SCIS 2005 The 2005 Symposium on Cryptography and Information Security
Maiko Kobe, Japan, Jan.25-28, 2005
The Institute of Electronics, Information and Communication Engineers

汎用的結合可能なグループ署名について
On Universally Composable Group Signatures

牧田 俊明 * 真鍋 義文 † 岡本 龍明‡
Toshiaki Makita Yoshifumi Manabe Tatsuaki Okamoto

あらまし グループ署名とは，グループのメンバのみが匿名で署名でき，不正があったときなどにはグループの管理者が署名者を特定できる機能を持つ署名方式である．本稿ではこのようなグループ署名をCanetti により提案された汎用的結合可能性 (Universal Composability: UC) の概念を用いて実現する方法を提案する．まず，グループ署名の理想的機能 (functionality)$\mathcal{F}_{\mathrm{GSIG}}$ を定義し，汎用的結合可能な知識の署名等を用いて，$\mathcal{F}_{\mathrm{GSIG}}$ を安全に実現する方法を提案する．

キーワード 汎用的結合可能性，グループ署名

1 はじめに

グループ署名は，Chaum と van Heyst[4] によって提案された．この手法は，グループのメンバは誰でも，グループ内の誰であるかは秘密のままにグループの代表として署名をすることができるという匿名性を持つ．しかし，不正があった場合など，必要なときには，グループの管理者が署名者を特定することができる．グループ署名は，投票，入札や内部告発など，様々なアプリケーションに応用が可能である．

グループ署名の安全性の定義については，今までに様々な議論がなされてきた．その多くはいくつかの独立する要件で安全性を定義するものであったが，Bellare, Micciancio, Warinschi[6] によりこれらの多様な要件が 2 つにまとめられた．この 2 つの要件は，既存の主な安全性の要求条件を満たすものであったが，その半面要求が過度に厳しすぎるとも考えられ，Camenisch, Groth[7] により弱められた条件も提案されている．

本稿では，Canetti[1] によって提案された汎用的結合可能性 (Universal Composability) の枠組みを用いて安全性を定義する．この枠組みで安全であると証明されたプロトコルは，どのような他のプロトコルと同時に用いても，その安全性が維持される．この枠組みの中で定義される安全性の概念を，汎用的結合可能（UC-安全）であるという．

本研究の目的は，まずグループ署名の理想的な機能 $\mathcal{F}_{\mathrm{GSIG}}$ を定義することである．そして，知識の署名の理想機能 $\mathcal{F}_{\mathrm{SPK}}^{R}$ とゼロ知識証明の理想機能 $\mathcal{F}_{\mathrm{ZK}}^{R'}$ を用いて，汎用的結合可能なグループ署名を構成する．そのため，知識の署名の理想機能を定義し，それとゼロ知識証明の理想機能，公開鍵暗号，電子署名を用いて汎用的結合可能なグループ署名を構成する方法を示す．

2 グループ署名

2.1 署名方式

グループ署名として，グループメンバ，非グループメンバ，グループ管理者の 3 種類のパーティからなり，また，以下の，KEYGEN, JOIN, SIGN, VERIFY, OPEN, REVOKE の 6 つのアルゴリズムからなる方式を用いる．

KEYGEN: グループ公開鍵とグループ管理者の秘密鍵を出力する確率的アルゴリズム．

JOIN: 非グループメンバをグループに加入させるプロトコル．

SIGN: グループメンバの秘密鍵と文書を入力として，署名を出力する確率的アルゴリズム．

VERIFY: グループ公開鍵，文書，署名を入力として 1 または 0 を返す決定的アルゴリズム．

OPEN: グループ管理者の秘密鍵，文書，署名を入力として，署名の作成者か，OPEN の失敗を表す $\perp$ を返す決定的アルゴリズム．

* 京都大学大学院情報学研究科社会情報学専攻, 〒 606-8501 京都市左京区吉田本町, Department of Social Informatics, Graduate School of Informatics, Kyoto University, Yoshidahonmachi, Sakyo-ku, Kyoto-shi, Japan, makita@lab7.kuis.kyoto-u.ac.jp
† NTT サイバースペース研究所, 〒 239-0847 横須賀市光の丘 1-1, NTT Cyber Space Laboratories, 1-1 Hikarinooka, Yokosuka-shi, Japan, manabe.yoshifumi@lab.ntt.co.jp
‡ NTT 情報流通プラットフォーム研究所, 〒 239-0847 横須賀市光の丘 1-1 NTT Information Sharing Platform Laboratories, 1-1 Hikarinooka, Yokosuka-shi, Japan, okamoto@sucaba.isl.ntt.co.jp

REVOKE: グループメンバをグループから脱退させるプロトコル.

