \not\equiv_m 0}\Big)$$

$$= \Big(1 - \text{prob}\Big(\bigvee_{i=1}^{q} K(v_i) = \text{true}\Big)\Big) \cdot \text{prob}\Big(x' + \sum_{v_i^*=1} x_i = km \,\Big|\, \bigwedge_{i=1}^{q} K(v_i) = \text{false}\Big)$$

$$\geq \Big(1 - \sum_{i=1}^{q} \underbrace{\text{prob}\Big(x' + \sum_{v_j^{(i)}=1} x_j \equiv_m 0\Big)}_{\frac{1}{m}}\Big).$$

$x' \in_R \mathbb{N}_{<m}$.

$$= \Big(1 - \frac{q}{m}\Big) \underbrace{\text{prob}\Big(x' + \sum_{v_i^*=1} x_i = km \,\Big|\, \underbrace{x' + \sum_{v_j^{(i)}=1} x_j \equiv_m 0}_{K(v^*)} \wedge \bigwedge_i \overline{K(v^{(i)})}\Big)}_{\geq \frac{1}{n+1}}$$

$\geq \frac{1}{n+1}$ (because $k$ is independent of $x'$, $(x_i)$)

$$\geq \Big(1 - \frac{q}{m}\Big)\frac{1}{n+1} \, \text{prob}\Big(K(v^*) \,\Big|\, \bigwedge_i \overline{K(v^{(i)})}\Big)$$

$$= \Big(1 - \frac{q}{m}\Big)\frac{1}{n+1} \frac{\text{prob}(K(v^*))}{\text{prob}\big(\bigwedge_i \overline{K(v^{(i)})}\big)} \, \text{prob}\Big(\bigwedge_i \overline{K(v^{(i)})} \,\Big|\, K(v^*)\Big)$$

$$\geq \Big(1 - \frac{q}{m}\Big)\frac{1}{n+1} \frac{1/m}{1} \cdot \Big(1 - \text{prob}\Big(\bigvee_i K(v^{(i)}) \,\Big|\, K(v^*)\Big)\Big)$$

$$\geq \frac{1}{(n+1)m}\Big(1 - \frac{q}{m}\Big)\Big(1 - \sum_i \underbrace{\text{prob}\big(K(v^{(i)}) \mid K(v^*)\big)}_{\frac{1}{m}}\Big)$$

because $v^{(i)} \neq v^*$.

$$= \frac{1}{(n+1)m}\Big(1 - \frac{q}{m}\Big)^2$$

$$\geq \frac{1}{(n+1)m}\Big(1 - 2\frac{q}{m}\Big)$$

Choosing $m = 4q$ gives $\text{prob}(\overline{\text{ABORT}}) \geq \frac{1}{2(n+1)m} = \frac{1}{8(n+1)q} = \lambda$. $\square$

To finish we calculate

$$PR = \text{prob}(\gamma' = 0 \mid T \text{ random})$$

$$PQ = \text{prob}(\gamma' = 1 \mid \underbrace{T = e(P,P)^{abc}}_{T \text{ good}})$$

$$PR = \frac{1}{2}$$

For the other we consider

$$\text{prob}(\gamma' = 1 \mid \text{ABORT}\,\gamma \wedge T \text{ good}) = \frac{1}{2}$$

and

$$\text{prob}(\gamma' = 1 \mid \overline{\text{ABORT}}\,\gamma \wedge T \text{ good})$$

This ~~would be~~ is

$$\text{prob}(\beta' = \beta \mid \overline{\text{ABORT}} \wedge T \text{ good})$$

$$\text{prob}(\beta' = \beta \mid \overline{\text{ABORT}})$$

We know $\text{prob}(\beta' = \beta) - \frac{1}{2} = \varepsilon$ (assume $\varepsilon > 0$).

