Vol.2013-CSEC-62 No.14
Vol.2013-SPT-6 No.14
2013/7/18

順序付き ID ベースアグリゲート署名についての安全性評価

岩崎 友哉　稲村 勝樹　岩村 惠市

Gap- Diffie-Hellman(GDH)グループに基づく ID ベース署名の順序付きアグリゲート署名についての安全性について考察する。ID 情報を検証鍵 とする ID ベース署名を署名方式として適応することで、検証鍵の持ち主の特定を容易にすることができる。さらに ID ベース署名は複数の署名者が作成した署名を効率よく合成するアグリゲート署名が実現できることが示されており、さまざまな用途への利用が期待できる。本稿ではこの ID ベース署名に基づくアグリ ゲート署名について，連続する前者と後者を紐付ける署名作成の処理を行い，この処理結果を順次合成していくことで，順序付きアグリゲート署名が実現可能で あることを示し、さらに安全性の考察を示す。

Security Evaluation of an Order-specified ID-based Aggeregate Signature Scheme

IWASAKI TOMOYA　INAMURA MASAKI　IWAMURA KEIICHI

I consider provable security of the ordered aggregate signature of ID-based signature based on (GDH) group Gap-Diffie-Hellman. By adapting a signature scheme the ID-based signature verification key and the ID information, it is possible to facilitate the identification of the owner of the verification key. That the aggregate signature to synthesize efficiently signature of multiple signers have created can be achieved has been shown to ID-based signature further, to the use of a variety of applications can be expected. By about aggregate signature based on this ID-based signature, and do the signature creation to give string the latter and the former consecutive, will continue to sequentially combined this process result, ordered aggregate signature can be achieved in this paper indicates that, to show the consideration of safety further.

1 はじめに

近年のコンピューターおよびネットワーク環境の発達により，場所や時間の制限に囚われずにコンテンツファイルを複数の人により作成・使用する場面が増加している．一例として,一般ユーザーが誰でもコンテンツを作成しインターネットで流通させることが可能な消費者生成メディア（CGM:Consumer Generated Media）という概念が発生し,YouTube [1] などの CGM サービスが急速に広まってきている．この CGM サービスにおいて,“マッシュアップ”と呼ばれるコンテンツの二次利用，三次利用，…によるコンテンツ作成が行われており，マッシュアップのための表記法を規定したクリエイティブ・コモンズ [2] のような活動も始まっている．別な例として,紙の消費量の削減や処理の効率化などを目的として,電子ファイルの状態でイントラネットを経由して文書回覧・稟議決裁を行うといったシステム [3] を取り入れている企業が増加している．今後は,このような CGM サービスにおいて，マッシュアップによるコンテンツ作成時に複数の著作権者の権利を保障する方式，あるいは文書回覧・稟議決裁システムにおいて，回覧確認・稟議承認を安全に行う方式の検討が急務であると考えられる．著作権者の権利主張の証拠,あるいは文書に対する承認確認用として電子署名の利用は有効な手段の一つであり，上記の例で示した場面において，複数の署名者による電子署名の作成，およびその署名の検証を効率的に行うことができる多重署名・アグリゲート署名方式の適用は検討に値する．しかし，一般的な多重署名・アグリゲート署名方式は署名者の関係性の区別を表現できない．CGM コンテンツ作成の場合は，マッシュアップのコンテンツ引用順，あるいは素材提供者や編集者といったグループ 毎の区別が，決裁承認の場合は，職位によって承認者の権限が異なることによる責任の範囲の区別が必要となる ことが想定されるが，上記の理由からこれらの署名方式をそのまま適用することは難しい．また，署名者の署名順序を規定できる順序付き多重署名方式も存在するが，この方式は全署名者の署名順が一列である時にその順序を保証するものであり，ある同一グループに属する署名者を対等と見なし，グループ毎の区別を行うといった表現はできない．