2.2 グループ署名の安全性

グループ署名について，それが満たすべき様々な安全性の要求条件が提案されてきた．その主なものが [5] で挙げられており，以下の通りである．

Correctness: SIGN により作成された署名は，VERIFY によって受理されなければならない.

Unforgeability: グループメンバのみがグループの代表として SIGN することができる.

Anonymity: 正当な署名が与えられたとき，管理者以外がその作成者を特定することは，計算量的に困難である.

Unlinkability: 2 つの異なる正当な署名が与えられたとき，これらが同一のメンバによって作成されたかどうかを決定することは計算量的に困難である.

Exculpability: メンバも管理者も，OPEN の結果が他のメンバとなるような署名を作成できない.

Traceability: 管理者はいつでも正当な署名を OPEN して，実際の作成者を特定することができる.

Coalition-resistance: グループメンバが結託しても（グループ全員でも），OPEN の結果が結託したグループメンバの誰かにならないような正当な署名を作成することができない.

Revocability: REVOKE されたメンバに作成された署名は VERIFY により拒否されなければならない.

[6] により，グループ署名の安全性として full-anonymity と full-traceability の 2 つの要求条件が提案されている．これらの定義は，KEYGEN が信用できるパーティによって行われることを想定している.

Full-traceability: OPEN の結果があるグループメンバとなるような署名を，そのメンバの秘密鍵なしに作成することはできない．この性質は，管理者の鍵と，任意のグループメンバの鍵が敵に知られていても保たれなければならない.

Full-anonymity: 敵が任意に選んだ 2 人のグループメンバと文書に対し，その 2 人からランダムに選ばれた 1 人が作成した署名が与えられたとき，敵がその署名からどちらのメンバが作成者であるか識別することができない．この性質は，署名の作成者を含めて全てのグループメンバが敵にコラプトされていても保たれなければならない．また，敵は OPEN オラクルを自由に利用することができる.

full-anonymity は anonymity, unlinkability を含む．また，full-traceability は，unforgeability, traceability, exculpability, coalition-resistance を含む.

full-anonymity を満たすためには，片方のメンバが敵にコラプトされていても匿名性が保たれなければならない．これは，グループメンバが適応的にコラプトされ，かつローカルのデータを消去できる場合にはよい定義である．しかし，敵が適応的でない場合には，全ての署名が作成される前にコラプトされているか，敵にコラプトされることは全くないかのどちらかである．また，敵が適応的でも，グループメンバがローカルのデータを消去できない場合，敵にコラプトされた際にグループメンバの行動の履歴から作成した署名が必ず知られるため，この安全性の定義は意味がないものとなる.

これらを考慮したうえで，full-anonymity を弱めた匿名性が [7] で提案されている．この定義では，管理者と，敵が署名作成者として選んだ 2 人のグループメンバはコラプトされないという前提になっている.

3 汎用的結合可能性

任意のプロトコルとの結合において安全性を保証するためには，そのようなプロトコルを表現し，議論するための適切な枠組みを用いる必要がある．本稿では Canetti により提案された，汎用的結合可能性の枠組みを用いる.

与えられたタスクの安全性の要求条件は，参加者の入力から望まれる出力を提供する，“ 信頼できるパーティ ” への命令集合で表すことができる．ここで，信頼できるパーティに参加者が入力を渡し，他のやりとりを全くせずにそのパーティから出力を得るような理想的な世界を考える．そして，あるプロトコルを実行することが，その理想的な世界を模倣することになっていれば，そのプロトコルは与えられたタスクを安全に実行しているといえる．この信頼できるパーティにより実行されるアルゴリズムを理想機能と言う.

計算モデルにはプロトコルを実行するパーティと敵 $\mathcal{A}$，そして敵対者的なエンティティである環境 $\mathcal{Z}$ が含まれる．環境 $\mathcal{Z}$ は全てのパーティに対して入力を生成し，全ての出力を読む．そして，さらにプロトコル実行の間を通してずっと任意の手段で敵と対話する．あるプロトコルが，与えられた理想機能 $\mathcal{F}$ を安全に実現するということは，すなわち，どのような敵 $\mathcal{A}$ に対しても，ある理想世界の敵 $\mathcal{S}$ が存在し，どのような環境 $\mathcal{Z}$ も，$\mathcal{A}$ やプロトコルを実行しているパーティと対話しているのか，理想世界で $\mathcal{S}$ や，$\mathcal{F}$ と対話しているパーティと対話しているのかが，識別できないことである.

