マルチメディア通信と分散処理 111－27
コンピュータセキュリティ　20－27
（２００３．２．２８）

リング署名プロトコルにおける署名者開示

菊池 浩明* 多田 美奈子† 中西 祥八郎 ‡

{kikn,lukia}@ep.u-tokai.ac.jp

概要: リング署名とは，グループメンバなら誰でも署名が可能で，かつ検証者に対して，匿名性を保証できるグループ署名方式の一つである．しかしこの匿名性が完全に保証されているため，署名後何らかの問題が発生し，どのメンバによる署名であるかを特定する必要がある場合にも，署名の開示は不可能である．そこで本稿では，リング署名に管理者の存在を仮定して必要に応じて管理者または署名者本人が署名者を開示できるリング署名プロトコルを提案する．提案プロトコルの安全性は，離散対数問題の困難性と安全なハッシュ関数の存在に基づいている．

Proof of Signer and Privacy Revocation in Ring Signature Protocol

Hiroaki Kikuchi * Minako Tada † Shohachiro Nakanishi ‡

Abstract: A ring signature is one of the group signature scheme which allows member of a group to sign message such that the resulting signature does not reveal their identity for all users. But, even if a problem occurs and the signer is necessary to be detected, anyone can't determine who signed it. In this paper, we propose an extended protocol of ring signature in which both the group administrator and a signer can indicate who signed on the message. The security of the proposed protocol is based on an assumption of the discrete logarithm problem and a secure hash function.

1 はじめに

グループ署名とは，グループに属している全てのグループメンバがグループを代表しての署名が可能であり，なおかつどのメンバが署名したのかを秘密にできる署名方式である．これは 1991 年に Chaum によって初めて提案された [CvH 91]．その後も Camenisch らによって効率的なグループ署名方式 [CS 97] が提案されている．これらの方式は，グループ管理者の存在を仮定しており，グループ管理者のみ署名者の開示が行うことができる．

一方，Rivest らによってリング署名 (Ring Signature) と呼ばれる署名方式が提案されている [RST 01].

* 東海大学電子情報学部情報メディア学科,
Department of Information Media Technology, School of Information Technology and Electronics, Tokai University
† 東海大学大学院工学研究科,
Graduate School of Engineering, Tokai University
‡ 東海大学電子情報学部情報科学科,
Department of Human and Information Science, School of Information Technology and Electronics, Tokai University

共通鍵暗号と落し戸つき一方向性置換関数を用いたプロトコル [RST 01] を基に，大久保らによって離散対数問題に基づくプロトコル [OAST 02] が提案されている．これらの方式は，前述のグループ署名同様，匿名性を保証しながらグループの代表での署名が可能であるが，管理者を持たず，署名から署名者の同定を行うのは，たとえ署名者本人でも不可能である．このように，リング署名から真の署名者の証拠を示すことを，署名者の**開示**とよぶ．

署名者の開示の要求には二種類が考えられる．一つは署名者自身が署名者であることを証明したい場合である．これは，オークションなどで落札前は自分が入札者であることを秘密にしておき，落札後に自分が落札者であることを示したい場合などに有効である．もう一つは権限をもった管理者が署名の匿名性を破棄して，真の署名者を特定したい場合である．例えば，署名が承認の性質を持ち，その内容に何らかの問題がある時に，それを許可した (署名した) メンバの特定を行う状況が考えられる．しかし従来のリング署名においてはこの二つの要求を満たすことはできなかった．

本稿では，この問題に対し，[OAST 02] で提案されたプロトコルを基に，管理者を導入し上記の二種類の要求に対して開示可能なリング署名プロトコルを提案する．

本稿の構成は次の通り．2 章では，[OAST 02] を説明する．3 章では，自己開示可能リング署名プロトコルと，開示可能なリング署名プロトコルを提案し，4 章で比較，5 章にてまとめる．

2 基本プロトコル

本章では，提案プロトコルの基本となる大久保らによるリング署名 [OAST 02] について説明する．本プロトコルは，離散対数問題に基づいている．

2.1 準備

エンティティを次のように示す．

$\mathcal{G}$ ：グループメンバの集合
$\mathcal{U}_j$ ：$\mathcal{G}$ に属するメンバ $(j = 1, \ldots, n)$
$\mathcal{U}_i$ ：署名者

グループメンバは，$q \mid p-1$ を満たす大きな素数 p, q と，p の位数 q の部分群の生成元となる g を生成し，p, q, g を公開する[1]．また，これを元に，グループメンバ $\mathcal{U}_j$ は $x_j \in Z_q$ を秘密鍵，$y_j = g_j^{x_j} \bmod p$ を公開鍵として生成し，y_j を公開する．

