Pairing Based Cryptography

An Introduction

Seminar, e-Voting Group, BFH

Biel/Bienne, May 24, 2011

Stephan Krenn[1]

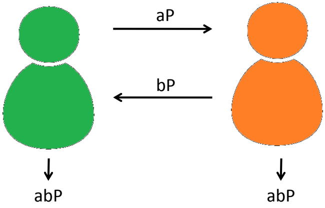
[1]Bern University of Applied Sciences

First of all ...

2002 :	652
2003 :	815
2004 :	1'113
2005 :	1'398
2006 :	1'650
2007 :	1'655
2008 :	1'779
2009 :	1'288
2010 :	525
2011 :	94

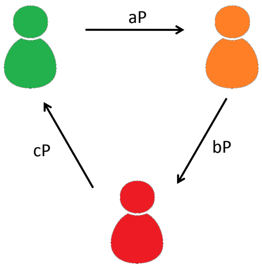
More than
11'000
publications
within 10 years!

Two-Party Key Exchange

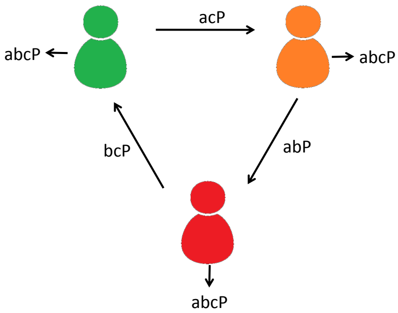


- Standard Diffie-Hellman key exchange on elliptic curves.

Two-Round Three-Party Key Exchange

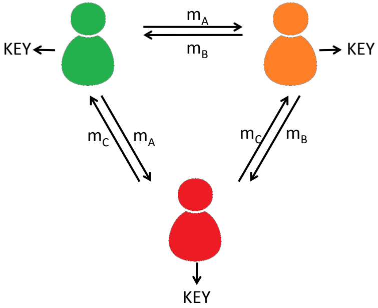


Two-Round Three-Party Key Exchange



- Assumed to be as secure as Diffie-Hellman
- Two *synchronized* messages per party.

What we Want to Have



Outline

Definition

For the whole talk let $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_T$ be groups of prime order q.

Definition

A mapping $e(.,.) : \mathcal{G}_1 \times \mathcal{G}_2 \to \mathcal{G}_T$ is called a *(bilinear) pairing*, if the following conditions are satisfied:

Bilinearity:
$$e(P+Q,R) = e(P,R)e(Q,R) \quad \forall P,Q \in \mathcal{G}_1, \forall R \in \mathcal{G}_2,$$
$$e(P,R+S) = e(P,R)e(R,S) \quad \forall P \in \mathcal{G}_1, \forall R,S \in \mathcal{G}_2.$$

Non-degeneracy: $\exists (P,Q) \in \mathcal{G}_1 \times \mathcal{G}_2 : e(P,R) \neq 1.$

Computability: $e(.,.)$ can be evaluated efficiently.

Basic Properties

Lemma

Let $e(.,.) : \mathcal{G}_1 \times \mathcal{G}_2 \to \mathcal{G}_T$ be a bilinear pairing. Then the following holds for all $P, Q \in \mathcal{G}_1$ and $R, S \in \mathcal{G}_2$:

(a) $e(P, \infty) = e(\infty, R) = 1$,

(b) $e(P, -R) = e(-P, R) = e(P, R)^{-1}$,

(c) $e(aP, bR) = e(P, R)^{ab}$ *for all* $a, b \in \mathbb{Z}$,

(d) $\langle P \rangle = \mathcal{G}_1$ *and* $\langle R \rangle = \mathcal{G}_2 \quad \Rightarrow \quad \langle e(P, R) \rangle = \mathcal{G}_T$,

(e) $f(X) = e(X, R)$ *is a homomorphisms from* $\mathcal{G}_1$ *to* $\mathcal{G}_T$*, and an isomorphism for* $R \neq \infty$.

(f) *For an isomorphism* $\psi : \mathcal{G}_1 \to \mathcal{G}_2$, $e(P, \psi(Q)) = e(Q, \psi(P))$.

From now on, we assume that $\mathcal{G}_1 = \mathcal{G}_2$.

Bilinear Diffie-Hellman Assumptions

Definition (Bilinear Diffie-Hellman (BDH) Assumption)

Given: (P, aP, bP, cP) with $a, b, c \in_R \mathbb{Z}_q^*$
Required: $e(P, P)^{abc}$
The *BDH assumption* says that the advantage of every PPT algorithm is at most negligibly better than guessing.