$$PQ = \text{prob}(\gamma' = 1 \mid \overline{\text{ABORT}} \wedge T \text{ good})$$

$$= \frac{1}{2} + \frac{1}{2}\Big(\text{prob}(\overline{\text{ABORT}} \mid \beta' = \beta)\,\underbrace{\text{prob}(\beta' = \beta)}_{\frac{1}{2}+\varepsilon} - \text{prob}(\overline{\text{ABORT}} \mid \beta' \neq \beta)\,\underbrace{\text{prob}(\beta' \neq \beta)}_{\frac{1}{2}-\varepsilon}\Big)$$

It's true! Compute ~~all terms~~ a bit...

$$\geq \frac{3}{2}\lambda\varepsilon \quad \text{(see appendix...)}$$

$$\geq \frac{1}{2} + \frac{3}{4}\lambda\varepsilon,$$

ie.

$$(PQ - PR) \geq \frac{3}{4}\lambda\varepsilon \geq \frac{\varepsilon}{32(n+1)q}$$

Gentry (2006)

Rus-ec
22.1.13
1

For ANON-IND-ID-CPA:

**Setup**

$G_1$, $G_T$ of prime order $\ell$, $e: G_1 \times G_1 \to G_T$ bil., efficient,

$P, Q \in_R G_1$, $\alpha \in_R \mathbb{Z}_\ell$, $P_1 = \alpha P$.

Public: $G_1, G_T, e, P, P_1, Q$.

Master key: $\alpha$.

**Key Gen**

Input: $ID \in \mathbb{Z}_\ell$

Output: $d_{ID} \in \mathbb{Z}_\ell \times G_1$ Optium remember.

1. Pick $r \in_R \mathbb{Z}_\ell$.
2. Compute $H = \frac{1}{\alpha - ID}(Q - rP)$.
3. Return $d_{ID} = (r, H)$.

**enc**

Input: $m \in G_T$, $ID \in \mathbb{Z}_\ell$.

Output: $C \in G_1 \times G_T \times G_T$.

1. Pick $s \in_R \mathbb{Z}_\ell$.
2. Compute

$$C = (sP_1 - s \cdot ID \cdot P,\ e(P,P)^s,\ m \cdot e(P,Q)^{-s})$$

**dec**

Input: $C = (C_1, C_2, C_3)$, $d_{ID} = (r, H)$

Output: FAIL or $m' \in G_T$.

1. Compute

$$m' = C_3 \cdot e(C_1, H) \cdot C_2^{\,r}.$$

2. Return $m'$.

How did Boneh arrive at this system?



Naor: IBE secure against an adaptive ID-attacker

$\Rightarrow$ signature scheme, secure against EF-CMA.

Here: the PKG is the signer, the private keys are the signatures on IDs.

Old strategy:

$\text{prob}(\overline{\text{ABORT}}) \approx \delta^q(1-\delta)$

This is maximized for $\delta \approx 1-\frac{1}{q}$ and leads to $\text{prob}(\overline{\text{ABORT}}) \in \Theta(\frac{1}{q})$.

New:
- Simulator will be able to answer all private key queries and challenges. Our simulator however will only be able to give one answer (rather than $\ell$ possible ones). $\rightsquigarrow$ tight reduction
- short parameters ✓
- recipient anonymous (✓)

Roadmap:
- Simulator picks $f \in_R \mathbb{Z}_q[x]$, $\deg f = q$, $Q = H(\ldots)$
- Private key queries by putting:

$$r_{ID} = f(ID), \quad H_{ID} = \frac{1}{x-ID}(Q - r_{ID}P) = \underbrace{\frac{f(x)-f(ID)}{x-ID}}_{\deg = q-1} \cdot P.$$

Notice that $r_{ID}$ appears random to the attacker, since $f$ was chosen randomly.