我々は上記の課題に対し，これまで Short Signature と呼ばれる Gap Diffie-Hellman（GDH）署名方式 [4]，およびこの GDH 署名に基づく多重署名方式 [5,6] を拡張し，署名者の立場を木構造に配置できる時に，その木構造で表現できる関係性も検証できる木構造表記型多重署名方式[7]およびグループ間の区別のみに特化した階層表記型多重署名方式[8]を提案した ．これらの提案方式は，署名対象となるメッセー

ジが全ての署名者で同一である多重署名方式を拡張したものであり，従って，作成が完了したコンテンツに全著作権者が署名を行う方法，あるいは変更のない文書の回覧確認に有効である．さらに，すでにあるコンテンツを引用・編集して新たなコンテンツを作成し，その引用者が追加で署名を行う，あるいは文書を追加・変更していきながら回覧確認を行っていくような場面を想定し，署名対象のメッセージが署名者によって異なることを想定したアグリゲート署名をベースとした順序付き・木構造表記型アグリゲート署名も提案している[11]．一方で，CGM コンテンツサービスは誰が作ったかと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 GDH 署名に基づくアグリゲート署名方式を拡張し，隣接する署名者間の前後において， 自分とその前者のメッセージに署名し，この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実現した．この方式をもとに検証鍵に ID 情報を使用する ID ベース署名を準じ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実現した．この方式を元に検証鍵に 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について検討し，順序付き ID ベースアグリゲート署名を考案した．本稿では，2 章で GDH グループの定義と，ID ベース署名，ID ベースアグリゲート署名について説明し，3 章で既存方式，4 章で提案方式の説明を行う．5 章では提案方式の安全性証明を考察し，6 章でまとめとする．

2 関連研究

2.1 GDH グループ

岡本らにより定義された問題 [10] を基に，GDH グループが定義された [4, 5, 11]．最初に Computational-Diffie-Hellman（CDH）問題，および Decisional Diffie- Hellman（DDH）問題の 2 種類の Diffie-Hellman 問題について整理する．ここで，$\mathbb{G}'$を位数 p の巡回群とした時，これらの 2 種類の問題は以下の通りとなる．

CDH 問題: $a, b \in \mathbb{Z}_p^*$および$g \in \mathbb{G}'$があり，(g, g^a, g^b)の組が与えられた時，g^{ab} を求める問題.

DDH 問題: $a, b, c \in \mathbb{Z}_p^*$および$g \in \mathbb{G}'$があり，$(g, g^a, g^b, g^c)$の組が与えられた時，$c = ab$であるかを判定する問題.

ここで，CDH 問題は難しい問題である一方，DDH 問題は簡単な問題であるという条件が満たされる場合，この$\mathbb{G}'$をGDH グループと定義する．

2.3 BLS 署名

Boneh らによって提案された GDH グループの特性と双線形写像を用いた署名方式である BLS 署名のアルゴリズムを以下に示す.BLS 署名は鍵生成,署名,検証の 3 つのアルゴリズムからなる.ただし,(G_1, G_2)における GDH グループ上での定義を利用する.

1. 鍵生成：$x \epsilon Z_p$を選択し,$v = g_2^x$を計算する.x を署名に使用する秘密鍵とし,v をその公開鍵とする.
2. 署名：一方向性ハッシュ関数$H: \{0,1\} \rightarrow G_1$を定義する.$m$を署名対象となる平文として,$\sigma = H(m)^x$を計算する.そしてσを$m$に対するデジタル署名とする.
3. 検証:検証者に公開鍵v,平文mと署名が与えられているとして$e(\sigma, g_2) = e(H(m), v)$であるかを検証する.