署名者 $\mathcal{U}_i$ は，n 個の公開鍵 $y_j (j = 1, \ldots, n)$ の内，少なくとも一つのある y_i に対応する秘密鍵 x_i を知っていることを，i を秘密にしたまま証明し，これを署名とする．ここで，H を一方向性セキュアハッシュ関数とする．

2.2 署名生成

文書 m に対する署名は，以下の手順で生成する．

Step 1 i について，

$$\begin{aligned} T_i &= g^{\alpha} \bmod p, \\ c_{i+1} &= H(m \parallel T_i) \end{aligned}$$

を求める．ただし，$\alpha \in_U Z_q$ とする．

Step 2 $j = i+1, \ldots, n, 1, \ldots, i-1$ について，

$$\begin{aligned} T_j &= g^{s_j} y_j^{c_j} \bmod p, \\ c_{j+1} &= H(m \parallel T_j) \end{aligned}$$

を順次計算する．ここで，$s_j \in_U Z_q$ とする．

Step 3 (秘密鍵 x_i を知っている)i について，

$$s_i = \alpha - x_i c_i \bmod q \qquad (1)$$

を計算する．$(c_1, s_1, s_2, \ldots, s_n)$ を m に対する署名として出力する．

2.3 署名検証

$j = 1, \ldots, n$ まで以下を繰り返す．

$$\begin{aligned} T_j &= g^{s_j} y_j^{c_j} \bmod p \\ c_{j+1} &= H(m \parallel T_j) \end{aligned}$$

$c_1 = c_{n+1}$ ならば受理し，そうでなければ棄却する．

2.4 考察

基本プロトコルは，リングを閉じてしまった後は，たとえ秘密情報を公開したとしても，その署名がどの

[1] (p, q, g) とハッシュ関数 H に関しては，各ユーザ毎に各々で設定することが可能であるが，ここでは簡単の為に共通とした．

メンバによるものであるかの証拠にはならない．以下に署名の証明が出来ない例を示す．

$\mathcal{U}_j(j=1,2,3)$ とする．真の署名者は $\mathcal{U}_2$ とし，2.1 節のプロトコルに従い，署名を生成する．$\mathcal{U}_2$ は，Step 1 で用いた α を署名の証拠として提示するかもしれない．しかし，真の署名者ではない別のメンバ $\mathcal{U}_3$ が秘密鍵 x_3 を用いると，$s_3 = \alpha' - x_3 c_3 \bmod q$ を満たすような α' は必ず存在して一意に決まる．これを用いて，次のような検証が成り立つ．

$$\begin{aligned} T_3 &= g^{s_3} y_3^{c_3} \bmod p \\ &= g^{\alpha' - x_3 c_3} y_3^{c_3} \bmod p \\ &= g^{\alpha'} \bmod p \end{aligned}$$

T_3 は真の署名者 $\mathcal{U}_2$ が行った証明と同じであり，調停者には α と α' のどちらが真の情報か (情報量的に) 区別がつかない．つまり，秘密情報を公開しても，署名の証拠になりえない．

3 提案プロトコル

3.1 概要

前章の基本プロトコルでは，チャレンジをハッシュでつなぎ，最後に自分のみが知る秘密鍵によってリングを閉じて，グループ署名を実現していた．そこで，我々は署名生成時に用いる T_j の生成順が，署名者を特定していることに着目し，この順序を示す情報を乱数である s_j へ埋め込むことで，署名者自身による開示を実現する．

本章では，まず初めに自己開示可能なプロトコルを示す．その後，管理者によって追跡可能な提案方式を示す．

3.2 自己開示可能リング署名

3.2.1 署名生成

H_2 を一方向性の性質を持つ関数と定義する．他のエンティティ，パラメータ等は，基本プロトコル同様．署名者 $\mathcal{U}_i$ は Step 2 において，$j = i+1, \ldots, n, 1, \ldots, i-1$ について，乱数 r_j を選び，それを用いて，

$$s_j = H_2(r_j, c_j) \tag{2}$$

を定め，T_j, c_{j+1} を同様に計算する．また，$r_1, \ldots, r_n$ を安全に管理しておく．その他，署名検証までは 2.1 節の基本プロトコル同様．

