SCIS 2016 2016 Symposium on
Cryptography and Information Security
Kumamoto, Japan, Jan. 19 - 22, 2016
The Institute of Electronics,
Information and Communication Engineers

対称ペアリングに基づく方式から非対称ペアリングに基づく方式への最適変換
Optimal Conversion from Symmetric Pairing-based Scheme to Asymmetric One

星野 文学 * 阿部 正幸 * 大久保 美也子 †
Fumitaka Hoshino Masayuki Abe Miyako Ohkubo

あらまし 非対称ペアリングは対称ペアリングと比較して代数的構造は複雑だが実装の効率はずっと良い. 従ってペアリングに基づく暗号方式を構成する際一旦は対称ペアリング上で設計を行い, 後に非対称ペアリングの代数的構造に合わせ再設計し実装を得る事が多くなっている. 2014 年阿部らは方式の安全性を維持したままこのタスクを自動実行するフレームワークを発表した. 2015 年に整数計画法を直接用いるアルゴリズムが提案されたことにより, かなり大規模な暗号方式の自動変換が可能となった (未発表). 一方 Groth-Sahai 証明などを用いて設計された大規模な方式を対称ペアリングを使って具体的に記述すると, 方式自体が証明する述語の数に対して指数的多数の変種を持ってしまう事がある. 本発表ではそのような場合でも最適な変換を見つけるアルゴリズムを提案する.

キーワード ペアリング, 整数計画法, Groth-Sahai 証明

1 はじめに

1.1 研究背景

一般に楕円曲線を用いたペアリングの実装では楕円曲線上の $\mathbb{F}_q$ 有理点の部分群 $E(\mathbb{F}_q)[\ell]$ およびそれと同型な群から有限体の乗法群 $\mathbb{F}_{q^k}^*$ への双準同型を利用してペアリングを構成する. この時ペアリングの安全性は少なくとも楕円曲線 $E(\mathbb{F}_q)[\ell]$ 上の楕円離散対数問題および有限体の乗法群 $\mathbb{F}_{q^k}^*$ 上の離散対数問題よりは易しい. k は埋め込み次数 (embedding degree) と呼ばれる整数で, k が小さいと解析計算量の漸近挙動が準指数的な乗法群の離散対数問題に引きずられ q を大きくする必要があり $E(\mathbb{F}_q)$ の演算効率が落ちるし, k が大き過ぎると $\mathbb{F}_{q^k}^*$ の演算効率が落ちる. 従ってペアリングの実装を行なう場合は楕円離散対数問題および乗法群の離散対数問題の解析計算量が両方とも適当な大きさとなるよう q, k を適切な値に設定する事が望ましい [5]. かつては小標数の楕円曲線を用いた対称ペアリングが実装に優れた性質を持つと期待されていたが [25, 14, 23, 28], 近年攻撃技術の進展によりあまり安全で無い事が分かってきた [18, 6, 22]. 従って現在では対称ペアリングを実装に用いる際は大きい素体の楕円曲線を使う必要があり, その場合埋め込み次数が 2 で固定されてしまう. 一方非対称ペアリングでは, より大きい埋め込み次数や豊かな代数的構造を利用する事によりサイズの小さい楕円曲線が使用できる [7]. しかし非対称ペアリングは対称ペアリングと比較すると代数的構造が複雑なので, 暗号方式の設計には対称ペアリングが良く用いられる. 従って方式の設計を一旦は対称ペアリング上で行い非対称ペアリングの複雑な代数的構造に合わせて方式を再設計し実装を得る事が増えており [21, 16, 15], そのような設計方針自体が設計者にとってメリットの有る新たなアプローチとなっている. さらに将来システムが大規模化して行くと非対称ペアリングを直接用いて最適な実装を得る仕事はいずれ人手では困難になると考えられる.

1.2 関連研究

もし, 対称ペアリングに基づく設計から非対称ペアリングに基づく実装が自動的に得られるなら, 煩わしい再設計の手間が省ける上, かつて対称ペアリング用に設計された大量の方式を見捨てる必要が無くなるであろう. そのような事は可能であろうか？こうした疑問から自然に

(A) 変換可能性の判定 (そもそも変換が可能か否か？)
(B) 実行可能解の求解 (実際にどう変換すれば良いか?)
(C) 最適解の求解 (実装に最も適した変換は何か？)

* NTT セキュアプラットフォーム研究所, 〒 180-8585 東京都武蔵野市緑町 3-9-11, NTT Secure Platform Laboratories, 3-9-11, Midori-cho Musashino-shi, Tokyo 180-8585 Japan
† NICT ネットワークセキュリティ研究所, 〒 184-8795 東京都小金井市貫井北町 4-2-1, Security Fundamentals Lab, NSRI, NICT, 4-2-1 Nukui-Kitamachi, Koganei, Tokyo 184-8795, Japan

といった問題が生じ, それらを解くアルゴリズムの研究が始まった. [27, 10, 26, 9, 11, 24] のような初期の研究では (A),(B) に関して発見的な指標を与えるに留まっていた. 2013 年 Akinyele らは CCS2013 にて背景理論付き充足可能性判定 (satisfiability modulo theory, SMT) ソルバを用いて, (A)〜(C) を解く自動変換アルゴリズムおよびツール (AutoGroup) を発表したが, この変換は変換後の方式の安全性を保証するものではなかった [4]. 2014 年阿部らは CRYPTO2014 にて, 方式および安全性証明を抽象した依存関係グラフを分割する事によって, 方式の安全性を維持したまま (A)〜(C) を解く為のフレームワーク及びアルゴリズムを発表したが, このアルゴリズムは方式の規模に対して指数時間であった [2]. 2015 年丹後らは SCIS2015 にて上記フレームワークの下 (A) を解く多項式時間アルゴリズムを提案したが (B),(C) については未解決のままであった [30]. 同年 Akinyele らは CCS2015 にて SMT ソルバを使って上記フレームワークの下 (A)〜(C) を解くアルゴリズムおよびツール (AutoGroup+) を発表した [3]. 同年阿部らは未発表論文 [1] にて整数計画法 (integer programming, IP)[13] を使って上記フレームワークの下 (A)〜(C) を解く実用的なアルゴリズムを提案した.