2.4 Aggregate 署名

Aggregate 署名は BLS 署名と同様に Boneh らによって提案された.この署名方式では複数の署名者が各自の文書に対して生成した署名を 1 つに集約することが可能であり,検証者はその集約された署名の検証を通じて全ての個別署名を検証することができる.Aggregate 署名の説明を以下に示す.Uを署名に参加するユーザの集合とし,それぞれのユーザ$u_i \epsilon U (1 \leq i \leq n)$は 1 つの鍵ペア$(pk_{ui}, sk_{ui})$を持つとする.各ユーザは署名対象である$m_{ui}$を選び,それに対して署名$\sigma_{ui}$を生成する.そして,これらの署名は 1 つの Aggregate 署名へと結合される.Aggregate 署名は GDH 署名に基づいたもので,安全性は co-CDH 問題に依存している.そして,そのアルゴリズムは鍵生成,署名,集約,検証の 4 つからなる.また BLS 署名の説明と同様(G_1, G_2)における GDH グループ上での定義を利用する.以下にそのアルゴリズムを示す.

1. 鍵生成：署名者$u_i \in U$について$x_{ui} \in Z_p$を選択し,$v_{ui} = g_2^{x_{ui}}$を計算する.x_{ui}を署名に使用する秘密鍵とし,v_{ui}をその検証鍵とする.
2. 署名：一方向性ハッシュ関数$H: \{0,1\} \rightarrow G_1$を定義する.$m_{ui}$を各ユーザの署名対象となる平文として,$\sigma_{ui} = H(m_{ui})^{x_{ui}}$を計算する.そしてσ$_{ui}$を$m_{ui}$に対する

デジタル署名とする.)

3. 集約：個別署名σ_{ui}を全て集め$\prod_{i=1}^{n}\sigma_{ui}$を計算する.
4. 検証:検証者が$1 \le i \le$ nまでの公開鍵をv_{ui},平文m_{ui}と集約された署名σを得ているとき,$h_{ui}=H(m_{ui})$を計算する.そして$e(\sigma,g_2)=\prod_{i=1}^{n}e(h_{ui},v_{ui})$であるかを判定する.

2.5 ペアリングによる ID ベース署名

楕円曲線上において双線形性の特徴を持つペアリングと呼ばれる関数を利用することで ID ベース署名が実現可能であることが示された.

$\mathbb{G}$をペアリングの演算が可能な楕円曲線上の点の集合，eをペアリング関数とすると, ペアリングにおける双線形性により，以下の式が成立する.

・$P_1,P_2,Q \in \mathbb{G}$に対し,

$$e(P_1+P_2,Q)=e(P_1,Q)e(P_2,Q)$$

・$P,Q_1,Q_2 \in \mathbb{G}$に対し,

$$e(P,Q_1+Q_2)=e(P,Q_1)e(P,Q_2)$$

・$a,b \in \mathbb{Z}_p^*$および$P,Q \in \mathbb{G}$に対し,

$$e(aP,bQ)=e(bP,aQ)=e(abP,Q)=e(P,abQ)=e(P,Q)^{ab}$$

このペアリングの特徴を用いて構成される ID ベース署名は以下の通りである.

準備:$P \in \mathbb{G}$,を生成元とする.$s \in \mathbb{Z}_q^*$を選び,第 3 者の秘密鍵発行センターTA が$P_{pub}=sP$．s をマスターキーとする.

鍵生成：一方向性ハッシュ関数$H_1:\{0,1\}^* \to \mathbb{G}'$を定義する．署名者のユーザーID 情報を ID とした時, TA は$Q_{ID}=H_1(ID)$を計算し，$d_{ID}=sQ_{ID}$を発行し、署名者に渡す。d_{ID}は秘密鍵とする.

署名作成:一方向性ハッシュ関数$H_2:\{0,1\}^* \to \mathbb{G}'$を定義する．$m$を署名対象となる平文とした時，署名者は，$r \in \mathbb{Z}_q^*$を選び,$U=rP$を計算する．その後，$h=H_2(ID,m,U)$を計算し，$V=d_{ID}+rh$を生成する．$m$の署名に対する署名を$\sigma =< U,V >$とする.

署名検証：検証者にP,P_{pub},V,U,m,IDが与えられた時，検証者は,$h=H_2(ID,m,U)$を計算し, $e(P,V)=e(P_{pub},Q_{ID})e(U,h)$であるかを判定する.

2.3 ID ベースアグリゲート署名

2.2 節で説明した ID ベース署名を基にした ID ベースアグリゲート署名が提案されている[10]．本節ではその方式について説明する.