3.2.2 署名者の証明

署名者 $\mathcal{U}_i$ は問題の署名について，$r_1, \ldots, r_{i-1}, r_{r+1}, \ldots, r_n$ を示す．検証者は，$j = 1, \ldots, i-1, i+1, \ldots, n$ について，

$$s'_j = H_2(r_j, c_j)$$

を計算し，$s'_j = s_j$ ならば，署名者の証明を受理し，そうでなければ，棄却する．

3.3 考察

真の署名者ならば，s_i 以外は式 (2) を満たす $r_1, \ldots, r_{i-1}, r_{i+1}, \ldots, r_n$ を必ず示すことができる，真の署名者 $\mathcal{U}_i$ が，ほかの署名者 $\mathcal{U}_j$ に署名者 (の証拠) をなすりつけようと思っても，s_i と c_i から $s_i = H_2(r_i, c_i)$ を満たす r_i を作ることが (H_2 の一方向性より) 不可能なので，失敗する．逆に，偽の署名者が署名者になりすますことも，同様に不可能である．また第三者が署名から，s_i が式 (1) と式 (2) のどちらで生成されているかは，一方向性関数が十分な冗長度を持っているという仮定の下で，区別できない．また，署名自体の安全性は，ベースとなるリング (Schnorr) 署名のものと同等である．

一方向性関数の例には，一方向性ハッシュ関数や離散対数問題，またはタイムスタンプなどがあげられる．ハッシュ関数を用いた場合は，検証する為に証拠 $r_1, \ldots, r_n$ を公開しなくてはならない．しかしたとえ，この情報を用いても s_i の生成順序は変えられないので開示は安全である．一方，H として次の様な離散対数

$$H'_2(r_j, c_j) = g^{r_j c_j} \bmod p'$$

を用いると，知識の証明 $PK\{\alpha | H_2 = (g^{c_j})^{\alpha}\}$ によって証拠を隠蔽したまま開示することができる．

不正な署名者は式 (2) を守らないかもしれない．しかし，署名の証明は自らのために行うことになるので，開示が必要となる前に情報を埋め込んであることの証明に対する要求は弱いと考える．

署名を開示することを署名者以外から要求される場合，例えば署名の乱用など悪質なケースが露呈した際には，特定の条件の下で署名の開示が行えることが必要とされる．そこで以降の方式では，署名開示の権限

を持たせた管理者の存在を仮定した方式について述べる．

3.4 追跡可能リング署名

3.4.1 準備

新しいエンティティとして失効管理者$\mathcal{RM}_1,\ldots,\mathcal{RM}_l$を設ける．失効管理者は$l$人中，$k$人で協力して，安全な方法で$k-1$次多項式$f(x)$を作り，公開鍵$h=g^{f(0)}$を公開し，各$\mathcal{RM}_i$へシェア$f(i)$を秘密に分散する．他は，基本プロトコル同様．

3.4.2 署名生成

署名者$\mathcal{U}_i$はiについて，

$$\begin{aligned} T_i &= g^{\alpha} \bmod p, \\ c_{i+1} &= H(m \parallel T_i), \end{aligned}$$

また，同じαを用いて，

$$U = h^{\alpha} \bmod p$$

を求める．ただし，$\alpha \in_U Z_q$とする．その他，基本プロトコル同様．署名は，$(c_1, s_1, \ldots, s_n, U)$とする．

3.4.3 知識証明

正しく情報を埋め込んだことの証拠として，リング署名を生成した後，ゼロ知識証明により

$$\begin{aligned} \log_g T_1 &= \log_h U \\ \vee \log_g T_2 &= \log_h U \\ &\vdots \\ \vee \log_g T_n &= \log_h U \end{aligned}$$

である事を示し，これを知識の証明SKとして，署名に添付する．具体的な手順を次に示す．

Step 1 $j=1,\ldots,i-1,i+1,\ldots,n$について，乱数$z_j \in_U Z_q^*$と，$e_j \in \{0,1\}^u$($u$はセキュリティパラメータ)を生成し，

$$\begin{aligned} a_j &= g^{z_j} T_j{}^{e_j}, \\ b_j &= h^{z_j} U^{e_j} \end{aligned}$$

を求める．また，真の署名のiについては，乱数r_iを選び，

$$\begin{aligned} a_i &= g^{r_i}, \\ b_i &= h^{r_i} \end{aligned}$$

とする．

Step 2 一方向性セキュアハッシュ関数$F:\{0,1\}^* \to \{0,1\}^u$を用い，

$$\begin{aligned} e &= F(m \parallel g \parallel h \parallel a_1 \parallel b_1 \parallel \cdots \parallel a_n \parallel b_n), \\ e_i &= \left(\bigoplus_{j \in \mathcal{G} \setminus \{i\}} e_j \right) \oplus e \end{aligned}$$

を求める．

Step 3 iについて，$z_i = r_i - \alpha e_i \bmod q$を計算する．$SK = (e, e_1, a_1, b_1, z_1, \ldots, e_n, a_n, b_n, z_n)$とする．