1.3 研究動機

整数計画法を直接用いるアルゴリズムにより, かなり大規模な暗号方式の自動変換が可能となった [1]. ところで大規模な方式の設計を行なう場合ペアリングやべき乗といった概念よりも抽象度が高い概念を暗号プリミティブとして利用する事がある (例えば Groth-Sahai 証明 [19, 20] など). こうした抽象度の高いプリミティブ (上位プリミティブ) をペアリングやべき乗といった具体的プリミティブで実装する時, その実装方法は必ずしも一意であるとは限らない. しかもその最適な実装は上位プリミティブそのものでは無くそれを用いて設計している方式に依存して変化し得る. 即ち上位プリミティブを用いると対称ペアリングによって記述されるアルゴリズム自体が一意に定まらない事がある. そのような上位プリミティブが方式の規模に比例した数存在すると, 変換前の対称ペアリングで記述した方式自体が指数的多数となってしまい, その中で最適な変換を見つけるタスクは事実上計算不能となってしまう. 本稿ではこの問題を回避する方法を研究する.

2 準備

2.1 ペアリング

暗号学においてペアリングとは概ね次のような代数的構造を持つ符号を生成する確率的多項式時間アルゴリズム $\mathcal{P}$ の事である.

$$\mathcal{P} : 1^\lambda \xmapsto{\$} (\ell, \mathbb{G}_0, \mathbb{G}_1, \mathbb{G}_T, e)$$

1. λ は安全変数.
2. ℓ は $\ell > 2^{\Theta(\lambda)}$ なる整数.
3. $\mathbb{G}_0, \mathbb{G}_1, \mathbb{G}_T$ はそれぞれ同型の位数 ℓ の巡回群 (の符号) で, それぞれ多項式時間の標本演算および群演算を持つ.
4. $e : \mathbb{G}_0 \times \mathbb{G}_1 \to \mathbb{G}_T$ は多項式時間非退化双準同型.
5. $\mathbb{G}_0, \mathbb{G}_1, \mathbb{G}_T$ 上の CDH 問題は難しい.

$\mathbb{G}_0, \mathbb{G}_1$ をソース群 (source group), $\mathbb{G}_T$ をターゲット群 (target group) と呼ぶ. 一般にソース群は $\mathbb{G}_1, \mathbb{G}_2$ と記述される事が多いが, 本稿では都合により $\mathbb{G}_0, \mathbb{G}_1$ と記述する. また以降特に明示しない限り単に群要素や群演算などと言う時は, ソース群の要素や演算を意味するとする. Galbraith らは, 暗号方式に用いられるペアリングを大雑把に以下の 3 つの型に分類した [17].

Type 1: $\mathbb{G}_0 = \mathbb{G}_1$

Type 2: $\mathbb{G}_0 \neq \mathbb{G}_1$, $\phi : \mathbb{G}_1 \to \mathbb{G}_0$ なる多項式時間同型写像が存在する.

Type 3: $\mathbb{G}_0 \neq \mathbb{G}_1$, $\mathbb{G}_0, \mathbb{G}_1$ の間に多項式時間同型写像が存在しない.

一般に Type 1 を対称ペアリングと呼び, Type 2, Type 3 を非対称ペアリングと呼ぶ. Type 2 は Type 3 を利用して実装できるので, 方式や安全性証明に ϕ を積極的に利用する時以外は Type 2 が必要となる事はあまり無い. 本稿では非対称ペアリングとして特に Type 3 を想定する. 演算速度や群要素のサイズといった実装上の問題に関して, 対称ペアリングよりもずっと効率的な非対称ペアリングの存在が知られている [7].

2.2 阿部らのフレームワーク [29, 2]

2.2.1 依存関係グラフ

暗号方式に対して, その依存関係グラフ (dependency graph) とは暗号方式を構成する各アルゴリズムや安全性帰着で使用される群要素変数の依存関係を表現する有向グラフである. 図 1(左) は群要素 A, B, D を入力とし群演算 (乗算およびべき乗) を使って C, E を計算しそのペアリング $e(C, E)$ を出力するアルゴリズムの例であり, 図 1(右) はその依存関係グラフである. 依存関係グラフの各ノードはそれぞれ群要素変数を表現しており, 各エッジは群演算による依存関係を表現している (即ち各エッジの目的ノードはソースノードに依存する). ペアリングへの入力はペアリングノード $\mathsf{P}_{\mathsf{CE}}[0]$ および $\mathsf{P}_{\mathsf{CE}}[1]$ への結線により表現される. 依存関係グラフには群演算を介した群要素同士の関係のみが記述され, "`if-then-else`" 命

令のようなプログラムの構造や $a \in \mathbb{Z}/p\mathbb{Z}$ のような群要素以外の変数あるいはターゲット群上の演算等は全て捨象される．阿部らは対称ペアリングに基づく暗号方式を

```
Sample(a, A, B, D):
  a ∈ Z/pZ;
  A, B, C, D, E ∈ G;
  if a = 0 then
    C := A · B; E := D;
  else
    C := D^a; E := D^3;
  endif ;
  Output e(C, E);
```

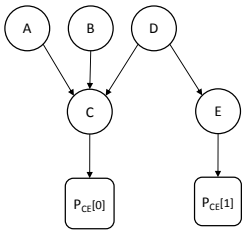
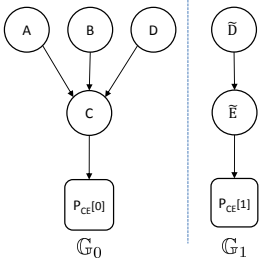


図 1: アルゴリズム例 (左) とその依存関係グラフ (右)

安全性を維持したまま非対称ペアリングに基づく暗号方式に自動変換する為のフレームワークを提案した [2, 29]．阿部らのフレームワークでは, 対称ペアリングに基づく暗号方式を表現した依存関係グラフを, ある制約に従った 2 つの部分グラフに分割する．分割後の 2 つのグラフは変換後の方式を抽象しており, それぞれ $\mathbb{G}_0$ および $\mathbb{G}_1$ の群要素の依存関係グラフとなる (図 2)．このアルゴリ




```
Sample(a, A, B, D, D̃):
  a ∈ Z/pZ;
  A, B, C, D ∈ G_0; D̃, Ẽ ∈ G_1;
  if a = 0 then
    C := A · B; Ẽ := D̃;
  else
    C := D^a; Ẽ := D̃^3;
  endif
  Output e(C, Ẽ)
```