以下に汎用的結合定理を記述する．パーティが現実世界で用いる通信に加えて，何らかの理想機能 $\mathcal{F}$ の無制限数のコピーに対して理想的なアクセスが許されているモデルを，$\mathcal{F}$-hybrid モデルと言う．この $\mathcal{F}$-hybrid モデ

ルで実行されるあるプロトコル π を考える．ρ を，$\mathcal{F}$ を安全に実現するプロトコルであるとし，π^ρ を結合されたプロトコルとする．すなわち，π^ρ は，π における $\mathcal{F}$ のコピーとの対話を ρ のインスタンスの呼び出しに置き換えたものに等しい．同様に，ρ の出力は，$\mathcal{F}$ のコピーから与えられた値として扱われる．そうすると，π と π^ρ は同じ入出力をすることとなる．特に，$\mathcal{F}$-hybrid モデルで π がある理想機能 $\mathcal{G}$ を安全に実現するならば，π^ρ は現実世界で $\mathcal{G}$ を安全に実現する．

$\mathcal{F}$ の各コピーは，固有の識別子 SID(session identifier) によって識別される．あるコピーに送られる全てのメッセージと，そのコピーが送信する全てのメッセージは，該当する SID を含む．本稿では，特に SID が重要である $\mathcal{F}_{\mathrm{GSIG}}$ の KEYGEN の記述を除いて，$\mathcal{F}$ が送受信するメッセージ中の SID の表記を省略する．

4 グループ署名の理想機能 $\mathcal{F}_{\mathrm{GSIG}}$

本稿では，適応的な敵について考える．また，汎用的結合可能性の考え方では，基本的にはパーティはローカルのデータを効率的に消去することができないため，グループ署名の理想機能が持つべき性質として，full-traceability と，弱められた full-anonymity（敵が選んだ 2 人のメンバはコラプトできないという前提を持つ）を用いることとする．さらに，グループの管理者は敵にコラプトされることはないとする．

グループ署名の理想機能を考えるため，電子署名の理想機能 [3] を参考にする．グループ署名の理想機能は，署名された文書の保管場所として働く．メンバがある文書に SIGN したときは，理想機能はその文書を SIGN されたものとして記録しておく．文書に対して VERIFY が行われれば，その文書が記録されているかを答える．OPEN を行えば，署名者が明らかにされる．

しかし，この機能は検証鍵と署名が存在せず実現が難しいので，これらを理想機能に導入する．

図 1 にグループ署名の理想機能を示す．この理想機能の 1 つのコピーは，1 つのグループを表現する．新たなグループを作成する場合は，また別のコピーが作成される．

5 $(\mathcal{F}^R_{\mathrm{SPK}}, \mathcal{F}^{R'}_{\mathrm{ZK}})$-hybrid モデルでの $\mathcal{F}_{\mathrm{GSIG}}$ の実現

$\mathcal{F}^R_{\mathrm{SPK}}$ を，知識の署名の理想機能とする．これを用いて，$(\mathcal{F}^R_{\mathrm{SPK}}, \mathcal{F}^{R'}_{\mathrm{ZK}})$-hybrid モデルで $\mathcal{F}_{\mathrm{GSIG}}$ を安全に実現するプロトコルを作る．具体的には，汎用的結合可能な知識の署名とゼロ知識証明，選択暗号文攻撃に対して強秘匿な公開鍵暗号，選択文書攻撃に対して存在的偽造不可な電子署名を用いて，汎用的結合可能なグループ署名を構成する．

$\mathcal{F}^{R'}_{\mathrm{ZK}}$ は，[1] で定義されているものを用いる．この理想機能は与えられた関係 $R'(x,w)$ に対して定義され，証明者 P_i が $\mathcal{F}^{R'}_{\mathrm{ZK}}$ に $(\mathtt{prover}, P_i, P_j, x, w)$ を送信すると，検証者 P_j と敵 $\mathcal{S}$ に $(P_i, x, R'(x,w))$ が送られる．ここで w は秘密情報を表す．

5.1 知識の署名

5.1.1 知識の署名の理想機能 $\mathcal{F}^R_{\mathrm{SPK}}$

知識の署名とは，ある関係 $R(v,s)$ と公開情報 v に対し s を知っていることを，s に関する情報を漏らすことなく証明するための署名方式である．署名者が検証鍵 v について，関係 $R(v,s)$ を満たす s を持っていれば，署名者から送られてきた (v,s,m) の組に対して署名 σ を返すと同時に，文書 m を v について署名されたものとして記録する．また，検証者が m について σ を v で検証したいときは，署名された記録があれば 1 を返し，なければ 0 を返す．