新たに$\mathbb{U}=\{u_1,\cdots,u_n\}$を署名作成が可能な署名者のグループとして定義し, さらに$\mathbb{L}=\{u_{i1},\cdots,u_{il}\}$を実際にアグリゲート署名作成に参加した署名者のグループと定義する. さらに$\mathbb{J}=\{i_1,\cdots,i_l\}$を，この署名参加者全員の符号とする．この時，ID ベースアグリゲート署名は以下のとおりに構成される.

準備:$P \in \mathbb{G}$,を生成元とする.$s \in \mathbb{Z}_q^*$を選び,第 3 者の秘密鍵発行センターTA が$P_{pub}=sP$．s をマスターキーとする.

鍵生成：一方向性ハッシュ関数$H_1:\{0,1\}^* \to \mathbb{G}'$を定義する．署名者$u_i$のユーザーID 情報を$ID_i$とした時，TA は$Q_{ID_i}=H_1(ID_i)$を計算し，$d_{ID_i}=sQ_{ID_i}$を発行する.

署名作成:一方向性ハッシュ関数$H_2:\{0,1\}^* \to \mathbb{G}'$を定義する．$m_i$を署名対象となる平文とした時，署名者は，$r_i \in \mathbb{Z}_q^*$を選び,$U_i=r_iP$を計算する．その後，$h_i=H_2(ID_i,m_i,U_i)$を計算し，$V_i=d_{ID_i}+r_ih_i$を生成する．$m_i$の署名に対する署名を$\sigma_i =< U_i,V_i >$とする.

アグリゲート:アグリゲート署名作成に参加する全ての署名参加者のV_iを集め,$V=\sum_{i=1}^{n}V_i$を計算する．署名を$\sigma =< U_1,U_2,\cdots,U_n,V >$とする.

署名検証：検証者にP,P_{pub},V,U_i,m_i,ID_iが与えられた時，検証者は,$h_i=H_2(ID_i,m_i,U_i)$を計算し,
$e(P,V)=\prod_{i=1}^{n}e(P_{pub},Q_{ID_i})e(U_i,h_i)$であるかを判定する.

3 既存方式~順序付きアグリゲート署名~

$\mathbb{G}$をペアリングの演算が可能な楕円曲線上の点の集合，eをペアリング関数とする．また，u_iをアグリゲート署名作成に参加した署名者とし, iはその署名者の識別符号とする.m_iを署名対象となる平文，$H:\{0,1\}^* \to \mathbb{G}$を提案方式で用いる一方向性ハッシュ関数と定義する.その他の記号については，文中で説明する.

・公開鍵基盤（PKI:Public Key Infrastructure）は 整備されており，全ての署名者の鍵ペアが正当に発行されている.

・署名者に発行されている鍵ペア以外に, 新たな鍵ペアの発行は行わない.

・正当なアグリゲート署名の参加者は正しく署名作成を行うものとし, 他の参加者との結託は行わないものとする.

・アグリゲート署名作成中において，署名者間の通信は安全に行われ, 作成中の中間情報を第三者が入手することはできないものとする.

3.1 鍵生成$g \in G$を生成元とする．署名者u_iについて$x_i \in Z_p^*$を選び（全ての署名者の署名鍵は各々異なるものとする），$v_i = x_i g$を計算する．x_iを署名者u_iの署名鍵，v_iを署名者u_iの検証鍵とする．

3.2 署名作成は以下の手順で,アグリゲート署名が作成される．

1. 第 1 署名者u_1は,平文m_1から$h_1 = H(m_1)$を求め 2.2 節で説明した BLS 署名と同様な署名作成処理を行うことで$\sigma_1 = x_1 h_1$を計算する.σ_1とm_1(またはh_1)を第 2 署名者u_2に送信する.

