2Z-8

暗号アプリケーション開発のためのペアリング演算フレームワーク

伴　拓也†　　毛利　公美‡　　白石　善明†

†名古屋工業大学　　‡岐阜大学

1. はじめに

クラウドコンピューティングを利用したシステムの開発が盛んに行われている．今後は，M2M，ライフログなどのビッグデータ処理システムや医療情報の共有システム等が実装されていくことになる．これらのシステムでは，機密情報やプライバシー情報をインターネット上で扱うので，高度な暗号技術が求められている．

このようなシステムで利用が期待される，従来の暗号技術では実現が困難な特徴を持つ暗号方式が提案されている．例えば，ID を公開鍵として利用できる ID ベース暗号，暗号化したまま演算ができる秘匿計算等が挙げられる．ID ベース暗号は，検索可能暗号，タイムリリース暗号，放送暗号等の応用につながっている．

ID ベース暗号と秘匿計算では楕円曲線上で定義されるペアリングにより構成されるものがある．例えば，ID ベース暗号では BF 方式[1]，秘匿計算では暗号化したまま任意回数の加算と 1 回の乗算を可能にする BGN 方式[2]などがある．

ペアリング演算では，入力となる楕円曲線上の点とその演算を利用し，出力となる拡大体上の元とその演算を利用する．ペアリング演算のコードを再利用するために，ペアリング演算ライブラリ[3][4]が開発されている．これらのライブラリでは，拡大体上の演算，楕円曲線上の演算，ペアリング演算機能を提供している．ペアリング暗号を実装するには，これらの演算の他にも必要となる機能がある．

他方で，ペアリング演算の種類は高速なものや安全性の高いものなど複数ある．楕円曲線上の有理点を選択する演算は，楕円曲線の種類によって異なる．

本稿では，演算の拡張性を持つペアリング演算フレームワークの設計と実装について述べる．本フレームワークでは，一般的な暗号演算ライブラリが提供する拡大体上の演算，楕円曲線上の演算，ペアリング演算だけではなく，ペアリング暗号に必要な機能を含める．本フレームワークを利用して BF 方式と BGN 方式を実装し，全ステップのうちフレームワークの利用者である開発者が記述するステップ数で評価する．

2. ペアリング暗号とフレームワークの必要性

2.1　双線形写像を利用した暗号

ペアリングeは 2 入力 1 出力の関数で，双線形写像(Bilinear maps)である．G_1を位数nの有限巡回群とすると，双線形写像は次の性質を満たす．

i. 双線形性(Bilinear)
$e(aP, bQ) = e(P, Q)^{ab}\ \forall P, Q \in G_1, \forall a, b \in Z$

ii. 非縮退性(Non-degenerate)
$e(P, P) \neq 1\ \forall P \in G_1 \setminus \infty$

iii. 計算可能性(Computable)
eを計算する効率的なアルゴリズムが存在する

例えば，BF 方式では，暗号化と復号でペアリングが利用される．BGN 方式では，秘匿計算の乗算でペアリングが利用される．

2.2　ペアリング演算フレームワークの必要性

ペアリング暗号の実装では，ペアリング関数，その入力となる楕円曲線上の点およびその演算，出力となる拡大体上の元およびその演算を利用するので，コードは相当な行数になる．

ペアリング暗号を短時間で実装できるように，ペアリング演算とそれに利用する演算のライブラリが開発されている．公開されているものとして C 言語の PBC[3]や Java の jPBC[4]などがある．

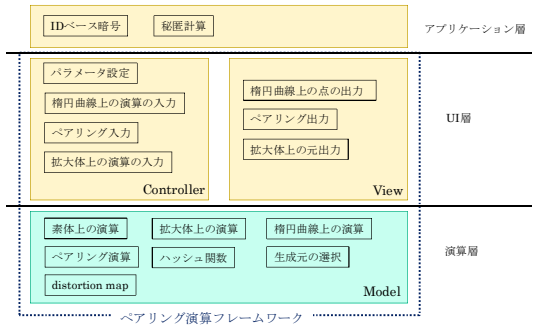

図 1　ペアリング演算フレームワークの階層

本フレームワークは，一般のライブラリに含まれているような拡大体上の演算，楕円曲線上の演算，ペアリング演算に加え，ペアリング暗号の開発に必要となる機能を提供する．さらに，開発者であるフレームワークの利用者に，ペアリングの種類やパラメータの変化による内部処理の違いを吸収し，同じ操作で利用できるインタフェースを提供する．以上の 2 点ができればペアリング暗号の実装を容易にすると考えられる．

