ペアリング暗号の実用化に向けて

Research on Pairing Cryptosystem and Its Applications

岡本 栄司（Eiji OKAMOTO, Ph. D.）
筑波大学教授
（ Professor, Ph. D., University of Tsukuba ）
IEEE, ACM, 電子情報通信学会、情報処理学会　会員
受賞：情報処理学会功績賞(2013), CHES Best Paper Award(2007, 2009), 情報処理学会論文賞(2008), 電子情報通信学会論文賞(1990)
著書：暗号理論入門(共立出版 1993), 電子マネー(岩波書店 1997)
研究専門分野：情報セキュリティ

あらまし　ペアリング関数を用いた暗号システムは広くペアリング暗号と呼ばれているが、今まで数多く提案されている。その代表例が ID ベース暗号であり、メールアドレスなど好きな文字列を暗号化鍵とすることができる。その他にも、属性暗号、放送型暗号、キーワード検索暗号など、従来の暗号では実現困難なシステムが可能になっている。

これらのペアリング暗号を実際に実用化する際に問題となるのが、ペアリング関数計算である。これまでの実装では、演算速度の有効性評価のために開発されるものが多く、汎用的に利用できるものはほとんどなかった。汎用的に利用できる代表的なライブラリである PBC Library も、その暗号強度と速度のバランスが優れているとはいいがたい。そこで、我々は高速性が確認されている Barret-Naehrig(BN)曲線上の Optimal Ate Pairing を利用可能な汎用的なペアリング暗号ライブラリを開発し、電子情報通信学会の 2013 年暗号と情報セキュリティシンポジウム(SCIS2013)で発表したので、それらをベースに応用も踏まえて、本稿で報告する[1]。本ライブラリは様々なプラットフォームで利用可能であり、またオープンソースでの提供や自由度の高いライセンスなど、幅広い利用が可能となっている。本文では、PBC Library との機能・速度比較やさまざまなプラットフォームでの動作評価の報告も行う。今後、大いに利用してもらいたい。

１．研究の目的

楕円曲線上の点の２つを入力とする２入力１出力の関数で各入力に対して線形性を提供するペアリングは、ID ベース暗号をはじめさまざまな暗号プロトコルの中で用いられており、近年の暗号研究における大きな位置を占めている。

ペアリングに関する研究は、ペアリングそのものの効率化や、効率的なペアリングを実現するための曲線の探索、またペアリングを高速に実装する技術など幅広く行われている。また、ペアリングを用いた暗号プロトコルは、現在においても盛んに研究が行われており、今後幅広い分野において実用化にむけて進んでいくものと考えられる。

このペアリングを用いようとした場合、ペアリングを専門にして研究開発を行っている組織であれば、自前でペアリングとその周辺の機能の実装を行い、アプリケーション開発やシステム開発をすることが可能である。しかし、ペアリング自身は研究していないものの、その有用性に目を向け、さまざまな暗号プロトコルの応用アプリケーションやシステムの開発を望む組織にとって、その実装は大きな障壁となる。

ペアリングの実装は多くの研究者や組織において行われている。しかしペアリングを用いた暗号プロトコルを実際にアプリケーションやシステムとして実現するときには、ペアリングの演算だけではなく、楕円曲線上の点の演算や、任意のデータを楕円曲線上の点にマッピングする機能、ペアリングの出力の値をさらに加工するなどのさまざまな周辺機能が必要となる。多くのペアリング実装にはそういった周辺機能の実現はされておらず、アプリケーションやシステムの開発に際しては不足している周辺機能を自前で用意しなければならなくなる。

ペアリングに関する機能が整備されたソフトウェアライブラリがあれば、そういった要望にこたえることが可能になる。その代表的な例として、Stanford 大学では PBC Library というライブラリを提供しており[2]、ペアリングの演算方法などの詳細を知らずとも、その機能を利用しアプリケーションやシステムの開発を行うことを容易にしている。

ライブラリとしての提供は、単にシステムやアプリケーションの開発を容易にするだけではなく、相互運用の視点からも重要となる。ペアリングを用いた暗号プロトコルにおいては、データの送信者と受信者、また、それら 2 者の媒介や信頼できる機関などの第 3 者が存在することが多い。暗号プロトコルを用いたアプリケーションやシステムの開発では、そういったそれぞれの役割を独立して実装しなければならない。現実的な応用を考慮すると、それぞれが異なるプラットフォームでアプリケーションが動作していることを考慮することが必要となる。たとえば、データ送信者側が iOS や Android で動作するスマートフォンであり、受信者側は Windows が搭載されている PC、そして信頼できる機関（鍵発行など）は Linux で動作するサーバとなるケースが考えられる。そういった異なるプラットフォーム上で開発されたアプリケーション同士でも、データの互換性などの相互運用性は、プラットフォームへの依存がすくないソフトウェアライブラリが存在することでより容易に実現可能になる。

