TEPLA1.0 と TEPLA2.0 の変更点

作成：石井, 神原

- 目次

1. buffer overflow について

- 概要

  関数 bn254_fp2_set_str, bn254_fp6_set_str, bn254_fp12_set_str において, 引数 s(文字列)に対するバッファサイズのチェックが無いので, ローカル変数を壊す可能性があったため修正を行った.

- 修正点
  - ✧ bn254_fp2_set_str：引数 s の長さが 140 より大きい場合エラー
  - ✧ bn254_fp6_set_str：引数 s の長さが 400 より大きい場合エラー
  - ✧ bn254_fp12_set_str：引数 s の長さが 790 より大きい場合エラー
- 修正が関係ある関数
  - ➢ bn254_fp2_set_str, bn254_fp6_set_str, bn254_fp12_set_str, ec_bn254_fp_point_set_str, ec_bn254_fp2_point_set_str

2. 文字列の入出力の順番の変更

- 概要

  関数 bn254_fp6_set_str, bn254_fp6_get_get_str, bn254_fp12_set_str, bn254_fp12_get_str における文字列の入出力の順番の変更を行った.

- 修正点

  - bn254_fp6_set_str, bn254_fp6_get_str

    $a = (a_{00} + a_{01}u) + (a_{10} + a_{11}u)v + (a_{20} + a_{21}u)v^2,$

    $a_{ij} \in \mathbb{F}_p (i = 0,1,2\ j = 0,1\ )$

    修正前 ： $str = \{a_{00}, a_{01}, a_{10}, a_{11}, a_{20}, a_{21}\}$

    修正後 ： $str = \{a_{00}, a_{10}, a_{20}, a_{01}, a_{11}, a_{21}\}$

  - bn254_fp12_set_str, bn254_fp12_set_str

    $a = \{(a_{000} + a_{001}u) + (a_{010} + a_{011}u)v + (a_{020} + a_{021}u)v^2\} +$
    $\{(a_{100} + a_{101}u) + (a_{110} + a_{111}u)v + (a_{120} + a_{121}u)v^2\}w$

    $a_{ijk} \in \mathbb{F}_p (i = 0,1, j = 0,1,2, k = 0,1)$

    修正前 ： $str = \{a_{000}, a_{001}, a_{010}, a_{011}, a_{020}, a_{021},$
    $a_{100}, a_{101}, a_{110}, a_{111}, a_{120}, a_{121}\}$

    修正後 ： $str = \{a_{000}, a_{010}, a_{020}, a_{100}, a_{110}, a_{120},$
    $a_{001}, a_{011}, a_{021}, a_{101}, a_{111}, a_{121}\}$

- 修正が関係ある主な関数

  - bn254_fp6_set_str, bn254_fp6_get_str,
    bn254_fp12_set_str, bn254_fp12_get_str,
    ec_bn254_fp12_new(既約多項式の定義時に使用)

3. オクテット列の入出力の変更

- 概要

  bn254_fp2_to_oct, bn254_fp2_from_oct, bn254_fp6_to_oct, bn254_fp6_from_oct, bn254_fp12_to_oct, bn254_fp12_from_oct におけるオクテット列の入出力の変更を行った.

- 修正点

  - bn254_fp2_to_oct, bn254_fp2_from_oct

    $$a = a_0 + a_1 u, a_i \in \mathbb{F}_p (i = 0,1)$$

    修正前 : $oct(a_0) \| oct(a_1)$

    修正後 : $oct(a_0 + a_1 p)$

  - bn254_fp6_to_oct, bn254_fp6_from_oct

    $$a = (a_{00} + a_{01}u) + (a_{10} + a_{11}u)v + (a_{20} + a_{21}u)v^2$$
    $$a_{ij} \in \mathbb{F}_p (i = 0,1,2, j = 0,1)$$

    修正前 : $oct(a_{00}) \| oct(a_{01}) \| oct(a_{10}) \| oct(a_{11}) \| oct(a_{20}) \| oct(a_{21})$

    修正後 : $oct(a_{00} + a_{10}p + a_{20}p^2 + a_{01}p^3 + a_{11}p^4 + a_{21}p^5)$

  - bn254_fp12_to_oct, bn254_fp12_from_oct

    $$a = \{(a_{000} + a_{001}u) + (a_{010} + a_{011}u)v + (a_{020} + a_{021}u)v^2\} + \{(a_{100} + a_{101}u) + (a_{110} + a_{111}u)v + (a_{120} + a_{121}u)v^2\}w$$
    $$a_{ijk} \in \mathbb{F}_p (i = 0,1, j = 0,1,2, k = 0,1)$$