2. 第 2 署名者u_2は,第 1 署名者u_1から受信したm_1を用いて$h_1 = m_1$を求める．さらに署名者u_2は,自分が本来署名したい平文m_2から$h_2 = H(m_2)$を求める．これと署名者u_1から受信したσ_1を用いて,$\sigma_2 = \sigma_1 + x_2 h_1 + x_2 h_2 = x_1 h_1 + x_2 h_1 + x_2 h_2$を計算する．また$L_2 = \{(u_1, u_2)\}$を作成する．この$\sigma_2, L_2$および$m_2$を第 3 署名者$u_3$に送信する．

3. 第 n 署名者u_nは,第 n-1 署名者u_{n-1}から受信したm_{n-1}を用いて$h_{n-1} = H(m_{n-1})$を求める．さらに署名者u_nは,自分が本来署名したい平文m_nから$h_n = H(m_n)$を求める．これと署名者u_{n-1}から受信したσ_{n-1}を用いて,$\sigma_n = \sigma_{n-1} + x_n h_{n-1} + x_n h_n = x_1 h_1 + \sum_{j=2}^{n} x_j h_{j-1} x_j h_{j-1}$を計算する．また,受信した$L_{n-1}$を用いて,$L_n = L_{n-1} + \{(u_{n-1}, u_n)\}$を作成する．この$\sigma_n, L_n$および$m_n$を．署名者$u_1$~$u_n$の，署名対象$m_1$~$m_n$に対するアグ リゲート署名として公開する．

4.3 署名検証 以下の手順で，アグリゲート署名の検証を行う．

1. 検証者は,L_nに示されている全ての署名者の検証鍵v_1~v_nおよび署名対象となる全ての平文m_1~m_nを集める．

2. 検証者は，集めた平文から$h_i = H(m_i)$を求める．

3. 検証者は,

$e(v_1, h_1)\left(\prod_{j=2}^{n} e(v_j, h_{j-1} + h_j\right) = e(v_1, h_1) e(v_2, h_1 + h_2) \cdots e(v_n, h_{n-1} + h_n)$を計算し,$e(g, \sigma_n)$と値が一致することを確認する．

4 提案方式~順序付き ID ベースアグリゲート署名~

本章では，2.2 節，および 2.3 節で説明した署名方式を拡張し，順序付き ID ベースアグリゲート署名の提案方式について説明する．

4.1 前提条件

$\mathbb{G}$をペアリングの演算が可能な楕円曲線上の点の集合，eをペアリング関数とする．また，u_iをアグリゲート署名作成に参加した署名者とし,iはその署名者の識別符号とする．m_iを署名対象となる平文，$H_1, H_2: \{0,1\}^* \to \mathbb{G}$を提案方式で用いる一方向性ハッシュ関数と定義し，$h_i = H_2(ID_i, m_i, U_i)$とする．その他の記号については，文中で説明する．

4.2 準備

$P \in \mathbb{G}$,を生成元とする.$s \in \mathbb{Z}_q^*$を選び,第 3 者の秘密鍵発行センターTA が$P_{pub} = sP$．sをマスターキーとする．

4.3 鍵生成

署名者u_iのユーザーID 情報をID_iとした時，TA は$Q_{ID_i} = H_1(ID_i)$を計算し，$d_{ID_i} = sQ_{ID_i}$を発行する．

4.4 署名作成

1．第 1 署名者u_1は，$r_1 \in \mathbb{Z}_q^*$を選び,$U_1 = r_1 P$を計算する．平文m_1から$h_1 = H_2(ID_1, m_1, U_1)$を計算し，$V_1 = d_{ID_1} + r_1 h_1$を生成する．$m_1$の署名に対する署名を$\sigma_1 =< U_1, V_1 >$とする．この$U_1, V_1, m_1, ID_1$を第 2 署名者$u_2$に送信する．

2. 第 2 署名者u_2は第 1 署名者u_1から受信した情報を用いて，$h_1 = H_2(ID_1, m_1, U_1)$を求める．さらに署名者$u_2$は$r_2 \in \mathbb{Z}_q^*$を選び,$U_2 = r_2 P$，自分が本来署名したい平文$m_2$から$h_2 = H_2(ID_2, m_2, U_2)$を求める．これと署名者$u_1$から受信した情報を用いて，

$$V_2 = V_1 + r_2 h_1 + r_2 h_2$$
$$= d_{ID_1} + d_{ID_2} + r_1 h_1 + r_2 h_1 + r_2 h_2$$

