Provably secure non interactive key distribution based on pairings

Régis Dupont (joint work with Andreas Enge)

http://www.lix.polytechnique.fr/Labo/Regis.Dupont

1. Key agreement and identity based cryptosystems
2. Pairings and elliptic curves
3. Non interactive key exchange
4. Security proof

Key agreement and identity based cryptosystems

Key agreement issues

Diffie–Hellman (1976):

solves the key agreement/distribution problem

Key agreement issues

Diffie–Hellman (1976):

solves the key agreement/distribution problem

BUT vulnerable to man-in-the-middle attacks.

Key agreement issues

Diffie–Hellman (1976):

solves the key agreement/distribution problem

BUT vulnerable to man-in-the-middle attacks.

$\longrightarrow$ authentication is needed

Key agreement issues

Diffie–Hellman (1976):

solves the key agreement/distribution problem

BUT vulnerable to man-in-the-middle attacks.

$\longrightarrow$ authentication is needed

$\longrightarrow$ certificates are a solution

Key agreement issues

Diffie–Hellman (1976):

solves the key agreement/distribution problem

BUT vulnerable to man-in-the-middle attacks.

$\longrightarrow$ authentication is needed

$\longrightarrow$ certificates are a solution

What if Alice and Bob do not want extra communication?

Identity based cryptosystems (Shamir, 1984)

Alice

Public key: $A = H(\text{"Alice"})$

Bob

Public key: $B = H(\text{"Bob"})$

(H is a public hash function)

Identity based cryptosystems (Shamir, 1984)

Private Key Generator

Master key s (secret)

Alice

Public key: $A = H(\text{"Alice"})$

Bob

Public key: $B = H(\text{"Bob"})$

(H is a public hash function)

Identity based cryptosystems (Shamir, 1984)

Private Key Generator

Master key s (secret)

$K_A = f(A, s)$

$K_B = f(B, s)$

Alice

Public key: $A = H(\text{"Alice"})$

Private key: K_A

Bob

Public key: $B = H(\text{"Bob"})$

Private key: K_B

(H is a public hash function)

Previous work

Identity based cryptography:

- Shamir, 1984

Identity based cryptosystems:

- Maurer–Yacobi, 1992
- Kügler, 1998
- Cocks, 2001

Pairings and elliptic curves

Pairings

$(G,+)$, $(G',+)$, $(V,\times)$ groups of prime order ℓ

$$e : G \times G' \to V$$

Needed properties:

- bilinearity: $e(aP, bQ) = e(P,Q)^{ab}$
- non-degeneracy: $\exists P, Q$ s.t. $e(P,Q) \neq 1_V$
- computability