Definition (Decisional BDH (DBDH) Assumption)

Given: (P, aP, bP, cP, r) with $a, b, c \in_R \mathbb{Z}_q^*$, and $r \begin{cases} \in_R \mathcal{G}_T \\ = e(P, P)^{abc} \end{cases}$
Required: $e(P, P)^{abc} \stackrel{?}{=} r$
The *DBDH assumption* says that the success probability of every PPT algorithm is at most negligibly larger than $1/2$.

Co-Gap Diffie-Hellman Groups

Let $P \in \mathcal{G}_1$, $R \in \mathcal{G}_2$ be generators, and $\psi(.) : \mathcal{G}_1 \to \mathcal{G}_2$ be an isomorphism with $\psi(P) = R$.

Definition (Co-Diffie-Hellman (Co-DH) Problems)

- **Decisional Co-DH (D-Co-DH) Problem**

  Given: (P, R, aP, bR, cR) with $a, b \in_R \mathbb{Z}_q^*$ and $c \begin{cases} \in_R \mathcal{G}_2 \\ = ab \mod q \end{cases}$

  Required: $ab \stackrel{?}{=} c \mod q$

- **Computational Co-DH (C-Co-DH) Problem**

  Given: (P, R, aP, bR) with $a, b \in_R \mathbb{Z}_q^*$

  Required: abR

Definition (Co-Gap Diffie-Hellman (Co-GDH) Groups)

$\mathcal{G}_1, \mathcal{G}_2$ are said to be *Co-GDH groups* if D-Co-DH can be solved efficiently but C-Co-DH can not.

Other Problems

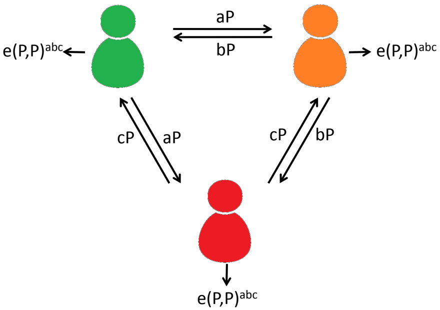
- k-Bilinear Diffie-Hellman Inversion:
  Given $P, aP, a^2P, \ldots, a^kP$, compute $e(P,P)^{\frac{1}{a}}$.
- k-Decisional Bilinear Diffie-Hellman Inversion:
  Distinguish $P, aP, a^2P, \ldots, a^kP, e(P,P)^{\frac{1}{a}}$ from
  $P, aP, a^2P, \ldots, a^kP, e(P,P)^b$
- Decisional Hash Bilinear Diffie-Hellman Problem:
  Given P, aP, bP, cP, r and a hash function H decide whether
  $r = H(e(P,P)^{abc})$.
- $\ldots$

Outline

- Encryption schemes
  - (Hierarchical) ID-based encryption
  - **Searchable public key encryption**
  - (ID-based) Threshold decryption
- Signature schemes
  - Blind signatures
  - **Short signatures**
  - **Ring signatures**
  - Verifiable committed signatures ($\approx$non-interactive fair exchange)
  - (Hierarchical) ID-based variants of the above
  - Threshold signatures
- Miscellaneous
  - **Key exchange**
  - Signcryption
  - Identification schemes
  - (ID-based) chameleon hashes

One-Round Three-Party Key Exchange

Joux



- No synchronization needed any more, thus "*one round*".

Short Signatures

Boneh, Lynn, Shacham

Let $e(.,.) : \mathcal{G}_1 \times \mathcal{G}_1 \rightarrow \mathcal{G}_T$ be a bilinear pairing and $H(.) : \{0,1\}^* \rightarrow \mathcal{G}_1$ a hash function.

KeyGen Let $\langle P_1 \rangle = \mathcal{G}_1$. Let $x \in_R \mathbb{Z}_q^*$ be the secret key, and $Y = xP_1$ be the public key.

Sign To sign a message M, the user computes $\sigma = xH(M)$.

Verifiy The receiver accepts, iff $(P_1, Y, H(M), \sigma)$ is a Diffie-Hellman tuple, i.e., iff $e(P_1, \sigma) = e(Y, H(M))$.

Lemma

If $\mathcal{G}_1$ is a GDH group, the scheme is secure against existential forgery under adaptive chosen message attacks in the ROM.

Searchable Public Key Encryption

Boneh, Crescenzo, Ostrovsky, Persiano

Idea: add a list of encrypted tags to a ciphertext such that, e.g., a mail gateway can route an email to the right device. That is, for a list of tags $W_1, \ldots, W_n$, Bob sends

$$E_{A_{pub}}(M) \| S(A_{pub}, W_1) \| \ldots \| S(A_{pub}, W_n)$$

to Alice.
The gateway can check whether $W_i = W$ for a predefined key word, but does not learn anything if this is not the case.