さまざまなプラットフォームで利用可能なソフトウェアライブラリは、今後のペアリングを用いた暗号プロトコルの実用化に向けては欠かせないものとなると考えられる。現状では PBC Library がほぼ唯一のソフトウェアライブラリと言ってよいが、そういったライブラリの多様性を持たせることも重要であろう。そこで我々は新たに汎用的に利用可能なソフトウェアライブラリである TEPLA を開発した[1]。

TEPLA は、高速性が確認されている BN 曲線上の Optimal Ate ペアリングを利用可能な汎用的なペアリング暗号ライブラリであり、様々なプラットフォームで利用可能となっている。またオープンソースでの提供や自由度の高いライセンスなど、幅広い利用が可能となっている。本稿では TEPLA の概要や設計思想、また提供する機能を解説する。また、各種演算で用いられている手法や他のライブラリなどの紹介も行う。TEPLA は汎用的に利用可能とすることに大きな目的を置いているため、その高速性は必ずしももっとも重要な要素ではなかった。しかし、PBC Library とのパフォーマンス評価の結果、高いセキュリティレベルながら高速な実装となっていることが示された。これは BN 曲線上の Optimal Ate ペアリングの高速性によるものではあるが、汎用性の高さとともに高い効率も実現したライブラリであることが示されている。また、公開する Web サイトでは日本語によるマニュアルも整備し、より開発者への障壁が低くなることも考慮している。

TEPLA により、ペアリングを用いた暗号プロトコルの応用が進むことが大きく期待される。

２．研究の背景

ペアリングを用いた暗号アルゴリズムを使ったシステムやアプリケーションを開発するために、利用可能なライブラリとして代表的なものに PBC Library がある[2]。PBC Library は C 言語ライブラリであり、無償で利用可能である。ソースも公開されている。また LGPL (GNU Lesser General Public License) で提供されている。

PBC Library はペアリング関数だけでなく、有限体の演算や楕円曲線上の点の演算などの API を提供しており、アプリケーションを開発するための関数が網羅的に提供されている。著者らの知っている限りでは、ペアリングを用いた暗号プロトコルを開発可能なライブラリとしては唯一のものである。多倍長演算については同じく無償利用可能な GMP Library[3]を利用している。

PBC Library は開発者や有志の協力により、Windows、Mac をはじめさまざまなプラットフォームで利用可能な形式としても配布されている。

PBC Library では、曲線やペアリングが複数選択可能となっている。それらを PBC Library では「Type」と呼び、Type A から Type G まで存在する。また 1 つの Type の中に複数のパラメータが選択可能なものも存在する。

PBC Library 以外にも利用可能なライブラリや実装がいくつか存在する。Aranha らによる RELIC はペアリング以外にもさまざまな関数を含んでいる C 言語ライブラリである[4]。センサーネットワークのノード上でのペアリング暗号の実装を提供する TinyPBC[5]では RELIC が採用されている。Beuchat らによる Pairing2010 で発表された実装は、BN 曲線上の

Optimal Ate ペアリング演算と BN 曲線上の点のスカラ倍算を提供している[6][7]。同 Web サイトでは他にも $F_{3^{97}}$ と $F_{3^{193}}$ 上の η_T ペアリングと、$F_{3^{509}}$ と $F_{2^{1223}}$ 上の η_T ペアリングの実装も公開されている。

公開の有無は不明であるものの、高速な実装やある特定環境でのペアリング実装などは多くの研究が行われている[8][9][10][11][12][13][14][15]。

３．研究成果

楕円曲線・ペアリング汎用ライブラリ TEPLA

３．１　概要と設計思想

ペアリングを用いた様々な暗号プロトコルが実用的になるためには、プロトコルで用いられる様々な機能が１つにまとまったソフトウェアライブラリが提供されることが非常に重要である。ソフトウェアやシステムの開発者がそれぞれそれらの機能を実現すると、開発コストが高くなるだけでなく、独自に作られたそれらのソフトウェアやシステム同士の互換性を保つことが難しくなる。共通のライブラリを用いることで開発コストを下げるだけでなく、容易に互換性を持ったアプリケーションやシステムを開発・構築することが可能になる。