図 2: 分割の例 (左) と変換後のアルゴリズム (右)

ズムを説明する前に, 概ね [29, 2] に従っていくつか用語や記法を定義する．以下依存関係グラフを $\mathcal{G} = (V, E)$ とする．

定義 1 (祖先グラフ $\mathtt{Anc}(\mathcal{G}, x)$) ノード $x \in V$ に対して x に到達可能な全ての経路を含む $\mathcal{G}$ の部分グラフを $\mathcal{G}$ のノード x に関する祖先グラフと呼び $\mathit{Anc}(\mathcal{G}, x)$ と記述する．

定義 2 (祖先ノード) ノード $x \in V$ に対して x に到達可能なノード $y \in V$ *(*但し x を含まない*)* を x の祖先ノードと呼ぶ．このときノード x はノード y を祖先に持つと言う．

$\mathtt{Anc}(\mathcal{G}, x)$ に含まれるノードの全体は x および x の祖先ノードの全体と一致する．

定義 3 (子孫ノード) ノード $x \in V$ に対して x から到達可能なノード $y \in V$ *(*但し x を含まない*)* を x の子孫ノードと呼ぶ．このときノード x はノード y を子孫に持つと言う．

定義 4 (ペアリングノード) 依存関係グラフにおいてペアリングへの入力を表現するノードをペアリングノードと呼ぶ．

全てのペアリング演算が, 対となる 2 つのペアリングノードを持つ．また全てのペアリングノードは一つの入力エッジを持ち出力エッジは無い．

定義 5 (レギュラーノード) ペアリングノードでないノードをレギュラーノードと呼ぶ．

定義 6 (ボトムノード) レギュラーノードのうち出力が全く無いか, 出力が無いループを代表するノードをボトムノードと呼ぶ．

以下 $\mathcal{G}$ から分割によって 2 つの部分グラフ $\mathcal{G}_b = (V_b, E_b)$, $b \in \{0, 1\}$ が生成されるとする．

定義 7 (述語 $x \in \mathbb{G}_b$) 簡単のため $b \in \{0, 1\}$ について, 述語 $x \in V_b$ を $x \in \mathbb{G}_b$ または $\mathbb{G}_b \ni x$ と記述し, 述語 $x \notin V_b$ を $x \notin \mathbb{G}_b$ または $\mathbb{G}_b \ni x$ と記述する．そしてそれらの述語が真であるときその値を 1, 偽であるときその値を 0 と定義する．即ち $(x \in \mathbb{G}_b) \in \{0, 1\}$ である．

定義 8 (二重化ノード) あるノード x が

$$x \in \mathbb{G}_0 \quad \wedge \quad x \in \mathbb{G}_1$$

を満たすなら x を二重化ノードと呼ぶ．

定義 9 (二重化禁止ノード) 何らかの理由により分割後に二重化ノードとなる事を禁止されたノードを二重化禁止ノードと呼ぶ．

二重化禁止ノードはアルゴルズム中の `HashToPoint()` 演算の出力を表現したり, データサイズを節約する為にユーザーによって設定される．以下 $V_p \subset V$ を二重化禁止ノードの全体とし, $P = \{P_i[b] : i \in \{1, \ldots, n_p\}, b \in \{0, 1\}\}$ を $\mathcal{G}$ 中のペアリングノードの全体とし, $P_i[0], P_i[1]$ が i 番目のペアリングの対となる入力とする． また $b \in \{0, 1\}$ に対して $\bar{b} = 1 - b$ とする．

定義 10 (有効分割) $\mathcal{G}_0, \mathcal{G}_1$ が次の性質を満たす時その分割を有効分割 *(valid split)* と呼ぶ．

1. $\mathcal{G}_0 \cup \mathcal{G}_1 = (V_0 \cup V_1, E_0 \cup E_1) = \mathcal{G}$.

2. $\forall b \in \{0,1\}, \forall x \in \mathbb{G}_b$, $\mathit{Anc}(\mathcal{G}, x) \subseteq \mathcal{G}_b$.
3. $\forall i \in \{1, \ldots, n_p\}, \exists b \in \{0,1\}$ |
$(\mathbb{G}_0 \ni P_i[b] \notin \mathbb{G}_1) \wedge (\mathbb{G}_0 \not\ni P_i[\bar{b}] \in \mathbb{G}_1)$.
4. $V_0 \cap V_1 \cap V_p = \emptyset$.

依存関係グラフが有効分割を持つ時, 変換後の方式は元の方式と同様に機能し安全性が保たれる事が知られている [2]. 本稿では (有効) 分割のみを取り扱うので如何なるノード x に関しても

$$x \in \mathbb{G}_0 \quad \vee \quad x \in \mathbb{G}_1$$

が要請される.

定義 11 (クリティカルノード) ペアリングノード, ボトムノード, 二重化禁止ノードをまとめてクリティカルノードと呼び, クリティカルノードでないノードを非クリティカルノードと呼ぶ.

定義 12 (排中律) あるノード x について

$$x \notin \mathbb{G}_0 \quad \vee \quad x \notin \mathbb{G}_1$$

が成立するとき, x は排中律を満たすと言う.

本稿で考える依存関係グラフではクリティカルノードは必ず排中律を満たすとして良い. 一方非クリティカルノードは必ずしも排中律を満たさない.

定義 13 (変数) クリティカルノード x および $b \in \{0,1\}$ に関して改めて論理変数 x の値を

$$x = b \quad \Leftrightarrow \quad \mathbb{G}_b \ni x \notin \mathbb{G}_{\bar{b}}$$

と定義する.

本稿ではクリティカルノードは論理変数とほとんど同義と考える. 非クリティカルノードに関しても似たような定義が可能だが排中律が無いので古典論理とはならない. そのような論理は本稿では扱わない.