    修正前 : $oct(a_{000}) \| oct(a_{001}) \| oct(a_{010}) \| oct(a_{011}) \| oct(a_{020}) \| oct(a_{021}) \| oct(a_{100}) \| oct(a_{101}) \| oct(a_{110}) \| oct(a_{111}) \| oct(a_{120}) \| oct(a_{121})$

    修正後 : $oct(a_{000} + a_{010}p + a_{020}p^2 + a_{100}p^3 + a_{110}p^4 + a_{120}p^5 + a_{001}p^6 + a_{011}p^7 + a_{021}p^8 + a_{101}p^9 + a_{111}p^{10} + a_{121}p^{11})$

- 修正が関係ある主な関数

  bn254_fp2_to_oct, bn254_fp2_from_oct,
  bn254_fp6_to_oct, bn254_fp6_from_oct,
  bn254_fp12_to_oct, bn254_fp12_from_oct,
  ec_bn254_fp_to_oct, ec_bn254_fp_from_oct,
  ec_bn254_fp2_to_oct, ec_bn254_fp2_from_oct

4. 構造体「Field」のオクテット列の長さの変更

- 概要

  構造体「Field」を初期化する際に設定するオクテット列の長さ(int oct_len)を変更した.

- 修正点
  - ec_bn254_fp6_new

    修正前 ：bn254_fp6->oct_len = 192

    修正後 ：bn254_fp6->oct_len = 190
  - ec_bn254_fp12_new

    修正前 ：bn254_fp6->oct_len = 384

    修正前 ：bn254_fp6->oct_len = 380
- 修正が関係ある主な関数

  ec_bn254_fp6_new, ec_bn254_fp12_new

5. tepla で使用されている mpz_export, mpz_import の引数の変更

- 概要

  tepla の関数内で使用されている mpz_export, mpz_import における endian に関する引数(関数の 5 番目)を変更した.
- 修正点(mpz_import も同様)

  修正前 ：mpz_export(os, size, -1, sizeof(* os), 0, 0, a) (CPU に依存)

  修正後 ：mpz_export(os, size, -1, sizeof(* os), 1, 0, a) (常時 big endian)
- 修正が関係ある主な関数

  bn254_fp2_to_oct, bn254_fp2_from_oct,

  bn254_fp6_to_oct, bn254_fp6_from_oct,

  bn254_fp12_to_oct, bn254_fp12_from_oct,

  ec_bn254_fp_to_oct, ec_bn254_fp_from_oct,

  ec_bn254_fp2_to_oct, ec_bn254_fp2_from_oct

  IHF1_SHA, cat_int_str (map to point に使用)

6. 2 次拡大体の元の比較を行う関数のバグ修正

- 概要

  2 次拡大体の 2 つの元を比較する際, $a_0 + a_1 u = b_0 + a_1 u$という形式であれば判定を通ってしまっていた.

- 修正点 bn254_fp2_cmp

```
int bn254_fp2_cmp(const Element x, const Element y)
{
    if( bn254_fp_cmp(rep1(x), rep1(y) ) == 0)
    {
        if( bn254_fp_cmp(rep1(x), rep1(y) ) == 0)
            return 0;
    }
    return 1;
}
```

- 修正後

```
int bn254_fp2_cmp(const Element x, const Element y)
{
    if( bn254_fp_cmp(rep1(x), rep1(y) ) == 0)
    {
        if( bn254_fp_cmp(rep0(x), rep0(y) ) == 0)
            return 0;
    }
    return 1;
}
```

- 修正が関係する主な関数

  bn254_fp2.c

7. 新たなパラメータの追加について

- 概要

  Aranha らの Eurocrypt2011 における論文”Faster Explicit Formulas for Computinc Pairings over Ordinary Curves”[1]を元に新たなパラメータを作成. パラメータの追加に従って, 全てのパラメータ名を変更. 1.0 までのパラメータは最後に”a”をつけ, 新たに追加したパラメータは最後に”b”をつけた. パラメータ名に関しては[2]を参考にした.