図 2 に，$\mathcal{F}^R_{\mathrm{SPK}}$ を示す．

5.1.2 $\mathcal{F}^R_{\mathrm{SPK}}$ を安全に実現するプロトコル

この理想機能 $\mathcal{F}^R_{\mathrm{SPK}}$ は，通常用いられる知識の署名のプロトコルを用いても安全に実現することはできない．通常のプロトコルでは，(v,s) の組から多項式時間アルゴリズムを用いて署名 σ を作成し，(m,σ,v) から多項式時間アルゴリズムで検証することができる．アルゴリズムは公開されているため，環境は (v,s) の組から現実世界で受理可能な σ を作ることができる．しかし，理想世界では (m,σ,v) の組が署名されたものとして記録されていなければ受理されることはない．従って，$\mathcal{Z}$ は現実世界と理想世界を識別することができる．

この問題は，プロトコルが環境 $\mathcal{Z}$ の作成した署名を受理しなければ解決する．そのためには，通常のプロトコルに署名の作成者を限定する機能を追加する必要がある．このプロトコルを構成するため，以下のような $\mathcal{F}^R_{\mathrm{NIZK}}$ を考える．

$\mathcal{F}^R_{\mathrm{NIZK}}$ は，$\mathcal{F}^R_{\mathrm{SPK}}$ の検証手続きを変更して，図 2 中 VERIFY の 2 を削除したものである．すなわち，$(\mathtt{Verify}, m, \sigma, v)$ を受け取ると，(m,σ,v,f') が記録されていれば f' を返すが，それ以外の場合は全て $\mathcal{S}$ に任せるようにする．こうすることで，Lapidot-Shamir 方式等の通常用いられている非対話ゼロ知識証明プロトコルや知識の署名プロトコルで安全に実現することが可能である．この $\mathcal{F}^R_{\mathrm{NIZK}}$ はどんな σ を検証しても 1 を返すようなプロトコルでも安全に実現してしまえるため，知識の署名としては理想的ではない．しかし，$\mathcal{F}^R_{\mathrm{NIZK}}$ と電子署名の理想機能の拡張 $\mathcal{F}_{\mathrm{SIG'}}$ を用いることで，$\mathcal{F}^R_{\mathrm{SPK}}$ を安全に実現するプロトコルを作ることができる．

以下に，$(\mathcal{F}_{\mathrm{SIG'}}, \mathcal{F}^R_{\mathrm{NIZK}})$-hybrid モデルで $\mathcal{F}^R_{\mathrm{SPK}}$ を安全に実現するプロトコル π_{SPK} を構成する方法を示す．$\mathcal{F}_{\mathrm{SIG'}}$ は，[3] で定義されている電子署名の理想機能の拡

理想機能 $\mathcal{F}_{\mathrm{GSIG}}$

$\mathcal{F}_{\mathrm{GSIG}}$ はパーティ P_1, ..., P_n, GM, 敵 $\mathcal{S}$ とともに実行され，以下のように動作する．

KEYGEN: GM から $(\mathtt{KeyGen}, sid)$ を受け取ると，ある sid' について $sid = (GM, sid')$ となっているか確認する．もしなっていなければ，要求は無視する．そうでなければ，$(\mathtt{KeyGen}, P_1, ..., P_n)$ を $\mathcal{S}$ に渡す．$\mathcal{S}$ から $(\mathtt{VerificationKey}, v)$ を受け取り，$(\mathtt{VerificationKey}, v)$ を GM に出力する．また，グループ管理者と検証鍵 (GM, v) を記録する．

JOIN: GM から $(\mathtt{Join}, P_i)$ を受け取ると，P_i をメンバリストに登録する．また，$\mathcal{S}$ にこれを通知する．

SIGN: あるパーティ P_i から $(\mathtt{Sign}, m)$ を受け取ると，P_i がメンバリストに登録されているかどうか確認する．登録されていなければ，要求は無視する．登録されていれば，$(\mathtt{Sign}, m)$ を $\mathcal{S}$ に送信する．$\mathcal{S}$ から $(\mathtt{Signature}, m, \sigma)$ を受け取り，$(\perp, m, \sigma, v, 0)$ が記録されているか確かめる．記録されていれば，P_i にエラーメッセージを送信し，停止する．そうでなければ，$(\mathtt{Signature}, m, \sigma)$ を P_i に送り，$(P_i, m, \sigma, v, 1)$ を記録する．

VERIFY: あるパーティ P_j から $(\mathtt{Verify}, m, \sigma, v')$ を受け取ると，$(\mathtt{Verify}, m, \sigma, v')$ を $\mathcal{S}$ に渡す．$\mathcal{S}$ から $(\mathtt{Verified}, m, \phi, P_x)$ を受け取り，以下のように続ける．