3 関連研究の詳細

3.1 阿部らのアルゴリズム [29, 2]

阿部らは暗号方式から最適な有効分割を求める次のアルゴリズムを提案した [2, 29].

Step 1: 暗号方式を構成する各アルゴリズムおよび安全性証明で使用する帰着アルゴリズムにおける群演算の依存関係グラフをそれぞれ構成し, それを一つに統合する.
Step 2: 全てのペアリングノードの対に対して $\mathbb{G}_0 \times \mathbb{G}_1$ または $\mathbb{G}_1 \times \mathbb{G}_0$ のどちらか一方を割り当て, 全てのボトムノードに対して $\mathbb{G}_0$ または $\mathbb{G}_1$ のどちらか一方を割り当てる.
Step 3: 全てのペアリングノードおよびボトムノードに関してその全ての祖先ノードに Step 2 で決めたのと同じ割り当てを行なう. この時 $\mathbb{G}_0$ と $\mathbb{G}_1$ の両方の割り当てが起こった場合は二重化ノードと解釈する.
Step 4: 予め設定されていた二重化禁止ノードが二重化ノードとなっている場合はその割り当ては無効とする. それ以外の場合はグラフの全体の割り当て状況を何らかの評価関数に入力して評価する.
Step 5: 最も評価が良い割り当てに従ってグラフを分割して出力する.

このアルゴリズムでは Step 2 における全ての割り当ての数に比例して Step 3, Step 4 を実行しなくてはならない. これはペアリングの数とボトムノードの数の和を n とすると, 2^n 回の演算を必要としており, n が大きい時には実行困難となる.

3.2 整数計画法を用いた分割 [1]

3.2.1 (A) について

非対称ペアリングにおいては, ペアリングへの入力は一方に $\mathbb{G}_0$ が割当たってるとするなら対になるノードには $\mathbb{G}_1$ が割当たっている必要がある. 従ってペアリングの対になっている (クリティカル) ノード対 $x, y \in \{0,1\}$ について

$$x + y = 1 \tag{1}$$

が成立する. また, あるクリティカルノード x の祖先ノードにクリティカルノード (二重化禁止ノード) y がある場合はそれらのノードの割り当ては等しい, 即ち

$$x - y = 0 \tag{2}$$

が成立する. [1] ではこれらのクリティカルノード間の関係式を $\mathbb{F}_2$ 上線型関係と見なし, 線型方程式系の充足可能性判定による多項式時間変換可能性判定アルゴリズム (sanity checking) を提案している.

例 図 3 は Boneh らの検証者ローカル失効付きグループ署名 [8] の統合後の依存関係グラフである. g_1, g_2, ω がこの方式のグループ公開鍵に相当する. 詳細は [8] を参照のこと. $(p_1[0], p_1[1])$, ... , $(p_8[0], p_8[1])$ がそれぞれ対になるペアリングノードの対である. また出力ノードの無い $R_1, R_3, (g_1^*)^{\frac{1}{\gamma+x}}$ がボトムノードである. また, u, v が禁止ノードにあたる [29]. 従ってこのグラフには, u , v , R_1 , R_3 , $(g_1^*)^{\frac{1}{\gamma+x}}$, $p_1[0]$, $p_1[1]$, ... , $p_8[0]$, $p_8[1]$ の合計 21 個のクリティカルノードが存在する. これらのクリティカルノード間の $\mathbb{F}_2$ 上線型関係を考える. 上記の変数順序に従い係数および定数項の値を順に並べて一つの線型関係を表現する. 例えば $p_1[0], p_1[1]$ はペアリングの対になっているクリティカルノードであるから, $p_1[0] \oplus p_1[1] = 1$ が成立する. 上記の変数順序において,

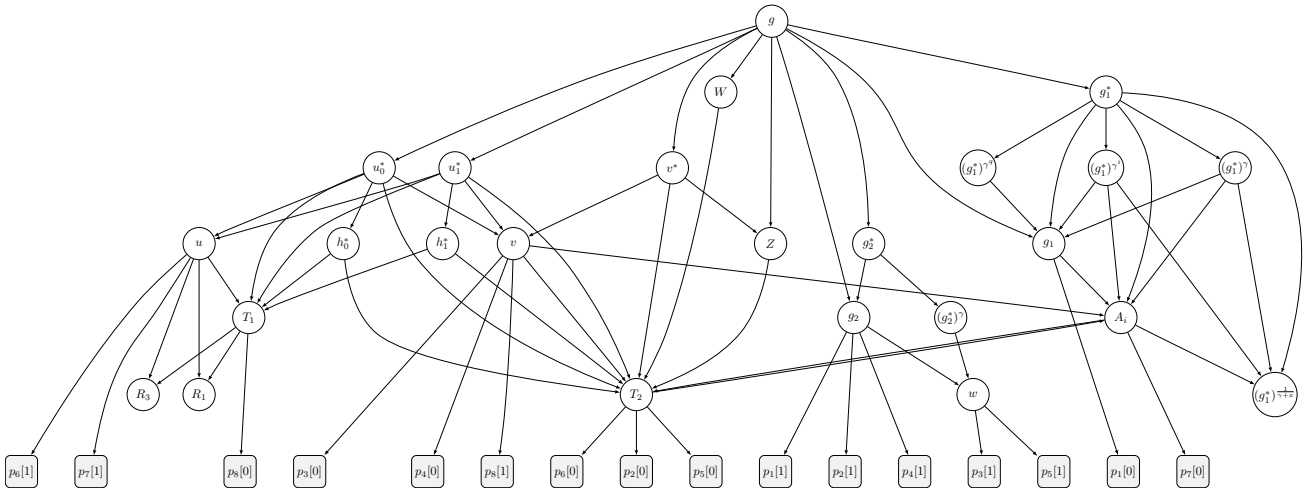


図 3: [8] の依存関係グラフ

6 番目の変数 $p_1[0]$ および 7 番目の変数 $p_1[1]$ の係数が 1 で他の係数が 0 であり, 定数項が 1 であるから, この式を係数および定数項の値だけで表現すると

$$0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1$$

となる. $\mathbb{F}_2$ 上の値は必ず一桁になるので ”,” を省略すると

$$0000011000000000000001$$

と表現できる. この方法で, 図 3 のグラフの線型関係を全て書き下すと, 図 4(左) となる. この行列を $\mathbb{F}_2$ 上の Gauss の消去法を用いて図 4(右) のような階段型 (echelon form) に変換する事が出来る. もし階段型の最下段