Pairings

$(G, +)$, $(G', +)$, $(V, \times)$ groups of prime order ℓ

$$e : G \times G' \to V$$

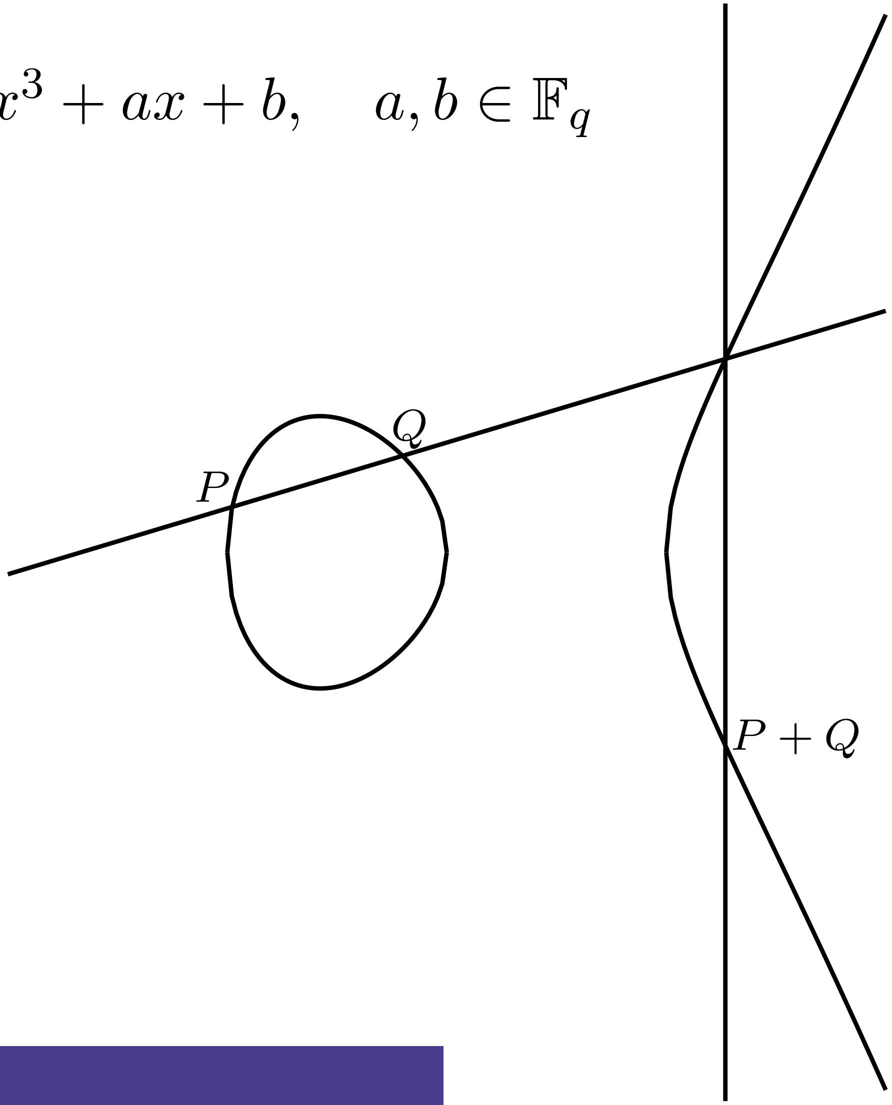
Needed properties:

- bilinearity: $e(aP, bQ) = e(P, Q)^{ab}$
- non-degeneracy: $\exists P, Q$ s.t. $e(P, Q) \neq 1_V$
- computability

$$\left.\begin{array}{r} H : \{0,1\}^* \to G \\ H' : \{0,1\}^* \to G' \end{array}\right\} \text{hash functions}$$

Elliptic curves

$$E : y^2 = x^3 + ax + b, \quad a, b \in \mathbb{F}_q$$



Weil and Tate pairings on elliptic curves

ℓ prime, $\ell \mid \#E$ and $\ell \mid q^k - 1$,

$\Rightarrow \exists P \in E(\mathbb{F}_q), Q \in E(\mathbb{F}_{q^k})$ that generate $E[\ell]$

Weil and Tate pairings:

$$e : \langle P \rangle \times \langle Q \rangle \to \mu_\ell \subseteq \mathbb{F}_{q^k}^\times$$

- bilinear
- non-degenerate
- efficiently computable if k is small

Previous work

Use of pairings in cryptosystems:

- Cryptosystems based on pairings (Sakaï–Ohgishi–Kasahara, 2000)
- Tripartite Diffie–Hellman (Joux, 2000)
- Identity based encryption (Boneh–Franklin, 2001)

