TEPLAでIDベース暗号プログラミング

ID-based cryptography programming with TEPLA

注意

- 必要な前提知識
  - 楕円曲線暗号
  - ペアリングとIDベース暗号

IDベース暗号概要

IDベース暗号とは

- 公開鍵は秘密鍵を生成してから、公開鍵を生成していた
  - 点 P と秘密鍵 a から公開鍵 aP を生成
- IDベース暗号は任意の文字列が公開鍵
  - メールアドレスとか
- とりあえず鍵共有の流れを説明
  - 暗号化・復号は後ほど

IDベース暗号の流れ

- 登場人物
  - 鍵生成局 / Aさん / Bさん

IDベース暗号の流れ(鍵生成)

- 鍵生成局が生成するもの
  - [公開] 楕円曲線 E
  - [公開] IDの集合からEへの中への ハッシュ関数 $H : \{ ID \} \rightarrow E$
  - [非公開] マスター秘密鍵 $s (\in Z)$
  - [秘密裏に配布] 秘密鍵 $K_x := sP_x = sH(X)$
    - XはXさんのID
    - $P_x := H(X)$ で誰でも計算できるE上の点

IDベース暗号の流れ(鍵共有)

- Aさん
  - 秘密鍵 K_A とP_B を使って$s_A = e(K_A, P_B)$ を計算する
- Bさん
  - 秘密鍵 K_B とP_A を使って$s_B = e(P_A, K_B)$ を計算する
- このとき、ペアリングのとり方の順序に注意
  - たとえば ID に順序関係を入れて小さい方の添え字を前にするといった取り決めが必要
  - 今回の場合は $A < B$

IDベース暗号の流れ(鍵共有)

$$s_A = e(K_A , P_B) = e(sP_A , P_B)$$
$$= \mathbf{e(P_A , P_B)^s}$$
$$= e(P_A , sP_B) = e(P_A , K_B)^s = s_B$$

s_Aとs_Bから$e(P_A, P_B)^s$を作れるのが重要

IDベース暗号の暗号化と復号

IDベース暗号の流れ(鍵生成)

- 鍵生成局が生成するもの
  - [公開] 楕円曲線 E
  - [公開] IDの集合からEへのハッシュ関数 $H_1 : \{ ID \} \rightarrow E$
  - [公開] Eから有限体へのハッシュ関数$h_2 : E \rightarrow \mathbb{F}_p$
  - [非公開] マスター秘密鍵 s ($\in$ Z)
  - [公開] E上の点PとsP
  - [秘密裏に配布] 秘密鍵 $K_x := sP_x = sH_1(X)$
    - XはXさんのID
    - $P_x := H_1(X)$ で誰でも計算できるE上の点

IDベース暗号の流れ(暗号化)

- ユーザAに送る / 平文m、乱数r を決める
- $\mathrm{Enc}(m) := (rP,\ m \oplus h_2(e(P_A,\ sP)^r))$
  - ⊕は排他的論理和(XOR)

IDベース暗号の流れ(復号)

- $Enc(m) := (rP, m \oplus h_2(e(P_A, sP)^r))$ -> 暗号文 C = (U, v)

- $Dec(C) = Dec(U, v) = v \oplus h_2(e(K_A, U))$

  $= (m \oplus h_2(e(P_A, sP)^r)) \oplus h_2(e(K_A, rP))$

  $= m \oplus h_2(e(K_A, rP)) \oplus h_2(e(K_A, rP))$

  $= m$

TEPLA

TEPLA

- ペアリングの実装に際し、TEPLAというライブラリを使用する
- TEPLA (University of Tsukuba Elliptic Curve and Pairing Library) はC言語で利用するソフトウェアライブラリ。ペアリングを使った暗号プロトコルの実装に必要となる様々な機能を備えている。

TEPLA

- 提供されている演算
  - 有限体上の元の演算（254ビットの素体と2次・12次拡大体）
    - 加算、減算、乗算（積）、逆元、2乗根、べき乗
      ランダムな元の生成
  - 楕円曲線上の点の演算（Barreto-Naehrig (BN) 曲線）
    - 加算、スカラ倍算、ランダムな点の生成
      任意データの曲線上の点へのマッピング
  - ペアリング演算（BN曲線上のOptimal Ate ペアリング）
    - ペアリング演算は、ペアリングの演算機能のみ

TEPLAでペアリング

- 必ず初期化をしましょう

```
pairing_init(p, "ECBN254a");
```

Reference

8.1 pairing_init(EC_PAIRING p, char *param)

利用するペアリングを指定します。戻り値はありません。

```
EC_PAIRING p;
pairing_init(p, "ECBN254a");
```

- ECBN254a Beuchat らによる手法
- ECBN254b Aranha らによる手法

TEPLAでペアリング

- この時、内部では3つの要素が生成されます

```
void ec_bn254_pairing_a_new(EC_PAIRING p)

    curve_init(p->g1, "ec_bn254_fpa");
    curve_init(p->g2, "ec_bn254_twa");
    field_init(p->g3, "bn254_fp12a");
```

BN曲線

6次Twist楕円曲線

12次拡大体

TEPLAでペアリング

```
/* --- e(P_A, sP)^r の計算 --- */
Element g;
element_init(g, p->g3);
pairing_map(g, sP, P_A, p);
    // 第2引数はg1，第3引数はg2で初期化する必要がある
```

Reference

8.3 pairing_map(Element g, const EC_POINT P, const EC_POINT Q, const EC_PAIRING p)

ペアリングを計算します。入力された EC_PONIT 型変数 P、Q のペアリングを入力された EC_PAIRING 型変数 p に基づき計算し、その結果を Element 型変数 g に格納します。($g = e(P, Q)$)