00000110000000000000001	10100000000000000000000
00000001100000000000001	01001000000000000000000
00000000011000000000001	00110000000000000000000
00000000000110000000001	00010000000000010000000
00000000000001100000001	00001001000000000000000
00000000000000011000001	00000110000000000000001
00000000000000000011001	00000001100000000000001
00000000000000000000111	00000000110000000000001
10100000000000000000000	00000000011000000000001
10010000000000000000000	00000000000110000000001
01001000000000000000000	00000000000101000000000
01000010000000000000000	00000000000011000000001
01000000100000000000000	00000000000010100000000
01000000010000000000000	00000000000000110000001
01000000000100000000000	00000000000000011000001
01000000000001000000000	00000000000000001100001
01000000000000010000000	00000000000000000110001
10000000000000000100000	00000000000000000011001
01000000000000000010000	00000000000000000001100
10000000000000000001000	00000000000000000000111
10000000000000000000100	00000000000000000000000
01000000000000000000010	00000000000000000000000

図 4: [8] の線型関係行列 (左) とその階段型 (右)

非ゼロ行が

$$0\ldots01$$

となった場合は, 全ての変数の係数が 0 であり, かつ定数項が 1 であるから, どのような変数の割り当てに対してもその式を充足する事は出来ない. 即ちそのグラフは分割不可能となる. 上記のグループ署名の例では最下段非ゼロ行が

$$00000000000000000000111$$

となっているので, 分割可能である. 即ち線型方程式系の充足可能性判定によって (A) を解く事が出来る [1].

3.2.2 (B) について

上記の計算過程の副産物を用いて (B) を解く事が出来る. 本稿では, 階段型の何れかの行 (に対応する関係式) において, 先頭項 (図 4(右) の赤い 1 に対応) となる変数を従属変数と呼ぶ. そして従属変数でない変数を独立変数と呼ぶ. 従って上記の例では $p_1[1], p_8[1]$ は独立変数それ以外は従属変数である. 全ての独立変数の値の割り当てを適当に 1 つ決めると, 階段行列の下の方の関係式から順次従属変数の値を一意に決定することが出来る. 即ち (B) を解く事が出来る (多項式時間).

3.2.3 (C) について

一般に独立変数の個数が n 個あるとすると, 値の割り当て方法 (実行可能解) は 2^n 個存在する. こうした割り当ての内, 何らかの評価基準を満たした最適な分割を見つけたいとする. 例えば図 3 のグループ署名の場合, g_1, g_2, ω が公開鍵に相当する. 従って公開鍵のサイズが最も小さい分割を見つけたいなら g_1, g_2, ω の合計サイズが最も小さいグラフを見つければ良い. 本稿では, このような評価基準をノード x に対する重み w_x を用いて表現する. 例えば図 3 の場合, ノード g_1, g_2, ω は重み 1, 残りのノードについては重み 0 等と設定する.
$|\mathbb{G}_b|$, $b \in \{0,1\}$ を $\mathbb{G}_b$ の元を表現するのに必要なビット数と定義し, $w_{x,b} = w_x \cdot |\mathbb{G}_b|$ と定義する. そしてグラフ

$\mathcal{G}$ に含まれる頂点の集合を $V_{\mathcal{G}}$ とする．これらの記法を用いて，各述語の値をそのまま整数と見なすと上記の公開鍵のサイズを最小化したい問題等は，

$$f(\mathcal{G}) = \sum_{x \in V_{\mathcal{G}}} w_{x,0} \cdot (x \in \mathbb{G}_0) + w_{x,1} \cdot (x \in \mathbb{G}_1)$$

なる目的関数を最小化する問題であると捉えることが出来る．ノード x が子孫に持つクリティカルノードの集合を D_x と定義すると

$$(x \in \mathbb{G}_j) = \bigvee_{d \in D_x} (d \in \mathbb{G}_j)$$

とする事ができる．従って

$$\begin{aligned} (x \in \mathbb{G}_0) &= \textstyle\bigvee_{d \in D_x} \neg d, \\ (x \in \mathbb{G}_1) &= \textstyle\bigvee_{d \in D_x} d \end{aligned}$$

と書くことが出来る．ところで $\{0,1\}$ 上の変数 x, x_1, x_2, y について，

$$\begin{aligned} y = x_1 \wedge x_2 \quad &\Leftrightarrow \quad \begin{cases} y - x_1 - x_2 + 1 \geq 0, \\ x_1 - y \geq 0, \\ x_2 - y \geq 0, \end{cases} \\ y = \neg x \quad &\Leftrightarrow \quad y = 1 - x. \end{aligned} \tag{3}$$

が成立する．即ち $\{0,1\}$ 変数の任意の連言や否定は線型制約の元で一つの変数に置き換え可能である．一般に 0-1 整数計画問題は目的関数や制約式に高次多項式や複雑な論理式が含まれていても，この置き換えを繰り返す事によって線型制約付き線形目的関数の最適化問題に変換出来る事が良く知られている [12]．特に $x_1, \ldots, x_k, y \in \{0,1\}$ の時

$$\begin{aligned} y = \bigwedge_{i=1}^{k} x_i \quad &\Leftrightarrow \quad \begin{cases} y - \sum_{i=1}^{k} x_i + (k-1) \geq 0, \\ y - x_i \leq 0, \forall i \in \{1, \ldots, k\}, \end{cases} \\ y = \bigvee_{i=1}^{k} x_i \quad &\Leftrightarrow \quad \begin{cases} y - \sum_{i=1}^{k} x_i \leq 0, \\ y - x_i \geq 0, \forall i \in \{1, \ldots, k\} \end{cases} \end{aligned}$$