Complexity assumption

q-BDHE (bilinear DH exponent)

Given: $(Q, P, \alpha P, \alpha^2 P, \dots \alpha^q P, \alpha^{q+2} P, \dots, \alpha^{2q} P) \in G_1^{2q+1}$.

find: $e(Q, P)^{\alpha^{q+1}} \in G_T$.

q-ABDHE (augmented bilinear DH exponent)

Given: $(Q, \alpha^{q+2} Q, P, \alpha P, \alpha^2 P, \dots \alpha^q P, \alpha^{q+2} P, \dots, \alpha^{2q} P) \in G^{2q+2}$

Find: $e(Q, P)^{\alpha^{q+1}} \in G_T$.

We'll use the decisional variant of q-ABDHE.



Attack scenario

... as before only the challenge is a bit more complex:

$\mathcal{A}$ supplies $ID_0, ID_1, m_0, m_1$

The challenge oracle picks two bits $b, c \in \{0,1\}$ and encrypts $m_c$ for $ID_b$

Theorem 1

Let $q$ be the number of private key queries plus one.

Assume truncated decision $(t, \varepsilon, q)$-ABDHE assumption holds for $(G_1, G_T, e)$.

Then the scheme IBE system is $(t', \varepsilon', q-1)$ ANON-IND-ID-CPA secure

with $t' = t - O(q^2 \cdot \text{time}(\text{exp. in } G_1))$

$\varepsilon' = \varepsilon + \frac{2}{\ell}$.

Pf

We assume that $\mathcal{B}$ obtains ~~either~~

$$(Q, \alpha^{q+2}Q, P, \alpha P, \alpha^2 P, \dots \alpha^q P, Z)$$

with either $Z = e(P, Q)^{\alpha^{q+1}}$

or $Z \in_R G_T$,

and shall decide which is the case.

~~Whenever $\mathcal{A}$ queries a private key for ID then $\mathcal{B}$~~

$\mathcal{B}$ supplies the setup to $\mathcal{A}$ with

$Q = f(\alpha) P$ where it chooses $f \in \mathbb{Z}_\ell[x]$, $\deg f = q$.

When $\mathcal{A}$ queries a private key $\mathcal{B}$ computes

$$r_{ID} = f(ID),$$

$$H_{ID} = \frac{1}{\alpha - ID}(Q - r_{ID} P) = \underbrace{\frac{f(\alpha) - f(ID)}{\alpha - ID}}_{\deg = q-1 \text{ poly in } \alpha} P$$

When $\mathcal{A}$ asks for the challenge it supplies $ID_0, ID_1$ and $m_0, m_1$.

$\mathcal{B}$ picks $b, c \in_R \{0,1\}$.

Produce the private key $(r_{ID_b}, H_{ID_b})$ as above.

Let
$$f_2(x) = x^{q+2}$$
$$g_2(x) = \frac{f_2(x) - f_2(ID_b)}{x - ID_b}$$
← degree $q+1$ polynomial, normed.

and compute
$$C_1 = (f_2(\alpha) - f_2(ID_b))\, Q$$
$$= \underbrace{\alpha^{q+2} Q}_{\text{given}} - \underbrace{ID_b}_{\text{given}} \underbrace{Q}_{\text{given}}.$$

$$c_2 = Z \cdot e(Q, (g_2(\alpha) - \alpha^{q+1}) P)$$

$$c_3 = m_c \cdot e(C_1, H_{ID_b})^{-1} \cdot c_2^{-r_{ID_b}}.$$

Provided that $Z = e(P,Q)^{\alpha^{q+1}}$ this is a valid encryption of $m_c$ for $ID_b$.

To that end put
$$s = \frac{Q}{P} \cdot g_2(\alpha).$$

Then
$$C_1 = \underbrace{\frac{Q}{P} \cdot \underbrace{\frac{f_2(\alpha) - f_2(ID_b)}{\alpha - ID_b}}_{g_2(\alpha)}}_{s} \cdot (\alpha - ID_b) \frac{Q}{P} \cdot P$$

and
$$c_2 = e(P,Q)^{s} \cdot e(Q, (g_2(\alpha) - \alpha^{q+1}) P)$$
$$= e(P,Q)^{g_2(\alpha)} = e(P,P)^{s}.$$

Perfect simulation!