Non interactive key exchange

Non interactive key exchange

Alice

$P_A = H(\text{“Alice”})$

$Q_A = H'(\text{“Alice”})$

Bob

$P_B = H(\text{“Bob”})$

$Q_B = H'(\text{“Bob”})$

Non interactive key exchange

Private Key Generator

Master key $s \in_R [0, \ell - 1]$

sP_A, sQ_A

sP_B, sQ_B

Alice

$P_A = H(\text{"Alice"})$

$Q_A = H'(\text{"Alice"})$

Bob

$P_B = H(\text{"Bob"})$

$Q_B = H'(\text{"Bob"})$

Non interactive key exchange

Private Key Generator

Master key $s \in_R [0, \ell - 1]$

sP_A, sQ_A

sP_B, sQ_B

Alice

$P_A = H(\text{“Alice”})$

$Q_A = H'(\text{“Alice”})$

$e(P_B, sQ_A)$

Bob

$P_B = H(\text{“Bob”})$

$Q_B = H'(\text{“Bob”})$

$e(sP_B, Q_A)$

Non interactive key exchange

Private Key Generator

Master key $s \in_R [0, \ell - 1]$

sP_A, sQ_A

sP_B, sQ_B

Alice

$P_A = H(\text{"Alice"})$

$Q_A = H'(\text{"Alice"})$

Bob

$P_B = H(\text{"Bob"})$

$Q_B = H'(\text{"Bob"})$

$$e(P_B, sQ_A) = e(P_B, Q_A)^s = e(sP_B, Q_A)$$

Security proof

Elementary security

We need:

- DLP in G, G', V hard (otherwise master key s)

Elementary security

We need:

- DLP in G, G', V hard (otherwise master key s)
- Generalized bilinear Diffie–Hellman problem (GBDH)

  given P, Q, aP, bQ, cP, cQ, compute $e(P,Q)^{abc}$

  (otherwise shared keys)

  $\Rightarrow$ the pairing inversion problem

  given P and $e(P,Q)$, compute Q

  should be hard

Attack scenario (computational version)

- System parameters are known.
- Private key queries (adaptive):
  Attacker $\mathcal{C}$ chooses identities ID_i.
  $\mathcal{C}$ receives private keys $sH(\mathrm{ID}_i)$, $sH'(\mathrm{ID}_i)$.
- Guess:
  $\mathcal{C}$ chooses two identities $\notin \{\mathrm{ID}_i\}$
  and outputs their shared key.

Equivalence of GBDH and security

random oracle model

If an attacker $\mathcal{C}$ is able to forge a valid guess
with N private key queries, in time t and
with probability ε,

then GBDH problem can be solved
in time $t' = \mathrm{poly}(t)$ and
with probability $O(\varepsilon/(1+N)^2)$.

Solving GBDH from the attacker $\mathcal{C}$

random oracle model

- H, H' look like random functions to $\mathcal{C}$

Solving GBDH from the attacker $\mathcal{C}$

random oracle model

- H, H' look like random functions to $\mathcal{C}$
- to evaluate H and H', $\mathcal{C}$ consults an oracle/black box

Solving GBDH from the attacker $\mathcal{C}$

random oracle model

- H, H' look like random functions to $\mathcal{C}$
- to evaluate H and H', $\mathcal{C}$ consults an oracle/black box

We want to design an algorithm $\mathcal{G}$ that obtains

$$e(P,Q)^{abc} \text{ from } (P,Q,aP,bQ,cP,cQ)$$

using the attacker $\mathcal{C}$

- $\mathcal{G}$ plays the role of the oracle for H and H'

Solving GBDH from the attacker $\mathcal{C}$

random oracle model

- H, H' look like random functions to $\mathcal{C}$
- to evaluate H and H', $\mathcal{C}$ consults an oracle/black box

We want to design an algorithm $\mathcal{G}$ that obtains

$$e(P,Q)^{abc} \text{ from } (P,Q,aP,bQ,cP,cQ)$$

using the attacker $\mathcal{C}$

- $\mathcal{G}$ plays the role of the oracle for H and H'
- $\mathcal{G}$ plays the role of the PKG for $\mathcal{C}$ (simulates a master key c)