である．従って，$\{0,1\}$ 上の値 $y_{0,x}$ を $y_{0,x} = (x \in \mathbb{G}_0)$ と定義すると，$y_{0,x} = \bigvee_{d \in D_x} \neg d$ であるから $|D_x|$ を D_x に含まれるノードの数とすると

$$y_{0,x} = (x \in \mathbb{G}_0) \quad \Leftrightarrow \quad \begin{cases} y_{0,x} + \sum_{d \in D_x} d \leq |D_x|, \\ y_{0,x} + d \geq 1, \quad \forall d \in D_x \end{cases}$$

と出来る．同様に $y_{1,x} = (x \in \mathbb{G}_1)$ と定義すると，

$$y_{1,x} = (x \in \mathbb{G}_1) \quad \Leftrightarrow \quad \begin{cases} y_{1,x} - \sum_{d \in D_x} d \leq 0, \\ y_{1,x} - d \geq 0, \quad \forall d \in D_x \end{cases}$$

である．従って

$$f(\mathcal{G}) = \sum_{x \in V_{\mathcal{G}}} w_{x,0} \cdot (x \in \mathbb{G}_0) + w_{x,1} \cdot (x \in \mathbb{G}_1)$$

を最小化する問題は $\forall x \in V_{\mathcal{G}}$ に関して

$$\begin{aligned} &y_{0,x} + \textstyle\sum_{d \in D_x} d \leq |D_x|, \\ &y_{0,x} + d \geq 1, \quad \forall d \in D_x, \\ &y_{1,x} - \textstyle\sum_{d \in D_x} d \leq 0, \\ &y_{1,x} - d \geq 0, \quad \forall d \in D_x \end{aligned} \tag{4}$$

なる線型制約の下，

$$f(\mathcal{G}) = \sum_{x \in V_{\mathcal{G}}} w_{x,0} \cdot y_{0,x} + w_{x,1} \cdot y_{1,x} \tag{5}$$

なる線形目的関数を最小化する問題に帰着できる．即ち (1), (2), (4) なる線型制約および (5) なる目的関数を任意の 0-1 整数計画アルゴリズムに入力する事によって，依存関係グラフの最適分割の厳密解あるいは近似解を得ることが出来る．

4 提案技術

4.1 解くべき課題

1.3 節で述べた通り上位プリミティブを使用した方式を対称ペアリングによって記述する際，方式のアルゴリズム自体が一意に定まらない事がある．そのような場合は依存関係グラフは一意に定まらない．例えば Groth-Sahai 証明において

$$\prod_i e(A_i, [X_i]) \prod_{i,j} e([X_i], [X_j])^{a_{ij}} = \prod_i e(B_i, C_i)$$

なる等式の非対話零知識証明 (noninteractive zero-knowledge proofs, NIZK) は

$$\prod_i e(A_i, [X_i]) \prod_i e(B_i^{-1}, [C_i]) \prod_{i,j} e([X_i], [X_j])^{a_{ij}} = 1$$

なる等式の非対話証拠識別不能証明 (noninteractive witness-indistinguishable proofs, NIWI) を利用して構成される場合がある [20]．但し $[X]$ は群要素 X についてコミットされた群要素変数であるとする．この時利用しているペアリングが対称ペアリングである場合 $e(B_i^{-1}, [C_i])$ を用いる代わりに $e(C_i^{-1}, [B_i])$ を用いても構わない．即ち Groth-Sahai 証明を対称ペアリングで実装する際には，定数ペアリングの数を n とすると $N = 2^n$ 個の自由度が存在し，その依存関係グラフは一意に定まらない．従ってそのような上位プリミティブが方式内に m 個存在すると，全ての実装の中で最適な方式を得るには N^m 回の整数計画法を呼び出す必要があった．

4.2 提案法

方式の規模を表すパラメタを s とすると実装の自由度 N が s に対して多項式 (即ち $n = O(\log(s))$) であるような上位プリミティブが方式内に多項式個 ($m = O(s^c)$) ぐら

い存在するような状況を考える．このような状況下では依存関係グラフが一意に定まらない場合でも，下記のように仮想的な依存関係グラフを考える事によって単一の整数計画問題インスタンスを構成出来る．

Step.1: 各上位プリミティブに対して 1 つづつ整数変数を定義する．
Step.2: 上位プリミティブのあり得る実装を全て仮想的に実装し各仮想的実装に上記整数変数の値を 1 つずつ割り当てる．この時，元の方式に陽には含まれない群要素変数で，複数の仮想的実装から参照されるものはまとめて 1 つだけ (仮想的に) 実装する．
Step.3: 値が割り当てられた仮想的実装に含まれる群要素変数を評価関数に組み入れる際は，上記整数変数がその値と一致する時だけ評価されるよう，係数に上記整数変数の述語 (等式) を設定する．複数の仮想的実装から参照される仮想的な群要素変数については上記の述語を $\vee$ で繋いだものを係数に設定する．
Step.4: 必要なら整数変数はバイナリ変数に変換する．この時，整数変数の述語 (等式) はバイナリ変数の述語 (等式) を $\wedge$ で繋いだ式に変換される．さらにバイナリ変数の述語 (等式) はリテラル (バイナリ変数それ自身か，またはその否定) に変換される．最終的に式 (3) 等を使って評価関数を線型化する．

例 図 5 のような NIZK のシーケンスを考える．ここで

$g, X, A, B, C \in \mathbb{G}$;
$\mathtt{crs} := \mathtt{CrsGen}(g)$;
$[X] := \mathtt{Commit}(\mathtt{crs}, X)$;
$[B] := \mathtt{Commit}(\mathtt{crs}, B)$;
$\mathtt{proof} := \mathtt{NIZK.Prove}(\mathtt{crs}, e(A,[X]) = e(B,C))$;
$\mathtt{NIZK.Verify}(\mathtt{crs}, \mathtt{proof}, e(A,[X]) = e(B,C))$;
...