→ i.e. the view of the attacker is as if all $r_{ID}$ were randomly chosen!

Notice that $f$ is randomly chose and the attacker just sees

$$r := (f(\alpha), r_{ID_1}, \dots, r_{ID_q}, r_{ID}) = (f(\alpha), f(ID_1), \dots f(ID_q), f(ID))$$

This is interpolation! Given the $r$-s and $q+1$ different ID. this determines $f$ uniquely. In other words, there is a bijection between

$$r \in \mathbb{Z}_\ell^{q+1} \leftrightarrow f \in \mathbb{Z}_\ell[x]^{(q)}.$$

Remark:
should $\mathcal{A}$ ever query $ID_i = \alpha$ or $ID_0 = \alpha$ or $ID_1 = \alpha$ then $\mathcal{B}$ sees that $(\alpha P) = \alpha \cdot P$ and uses this value to answers its challenge.

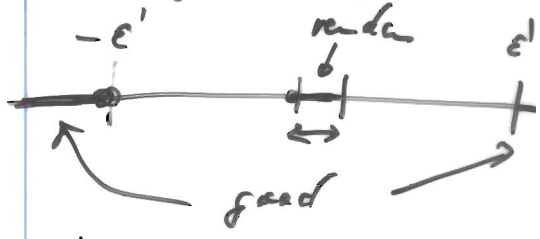
Advantage:
If $Z = e(P, Q)^{\alpha^{q+1}}$ then $\mathcal{A}$ will have advantage $\varepsilon'$ to answer its challenge → $b, c$.
In that $\mathcal{B}$ will answer that $Z$ is good.
Thus it has advantage $\varepsilon'$ here.
If $Z$ is random then $(c_1, c_2)$ are uniformly random (because $s$ is uniform and $Z$ is random
then $c_2 \neq e(C_1, P)^{\frac{1}{\alpha - ID_0}}$ and $c_2 \neq e(C_1, P)^{\frac{1}{\alpha - ID_1}}$
with prob. $1 - \frac{2}{p}$. Then $\frac{c_3}{m}$ is uniformly random

[ ... ]

Thus we loose $\frac{2}{e}$ in the advantage:

$$\left|\text{prob}\left(\mathcal{B}(\ldots) = 0\right) - \tfrac{1}{4}\right| \leq \frac{2}{e} \quad \text{if } z \text{ is random}$$

$$\left|\text{prob}\left(\mathcal{B}(\ldots) = 0\right) - \tfrac{1}{4}\right| > \varepsilon' \quad \text{if } z \text{ is good}$$




Thus

$$\left|\text{prob}\left(\mathcal{B}(\ldots, \text{good } z) = 0\right) - \text{prob}\left(\mathcal{B}(\ldots, \text{random } z) = 0\right)\right| \geq \varepsilon' - \frac{2}{e}.$$

runtime:
Just check that $\mathcal{B}$ essentially performs $q$ private key private queries and needs $q$ scalar multiplications for each, i.e. the overhead is

$$O(q^2) \text{ scalar mult. in } G_1.$$

$\square$

What to change for CCA?

Setup

...

public: $P$, $P_1 = \alpha P$, $Q_1, Q_2, Q_3$, $H$ hashfn.
masterkey: $\alpha$

KeyGen

... $d_{ID} = [\,(r_{ID,i}, H_{ID,i})\,]_{i \in \{1,2,3\}}$

with $r_{ID,i} \in_R \mathbb{Z}_e$,

$$H_{ID,i} = \frac{1}{\alpha - ID}\,(Q_i - r_{ID,i} P).$$

Enc

...

$$C = [\; sP_1 - s \cdot ID \cdot P,$$
$$e(P,P)^s,$$
$$m \cdot e(P, Q_1)^{-s},$$
$$e(P, Q_2)^s e(P, Q_3)^{s\beta} \;] =: [C_1, c_2, c_3, c_4]$$

where $\beta = H(C_1, c_2, c_3)$

$c_4$ is a signature that prevents $\mathcal{A}$ from malleating the ciphertext in order to be allowed to query the decryption oracle with the modified ciphertext.

