4V-05

順序付けIDベースアグリゲート署名についての実装と性能評価

玉井　裕太　　稲村　勝樹　　岩村　恵市
東京理科大学　工学研究科電気工学専攻

1. はじめに

近年,消費者生成メディアという概念が発生している.有名な CGM サービスのサイト例として You Tube などがある．CGM サービスにおいてはある一般ユーザーがその他のユーザーに対してコンテンツを提供し、それを２次利用して新たなコンテンツを作成するという流通形態をとっており.ユーザー自身によって様々なコンテンツの作成が行われる．このようなコンテンツに対し，特定の管理者のみがコピー作成やコンテンツ利用に制限を施す従来のデジタル著作権管理（DRM）技術とは異なる新たな著作権管理の仕組みが必要となる．そこで， CGM の利用シーンに適応した新しい著作権保護方式が電子署名を用いて提案された [1][2] ．

CGM コンテンツサービスはだれが作ったかといいうことを主張するため，特定の名前(ニックネームなど)でコンテンツを公開している．したがって，これらの名前などを用いて直接署名を検証できれば，コンテンツ制作者が誰であるかをより強固に主張することが期待できる.

そこで我々は，検証鍵に ID 情報を使用する ID ベース署名を順次合成していくことで署名順序まで検証できるアグリゲート署名方式を用いて上記[1][2]を実装することを検討している.

本論文では，最初に行った[2]に関する ID ベースアグリゲート署名のプログラム実装についてその現状を説明する.

2. 階層型アグリゲート署名

アグリゲート署名は異なる署名者がそれぞれ異なるコンテンツに署名したものを一つにまとめることができ，検証者はそのまとめた署名に対して 1 回検証をするだけで，署名者全員の署名を検証することができる方式である．これを 1 つのコンテンツを表す階層的な木構造に適用することで，そのコンテンツに関わる著作者の署名を 1 つにまとめることができる．よって、1 つのコンテンツに関わる著作者の関係の正当性（1 次著作者 A のコンテンツを 2 次著作者 B が 2 次利用しているという関係）を１つの署名によって検証できる.

3. 順序付け ID ベースアグリゲート署名

階層型アグリゲート署名を ID ベース署名に拡張したものを以下に示す．それぞれの署名者はユーザー情報 ID，平文 m 及びアグリゲート署名 U,V を公開情報として，n 番目の署名者は次の計算を行う.

$$U_n = r_n P \quad s.t. r_n \in \mathbb{Z}_q^*$$
$$h_n = H_2(ID_n, m_n, U_n)$$
$$V_n = V_{n-1} + r_n h_{n-1} + r_n h_i$$

検証者は，上記のユーザーの公開情報とセンターの公開情報 Q を収集し以下の式を検証することでできる.

$$\begin{aligned} e(P,V) &= e\left(P, \left(\sum_{i=1}^{n} d_{ID_i}\right) + r_1 h_1 + \sum_{i=2}^{n} r_i (h_{i-1} + h_i)\right) \\ &= \left(\prod_{i=1}^{n} e\left(P_{pub}, Q_{ID_i}\right)\right) \cdot e(U_1, h_1) \cdot \left(\prod_{i=2}^{n} e\left(U_i, h_{i-1} + h_i\right)\right) \end{aligned}$$

4. 性能評価

4.1. 実装方法

本研究では，上記の署名生成処理と検証処理を実現するプログラムを実装した．実装に用いる基本コンポーネントとして OpenSSL，GMP ライブラリ，および TEPLA を使用した.

OpenSSL：SSL プロトコル・TLS プロトコルの，オープンソースで，基本的な暗号化関数と様々なユーティリティ関数を提供する.