図 5: NIZK シーケンスの例

$\mathtt{crs}, [X], [B], \mathtt{proof}$ 等は各右辺の関数内部で具体的な群演算によって生成される有限個の群要素変数のリストとする．以下，それらのリストのビット長を $|[X]|$ 等と記述する．一般には $\mathtt{NIZK.Prove}$ と対応する $\mathtt{NIZK.Verify}$ は方式を構成する異なるアルゴリズム中で使用されるが，最終的に依存関係グラフは一つに統合されるのでここでは一つのシーケンスとして記述する．コミット $[B]$ は ... の部分で参照されているとする．上記 **Step.1** に従って図 5 のシーケンス中の上位プリミティブ $\mathtt{NIZK}$ に対し変数 $v \in \{0,1\}$ を準備する．そして **Step.2** に従い実装を図 6 のように列挙し，変数の値を割り当てる (コメント部)．ここで，元の方式には陽に含まれていないコミット $[C]$ は $v = 0$ の仮想的実装からのみ参照される．仮にコミット $[C]$ が他の仮想的実装から参照されるとしても 1 つだけ実装する．Groth-Sahai 証明において NIWI を使って NIZK を構成する為には，実際には追加の NIWI が幾らか必要となるが，図 6 では ... と記述して省略している．**Step.3** に従い目的関数が例えばコミットのサ

$g, X, A, B, C \in \mathbb{G}$;
$\mathtt{crs} := \mathtt{CrsGen}(g)$;
$[X] := \mathtt{Commit}(\mathtt{crs}, X)$;
$[B] := \mathtt{Commit}(\mathtt{crs}, B)$;
// extra commitment
$[C] := \mathtt{Commit}(\mathtt{crs}, C)$; // if $v = 0$
...
// $v = 0$
$\mathtt{proof}_1 := \mathtt{NIWI.Prove}(\mathtt{crs}, e(A,[X]) = e(B,[C]))$;
$\mathtt{NIWI.Verify}(\mathtt{crs}, \mathtt{proof}_1, e(A,[X]) = e(B,[C]))$;
...
// $v = 1$
$\mathtt{proof}_2 := \mathtt{NIWI.Prove}(\mathtt{crs}, e(A,[X]) = e([B],C))$;
$\mathtt{NIWI.Verify}(\mathtt{crs}, \mathtt{proof}_2, e(A,[X]) = e([B],C))$;
...

図 6: NIWI への展開

イズを最小化する関数であったとするなら

$$f(\mathcal{G}) = |[X]| + |[B]| + (v = 0) \cdot |[C]|$$

等と定義する．コミット $[X], [B]$ は元々の方式で定義されているので無条件に目的関数に組み入れられている．一方コミット $[C]$ は上位プリミティブの展開によって新たに発生するコミットなので条件付きの項となる．**Step.4** に従い $|[X]|, |[B]|, |[C]|$ は 3.2.3 節の方法に従ってクリティカルノードの論理式に展開される．

$$f(\mathcal{G}) = \sum_{x \in [X] \cup [B]} |\mathbb{G}_0| \cdot (x \in \mathbb{G}_0) + |\mathbb{G}_1| \cdot (x \in \mathbb{G}_1) + \sum_{x \in [C]} |\mathbb{G}_0| \cdot (\neg v \wedge (x \in \mathbb{G}_0)) + |\mathbb{G}_1| \cdot (\neg v \wedge (x \in \mathbb{G}_1))$$

そして最終的に (1), (2) および

$$\left.\begin{array}{l} z_{b,x} + v - y_{b,x} \geq 0, \\ 1 - v - z_{b,x} \geq 0, \\ y_{b,x} - z_{b,x} \geq 0, \end{array}\right\} \forall x \in [C], \forall b \in \{0,1\}$$

$$\left.\begin{array}{l} y_{0,x} + \sum_{d \in D_x} d \leq |D_x|, \\ y_{0,x} + d \geq 1, \quad \forall d \in D_x, \\ y_{1,x} - \sum_{d \in D_x} d \leq 0, \\ y_{1,x} - d \geq 0, \quad \forall d \in D_x \end{array}\right\} \forall x \in [X] \cup [B] \cup [C]$$

なる線形制約の下

$$f(\mathcal{G}) = \sum_{x\in[X]\cup[B]} |\mathbb{G}_0|\cdot y_{0,x} + |\mathbb{G}_1|\cdot y_{1,x} + \sum_{x\in[C]} |\mathbb{G}_0|\cdot z_{0,x} + |\mathbb{G}_1|\cdot z_{1,x}$$

なる目的関数を最小化する 1 つの 0-1 整数線形計画問題インスタンスに帰着される.

参考文献

[1] M. Abe, J. Groth, F. Hoshino, and M. Ohkubo. A fast and scalable bilinear-type conversion using integer programming. unpublished work, 2015.

[2] M. Abe, J. Groth, M. Ohkubo, and T. Tango. Converting cryptographic schemes from symmetric to asymmetric bilinear groups. In J. A. Garay and R. Gennaro, editors, *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 241–260. Springer, 2014.

[3] J. A. Akinyele, C. Garman, and S. Hohenberger. Automating Fast and Secure Translations from Type-I to Type-III Pairing Schemes. In I. Ray, N. Li, and C. Kruegel, editors, *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-6, 2015*, pages 1370–1381. ACM, 2015.

[4] J. A. Akinyele, M. Green, and S. Hohenberger. Using SMT solvers to automate design tasks for encryption and signature schemes. In A. Sadeghi, V. D. Gligor, and M. Yung, editors, *2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013*, pages 399–410. ACM, 2013.

[5] D. F. Aranha, P. S. L. M. Barreto, P. Longa, and J. E. Ricardini. The realm of the pairings. In T. Lange, K. E. Lauter, and P. Lisonek, editors, *Selected Areas in Cryptography - SAC 2013 - 20th International Conference, Burnaby, BC, Canada, August 14-16, 2013, Revised Selected Papers*, volume 8282 of *Lecture Notes in Computer Science*, pages 3–25. Springer, 2013.

[6] R. Barbulescu, P. Gaudry, A. Joux, and E. Thomé. A quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. *IACR Cryptology ePrint Archive*, 2013:400, 2013.

