楕円曲線計算ライブラリ ecpy の C++ による高速化

Shiho Midorikawa

自己紹介

- 緑川 志穂 (@elliptic_shiho)
- 0x11 歳
- ラボユース 5th(6th)
- CTFer (Crypto)
- 数学
- 暗号

ecpy (5th)

- 5th ラボユースでは ecpy というライブラリを作りました
  - 成果物: `https://github.com/elliptic-shiho/ecpy/`
  - 楕円曲線を扱うための Python2/3 ライブラリ
  - ペアリングにも対応
  - (Pure Python にしては) それなりの速度を持ち, 任意精度に対応

ecpy (5th)

- ペアリングとは?
  - 楕円曲線上の点 2 つから定まる写像 $e_n(P, Q)$
  - 便利な性質を持つ (双線形性)

$$a, b \in \mathbb{Z} \text{ のとき } e_n(aP, bQ) = e_n(P, Q)^{ab}$$

  - ちゃんとした定義は非常に長くなるので省略します

ecpy (5th)

- ペアリングは暗号に使える
  - 例えば相手の ID 情報を鍵とした公開鍵暗号 (ID-based encryption)
  - 例えば暗号化した状態での暗号文同士の演算 (Functional encryption)
  - 等々 詳しくは光成さんの本[1] へ
- 現在かなりホットな分野
- このライブラリはそれらをプログラム化する際のベースとして使えるようなものとして作りました

[1] https://herumi.github.io/ango/

ecpy (5th)

- Python 部分の実装
  - 5th の期間中に完成 (詳しい話は 5th スライドへ)
  - sage よりは速いものを作れたが, 体感的にまだ遅い...
  - 勤務開始が 2015/08/03 だったので最長の 2016/08/03 まで延長させて頂けることに

ecpy (6th)

- 高速化
- C++を使って外部モジュールとして組むことを考える
  - 実の所, C は分かっても C++の経験は皆無
  - クラス設計が曖昧なまま始めたため, 数回 0 から書き直すことに...

ecpy (6th)

- 一旦クラス設計から教わることに

  - 「インターフェースは極力シンプルに」

  - ctypes を利用して呼び出し, 任意精度を扱えるように数値を文字列としてやりとりする

  - その他 C++の規格書や他の実装を読みながら設計→実装→テストのサイクルで開発を進めた

- 設計ができてからは順調に進んだ

ecpy (6th)

- 8/3 までには完成できませんでしたが. 本社への出社によるラボユースとしての活動自体は一応終了.

  - アドバイス等は受けられるということで, 速度的な面やコード自体が C++でのルールに反している部分を指摘していただいた.

  - 実装自体が終わってコードレビューへの修正等が一段落付くと 2017 年に

ecpy (6th)

- 結果
  - Python のみ (5th 成果発表スライドより)

    ```
    weil: 71188.95 usec/pass
    tate: 15711.19 usec/pass
    ```

  - 今回

    ```
    weil: 14739.54 usec/pass
    tate: 3095.69 usec/pass
    ```

- それなりに速くなりました

ecpy (6th)

- 今後の課題
  - ライブラリとしての使い勝手はまだ改善の余地がある
  - 楕円曲線のパラメータを変える手段が無い, 公開されている関数が被る, ...
- C++部分を別のライブラリとして切り出すことも検討しても良いかもしれない
  - 任意長に対応しているライブラリはそんなにないと考えている

課外活動

- 5th の時点で大方楕円曲線・ペアリング周りの数学は道具としては使えていた

  - 実装したりなんだりを進めていくうちに「道具としては」の部分が頭打ちになってきていた

- ちゃんとした理解をしたい/数学的議論が出来るようになっておきたいよね, という話になり, 「ラボユースの延長としての活動」として数学書のゼミをして頂けることになった

  - 「代数概論」という本を読んでいます

  - 数学科出身の光成さん, 中谷さんと途中から星野さんが加わった 4 人でのゼミで, 発表者は全て私です.

課外活動

- 東京に常に居るわけではないので, 帰ってからもゼミが出来るようにと TV 会議用の Surface を貸与していただきました.
- 一ヶ月半で 2 ページしか進まない時もありましたが, その分得られるものも多かったです
- せっかくある程度読み進められたので今後も読んでいきたい

まとめ

- 楕円曲線の計算のためのライブラリ ecpy を開発した.
  - C++による高速化を実施, それなりに速度が出るようにした
- コード設計, 数式の実装, 実際の使い勝手を考えながらのリファクタリング等々良い経験が積めた.
- 人に使ってもらう前提のソフトウェアを書く事は難しい
  - 今後も長く使えるようなソフトウェアを書けることも必要だが難しい

余談

- ecpy のネイティブコード以外は純粋な Python コードなので, つい先日ちゃんと PyPI に登録しようと考えました

余談

- ecpy のネイティブコード以外は純粋な Python コードなので, つい先日ちゃんと PyPI に登録しようと考えました

  - 去年 (2016 年) の夏に同名のモジュールが登録済み (しかも用途が同じ) でした...

- 登録はしたいので, 名前を譲ってもらえるぐらいに実用的なライブラリとしていきたいです

参考文献

- J.H.Silverman. 2009. *The arithmetic of Elliptic Curves*
- 森田康夫. 1987. 代数概論
- 光成滋生. 2015. クラウドを支えるこれからの暗号技術
- 辻井 重男, 笠原 正雄, 有田 正剛, 境 隆一, 只木 孝太郎, 趙 晋輝, 松尾 和人. 2008. 暗号理論と楕円曲線
- I.F.Blake, G.Seroussi, N.P.Smart, N.J.Hitchin. 2011. *Advances in Elliptic Curve Cryptography*
- その他多くの Web サイト

ありがとうございました.