Searchable Public Key Encryption

Let $e(.,.) : \mathcal{G}_1 \times \mathcal{G}_1 \to \mathcal{G}_T$ be a bilinear map, and let $H_1(.) : \{0,1\}^* \to \mathcal{G}_1$, $H_2(.) : \mathcal{G}_2 \to \{0,1\}^l$ be hash functions.

KeyGen: Let P be a public generator of $\mathcal{G}_1$, and let $s \in \mathbb{Z}_q^*$ be Alice's secret key. Her public key is given by $A_{pub} = sP$. Give $T_W = sH_1(W)$ as a trapdoor to the gateway.

Encrypt: Draw $r \in_R \mathbb{Z}_q^*$ and set $S(A_{pub}, W) = (U, V) = (rP, H_2(e(H_1(W), A_{pub})^r))$.

Test: Output yes, iff $V = H_2(e(T_W, U))$.

Lemma

Under the BDH assumption, the above scheme is semantically secure against chosen keyword attacks in the random oracle model.

Bilinear Ring Signatures

Boneh, Gentry, Lynn, Shacham

Idea: A ring signature allows to sign a document on behalf of a group without revealing the identity of the signer while guaranteeing the correctness of the signature.

Bilinear Ring Signatures

Let $e(.,.) : \mathcal{G}_1 \times \mathcal{G}_2 \to \mathcal{G}_T$ be a bilinear map. Further, let $\psi(.) : \mathcal{G}_1 \to \mathcal{G}_2$ be a computable isomorphism, and $H(.) : \{0,1\}^* \to \mathcal{G}_2$ be a hash function.

KeyGen Let $\langle P_i \rangle = \mathcal{G}_i$ for $i = 1, 2$, $P_2 = \psi(P_1)$. Let $x_i \in \mathbb{Z}_q^*$ be the secret key and $V_i = x_i P_1$ be the public key of user $i = 1, \dots, n$.

Sign To sign message M, user j draws $a_i \in_R \mathbb{Z}_q^*$ for $i \neq j$, and outputs the signature $\sigma = (\sigma_1, \dots, \sigma_n)$, where $\sigma_j = \frac{1}{x_i}\left(H(M) - \psi(\sum_{i \neq j} a_i V_i)\right)$ and $\sigma_i = a_i P_2 \; \forall i \neq j$.

Verify The receiver accepts, iff $e(P_1, H(M)) = \prod_{i=1}^{n} e(V_i, \sigma_i)$.

Lemma

Under the Co-GDH assumption the above scheme unconditionally protects the signer's identity, and is resistent to forgery in the ROM.

Outline

Elliptic Curves

Definition

Let $\mathbb{K}$ be a finite field with $\operatorname{char}\mathbb{K} \neq 2, 3$. Let $\overline{\mathbb{K}}$ be the algebraic closure of $\mathbb{K}$, and let $a, b \in \overline{\mathbb{K}}$.

An *elliptic curve* $\mathcal{E}$ is given by ∞ and all $(x, y) \in \overline{\mathbb{K}}^2$ satisfying

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

Lemma

With the tangent-and-chord-method, $\mathcal{E}$ becomes a group.

Some further notation:

- $\mathcal{E}[n] = \{P \in \mathcal{E} : nP = \infty\}$
- $\overline{\mathbb{K}}[\mathcal{E}] = \overline{\mathbb{K}}[x, y]/(y^2 - x^3 - ax^2 - b)$ (ring)
- $\overline{\mathbb{K}}(\mathcal{E}) = \left\{ \frac{f(x,y)}{g(x,y)} : f, g \in \overline{\mathbb{K}}[\mathcal{E}] \right\}$ (field)

Zeros and Poles

For every $P \in \mathcal{E}$ there exists $u \in \overline{\mathbb{K}}(\mathcal{E})$ with $u(P) = 0$ such that for every $f \in \overline{\mathbb{K}}(\mathcal{E})$ there is $d \in \mathbb{Z}$ such that fu^d is defined and $\neq 0$.

Definition

For $P \in \mathcal{E}$ and $f \in \overline{\mathbb{K}}(\mathcal{E})$ we define $\mathrm{ord}_P(f) = d$.
If $d > 0$ we call P a *zero of multiplicity* d.
If $d < 0$ we call P a *pole of multiplicity* $-d$.