Implementation and Evaluating of an Order-specified ID-based Aggregate Signature Scheme
Yuta Tamai Masaki Inamura Keiichi Iwamura
Department of Electrical Engineering, Tokyo University of Science

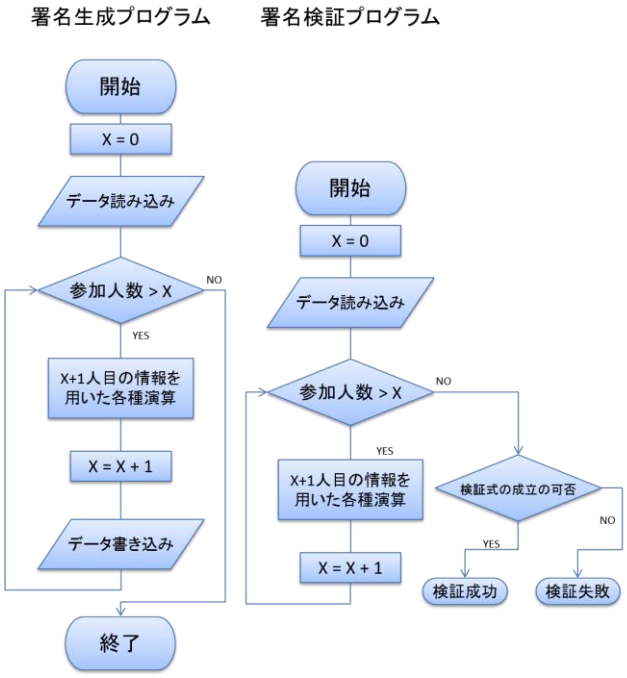
GMP ライブラリ：GNU Multi-Precision Library (GMP)は，多倍長整数など任意の精度の算術ライブラリを提供するフリーソフトウェアである．

TEPLA:楕円曲線上の演算や有限体の元の演算などを機能として備え，様々なプラットフォームにおいて暗号システムを構築可能にすることを目的としたC言語公開ライブラリである．

4.2. プログラムの構成

本研究において実装するプログラムにおいて，プログラム全体のフローチャートを以下に示す．

なお署名検証プログラムにおいては TEPLA における動的変数の定義が不可能な為に 3.に示した式を実現するためにループ処理を用いている．



5. 性能評価

以下に動作環境及び動作結果を示す．署名参加人数は一人，三人，五十人とし，全体の署名時間，そこから算出された署名参加者の平均署名時間並びに署名検証時間を示す．

5.1. 動作環境

CPU: Intel Core i7 920(3.20 GHz)
メモリ: 12.0GB RAM
実装環境: Linux(Ubuntu)
Linux カーネル: 3.19.0-33-generic
TEPLA: 1.0
GMP: 6.0.0
OpenSSL: 1.0.1f

5.2. 動作結果

署名時間[ms]	2459
検証時間[ms]	9961

署名妊数:一人

署名時間[ms]	8797
平均署名時間[ms]	2932
検証時間[ms]	27719

署名人数:三人

署名時間[ms]	162622
平均署名時間[ms]	3252
検証時間[ms]	494790

署名人数:五十人

6. 考察

署名時間は人数が増えるに従ってその平均署名時間が僅かに増えているのがわかる．その署名時間もおよそ 3 秒ほどであり，実用性を考えると少し時間がかかりすぎている．さらに，表には無いが個々の署名時間を見るとその最小値と最大値には 3 倍ほどの差があり，安定性に欠けていた．

また検証時間は署名者が増えるに従って，ほぼ線形で増えていることがわかる．これは署名者が増えることで，同じ比率で公開情報が増えるために計算量が増えていると考えられ，妥当な結果が得られているとわかる．

7. 終わりに

今回 Linux 環境下において順序付け ID ベースアグリゲート署名を実装し，その性能を検討することができた．

今後の課題として，6.で述べたように個々の署名時間に大きな差が生じているので，その差を少なくし安定性を求めることで平均署名時間の微小な増加もなくなると考えられる．

参考文献

[1] 稲村勝樹，田中俊昭，“コンテンツの二次利用を実現する著作権保証方式,″暗号と情報セキュリティシンポジウム- SCIS2009, 1B2-4, 2009.
[2] 稲村勝樹，渡辺龍，田中俊昭，“GapDiffie-Hellman 署名に基づいた階層表記型多重署名方式,″信学技報，ISEC2009，Vol.109, No.271, pp.9-14, 2009.