TEPLA はそういったライブラリとして提供されるために開発が行われた。TEPLA は University of Tsukuba Elliptic Curve and Pairing Library の略称であり、汎用的に利用可能な楕円曲線とペアリング演算関連の機能を提供する C 言語のライブラリである。

TEPLA はサーバ側やクライアント側などさまざまな環境で利用可能になるべく、ハードウェアやプラットフォームの制限などが排除されるよう志向し、その実装はシンプルに行った。またオープンソースで広く利用可能な形で提供することを目的とし、各演算のアルゴリズムは独自に開発するのではなく、すでに成果が公開されているアルゴリズムを採用した。各演算の実装はアルゴリズムを改良せずに行い、その高速性はアルゴリズムに任せ、実装での高速化は本ライブラリの対象外とした。

TEPLA はオープンソースとして提供され、またライセンスは修正 BSD ライセンス（三条項 BSD ライセンス）で提供される。

３．２　TEPLA が提供する機能

TEPLA では以下の機能を提供する。

- 254 ビット素体と 2 次、12 次拡大体上の元の演算
  - 加算、減算、積、逆元、2 乗、平方根、べき乗、フロベニウス写像、ランダムな元の生成
- Barreto-Naehrig (BN) 曲線上の点の演算
  - 加算、スカラ倍算、フロベニウス写像、ランダムな点の生成､任意のデータの曲線上の点へのマッピング
- BN 曲線上の Optimal Ate ペアリング演算

その他、値の比較や文字列出力・入力などの機能を持つ。

BN 曲線は $y^2 = x^3 + b,\ b \in F_p$ であらわされ、標数 p と位数 r はそれぞれ、パラメータ z を基に

$$p = 36z^4 + 36z^3 + 24z^2 + 6z + 1$$
$$r = 36z^4 + 36z^3 + 18z^2 + 6z + 1$$

で表される。

TEPLA では $z = 2^{62} - 2^{54} + 2^{44}$ を利用した。

３．３　TEPLA の各演算に用いられたライブラリや手法

TEPLA では有限体の演算、楕円曲線上の演算、ペアリング演算など複数の演算が利用可能であるが、それぞれは独自に新規提案したアルゴリズムではなく、既存のライブラリの利用や既発表のアルゴリズムを採用した。

有限体の演算では、素体上の演算は GMP ライブラリを利用した。拡大体上の演算は Paring 2010 での Beuchat らの論文でのアルゴリズム[6]を用いて実装を行った。

楕円曲線上の点のスカラ倍算では､Hankerson らの書籍[16]に掲載されている手法と、Nogami らによる手法[17]を用いて実装を行った。任意のデータの曲線上の点へのマッピングに関しては、IEEE 1363.3 での草案文書[18]の手法を用いて実装した。

ペアリングでは、Beuchat らの論文のアルゴリズム[6]を用いて実装を行った。

３．４　TEPLA の評価

３．４．１　適用可能プラットフォーム

TEPLA は、プラットフォームやハードウェアアーキテクチャに依存しないという設計思想の下で実現されており、2012 年 12 月の時点で、TEPLA バージョン 1.0 は以下のプラットフォームでの動作が確認されている。

・Windows 7 Professitonal SP1 (64bit)　+ Visual C++ 2010 (Visual Studio 2010 Professional) + MPIR 2.6.0 + OpenSSL 1.0.1c
・Linux (Kernal 2.6.18-308.8.1.el5) + gcc 4.1.2 + GMP 5.0.5 + OpenSSL 0.9.8e-fips-rhel5
・Mac OS X 10.6.8 (Snow Leopard)　+ gcc 4.2.1 + GMP 5.0.4 + OpenSSL 0.9.8r

３．４．２　パフォーマンス比較

TEPLA のパフォーマンス評価のため、PBC Library の各パラメータにおけるいくつかの演算の計算速度を比較した。

評価は Linux (Cent OS) 上で行った。PBC Library はバージョン 0.5.12 を用いた。Linux カーネルのバージョンは 2.6.18-308.8.1.el5 であり、コンパイルに用いた gcc は 4.1.2、GMP は 5.0.5, また TEPLA でハッシュ計算を行うときに用いる OpenSSL はバージョン 0.9.8e-fips-rhel5 を利用した。

評価は Linux (Cent OS) 上で行った。PBC Library はバージョン 0.5.12 を用いた。Linux カーネルのバージョンは 2.6.18-308.8.1.el5 であり、コンパイルに用いた gcc は 4.1.2、GMP は 5.0.5, また TEPLA でハッシュ計算を行うときに用いる OpenSSL はバージョン 0.9.8e-fips-rhel5 を利用した。利用した計算機の CPU は Intel Core i7 920 を搭載していた。2.67GHZ のクロック数で、コア数は 4 である。また、RAM は 6GB 搭載していた。