Solving GBDH from the attacker $\mathcal{C}$

- Evaluation of H and H' at ID
  - $\mathcal{G}$ picks $k, k' \in_R [0, \ell - 1]$ and sets
  - $H(\mathrm{ID}) = kP$ (prob. δ) or $H(\mathrm{ID}) = kaP$ (prob. $1 - \delta$)
  - $H'(\mathrm{ID}) = k'Q$ (prob. δ) or $H'(\mathrm{ID}) = k'bQ$ (prob. $1 - \delta$)

Solving GBDH from the attacker $\mathcal{C}$

- Evaluation of H and H' at ID
  - $\mathcal{G}$ picks $k, k' \in_R [0, \ell - 1]$ and sets
  - $H(\mathrm{ID}) = kP$ (prob. δ) or $H(\mathrm{ID}) = kaP$ (prob. $1 - \delta$)
  - $H'(\mathrm{ID}) = k'Q$ (prob. δ) or $H'(\mathrm{ID}) = k'bQ$ (prob. $1 - \delta$)
- Private key queries:

  when $\mathcal{C}$ asks for the private key for ID
  - if $H(\mathrm{ID}) = kP$ and $H'(\mathrm{ID}) = k'Q$ (prob. δ^2),

    $\mathcal{G}$ returns $(kcP, k'cQ) = (cH(\mathrm{ID}), cH'(\mathrm{ID}))$
  - otherwise $\mathcal{G}$ fails

LIX

Solving GBDH from the attacker $\mathcal{C}$

- if $\mathcal{C}$ outputs a correct guess (prob. $\geq \varepsilon$)

$$\mathrm{ID}_1,\ \mathrm{ID}_2,\ e(H(\mathrm{ID}_1), H'(\mathrm{ID}_2))^c$$

  - if $H(\mathrm{ID}_1) = kaP$ and $H'(\mathrm{ID}_2) = k'bQ$ (prob. $(1-\delta)^2$)
    then $e(H(\mathrm{ID}_1), H'(\mathrm{ID}_2))^c = e(P, Q)^{kk'abc}$
    so $\mathcal{G}$ can output $e(P, Q)^{abc}$
  - otherwise $\mathcal{G}$ fails

Solving GBDH from the attacker $\mathcal{C}$

- if $\mathcal{C}$ outputs a correct guess (prob. $\geq \varepsilon$)

$$\mathrm{ID}_1,\ \mathrm{ID}_2,\ e(H(\mathrm{ID}_1), H'(\mathrm{ID}_2))^c$$

  - if $H(\mathrm{ID}_1) = kaP$ and $H'(\mathrm{ID}_2) = k'bQ$ (prob. $(1-\delta)^2$)
    then $e(H(\mathrm{ID}_1), H'(\mathrm{ID}_2))^c = e(P, Q)^{kk'abc}$
    so $\mathcal{G}$ can output $e(P, Q)^{abc}$
  - otherwise $\mathcal{G}$ fails

Global success probability: $p \geq ((1-\delta)\delta^N)^2\varepsilon$
Setting $\delta = N/(1+N)$ we get

$$p \geq \frac{\varepsilon}{2(\exp(1)(1+N))^2}$$

Conclusions and outlook

- efficient non interactive key distribution scheme

Conclusions and outlook

- efficient non interactive key distribution scheme
- security proof:
  reduction to a well identified computational problem (GBDH)
  (random oracle model)

Conclusions and outlook

- efficient non interactive key distribution scheme
- security proof:
  reduction to a well identified computational problem (GBDH)
  (random oracle model)
- distribution of the master key and "hierarchization" are possible (Gentry–Silverberg)

Conclusions and outlook

- efficient non interactive key distribution scheme
- security proof:
  reduction to a well identified computational problem (GBDH)
  (random oracle model)
- distribution of the master key and "hierarchization" are possible (Gentry–Silverberg)
- not limited to supersingular curves anymore
  ordinary curves with small k
  (Miyaji–Nakabayashi–Takano, Barreto–Lynn–Scott, D.–Enge–Morain)