1. $v = v'$ で，かつ，ある P_i について $(P_i, m, \sigma, v, 1)$ が記録されていれば，$f = 1$ とする．
2. 1 ではなく，$v = v'$ でメンバが誰も敵にコラプトされていなく，かつ，どの P_i と σ' についても $(P_i, m, \sigma', v, 1)$ が記録されていなければ，$f = 0$ とし，$(\perp, m, \sigma, v, 0)$ を記録する．
3. 1, 2 でなく，ある P_i について (P_i, m, σ, v', f') が記録されていれば，$f = f'$ とする．
4. 1, 2, 3 に当てはまらなければ，$f = \phi$ とし，$(P_x, m, \sigma, v', \phi)$ を記録する．

P_j に $(\mathtt{Verified}, m, f)$ を出力する．

OPEN: GM から $(\mathtt{Open}, m, \sigma, v')$ を受け取ると，$\mathcal{S}$ にこの要求を通知し，次にこの署名に対し VERIFY する．署名が正当であったならば，記録されている署名者 P_i を調べ，$(\mathtt{Opened}, m, P_i)$ を GM に出力する．署名が正当でなかった場合，$(\mathtt{Opened}, m, \perp)$ を GM に出力する．

REVOKE: GM から $(\mathtt{Revoke}, P_i)$ を受け取ると，メンバリストから P_i を削除し，S に通知する．

図 1: グループ署名の理想機能 $\mathcal{F}_{\mathrm{GSIG}}$

張版である．この $\mathcal{F}_{\mathrm{SIG'}}$ は秘密鍵を多人数で共有する電子署名を表す．GM とメンバが定義されており，GM が指定したメンバが署名可能となる．

1. GM が $(\mathtt{Join}, P_i)$ を入力として受け取ると，$\mathcal{F}_{\mathrm{SIG'}}$ に P_i をメンバ追加するメッセージを送る．
2. あるメンバ P_i が $(\mathtt{Sign}, v, s, m)$ を入力として受け取ると，$\mathcal{F}^R_{\mathrm{NIZK}}$ に $(\mathtt{Sign}, v, s, m)$ を送信して $(\mathtt{Signature}, m, \omega)$ を受け取る．次に $\mathcal{F}_{\mathrm{SIG'}}$ に $(\mathtt{KeyGen})$ を送信し，検証鍵 v_{sig} を受け取る．$\mathcal{F}_{\mathrm{SIG}}$ に $(\mathtt{Sign}, (m, \omega))$ を送信して，m と ω に対する署名 σ_{sig} を受け取る．知識の署名を $\sigma = (v_{sig}, \omega, \sigma_{sig})$ とする．
3. あるパーティ P_j が $(\mathtt{Verify}, m, \sigma, v)$ を入力として受け取ると，σ から v_{sig}, ω, σ_{sig} を取り出す．$\mathcal{F}_{\mathrm{SIG}}$ に $(\mathtt{Verify}, (m, \omega), \sigma_{sig}, v_{sig})$ を送信し，σ_{sig} が (m, ω) の署名であるかを v_{sig} で検証する．$\mathcal{F}_{\mathrm{SIG}}$ から 0 が返ってくれば，$f = 0$ とする．1 が返ってくれば，$\mathcal{F}^R_{\mathrm{NIZK}}$ に $(\mathtt{Verify}, m, \omega, v)$ を送信する．0 が返ってくれば $f = 0$ とし，1 が返ってくれば，$f = 1$ とする．検証結果として f を返す．

定理 1 π_{SPK} は $(\mathcal{F}_{\mathrm{SIG'}}, \mathcal{F}^R_{\mathrm{NIZK}})$-hybrid モデルで $\mathcal{F}^R_{\mathrm{SPK}}$ を安全に実現する．

略証: $\mathcal{A}$ を $(\mathcal{F}_{\mathrm{SIG'}}, \mathcal{F}^R_{\mathrm{NIZK}})$-hybrid モデルでの敵とする．このとき，どのような環境 $\mathcal{Z}$ も，$(\mathcal{F}_{\mathrm{SIG'}}, \mathcal{F}^R_{\mathrm{NIZK}})$-hybrid モデルで $\mathcal{A}$ やパーティと対話しているのか，$\mathcal{F}^R_{\mathrm{SPK}}$ の理想世界における敵 $\mathcal{S}$ と対話しているのか識別できないように $\mathcal{S}$ を構成する．$\mathcal{S}$ は $\mathcal{A}$ の行動をシミュレートし，以下のように続ける．

1. $\mathcal{Z}$ からの入力は $\mathcal{A}$ に，$\mathcal{A}$ の出力は $\mathcal{Z}$ に転送する．
2. $\mathcal{F}^R_{\mathrm{SPK}}$ からメッセージ $(\mathtt{Join}, P_i)$ を受け取ると，P_i をメンバに加える．