比較に用いた演算は以下の 4 種類である

・Pairing: ペアリング演算
・ScalarMultiplication: 楕円曲線上の点のスカラ倍算
・MapToPoint: 任意のデータの楕円曲線上の点へのマッピング
・PowOnG_T: G_T (ペアリングの出力となる体) 上の元のべき乗演算

各演算の計算速度の値を表 1 に示す。

また、各パラメータでの演算時間とビット数の関係性を示した図を演算ごとに図 1～4 に示す。図 3 では、PBC Library のパラメータ a と e の演算時間が他と比較し大きな値であったため、演算時間を対数軸であらわした。

表１　TEPLA と PBC Library のパフォーマンス比較 (msec)

	PBC Library パラメータ								TEPLA
演算種類	a	a1	d159	d201	d224	e	f	g149	bn254
Pairing	2.2078	64.1451	4.8378	7.3075	9.0325	9.0840	25.3030	14.6299	6.3980
ScalarMultiplication	2.5591	55.4810	0.7697	1.1023	1.4632	6.5124	0.7698	0.6912	0.6836
MapToPoint	5.8960	2.6394	0.0564	0.2136	0.2020	34.4924	0.0320	0.0335	0.3081
PowOnGT	0.2666	5.2082	1.4265	2.0826	2.5838	0.2853	5.7848	4.0275	6.6730
ビット数（安全性）	1024	2048	954	1206	1304	1024	1920	1490	3048