- 修正前 ： 利用可能パラメータ

有限体	bn254_fp bn254_fp2 bn254_fp6 bn254_fp12
楕円曲線	ec_bn254_fp ec_bn254_tw
ペアリング	ECBN254

- 修正後 ： 利用可能パラメータ

	Beuchat パラメータ	Aranha パラメータ
有限体	bn254_fpa	bn254_fpb
	bn254_fp2a	bn254_fp2b
	bn254_fp6a	bn254_fp6b
	bn254_fp12a	bn254_fp12b
楕円曲線	ec_bn254_fpa	ec_bn254_fpb
	ec_bn254_twa	ec_bn254_twb
ペアリング	ECBN254a	ECBN254b

- 修正が関係する関数

  全てのファイル

8. 有理点が$\mathbb{G}_2$上にあるかの判定について

- 概要

  曲線の定義式を満たしているかだけの判定になっており，正確には$\mathbb{G}_2$上にあるかどうかの判定を行えていない.

- 修正前 ：ec_bn254_lib.c (パラメータが増えたので両方に適用)

```
method->is_on_curve = ec_bn254_fp_is_on_curve;
```

- 修正後

```
method->is_on_curve = ec_bn254_fp2_is_on_curve;
```

- 追加した関数 ：ec_bn254_fp2.c

  ec_bn254_fp.c にある ec_bn254_fp_is_on_curve に以下の命令を追加，修正した.

  ec_bn254_fp2_mul_naf(R, curve(P)->order, P);

  hr = (element_cmp(x,y) == 0 && point_is_infinity(R));

  点Rが無限遠点になっていてかつ定義式を満たせば$\mathbb{G}_2$に属すと考えられる.

- 修正が関係する主な関数

  ec_bn254_fp2.c, ec_bn254_lib.c

9. 新たな関数の追加

- 概要

  新たに元，点の値を出力する関数を追加した.

- 追加した関数

  element_print(const Element), point_print(const EC_POINT)

  出力 ：“element(point)： 元(点)の値”

- 修正が関係する主な関数

  ec_lib.c

10. 楕円曲線の生成元について

- 概要

  $\mathbb{G}_1, \mathbb{G}_2$の生成元が無限遠点に設定されてしまっていて，生成元を使用することができない．そのため生成元を設定した．生成元の値に関しては[2]を参考にした．

- ec_bn254_lib.cに以下を追加．

  - ec_bn254_fpa_group_newに以下を追加．

    point_set_str(ec->generator, “[1,
    d45589b158faaf6ab0e4ad38d998e9982e7ff63964ee1460342a592677cccb0]”);

  - ec_bn254_fpb_group_newに以下を追加

    point_set_str(ec->generator,
    "[15F29C78629DD455F34C8D8E1B9C514FABAB45E7A5AD27E78C2B915
    DF4C6C264,1BCD3E98D7CAF1D0DA1524C2C07DE87B7D96C89B11B2E9
    27FE6DEB90B2F7FA5]");

  - ec_bn254_twa_group_newに以下を追加

    point_set_str(ec->generator,
    "[19b0bea4afe4c330da93cc3533da38a9f430b471c6f8a536e81962ed967909b5
    a1cf585585a61c6e9880b1f2a5c539f7d906fff238fa6341e1de1a2e45c3f72,17ab
    d366ebbd65333e49c711a80a0cf6d24adf1b9b3990eedcc91731384d2627
    0ee97d6de9902a27d00e952232a78700863bc9aa9be960C32f5bf9fd0a32d345]"
    );

  - ec_bn254_twb_group_newに以下を追加

    point_set_str(ec->generator,
    "[61a10bb519eb62feb8d8c7e8c61edb6a4648bbb4898bf0d91ee4224c803fb2b

516aaf9ba737833310aa78c5982aa5b1f4d746bae3784b70d8c34c1e7d54cf3,218
97a06baf93439a90e096698c822329bd0ae6bdbe09bd19f0e07891cd2b9a
ebb2b0e7c8b15268f6d4456f5f38d37b09006ffd739c9578a2d1aec6b3ace9b]");

- 参考文献

[1] Diego F. Aranha, Koray Karabina, Patrick Longa, Catherine H. Gebotys, and Julio Lopez. Faster explicit formulas for computing pairings over ordinary curves. In Kenneth G. Paterson, editor, *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, Vol. 6632 of *Lecture Notes in Computer Science*, pp. 48–68. Springer, 2011.

[2] K.Kasamatsu, S.Kanno, T.Kobayashi, Y.Kawahara, NTT Software Corporation, "Barreto-Naehrig Curves draft-kasamatsu-bncurves-01", https://tools.ietf.org/html/draft-kasamatsu-bncurves-01