結果として，mについての署名は，$(c_i, s_1, \ldots, s_n, h, SK)$となる．

3.4.4 署名検証

署名本体の検証は，基本プロトコル同様．SKの検証を次に示す．

$$\begin{aligned} e &= F(m \parallel g \parallel h \parallel a_1 \parallel b_1 \parallel \cdots \parallel a_n \parallel b_n) \\ &\stackrel{?}{=} e_1 \oplus \cdots, e_n \end{aligned}$$

ここで，$j=1,\ldots,n$について，

$$\begin{aligned} a_j &\stackrel{?}{=} g^{z_j} T_j^{e_j}, \\ b_j &\stackrel{?}{=} h^{z_j} U^{e_j} \end{aligned}$$

を行う．全ての検証が成功した場合のみ，署名を受理し，失敗した場合棄却する．

3.4.5 署名開示

管理者$\mathcal{RM}_k$は自分の持つ分散情報$f(k)$を用いて，$j=1,\ldots,n$について$T_j^{f(k)}$を求めてコミットした後に共有する．l人中の任意のk人が協力してLagrangeの補間法を用いて$T_j^{f(0)}$を求め，

$$U = T_j^{f(0)} \bmod p$$

表 1: 効率比較

	基本	自己開示	追跡可能	
			署名のみ	署名+SK
署名長	$\|q\|(n+1)$	$\|q\|(n+1)$	$\|q\|(n+1)+\|p\|n$	$2\|p\|n+\|q\|(n+1)$ $+\|u\|(n+1)$
検証コスト	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	
$\mathcal{RM}$ の管理量	N/A	N/A	$\mathcal{O}(1)$	
開示コスト	N/A	N/A	$\mathcal{O}(n)$	

が成り立つ j をもつ $\mathcal{U}_j$ をさがす．以下に具体的な手順を示す．

$\mathcal{RM}_1, \cdots, \mathcal{RM}_l$ は，$j = 1, \ldots, n$ について，

$$T_j^{f(0)} = \prod_{1 \le k \le l} T_j^{f(k)\lambda(k)}$$

を求める．ここで，

$$\lambda(k) = \prod_{1 \le k \le l, k \ne d} \frac{k}{k-d} \bmod q$$

とする．このうち，

$$U \stackrel{?}{=} T_j^{f(0)} \bmod p$$

が成り立つ j をもつ $\mathcal{U}_j$ が，署名者である．

3.5 考察

署名についての安全性は，リング (Schnorr) 署名と，知識証明については [CDS 94] と同等である．

この方式では，管理者が秘密を分散してもつことにより，署名の開示を行うときには管理者が協力する．また管理者を複数置くことにより，閾値までの不正な管理者による開示も防ぐことができる．

4 比較

ここでは，2 章の基本プロトコルと，3 章で挙げた提案方式について比較する．効率の比較は表 1 に示す通りである．

また，追跡可能リング署名は自己開示可能リング署名と同時に用いることが可能であるため，自己開示と追跡の二つの機能を持たすことができる．

5 おわりに

本稿では，大久保らによる離散対数問題に基づくリング署名を用いた，開示可能な方式を提案した．

参考文献

[BSS 02] E. Bresson, J. Stern and M. Szydlo, “Threshold Ring Signatures and Applications to Ad-hoc Groups,” Advances in Cryptology - CRYPTO 2002, pp.465-480, 2002.

[CvH 91] D. L. Chaum and E. Van Heyst, “Group Signatures,” Advances in Cryptology - EUROCRYPTO ’91, pp.257-264, 1991.

[CDS 94] R. Cramer, I. Damgård, and B. Schoenmakers, “Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols,” Advances in Cryptology - CRYPTO ’94,LNCS,vol.839, pp.174-187, 1994.

[CS 97] J. L. Camenisch and M. A. Stadler, “Efficient Group Signature Schemes for Large Groups, ” Advances in Cryptology Proceedings of CRYPTO ’97, pp.410-424, 1997.

[KT 02] 桑門，田中，“署名者の匿名性を有するディジタル署名方式”，情報処理学会コンピュータセキュリティ研究発表会-CSEC18-35，pp.239-244, 2002.

[OAST 02] 大久保，阿部，鈴木，辻井，“証明長が短い 1-out-of-n 証明”，暗号と情報セキュリティシンポジウム-SCIS2002，pp.189-193，2002.

[RST 01] R. Rivest, A. Shamir, and Y. Tauman, “How to leak a secret,”Advances in Cryptology - ASIACRYPT 2001, LNCS,vol.2248, pp.552-565, 2001.

[SKN 00] 崔，菊池，中西，“ブラインドグループ署名”，電子情報通信学会情報セキュリティ研究会-ISEC，2000.