Divisors

Definition

A *divisor* D is a formal sum $D = \sum_{P \in \mathcal{E}} n_P(P)$.

- support of D: $\mathrm{supp}(D) = \{P \in \mathcal{E} : n_P \neq 0\}$
- degree of D: $\deg(D) = \sum_{P \in \mathcal{E}} n_P$
- for $f \in \overline{\mathbb{K}}(\mathcal{E})$ we set $\mathrm{div}(f) = \sum_{P \in \mathcal{E}} \mathrm{ord}_P(f)(P)$
- we write $D_1 \sim D_2$: $\Leftrightarrow \exists f \in \overline{\mathbb{K}}(\mathcal{E}) : D_1 = D_2 + \mathrm{div}(f)$
- D is *principal*: $\Leftrightarrow \exists f \in K(\mathcal{E}) : \mathrm{div}(f) = D$

Lemma

D is a principal divisor, iff $\deg(D) = 0$ *and* $\sum_{P \in \mathcal{E}} n_P P = \infty$.

The Weil Pairing

Definition

For $\gcd(m, p) = 1$ and $(S, T) \in \mathcal{E}[m] \times \mathcal{E}[m]$ let A, B be divisors with

- $\sum_{p \in \mathcal{E}} n_{AP}(P) = A \sim (S) - (\infty)$,
- $\sum_{p \in \mathcal{E}} n_{BP}(P) = B \sim (T) - (\infty)$, and
- $\mathrm{supp}(A) \cap \mathrm{supp}(B) = \emptyset$.

Let further $f_A, f_B \in \mathcal{E}(\overline{\mathbb{K}})$ such that $\mathrm{div}(f_A) = mA$ and $\mathrm{div}(f_B) = mB$.

Then the *Weil pairing* is defined by

$$e_W : \mathcal{E}[m] \times \mathcal{E}[m] \to \mu_m : (S, T) \mapsto \frac{f_A(B)}{f_B(A)},$$

where $f_A(B) = \prod_{P \in \mathrm{supp}(B)} f_A(P)^{n_{BP}}$ and similar for $f_B(A)$, and $\mu_m \subseteq \mathbb{K}$ denotes the set of m^{th} roots of unity.

Comparison to Tate Pairing

- Tate pairing is much more complex to understand.
- Weil pairing has more restrictive conditions on curves (in theory).
- Weil pairing is twice as expensive as Tate pairing.
- Tate pairing maps to equivalence classes, not to single values.

Parameter Selection

- Let $q = p^i$ for $p \in \mathbb{P}$ and let $\mathcal{E}$ be defined over $\mathbb{F}_q$.
- Let $m \in \mathbb{P}$ and let k be the least integer with $\mathcal{E}[m] \subseteq \mathcal{E}(\mathbb{F}_{q^k})$.
- Then $\mathcal{G}_1 = \mathcal{G}_2 = \mathcal{E}[m]$ and $\mu_m \subseteq \mathbb{F}_{q^k}$.
- m, k should be large enough for DLP to be hard in $\mathcal{E}[m]$ and $\mathbb{F}_{q^k}$.
- k should be small enough for computations in $\mathbb{F}_{q^k}$ to be efficient.
- The smaller q, the shorter are elements of $\mathcal{E}[m]$.
- For 128 bit security: $m \approx 2^{256}$, $q^k \approx 2^{3072}$.

- Super-singular elliptic curves ($q + 1 - \#\mathcal{E}(\mathbb{F}_q) = 0 \mod p$) always have embedding degree ≤ 6.
- Elliptic curves for any k and any m can be generated using the Cocks-Pinch method.

Efficiency of Pairing Based Cryptography

- For 128 bit security one should (**very** roughly) use parameters such that:

	$\lvert\log(q)\rvert$	$\lvert P \in \mathcal{G}_1\rvert$	$\lvert R \in \mathcal{G}_2\rvert$	$\lvert T \in \mathcal{G}_T\rvert$
$\mathcal{G}_1 = \mathcal{G}_2$	512	512	512	$6 \cdot 512$
$\mathcal{G}_1 \neq \mathcal{G}_2$	256	256	$3 \cdot 256$	$6 \cdot 256$

- Costs for computing pairings is of the same order as exponentiation (cubic).
- A single pairing costs as much as 4 to 20 mod-exps.

Things are Getting Better

CPU Cycles per Pairing

(all implementations optimized for optimal Ate pairing on Core i5/i7)

IOS Press 2008	10’000’000
LATINCRYPT 2010	4’380’000
PAIRING 2010	2’333’000
EUROCRYPT 2011	1’688’000