Dec

... Put $\beta = H(C_1, c_2, c_3)$ and test $c_4 = e(C_1, H_{ID,2} H_{ID,3}^{\beta})\, c_2^{r_{ID,2} + r_{ID,3}\beta}$.
Output FAIL if this fails and otherwise return

$$m' = c_3 \cdot e(C_1, H_{ID,1}) \cdot c_2^{r_{ID,1}}.$$

    - Boneh & Boyen (2004). DOI 10.1007/978-3-540-28628-8_27. Full version: eprint 2004/173.
    - Waters (2005). DOI 10.1007/11426639_7.
    - Gentry (2006). DOI 10.1007/11761679_27.
5. More systems.
    - Short signatures (Pairing based...)
        - Boneh, Lynn & Shacham (2001) PS.
        - ...
        - Paterson & Schuldt (2006). DOI 10.1007/11780656_18.
    - Certificateless public key encryption.
        - Al-Riyami & Paterson (2003). Certificateless public key cryptography. Eprint 2003/126.
            - Zhang & Feng (2005). On the Security of a Certificateless Public-Key Encryption. @CiteSeer.
            - Au, Chen, Liu, Mu, Wong & Yang (2006). Malicious KGC Attack in Certificateless Cryptography. @CiteSeer.
            - Gorantla, Gangishetti, Das & Saxena (2005). An effective certificateless signature scheme based on bilinear pairings. @CiteSeer.
            - Huang & Wong (2007). Generic Certificateless Encryption in the Standard Model. @CiteSeer.
            - Shi, Li & Shi (2006). Constructing Efficient Certificateless Public Key Encryption with Pairing. @CiteSeer.
            - Sun & Zhang (2008). Secure Certificateless Public Key Encryption without Redundancy. @CiteSeer.
            - Lippold, Boyd & Nieto (2009). Efficient Certificateless KEM in the Standard Model. @CiteSeer.
            - Dent (2008). A survey of certicateless encryption schemes and security models. International Journal of Information Security 7(5), 349-377. eprint 2006/211.
            - Lippold & Nieto (2010). Certificateless Key Agreement in the Standard Model. AISC 2010. PDF. Preprint PDF.
    - Lee, Boyd, Dawson, Kim, Yang & Yoo (2004). Secure Key Issuing in ID-based Cryptography. @CiteSeer.
    - Waters (2009). Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. Eprint 2009/385.
6. End of course / does not fit any more:
    - Hierarchical ID based crypto
    - ...
    - Pereira, Simplício, Barreto (2011). A Family of Implementation-Friendly BN Elliptic Curves. @CiteSeer.

## Literature


- Kenny G. Paterson (2005). Cryptography from Pairings. In *I.F. Blake, G. Seroussi and N.P. Smart (eds.), Advances in Elliptic Curve Cryptography*, London Mathematical Society Lecture Note Series Vol. 317, Cambridge University Press, pp. 215-251.

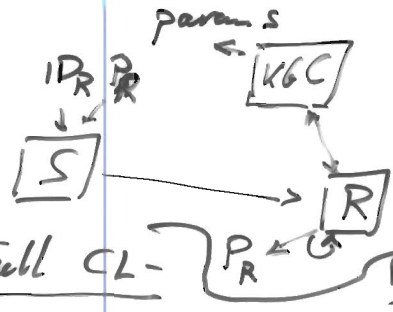

## Literature (restricted)

**Primary sources**

- ...