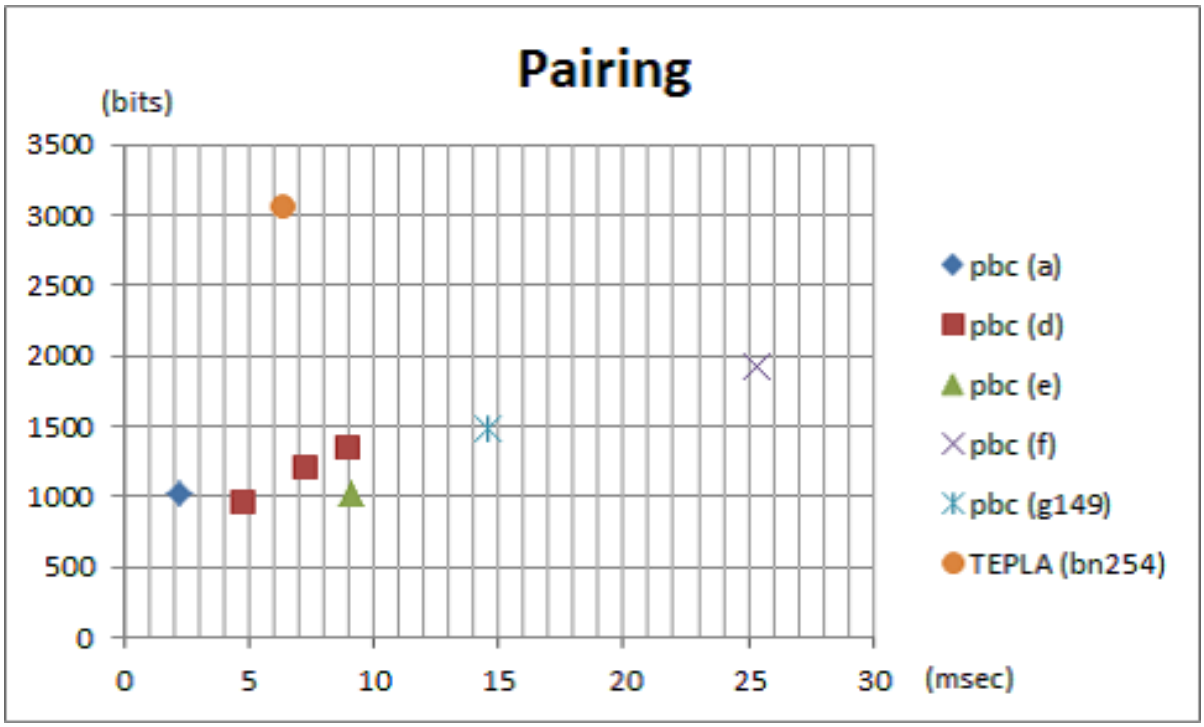

図１　ペアリング演算時間の比較

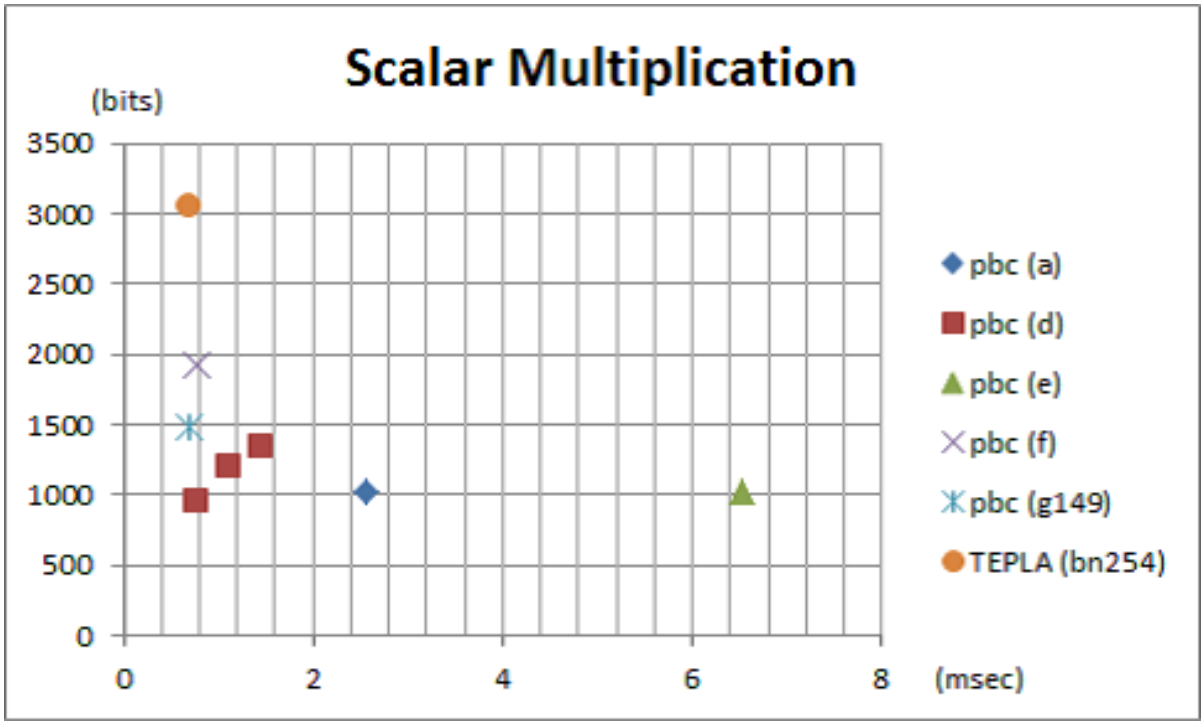

図２　楕円曲線上の点のスカラ倍算演算時間の比較

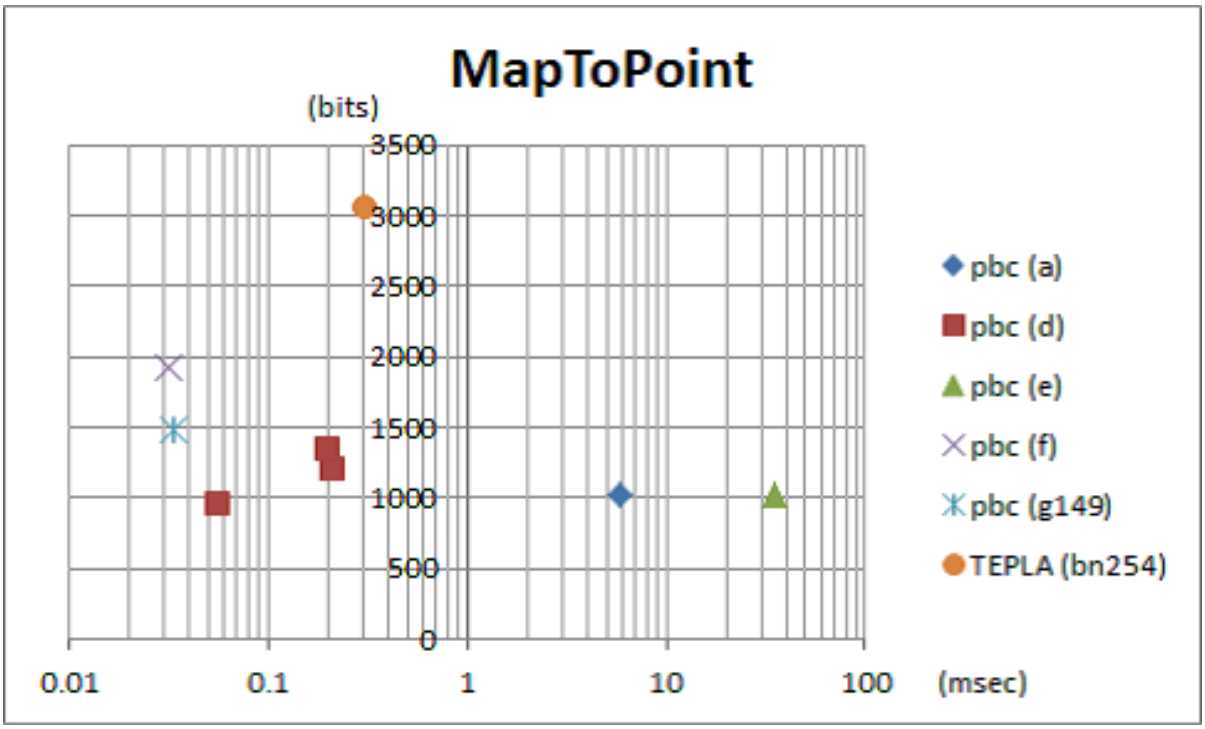

図３　任意の文字列を楕円曲線上の点にマッピングする関数の演算時間の比較

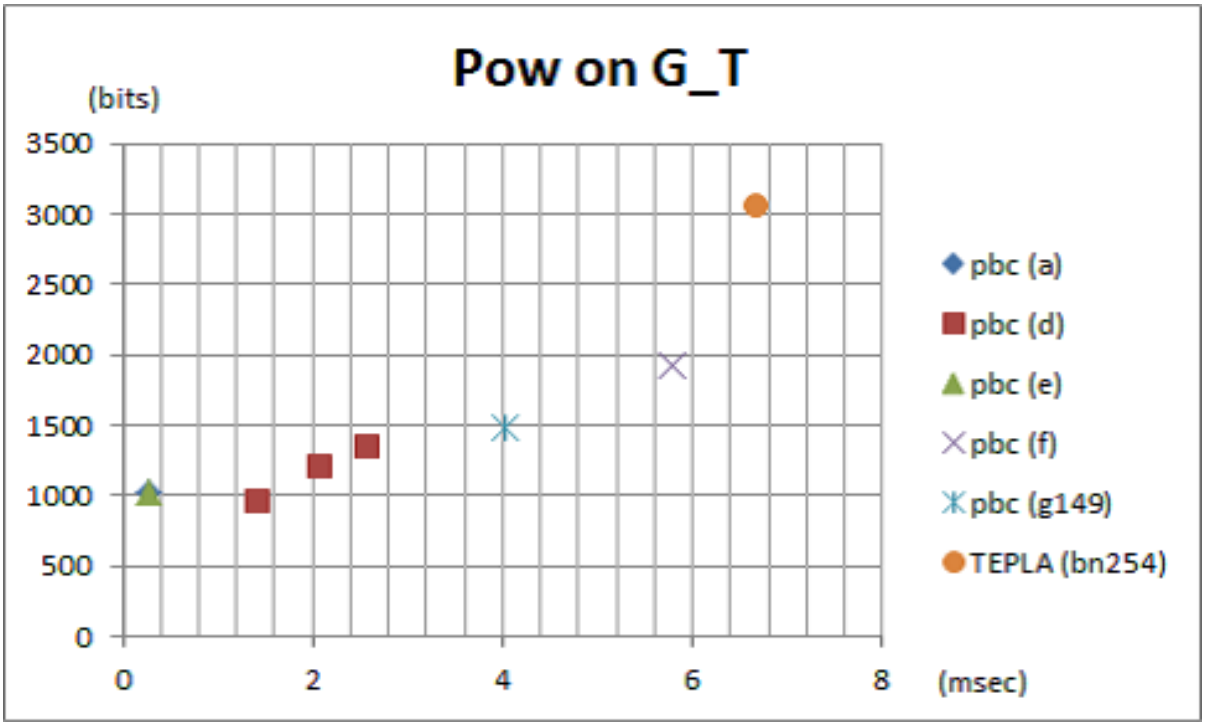

図４　ペアリングの出力となる体の元でのべき乗剰余演 算時間の比較

べき乗剰余演算では、TEPLA の結果と PBC Library の各パラメータの結果は強度と時間が比例している様子が見てとれるが、他の演算では TEPLA は高い強度にもかかわらず高速に処理できていることがわかる。これは実装の高速性を示すものではなく、主に BN 曲線と Optimal Ate ペアリングの効率性の高さに依るものである。

