Pairings II

Pairing based protocols

Tanja Lange

Eindhoven University of Technology

2MMC10 – Cryptology

One round tripartite key exchange

Joux, ANTS 2000

Let P, P' be generators of G_1 and G_2 respectively.
Users A, B and C compute joint secret from their secret contributions a, b, c as follows (A's perspective)

- Compute and send aP, aP'.
- Upon receipt of bP and cP' put $k = (e(bP, cP'))^a$

The resulting element k is the same for each participant as

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

- Only one user needs to compute in G_1 and G_2,
  but determining who would require communication round.
- Obvious saving in first step if $G_1 = G_2$.

ID-based cryptography

Idea: sender decides public key of receiver using receiver ID.

Consequences

- Advantage if recipient is not in system or sender wants to force use of a fresh key (e.g., using [ID,date]).
- Disadvantage: Set-up requires a trusted authority (TA) which can compute the secret key for a given public key.
  Some places see this as an advantage . . .

ID-based cryptography using pairings

Sakai-Ohgishi-Kasahara 2000, Boneh and Franklin, Crypto 2001

Let $H : \{0,1\}^* \rightarrow G_2$ be hash function.

Master secret key of TA is s, public key is $P_{pub} = sP$.

Encryption:

- Choose ID string ID and compute $H(ID) \in G_2$.
- Choose random nonce k and compute $R = kP$.
- Compute $c = \mathrm{KDF}((e(P_{pub}, H(ID)))^k) \oplus m$ and send (R, c).

Decryption:

- Obtain secret key $S' = s(H(ID)) \in G_2$ from TA.
- Compute $m' = \mathrm{KDF}(e(R, S')) \oplus c$

ID-based cryptography using pairings

Sakai-Ohgishi-Kasahara 2000, Boneh and Franklin, Crypto 2001

Let $H : \{0,1\}^* \rightarrow G_2$ be hash function.

Master secret key of TA is s, public key is $P_{pub} = sP$.

Encryption:

- Choose ID string ID and compute $H(ID) \in G_2$.
- Choose random nonce k and compute $R = kP$.
- Compute $c = \mathrm{KDF}((e(P_{pub}, H(ID)))^k) \oplus m$ and send (R, c).

Decryption:

- Obtain secret key $S' = s(H(ID)) \in G_2$ from TA.
- Compute $m' = \mathrm{KDF}(e(R, S')) \oplus c$

Decryption works:

$$e(R, S') = e(kP, s(H(ID))) = (e(P, H(ID)))^{ks} = (e(sP, H(ID)))^k$$

$$\Rightarrow m' = \mathrm{KDF}(e(R, S')) \oplus c = \mathrm{KDF}((e(P_{pub}, H(ID)))^k) \oplus c = m$$

Security assumptions

Clearly these systems require that the DLP is hard in the groups.

Additionally we define the following computational and decisional problems:

- Computational Bilinear Diffie-Hellman Problem (CBDHP): Compute $(abc)P$ given aP, bP, cP and P
- Decisional Bilinear Diffie-Hellman Problem (DBDHP): Decide whether $rP = (abc)P$ given P, aP, bP, cP and rP.

Distortion maps

An injective homomorphism $\phi : G_1 \to G_2$ is called a distortion map.

- Distortion maps allow to state the protocols as using bilinear map
$$G_1 \times G_1 \to G.$$
Systems are easier to state.
- Tripartite DH needs only public keys in G_1.
- Typically computations in G_1 cheaper than in G_2;
save computation time by computing in G_1 and using ϕ.
(If ϕ is efficient).
- Really short signatures (see next slide).

Distortion maps

An injective homomorphism $\phi : G_1 \to G_2$ is called a distortion map.

- Distortion maps allow to state the protocols as using bilinear map
$$G_1 \times G_1 \to G.$$
Systems are easier to state.
- Tripartite DH needs only public keys in G_1.
- Typically computations in G_1 cheaper than in G_2;
save computation time by computing in G_1 and using ϕ.
(If ϕ is efficient).
- Really short signatures (see next slide).
- DDHP in G_1 is broken, see Pairings I.

Short signatures

Boneh, Lynn, and Shacham, Asiacrypt 2001

Requires hash function $H : \{0,1\}^* \rightarrow G_1$.

KeyGen:

- Pick random $0 < a < \ell$, compute aP.
- Output public key $Q = aP$, private key a.

Sign message m:

- Compute $S = a(H(m))$.
- Send S.

Verify:

- Accept if

$$e(Q, H(m)) = e(P, S).$$

Can compress point to one coordinate and sign bit, so need only one $\mathbf{F}_p$ element + 1 bit, i.e. of half the length of ECDSA.