# Certificate less schemes

Al-Riyami & Paterson (2003)




12ws-ec
29.1.13
1

## Basic CL-PKE scheme

**Setup** in: $k$.
out: ... of prime order $\ell$

1. Prepare $G_1, G_T, e: G_1 \times G_1 \to G_T$.
2. Choose $P \in_R G_1$.
3. Select $\alpha \in_R \mathbb{Z}_\ell^\times$ and set $P_1 = \alpha P$.
4. Choose hash fns. $H_1: \{0,1\}^* \to G_1 \setminus \{0\}$,
   $H_2^T: G_T \to \{0,1\}^n$.

params $= (G_1, G_T, e, n, P, P_1, H_1, H_2^T)$

master key $= \alpha$

**Partial-Private-Key-Extract**

in: $ID \in \{0,1\}^*$, master secret
out: partial private key $D_{ID} \in G_1$.

~~1. $Q_{ID} = H_1(ID)$~~

1. return $D_{ID} = \alpha \cdot H_1(ID)$.

**Key-generate**

in: params, $ID$
out: secret $x_{ID}$, private key $S_{ID}$,
public key $P_{ID}$

1. $x_{ID} \in_R \mathbb{Z}_\ell^\times$.
2. ~~$S_{ID} = x_{ID} \cdot D_{ID} = x_{ID} \alpha H_1(ID)$~~
3. $P_{ID} = (x_{ID} P, x_{ID} P_1)$

**Encrypt**

in: $M \in \{0,1\}^n$, $ID \in \{0,1\}^*$, $P_{ID}$.
out: $C \in G_1 \times G_T$.

1. check that $P_{ID}$ is valid: $e(P_{ID,1}, P_1) = e(P_{ID,2}, P)$
   (abort if fails)
2. Pick random $r \in_R \mathbb{Z}_\ell^\times$
3. Compute and return
   $C = [rP, M \oplus H^T(e(H_1(ID), P_{ID,2}))]$

**Decrypt**

in: $C \in G_1 \times G_T$, secret key $x_{ID}, D_{ID}$
out: $M'$

$M' = C_2 \oplus H^T(e(x_{ID} D_{ID}, C_1))$

Correctness: ✓

## Full CL-PKE scheme

+5. Pick $H_3: \{0,1\}^n \times \{0,1\}^n \to \mathbb{Z}_\ell^\times$,
$H_4: \{0,1\}^n \to \{0,1\}^n$
cryptogr. hash fn.

=

=

1.
2. $\sigma \in_R \{0,1\}^n$, $r \leftarrow H_3(\sigma, M)$
3. $C = [rP, \sigma \oplus H^T(e(H_1(ID), P_{ID,2}))^r, M \oplus H_4(\sigma)]$

$\sigma' = C_2 \oplus H^T(e(x_{ID} D_{ID}, C_1))$
$M' = C_3 \oplus H_4(\sigma')$
$r' = H_3(\sigma', M')$, test $C_1 = r'P$.

That scheme is secure in the random oracle model for type I and type II attackers.

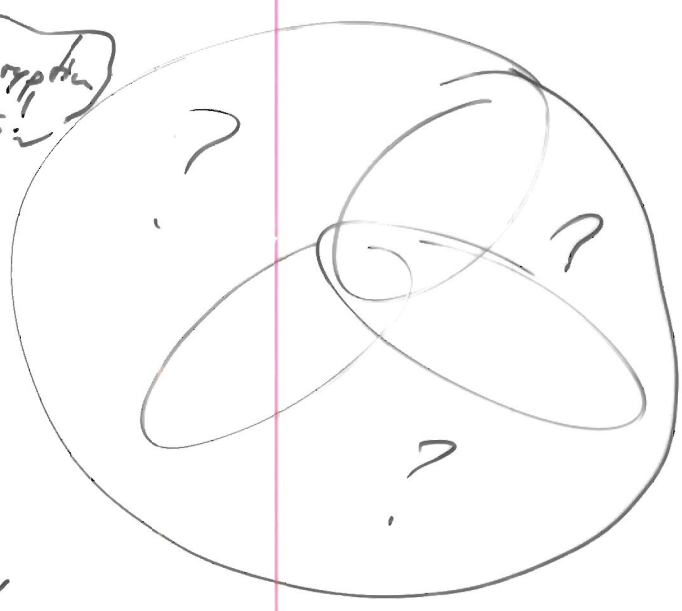
outsiders without access to the master key

insiders with access to the master key

Attackers are allowed to

| | I | II |
|---|---|---|
| • extract partial private keys D | not both for ID* | ✗ |
| • replace public keys P | not for C*, ID* | not for C*, ID* |
| • make decryption queries | not ID* | not ID* |
| • extract secret key x | not for ID whose public key has been replaced | |
| • extract master key | ✗ | ✓ |

No decryption queries!