理想機能 $\mathcal{F}_{\mathrm{SPK}}^{R}$

$\mathcal{F}_{\mathrm{SPK}}^{R}$ はある関係 R について定義され，パーティ $P_1, ..., P_n$, GM, 敵 $\mathcal{S}$ とともに実行され，以下のように動作する．

JOIN: GM から $(\mathtt{Join}, P_i)$ を受け取ると，P_i を SIGN 可能パーティリストに追加し，S に通知する．

SIGN: ある SIGN 可能パーティ P_i から $(\mathtt{Sign}, v, s, m)$ を受け取ると，(v, s) が関係 R を満たすかどうかを確認する．満たしていなければ，要求は無視する．満たしていれば，$(\mathtt{Sign}, v, m)$ を $\mathcal{S}$ に送信する．$\mathcal{S}$ から $(\mathtt{Signature}, m, \sigma)$ を受け取り，$(m, \sigma, v, 0)$ が記録されているか確かめる．記録されていれば，P_i にエラーメッセージを送信し，停止する．そうでなければ，$(\mathtt{Signature}, m, \sigma)$ を P_i に送り，$(m, \sigma, v, 1)$ を記録する．

VERIFY: あるパーティ P_j から $(\mathtt{Verify}, m, \sigma, v)$ を受け取ると，$(\mathtt{Verify}, m, \sigma, v)$ を $\mathcal{S}$ に渡す．$\mathcal{S}$ から $(\mathtt{Verified}, m, \phi)$ を受け取り，以下のように続ける．

1. $(m, \sigma, v, 1)$ が記録されていれば，$f = 1$ とする．
2. 1 ではなく，SIGN 可能パーティが誰も敵にコラプトされていなく，かつ，どの σ' についても $(m, \sigma', v, 1)$ が記録されていなければ $f = 0$ とし，$(m, \sigma, v, 0)$ を記録する．
3. 1, 2 でなく，$(m, \sigma, v, 0)$ が記録されていれば，$f = 0$ とする．
4. 1, 2, 3 に当てはまらなければ，$f = \phi$ とし，(m, σ, v, ϕ) を記録する．

P_j に $(\mathtt{Verified}, m, f)$ を出力する．

図 2: 知識の署名の理想機能 $\mathcal{F}_{\mathrm{SPK}}^{R}$

3. $\mathcal{F}_{\mathrm{SPK}}^{R}$ からメッセージ $(\mathtt{Sign}, v, m)$ を受け取ると，π_{SPK} の SIGN をシミュレートし，知識の署名 σ を作成して $\mathcal{F}_{\mathrm{SPK}}^{R}$ に送る．
4. $\mathcal{F}_{\mathrm{SPK}}^{R}$ からメッセージ $(\mathtt{Verify}, m, \sigma, v)$ を受け取ると，π_{SPK} の VERIFY をシミュレートする．得られた結果を ϕ とし，$\mathcal{F}_{\mathrm{SPK}}^{R}$ に送る．
5. $\mathcal{A}$ がパーティをコラプトしたときは，$\mathcal{S}$ も理想世界で該当するパーティをコラプトし，その内部状態を $\mathcal{A}$ に渡す．

以上により $\mathcal{S}$ は $\mathcal{A}$ の出力をシミュレートできるため，π_{SPK} は $(\mathcal{F}_{\mathrm{SIG}}, \mathcal{F}_{\mathrm{NIZK}}^{R})$-hybrid モデルで $\mathcal{F}_{\mathrm{SPK}}^{R}$ を安全に実現する．

5.2 $\mathcal{F}_{\mathrm{GSIG}}$ を安全に実現するプロトコル

以下のようにして，公開鍵暗号，電子署名を用いて，$(\mathcal{F}_{\mathrm{SPK}}^{R}, \mathcal{F}_{\mathrm{ZK}}^{R'})$-hybrid モデルで $\mathcal{F}_{\mathrm{GSIG}}$ を安全に実現するプロトコル π_{GSIG} を構成する．REVOKE を実装するため，脱退者リストを用いる．脱退者リストは，GM によって作成・保存され，常に公開されているものとする．