3. ペアリング演算フレームワークの設計

3.1　設計方針

本フレームワークは，2 つの方針に沿って設計する．

1 つは，一般のライブラリではサポートされていないペアリング暗号の実装に必要な機能を含めることである．本フレームワークでは，ID ベース暗号，属性ベース暗号，検索可能暗号，タイムリリース暗号，放送暗号，秘匿計算などの実装に必要となる機能を含める．例えば，ID ベース暗号の一つである BF 方式では，2 種類のハッシュ関数が必要になる．秘匿計算の一つである BGN 方式は BF 方式と共通して，楕円曲線上の原始元の選択機能が必要とされる．このような一般のライブラリではサポートされない機能を含める．

フレームワークを拡充させていくためには，未実装な機能を柔軟に含めていけるといった拡張性を考慮することが必要で，これをもう 1 つの方針とする．フレームワーク利用者である開発者に対してインタフェースを提供する機能と演算機能などの諸機能のすべてが混在し，密に結合している状態では拡張が容易とは言えない．そこで，本フレームワークでは，図 1 のように演算層と UI 層に分けたフレームワークの構成とする．MVC モデルを適用し，それぞれ次のように役割を持たせる．

[Model] 演算を実行する
[View] 演算結果を出力する
[Controller] パラメータ，演算の入力を受け付ける

演算の入出力および，演算が拡張できるようにインタフェース，抽象クラスを利用して設計する．

まず，基本的な機能として，ペアリングの種類に Weil ペアリング，Tate ペアリング，η_T ペアリングを本フレームワークに組み込むことを想定して設計する．そして，アプリケーション層には ID ベース暗号として BF 方式，秘匿計算として BGN 方式を組み込むことを想定して演算層と UI 層の基本的な部分を設計する．

3.2　フレームワークの機能と拡張性

ペアリング演算は，Weil ペアリング，Tate ペアリング，

A Pairing Framework for Developing Cryptographic Applications
† Takuya BAN and Yoshiaki SHIRAISHI・Nagoya Institute of Technology
‡ Masami MOHRI・Gifu University

η_T ペアリング等，複数の種類がある．拡大体上の演算，楕円曲線上の演算，ペアリング演算機能を拡張可能な形で設計する．例えば，η_T ペアリングは，Tate ペアリングで標数が小さいときに，ビット演算を利用することで拡大体上の演算および楕円曲線上の演算を高速化したものである．つまり，拡大体上の演算，楕円曲線上の演算はペアリングの種類によって変わるので，共通部分を抽象クラスで実装し各演算に拡張性を持たせ，演算を追加できるようにする．

View と Controller では，入出力のインタフェースおよび抽象クラスを用意し，演算の入力，演算結果の出力形式に拡張性を持たせる．

ペアリング暗号の実装において，一般のライブラリに含まれる演算以外で必要になる機能として例えば次のようなものがある．

BF 方式では，暗号化，復号で 2 種類のハッシュ関数を利用する．1 つは MapToPoint と呼ばれ，ID ベース暗号で利用される，ID(文字列)から楕円曲線上の点にマップするハッシュ関数である．もう 1 つは，拡大体上の元を n ビット整数にマップするハッシュ関数である．以降，これを FtoZn と呼ぶ．ペアリング暗号によっては，これら以外のハッシュ関数が必要になることがある．本フレームワークでは，ハッシュ関数の内部で共通して使う機能を抽象クラスとして用意し，その抽象クラスを継承することで機能が追加できるようにする．

BF 方式，BGN 方式では，鍵生成時に楕円曲線上の生成元の選択機能が必要になる．生成元の選択機能の実装は利用する楕円曲線により異なるので，入出力インタフェースを用意し，楕円曲線に合わせて実装できるようにする．

4. ペアリング演算フレームワークの実装

フレームワークの開発環境を表 1 に示す．UI 層は Java により実装した．演算層は演算コストの高い処理を多用するので C++で実装し，Java の JNI でラップした．素体上の演算，拡大体上の演算では，C++の暗号用の多倍長演算ライブラリ NTL[5]を利用した．開発環境では Windows を利用しているが，Linux，Android においても JNI を利用して C++のラップが可能である．

4.1 Model

Model では，Controller が操作する，拡大体上の演算，楕円曲線上の演算，ペアリング演算，ハッシュ演算，distortion map[6]，楕円曲線上の有理点の選択，楕円曲線上の生成元の選択の 7 つを演算インタフェースとして用意する．拡大体上の演算，楕円曲線上の演算は標数 2，3，4 以上の場合で実装が異なる．拡大体上の演算，楕円曲線上の演算のそれぞれの共通部分を抽象クラスで括る．

distortion map，楕円曲線上の有理点の選択，楕円曲線上の生成元の選択の演算は，楕円曲線の種類によって違いが生じる．インタフェースで入出力の型を定義する．