３．４．３　プラットフォームごとの TEPLA のパフォーマンス

ここでは TEPLA による各種演算がプラットフォームでどう変化するかを評価する。プラットフォームは 3 つ準備した。それぞれのプラットフォームの仕様を表 2 と表 3 に示す。

結果を表 4 に示す。Linux は Windows と Mac OS X と比較して低速であったものの、全般的に高速な値が出ていることが見て取れる。

表２　TEPLA と PBC Library のパフォーマンス比較 (msec)

OS	CPU	RAM(GB)
Windows 7 Professional SP1 (64bit)	Intel Core i7 3960X (3.30GHz)	8
Linux(Cent OS, 2.6.18-308.8.1.el5)	Intel Core i7 920 (2.67GHz)	6
Mac OS X 10.6.8(Snow Leopard)	Intel Core i7-2820QM (2.30GHz)	8
Android 4.2.1	Tegra3 (1.3GHz)	1
iOS6.0.1	Apple A5X (1.0)	1

表３　TEPLA パフォーマンス評価に用いたプラットフォームの情報(1)

OS	C コンパイラ	GMP	OpenSSL
Windows 7 Professional SP1 (64bit)	Visual C++ 2010	(MPIR 2.6.0)	1.0.1c
Linux(Cent OS, 2.6.18-308.8.1.el5)	gcc 4.1.2	5.0.5	0.9.8e-fips-rhel5
Mac OS X 10.6.8(Snow Leopard)	gcc4.2.1	5.0.4	0.9.8r
Android	Android NDK r8d	5.1.0	1.0.1c
iOS	Xcode 4.5.2	5.0.5	1.0.1c

表４　TEPLA パフォーマンス評価に用いたプラットフォームの情報(2)

	Windows	Linux	Mac OS X	Android	iOS
ペアリング演算	3.2313	6.3980	3.5610	25.02	38.06
スカラ倍算	0.3538	0.6836	0.3619	3.01	4.41
Map to Point	0.1661	0.3081	0.1662	2.94	2.00
G_T 上のベキ乗演算	3.2493	6.6730	3.613	26.25	39.56

４．TEPLA の応用

今の時点で、Windows と Linux、Mac OS X での動作が確認されており、非常に幅広く用いることが可能であることが示されている。それにより、図 5 に示すように、サーバ側で Linux により鍵管理サーバが実現され、暗号文送信者側が Mac OS X、受信者側が Windows を用いるなど、ユーザのプラットフォームを問わないシステムの実現が可能になる。

また、現在、ペアリングそのものやペアリングを用いた暗号プロトコルにいくつかは標準化が完了しているか、あるいは標準化に向けた作業が行われている。標準化と TEPLA のような汎用ライブラリがあることで、さまざまな応用が可能になる。

たとえば、用途をあまり限定しない ID ベース暗号用の鍵発行サーバを TEPLA を用いて開発し、データフォーマットや通信プロトコルは標準に即して行えば、ID をその用途ごとに変えてユーザや組織ごとに自由な用途が選択できる暗号アプリケーションとしてアプリケーションを用意しても、閉じた鍵発行サーバではなく広く公開されたシステムとしての利用可能な環境を提供することが可能になり、一般利用を促進することもできよう。

図５　TEPLA の応用

５．まとめ

本稿では、ペアリングを用いた暗号プロトコルのアプリケーションやシステムを容易に開発可能な、汎用的なペアリング暗号ライブラリ TEPLA の開発とその内容について報告を行った。オープンソースで提供され、さまざまなプラットフォームで利用可能であり、また、採用している楕円曲線やペアリングの高速性の影響もあり高いセキュリティレベルにありながら高速性を実現しており、その有用性は高い。

TEPLA を用いることで、より広範な利用を促進することが可能となる。TEPLA が、これまでペアリングを用いた暗号についての知識を持たなかったアプリケーション・システム開発者にとって新たな用途を模索する土台となることを期待する。

TEPLA はすでに使える状態になっており、筑波大学暗号と情報セキュリティ研究室サーバにおいてある[19]。

参考文献

[1] 石井健太、齋藤和孝、金岡晃、金山直樹、岡本栄司、"汎用的に利用可能なペアリング暗号 C 言語ライブラリの開発"、SCIS2013, 3E1-3, 2013 年