1. GM は $(\mathtt{KeyGen})$ を入力として受け取ると，GM 用の公開鍵暗号の鍵 (pk_e, sk_e), メンバ証明書作成用の電子署名の鍵 (pk_s, sk_s) を生成する．グループの公開鍵を $v = (pk_e, pk_s)$ とする．
2. GM は $(\mathtt{Join}, P_i)$ を入力として受け取ると，P_i の署名用公開鍵 pk_i と，pk_i に対する秘密鍵 sk_i を持っていることの証明を P_i に要求する．P_i は pk_i を GM に送ると同時に $\mathcal{F}_{\mathrm{ZK}}^{R'}$（ここでの $R'(pk_i, sk_i)$ は pk_i と sk_i が一組の鍵である関係）に $(\mathtt{prover}, P_i, GM, pk_i, sk_i)$ を送る．GM は P_i から pk_i と，$\mathcal{F}_{\mathrm{ZK}}^{R'}$ から $(P_i, pk_i, R(pk_i, sk_i))$ を受け取ると，$\langle P_i, pk_i \rangle$ に対し sk_s で署名を作成し，これをメンバ証明書 $cert_i$ とする．$cert_i$ を P_i の公開鍵で暗号化して P_i に送る．
3. あるグループメンバ P_i は $(\mathtt{Sign}, m)$ を入力として受け取ると，文書 m に対し sk_i を用いて電子署名 s を生成する．また，乱数 r を選んで $\langle P_i, pk_i, cert_i, s \rangle; r$ を pk_e で暗号化し，C を生成する．公開情報を (pk_e, pk_s, C), 秘密情報を $(P_i, pk_i, cert_i, s, r)$ として，$\mathcal{F}_{\mathrm{SPK}}^{R}$ に $(\mathtt{Sign}, (pk_e, pk_s, C), (P_i, pk_i, cert_i, s, r), m)$ を送信する．この $\mathcal{F}_{\mathrm{SPK}}^{R}$ は，関係 R として，以下のような関係を持つものを選ぶ．
   - $cert_i$ が，$\langle P_i, pk_i \rangle$ の署名であることを pk_s で検証できる．
   - s が m の署名であることを，pk_i で検証できる．
   - $\langle P_i, pk_i, cert_i, s \rangle; r$ を pk_e で暗号化すると C となる．
   - P_i が脱退者リストに含まれていない．

   これに対して $\mathcal{F}_{\mathrm{SPK}}^{R}$ が出力する知識の署名を σ_{SPK} とし，グループ署名を $\sigma = (C, \sigma_{SPK})$ とする．
4. あるパーティ P_j が $(\mathtt{Verify}, m, \sigma, v')$ を入力として受け取ると，σ から C と σ_{SPK} を取り出し，文書 m, 知識の署名 σ_{SPK}, 公開情報 (v', C) として $\mathcal{F}_{\mathrm{SPK}}^{R}$ に $(\mathtt{Verify}, m, \sigma_{SPK}, (v', C))$ を送信する．$\mathcal{F}_{\mathrm{SPK}}^{R}$ から返ってきた結果 f をグループ署名の検証結果とする．

5. GM が $(\texttt{Open}, m, \sigma, v')$ を入力として受け取ると，まず $(\texttt{Verify}, m, \sigma, v')$ の場合と同じように検証する．検証結果が 0 ならば $\perp$ を出力する．1 であったら，C を sk_e で復号し，$\langle P_i, pk_i, cert_i, s\rangle$ を求める．もし P_i が脱退者リストに含まれていたり，$cert_i$ を pk_s で検証した結果 $\langle P_i, pk_i\rangle$ の正当な署名でなかったり，m が pk_i により s の正当な署名でなかったと検証された場合は，OPEN の結果は $\perp$ とする．そうでなければ，署名者を P_i とする．

6. GM が $(\texttt{Revoke}, P_i)$ を入力として受け取ると，P_i を脱退者リストに登録する．

定理 2 π_{GSIG} は，$(\mathcal{F}_{\mathrm{SPK}}^{R}, \mathcal{F}_{\mathrm{ZK}}^{R'})$-hybrid モデルで $\mathcal{F}_{\mathrm{GSIG}}$ を安全に実現する．

略証: $\mathcal{A}$ を $(\mathcal{F}_{\mathrm{SPK}}^{R}, \mathcal{F}_{\mathrm{ZK}}^{R'})$-hybrid モデルでの敵とする．このとき，どのような環境 $\mathcal{Z}$ も，$(\mathcal{F}_{\mathrm{SPK}}^{R}, \mathcal{F}_{\mathrm{ZK}}^{R'})$-hybrid モデルで $\mathcal{A}$ やパーティと対話しているのか，$\mathcal{F}_{\mathrm{GSIG}}$ の理想世界における敵 $\mathcal{S}$ と対話しているのか識別できないように $\mathcal{S}$ を構成する．$\mathcal{S}$ は $\mathcal{A}$ の行動をシミュレートし，以下のように続ける．

1. $\mathcal{Z}$ からの入力は $\mathcal{A}$ に，$\mathcal{A}$ の出力は $\mathcal{Z}$ に転送する．

2. $\mathcal{F}_{\mathrm{GSIG}}$ からメッセージ $(\texttt{KeyGen})$ を受け取ると，GM 用の公開鍵暗号鍵ペアとメンバ証明書作成用の電子署名鍵ペアを作成し，それらの公開鍵 (pk_e, pk_s) をグループの公開鍵 v として，$\mathcal{F}_{\mathrm{GSIG}}$ に $(\texttt{VerificationKey}, v)$ を送信する．また，各パーティの電子署名用鍵ペア $(pk_1, sk_1), ..., (pk_n, sk_n)$ を作成しておく．