$\boxed{\text{CPA}}$ ✓

The basic scheme is secure

if GBDH for type I

if BDH for type II

assuming $H_1, H^T$ to be random oracles.

BDH:

$(P, aP, bP, cP) \mapsto [R, e(P,R)^{abc}]$

RWS-ec
29.1.13
3

**Thm**

Assume $H_1, H^T, H_3, H_4$ are random oracles.
Suppose that GBDH problem in groups produced by the setup is intractable.
Then Full CL-PKE scheme is IND-CCA secure.

This can be made quantitative.

Leap forward from time ...

30.1.13
1

Lippold, Boyd & Nieto (2009)
Certificate less KEM.

**Setup (k)**

$U_1, U_2, A \in_R G_1 \setminus \{0\}$,
$z \leftarrow e(P, A) \in G_T$
$H : \{0,1\}^n \to G_1$ Waters hashfn:
$x \mapsto H_0 + \sum_{x_i = 1} H_i$
where $H_0, H_i \in_R G_1$.
$mpk \leftarrow (U_1, U_2, z, H)$,
$msk \leftarrow A$.
return $(mpk, msk)$

**Key Derivation (msk, ID)**

$s \in_R \mathbb{Z}_q^\times$,
$sk_{ID} \leftarrow (A + sH(ID), sP)$
return $(sk_{ID})$

**User Keygen (mpk, ID)**

$x_{ID} \in_R \mathbb{Z}_q^\times$,
$\beta_{ID} \leftarrow z^{x_{ID}}$
return $(\beta_{ID}, x_{ID})$

**Enc(mpk, $\beta_{ID}$, ID)**

$r \in_R \mathbb{Z}_q^\times$
$C_1 \leftarrow rP$
$C_2 \leftarrow r\,H(ID)$, $t \leftarrow TCR(C_1)$
$C_3 \leftarrow r(tU_1 + U_2)$,
$K \leftarrow \beta_{ID}^r = (z^{x_{ID}})^r \in G_T$
$C \leftarrow (C_1, C_2, C_3)$
return $(K, C)$

**Dec ($sk_{ID}$, $x_{ID}$, C)**

Write $(D_1, D_2) = sk_{ID}$.
$r_1, r_2 \in_R \mathbb{Z}_q^\times$.
$t \leftarrow TCR(C_1)$
$$K' \leftarrow \left(\frac{e(C_1, D_1 + r_1(tU_1 + U_2) + r_2 H_2(ID))}{e(C_2, D_2 + r_2 P)\, e(r_1 P, C_3)}\right)^{x_{ID}}$$
return $(K')$

If $e(C_1, tU_1 + U_2) = e(C_2, P)$ and $e(C_1, H(ID)) = e(P, C_3)$ then $K'$ is independent of $r_1, r_2$

## Security game

Setup & oracles as usual.
The attacker's task is to distinguish

$$(K_1, C) \leftarrow \mathrm{Enc}(\ldots, \underset{\text{chosen by the attacker}}{ID^*})$$

and

$$(K_0, C) \text{ where } K_0 \in_R G_T \setminus \{0\}.$$

~~For the secu~~

**Theorem**

Assume TCR is a target collision resistant hash fn.
(i.e. 2nd preimage resistant / tagged hash fn),
and DBDH for $P \in G_1$.
Then this scheme is secure against CCA.

**Proof** Consider

Game 0 ...

⋮

Game 5

and analyze the differences...