を計算する．このV_2, U_2, ID_2, m_2を第 3 署名者u_3に送信する．

3．第i署名者u_iは，第$i-1$署名者u_{i-1}から受信した情報を用いて，$h_{i-1} = H_2(ID_{i-1}, m_{i-1}, U_{i-1})$を求める．さらに署名者$u_i$は，$r_i \in \mathbb{Z}_q^*$を選び,$U_i = r_i P$を求め，自分が本来署名したい平文$m_i$から，$h_i = H_2(ID_i, m_i, U_i)$を求める．これと署名者$u_{i-1}$から受信した情報を用いて，

$$V_i = V_{i-1} + r_i h_{i-1} + r_i h_i$$
$$= \sum_{j=1}^{i} d_{ID_j} + r_1 h_1 + \sum_{j=2}^{i} r_j h_{j-1} + r_j h_j$$

を計算する．この，このV_i, U_i, ID_i, m_iを第$i+1$署名者u_{i+1}に送信する．この手順を最後から 1 人手前の署名者まで再帰的に行う．

4．最後の署名者u_nは，第$n-1$署名者u_{n-1}から受信した情報を用いて，$h_{n-1} = H_2(ID_{n-1}, m_{n-1}, U_{n-1})$を求める．さらに署

名者u_nは，$r_n \in \mathbb{Z}_q^*$を選び,$U_n = r_n P$を求め，自分が本来署名したい平文m_nから，$h_n = H_2(ID_n, m_n, U_n)$を求める．これと署名者$u_{n-1}$から受信した情報を用いて，

$$V_n = V_{n-1} + r_n h_{n-1} + r_n h_i$$
$$= \sum_{i=1}^{n} d_{ID_i} + r_1 h_1 + \sum_{i=2}^{n} r_i h_{i-1} + r_i h_i$$

を計算する．最終的にσ =< $U_1, U_2, \cdots, U_n, V$ >をアグリゲート署名として公開する．

4.5 署名検証

以下の手順で，アグリゲート署名の検証を行う．

1．検証者は，全ての署名者の検証鍵$(U_1, U_2, \cdots U_n), V$と公開情報$(Q_{ID_1}, Q_{ID_2}, \cdots, Q_{ID_n}), (m_1, m_2, \cdots, m_n), (ID_1, ID_2, \cdots, ID_n)$を集める．

2. 検証者は, 集めた情報から, $h_i = H_2(ID_i, m_i, U_i)$を求める．

3．検証者は，

$$e(P,V) = e\left(P, \left(\sum_{i=1}^{n} d_{ID_i}\right) + r_1 h_1 + \sum_{i=2}^{n} r_i h_{i-1} + r_i h_i\right)$$

$$= \prod_{i=1}^{n} e\left(sP, Q_{ID_i}\right) e(r_1 P, h_1) \prod_{i=2}^{n} (r_i P, h_i + h_{i-1})$$

$$= \prod_{i=1}^{n} e\left(P_{pub}, Q_{ID_i}\right)) \cdot e(U_1, h_1) \cdot (\prod_{i=2}^{n} e\,(U_i, h_{i-1} + h_i)$$

の右辺と左辺の値が一致することを確認する．

5 安全性証明

アグリゲート署名正当性 : 提案方式では、検証鍵と対応するメッセージ(のハッシュ値)をペアリング関数に順次入力していくことで, V_nを検証する手順となっている。このときのアグリゲート書名の安全性について考察する。

攻撃者 A は偽装を行う署名者の検証鍵U_1'を入力値としたときに, 想定している順序に応じて残りの n-1 個の署名鍵と検証鍵のペアの出力を行うものとする. A の攻撃が成功するとは，上記の入力値，および出力した鍵ペアに対し，想定した順序に対応するV_n'出力できること，すなわちV_n'の偽造に成功することを意味する．この時，以下の定理が成立する．