3. $\mathcal{F}_{\mathrm{GSIG}}$ から P_i が JOIN したことを通知されると，π_{GSIG} の Join プロトコルをシミュレートし，$\langle P_i, pk_i\rangle$ に対する署名 $cert_i$（メンバ証明書）を作成する．

4. $\mathcal{F}_{\mathrm{GSIG}}$ からメッセージ $(\texttt{Sign}, m)$ を受け取ると，任意のグループメンバ P_i のメンバ証明書を用いて，公開鍵暗号，電子署名，$\mathcal{F}_{\mathrm{SPK}}^{R}$ をシミュレートしてグループ署名 $\sigma = (C, \sigma_{SPK})$ を作成し，$(\texttt{Signature}, m, \sigma)$ を $\mathcal{F}_{\mathrm{GSIG}}$ に送る．

5. $\mathcal{F}_{\mathrm{GSIG}}$ からメッセージ $(\texttt{Verify}, m, \sigma, v')$ を受け取ると，$\mathcal{F}_{\mathrm{SPK}}^{R}$ による署名検証をシミュレートして ϕ を得る．また，GM の秘密鍵による OPEN の結果 P_x を計算し，$(\texttt{Verified}, m, \phi, P_x)$ を $\mathcal{F}_{\mathrm{GSIG}}$ に送信する．

6. $\mathcal{F}_{\mathrm{GSIG}}$ から OPEN の通知を受けると，この OPEN をシミュレートする．

7. $\mathcal{F}_{\mathrm{GSIG}}$ から P_i が REVOKE されたことを通知されると，脱退者リストに P_i を追加する．

8. $\mathcal{A}$ がパーティをコラプトしたときは，$\mathcal{S}$ も理想世界で該当するパーティをコラプトし，その内部状態を $\mathcal{A}$ に渡す．

上記の $\mathcal{S}$ では，署名の作成者を任意に選んでグループ署名 σ を作成しているため，GM が敵にコラプトされるとその秘密鍵が $\mathcal{Z}$ に渡り，2 つの世界の区別がついてしまう．しかし，ここでは GM はコラプトされないと仮定しているため，この $\mathcal{S}$ に対しては $\mathcal{Z}$ は 2 つの世界を識別することができない．従って，π_{GSIG} は $(\mathcal{F}_{\mathrm{SPK}}^{R}, \mathcal{F}_{\mathrm{ZK}}^{R'})$-hybrid モデルで $\mathcal{F}_{\mathrm{GSIG}}$ を安全に実現する．

6 おわりに

グループ署名の汎用的結合可能性について考察した．そのために，まずグループ署名の安全性の定義について調査した．次に，それに基づいてグループ署名の理想的な機能 $\mathcal{F}_{\mathrm{GSIG}}$ を定義した．そして，$\mathcal{F}_{\mathrm{GSIG}}$ を安全に実現するプロトコルとして，$(\mathcal{F}_{\mathrm{SPK}}^{R}, \mathcal{F}_{\mathrm{ZK}}^{R'})$-hybrid モデルでの構成法を提案した．

参考文献

[1] R. Canetti, “Universally Composable Security: A New Paradigm for Cryptographic Protocols,” Proceedings of 42nd FOCS, pp. 136-145. IEEE, 2001. http://eprint.iacr.org/2000/067.

[2] R. Canetti and H.Krawczyk, “Universally Composable Key Exchange and Secure Channels,” Proceedings of Eurocrypt ’02, pp. 337-351, 2002. LNCS 2332.

[3] R. Canetti, “Universally Composable Signatures, Certification, and Authentication,” Proceedings of 17th CSFW, 2004.

[4] D. Chaum and E. van Heyst, “Group Signatures,” Proceedings of EUROCRYPT ’91, pp. 257-265. LNCS 547, 1991.

[5] G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik, “A Practical and Provably Secure Coalition-Resistant Group Signature Scheme,” Proceedings of CRYPTO ’00, pp. 255-270. LNCS 1880, 2000.

[6] M. Bellare, D. Micciancio, and B. Warinschi, “Foundations of Group Signatures: Formal Definitions, Simplified Requirements, and a Construction Based on General Assumptions,” Proceedings of EUROCRYPT ’03, pp. 614-629. LNCS 2656, 2003.

[7] J. Camenisch and J. Groth, “Group Signatures: Better Efficiency and New Theoretical Aspects,” SCN ’04.