Weil ペアリング，Tate ペアリングの共通部分の例として，両ペアリングで利用できる Miller のアルゴリズムがある．これも抽象クラスとして実装する．

4.2 View

View では，演算結果の値を保持し，Getter メソッドで値を出力する．拡大体上の元，楕円曲線上の点，整数の値を保持する抽象クラスと Controller で利用する演算の入力用インタフェースを用意する．

抽象クラスでは，インタフェースを実装し，演算結果の出力形式を定義する．

4.3 Controller

Controller では，パラメータ，演算の入力を受け付け，Model の演算機能を操作する．パラメータ，拡大体上の演算，楕円曲線上の演算，ハッシュ演算，ペアリング演算の 5 つの入力用の抽象クラスを用意する．

表 1　開発環境

OS	Windows 7 Professional SP1	
言語	Java, C++	
IDE	Java	Eclipse Java EE IDE Indigo SR1
	C++	Microsoft Visual C++ 2010
ライブラリ	NTL	

5. フレームワークによる暗号方式の実装・評価

5.1 評価方法

本フレームワークを利用してペアリング暗号を実装するときに全体のステップのうち，フレームワークの利用者が記述する処理ステップの割合を調べる．ペアリング暗号として BF 方式，BGN 方式を実装する．

評価の準備として，本フレームワークに次の 3 つの処理が組み込まれているものとする．

・標数$p > 3$での拡大体上の演算，楕円曲線上の演算
・次の楕円曲線に対応した有理点，生成元の選択機能，distortion map

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

・Weil ペアリング

5.2 ペアリング暗号の実装

5.2.1 BF 方式

BF 方式のセットアップ，鍵生成，暗号化，復号を実装した．各処理を次に示す．Gを位数nの有限巡回群とする．

Setup　$P \in G,\ sP, H_1, h_2, n$
Extract　公開鍵 $P_A = H_1(ID_A)$，復号鍵 $sP_A = sH_1(ID_A)$
Encrypt　$C = (c_1, c_2) = (xP, m * h_2(e(P_A, xsP)))$
Decrypt　$m = c_2 / h_2(e(sP_A, c_1))$

BF 方式では，楕円曲線上の演算，ペアリング演算の他にハッシュ関数H_1とh_2を利用する．ハッシュ関数H_1，h_2はそれぞれ，本フレームワークが提供する MapToPoint，FtoZn に対応している．

5.2.2 BGN 方式

BGN 方式の鍵生成，暗号化，復号，ペアリングを利用する秘匿計算の乗算を実装した．各処理を次に示す．G, G_1を位数nの有限巡回群とする．

KeyGen　$n = q_1 q_2,\ g, u \in_R G, h = u^{q_2}$
公開鍵：(n, G, G_1, e, g, h)，秘密鍵：q_1
Encrypt　$C = g^m h^r \in G$
Decrypt　$C^{q_1} = (g^m h^r)^{q_1} = (g^{q_1})^m,\ m = \log_{g^{q_1}} C^{q_1}$
Multiplicative homomorphic　$e(C_1, C_2){h_1}^r, h_1 = e(g, h)$

鍵生成で準備するg, uは楕円曲線上の生成元である．本フレームワークの生成元の選択機能を利用することで楕円曲線上の生成元をランダムに選択することができる．

5.3 評価結果

評価結果として暗号方式の実装に必要な全ステップのうち，2～3%の処理を記述することで，ペアリング暗号を実装できることを確認した．

6. おわりに

幅広い応用が期待されているペアリング暗号の共通する処理ステップを再利用することを目的として，ペアリング演算フレームワークを作成した．

本稿では，フレームワークの設計，実装について述べ，フレームワークを利用して BF 方式，BGN 方式を作成し，フレームワーク利用者が記述するステップの比率を求めた．フレームワークの利用により暗号方式の実装に必要な全ステップのうち 2～3%の処理を記述することで実装できることを確認した．

参考文献

[1] Boneh, D. and Franklin, M.: Identity-Based Encryption from the Weil Pairing, Appears in SIAM J. of Computing, Vol.32, No.3, pp.586-615 (2003).
[2] Boneh, D. Goh, E. and Nissim, K.: Evaluating 2-DNF Formulas on Ciphertexts, TCC2005, LNCS3378, pp.325-341 (2005).
[3] Lynn, B.: PBC Library - Pairing-Based Cryptography (online), available from <http://crypto.stanford.edu/pbc/> (accessed 2013-01-10).
[4] Caro, A.: Java Pairing-Based Cryptography Library (online), available from <http://gas.dia.unisa.it/projects/jpbc/> (accessed 2013-01-10).
[5] Shoup, V.: NTL: A library for doing number theory (online), available from <http://www.shoup.net/ntl/> (accessed 2013-01-10).
[6] Verheul, E.: Evidence that XTR is more secure than supersingular elliptic curve cryptosystems, EUROCRYPT2001, pp.195-210 (2001).