定理　ランダムオラクルモデルにおいて, 提案方式におけるV_n'の偽造可能性と ID ベース署名の偽造可能性は等価である．

証明　A をV_n'を偽造しようとする攻撃者とする．B を ID ベース署名を偽造する攻撃者とした時, A の攻撃が成功するなら，B の攻撃が成功することを示す（B の攻撃が成功するなら，A の攻撃が成功することは自明であるため省略する）．

B は 1 つの検証鍵U_1'を持っており，ランダムオラクル，および署名オラクルへの応答を行う．このU_1'に対して，$r_1' \in Z_p^*$を用いて$r_1' P$と表すことができるが，B にはx_1'の値は道であるものとする．

B は A を Honest Player として実行する. まず B は A にU_1'を与え，A はそれ以外の署名鍵と検証鍵のペア$\left(d_{ID_2}', r_2', U_2'\right), \cdots, (d_{ID_n}' r_n', U_n')$を出力する．($U_i' = r_i' P$，また，アグリゲート署名作成に参加した人数は n 人の想定). また，A はランダムオラクルと署名オラクルへの応答によりV_n'とその署名対象となる平文$m_1', \cdots, m_n'$を求め，B に返答する．B はV_n'を用いて，$V_n' - \sum_{i=2}^{n} d_{ID_i}' - \sum_{i=2}^{n} r_i' h_{i-1}' + r_i' h_i' = d_{ID_1}' + r_1' h_1'$の計算により，$U_1'$に対応する ID ベース署名を作成することができ，B の攻撃が成功する．

以上により ID ベース署名の偽造が困難であれば，順序付き ID ベースアグリゲート署名の偽造も困難である．

6 まとめ

署名者の前後において, 後者が自分とその前者のメッセージに署名し, この署名を順次合成していくことで構成された順序付きアグリゲート署名を説明し，その方式に ID ベース署名を拡張した順序付き ID ベース署名を提案した．検証鍵に ID 情報を用いた ID ベース署名を適応することで，検証の際に署名者の ID を用いることで署名者の特定が用意になり，CGM サービスにより適した形であることを示した．

さらにこの方式の安全性の議論に基づいて, 方式の安全性の考察を行った．

今後は, この方式を 1 対多の木構造表記型への拡張をすることで改良を行い, 既存の方式とのパフォーマンスの比較を行なっていく予定である．

参考文献

[1] YouTube, http://www.youtube.com/

[2] Creative Commons,
http://creativecommons.org/

[3] パソコン決裁,
http://www.shachihata.co.jp/interweb/index.php

[4] D.Boneh, B.Lynn, and H.Shacham, Short Signatures from theWeil Pairing," Advances in Cryptology - ASIACRYPT 2001, LNCS 2248, pp.514-532,Springer-Verlag, 2001.

[5] A.Boldyreva, Threshold Signatures, Multisignatures and

Blind Signatures Based on the Gap-Diffie-Hellman-Group Signature Scheme," Public Key Cryptography - PKC 2003, LNCS 2567,pp.31-46, Springer-Verlag, 2003.

[6] C.Y.Lin, T.C.Wu, and F.Zhang, “A Structured Multisignature Scheme from the Gap Diffie-Hellman Group," Cryptology ePrint Archive, Report 2003/090, http://eprint.iacr.org/2003/090, 2003.

[7] 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.

[8] 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.

[9] D.Boneh, C.Gentry, B.Lynn, and H.Shacham, “Aggregate and Veriably Encrypted Signatures from Bilinear Maps," Advances in Cryptology - EUROCRYPT 2003, LNCS 2656, pp.416-432, Springer-Verlag,2003.

[10]Jing Xu,Zhenfeng and Dengguo Feng, “ID-Based Aggregate Signatures from Bilinear Pairing”,CANS 2005,LNCS3810,pp.110-119, Springer-Verlag Berlin Heidelberg,2005.