[2] Ben Lynn, "PBC Library", Standord University, Applied Crypto Group, http://crypto.stanford.edu/pbc/, 参照 2012 年 12 月

[3] "The GNU Multiple Precision Arithmetic Library", http://gmplib.org/, 参照 2012 年 12 月

[4] D. F. Aranha and C. P. L. Gouv¥^{e}a, "RELIC is an Efficient LIbrary for Cryptography", http://code.google.com/p/relic-toolkit/, 参照 2012 年 12 月

[5] D. F. Aranha, C. P. L. Gouv¥^{e}a, J. Lopez, L. B. Oliveira, M. Scott, R. Dahab, "Fully Enabling PKC in SensorNets", https://sites.google.com/site/tinypbc/, 参照 2012 年 12 月

[6]Beuchat, J.L., Díaz, J.E.G., Mitsunari, S., Okamoto, E., Rodríguez-Henríquez, F., Teruya, T. "High-Speed Software Implementation of the Optimal Ate Pairing over Barreto-Naehrig Curves". In Proc. of Pairing 2010. LNCS 6487, pp. 21–39., 2010

[7] MITSUNARI Shigeo, "High-Speed Software Implementation of the Optimal Ate Pairing over Barreto-Naehrig Curves", http://homepage1.nifty.com/herumi/crypt/ate-pairing.html 、参照 2012 年 12 月

[8] G. Grewal, R. Azarderakhsh, P. Longa, S. Hu, and D. Jao, "Efficient Implementation of Bilinear Pairings on ARM Processors", Cryptology ePrint Archive: Report 2012/408, Jul. 2012

[9] X. Zhang, K. Wang, D. Lin, "On Efficient Pairings on Elliptic Curves over Extension Fields", Pairing 2012, May 2012

[10] I. Kim, S. O. Hwang, "An Efficient Implementation of Tate and Ate Pairings", International Journal of Multimedia and Ubiquitous Engineering, Vol. 7, No. 2, April 2012

[11] Aranha, D.F., L´opez, J., Hankerson, D.: High-speed parallel software implementation of the ηT pairing. In: Pieprzyk, J. (ed.) CT-RSA 2010. LNCS 5985, pp.89–105. Springer, Heidelberg (2010)

[12] Beuchat, J.-L., L´opez-Trejo, E., Mart´ınez-Ramos, L., Mitsunari, S., Rodr´ıguezHenr´ıquez, F.: Multi-core implementation of the Tate pairing over supersingular elliptic curves. In: Garay, J.A., Miyaji, A., Otsuka, A. (eds.) CANS 2009. LNCS 5888, pp. 413–432. Springer, Heidelberg (2009)

[13] Devegili, A.J., Scott, M., Dahab, R.: Implementing cryptographic pairings over Barreto–Naehrig curves. In: Takagi, T., Okamoto, T., Okamoto, E., Okamoto, T. (eds.) Pairing 2007. LNCS 4575, pp. 197–207. Springer, Heidelberg (2007)

[14] Grabher, P., Großsch¨adl, J., Page, D.: On software parallel implementation of cryptographic pairings. In: Avanzi, R.M., Keliher, L., Sica, F. (eds.) SAC 2008. LNCS 5381, pp. 34–49. Springer, Heidelberg (2008)

[15] Naehrig, M., Niederhagen, R., Schwabe, P.: New software speed records for cryptographic pairings. Cryptology ePrint Archive, Report 2010/186 (2010), http://eprint.iacr.org/2010/186.pdf, 参照 2012 年 12 月

[16] Hankerson, Darrel, Menezes, Alfred J., Vanstone, Scott ,"Guide to Elliptic Curve Cryptography", Springer, January 2004

[17] Nogami, Yasuyuki, Sakemi, Yumi, Okimoto, Takumi, Nekado Kenta, Akane Masataka, Morikawa, Yoshitaka, "Scalar Multiplication Using Frobenius Expansion over Twisted Elliptic Curve for Ate Pairing Based Cryptography", IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Volume E92.A, Issue 1, pp.182-189 (2009).

[18] IEEE P1363.3/D6, "Draft Standard for Public-key Cryptography"

[19] "TEPLA - LCIS, Univ. of Tsukuba", University of Tsukuba, Laboratory of Cryptography and Information Security, http://www.cipher.risk.tsukuba.ac.jp/tepla/, 参照 2012 年 12 月

この研究は、平成２０年度ＳＣＡＴ研究助成の対象として採用され、平成２１～２３年度に実施されたものです。