[7] P. S. L. M. Barreto and M. Naehrig. Pairing-friendly elliptic curves of prime order. In B. Preneel and S. E. Tavares, editors, *Selected Areas in Cryptography, 12th International Workshop, SAC 2005, Kingston, ON, Canada, August 11-12, 2005, Revised Selected Papers*, volume 3897 of *Lecture Notes in Computer Science*, pages 319–331. Springer, 2005.

[8] D. Boneh and H. Shacham. Group signatures with verifier-local revocation. In V. Atluri, B. Pfitzmann, and P. D. McDaniel, editors, *Proceedings of the 11th ACM Conference on Computer and Communications Security, CCS 2004, Washington, DC, USA, October 25-29, 2004*, pages 168–177. ACM, 2004.

[9] S. Chatterjee, D. Hankerson, E. Knapp, and A. Menezes. Comparing two pairing-based aggregate signature schemes. *Des. Codes Cryptography*, 55(2-3):141–167, 2010.

[10] S. Chatterjee and A. Menezes. On cryptographic protocols employing asymmetric pairings - the role of Ψ revisited. *IACR Cryptology ePrint Archive*, 2009:480, 2009.

[11] S. Chatterjee and A. Menezes. On cryptographic protocols employing asymmetric pairings - the role of Ψ revisited. *Discrete Applied Mathematics*, 159(13):1311–1322, 2011.

[12] D.-S. Chen, R. G. Batson, and Y. Dang. Transformation using 0-1 variables. In *Applied Integer Programming: Modeling and Solution*, chapter 3, pages 54–78. John Wiley & Sons, Inc., Hoboken, New Jersey, 2010.

[13] G. B. Dantzig. On the significance of solving linear programming problems with some integer variables. *Econometrica*, 28(1):30–44, 1960.

[14] I. M. Duursma and H. Lee. Tate pairing implementation for hyperelliptic curves $y^2 = x^p - x + d$. In C. Laih, editor, *Advances in Cryptology - ASIACRYPT 2003, 9th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, November 30 - December 4, 2003, Proceedings*, volume 2894 of *Lecture Notes in Computer Science*, pages 111–123. Springer, 2003.

[15] A. Fujioka, F. Hoshino, T. Kobayashi, K. Suzuki, B. Ustaoglu, and K. Yoneyama. id-eCK Secure ID-Based Authenticated Key Exchange on Symmetric and Asymmetric Pairing. *IEICE Transactions*, 96-A(6):1139–1155, 2013.

[16] A. Fujioka, K. Suzuki, and B. Ustaoglu. Ephemeral key leakage resilient and efficient id-akes that can share identities, private and master keys. In M. Joye, A. Miyaji, and A. Otsuka, editors, *Pairing-Based Cryptography - Pairing 2010 - 4th International Conference, Yamanaka Hot Spring, Japan, December 2010. Proceedings*, volume 6487 of *Lecture Notes in Computer Science*, pages 187–205. Springer, 2010.

[17] S. D. Galbraith, K. G. Paterson, and N. P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.

[18] F. Göloglu, R. Granger, G. McGuire, and J. Zumbrägel. On the function field sieve and the impact of higher splitting probabilities: Application to discrete logarithms in $\mathbb{F}_{2^{1971}}$. *IACR Cryptology ePrint Archive*, 2013:74, 2013.

[19] J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In N. P. Smart, editor, *EUROCRYPT*, volume 4965 of *Lecture Notes in Computer Science*, pages 415–432. Springer, 2008.

[20] J. Groth and A. Sahai. Efficient noninteractive proof systems for bilinear groups. *SIAM J. Comput.*, 41(5):1193–1232, 2012.

[21] D. Hofheinz. Algebraic partitioning: Fully compact and (almost) tightly secure cryptography. *IACR Cryptology ePrint Archive*, 2015:499, 2015.

[22] A. Joux. A new index calculus algorithm with complexity $L(1/4 + o(1))$ in very small characteristic. *IACR Cryptology ePrint Archive*, 2013:95, 2013.

[23] Y. Kawahara, M. Shirase, T. Takagi, and E. Okamoto. Efficient Software Implemenetation of η_T Pairing. In *Proc. of SCIS 2007 The 2007 Symposium on Cryptography and Information Security Sasebo, Japan, Jan. 23-26, 2007*. IEICE, 2012.

[24] T. Kobayashi, F. Hoshino, and K. Suzuki. Symmetric Pairing based on Asymmetric Pairing. In *Proc. of SCIS 2012 The 29th Symposium on Cryptography and Information Security Kanazawa, Japan, Jan. 30 - Feb. 2, 2012*. IEICE, 2012.

[25] D. Page and N. P. Smart. Hardware implementation of finite fields of characteristic three. In B. S. K. Jr., Ç. K. Koç, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, volume 2523 of *Lecture Notes in Computer Science*, pages 529–539. Springer, 2002.

[26] M. Shirase. Symmetric Pairing on Ordinary Elliptic Curves. In *Proc. of CSS 2010 Computer Security Symposium 2010 in Okayama, Japan, Oct. 19-21, 2010*. IPSJ, SIG CSEC, 2010.

[27] N. P. Smart and F. Vercauteren. On computable isomorphisms in efficient asymmetric pairing-based systems. *Discrete Applied Mathematics*, 155(4):538–547, 2007.

[28] G. Takahashi, F. Hoshino, and T. Kobayashi. Efficient GF(3^m) Multiplication Algorithm for η_T Pairing. In *Proc. of SCIS 2008 The 2008 Symposium on Cryptography and Information Security Miyazaki, Japan, Jan. 22-25, 2008*. IEICE, 2012.

[29] T. Tango, M. Abe, and T. Okamoto. Implementation of Automated Translation for Schemes on Symmetric Bilinear Groups. In *Proc. of SCIS 2014 The 31st Symposium on Cryptography and Information Security kagoshima, Japan, Jan. 21 - 24, 2014*. IEICE, 2014.

[30] T. Tango, M. Abe, T. Okamoto, and M. Ohkubo. Polynomial-Time Algorithm for Deciding Possibility of Converting Cryptographic Schemes from Type-I to III Pairing Groups. In *Proc. of SCIS 2015 The 32nd Symposium on Cryptography and Information Security Kokura, Japan, Jan. 20 - 23, 2015*. IEICE, 2015.