社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2006－AL－105（1）
2006／3／17

共有情報を用いた同時実行可能なゼロ知識認証について

松浦 恭平 †　上土井 陽子 ††　若林 真一 ††

† 広島市立大学院情報科学研究科
†† 広島市立大学情報科学部
〒 731-3194 広島市安佐南区大塚東 3 丁目 4-1

あらまし ネットワーク上の通信で問題となるなりすましを防ぐために，プロトコルが持つべき性質の 1 つにゼロ知識性がある．ゼロ知識プロトコルにおいて，証明者は検証者に対してのみ有効な証明記録しか与えない．つまり，ゼロ知識プロトコルにおいて，第三者に対して有効な証明記録を検証者は得ることができない．インターネットなどの同時実行環境下では，複数の検証者による共同攻撃を考慮しなければならない．現在，プロトコルを複数同時実行した場合にゼロ知識性を完全に保証するプロトコルは提案されていない．本研究では，同時実行におけるプロトコルのゼロ知識性について考察し，証明者と検証者間で共通鍵を保持する状況下で同時実行可能なゼロ知識認証プロトコルを提案する．

Concurrent Zero Knowledge Authentication Protocol with Secret Key

Kyohei Matsuura†　Yoko Kamidoi††　Shin'ichi Wakabayashi††

† Graduate School of Information Sciences, Hiroshima City University
†† Faculty of Information Sciences, Hiroshima City University
3-4-1, Ohzuka-higashi, Asaminami-ku, Hiroshima, 731-3194 Japan

Abstract When communication is performed in a network, we need a *zero knowledge* protocol to make sure that we are communicating with the intended person. In zero knowledge protocols, the prover P can authenticate the message m to just the verifier V, and V cannot convince a third-party that m is authenticated by P. In a concurrent environment such as Internet, we assume a coordinated attack in which many verifiers collude. There are two types of concurrent zero knowledge protocols, but we think that these protocols are not *perfect* concurrent zero knowledge. In this work, we examine concurrent zero knowledge protocols and propose a concurrent zero knowledge authentication protocol with secret keys.

1　はじめに

ネットワーク上での通信において，相手が意図した相手であるか，メッセージが改ざんされていないかを確認するため認証が要求される．認証において，証明者 P は情報 m を検証者 V に対して認証し，検証者 V はその証明を検証する．一般に用いられているデジタル署名認証において，証明者 P は自身の個人鍵で作成した署名を情報 m に添えることにより，情報の作成者であることを証明する．しかしながら、デジタル署名認証では防ぐことのできない問題がある．デジタル署名認証では第三者は署名から情報 m の作成者が証明者 P であることを特定可能であり，証明者 P は検証者 V に対して秘密に情報 m を教えることができない．コンテンツ配信システムにおけるコンテンツ認証にデジタル署名を用いた場合，証明者に認証されたコンテンツを，検証者が証明者になりすまして再配布する著作権侵害行為が考えられる．そのため，証明者が自身の意図しない相手に対して認証しないよう，認証相手を限定可能なゼ

ロ知識プロトコルが要求される．ゼロ知識プロトコルにおいて，検証者 V が証明者 P から得る証明記録は検証者 V に対してのみ有効であり，その証明記録を用いて，情報 m が証明者 P に認証されていることを検証者は第三者に対して証明できない．ゼロ知識プロトコルは，文献 [6] によって最初に定義され，NP 問題を用いた証明プロトコルが文献 [1, 2] で紹介されている．文献 [5] の公開鍵認証プロトコルを基に，文献 [1] でゼロ知識認証プロトコルが提案された．検証者に対しても公開鍵の所有を要求するプロトコルが文献 [7, 8] で提案された．本研究の提案プロトコルは，証明者と検証者間で共通鍵を持つ状況下で，共通鍵を持つ検証者に対してのみ認証する．

本研究では，インターネットなどの同時実行可能環境を想定している．この環境下では，通信相手のメッセージ送信タイミングが分からない．複数のプロトコルを同時実行した場合，複数の検証者が何らかの方法でプロトコルを攻撃したとしても，証明者の認証している情報の出所を明かさないためにはプロトコルはゼロ知識性を維持しなければならない．

同時実行可能環境における複数の検証者による共同攻撃を考慮するため，n 人の検証者をコントロール可能な攻撃者を想定する．攻撃者は，各検証者 $V_i(1 \leq i \leq n)$ に対して，メッセージの送信タイミングと内容の決定権を持つ．攻撃者は，検証者が受け取る証明記録を統合したとき，その記録が第三者に対して有効である状況を作り出そうとする．この攻撃者が存在したとしてもプロトコルがゼロ知識性を維持することができるならば，プロトコルは同時実行可能なゼロ知識性を持つ．

本研究では，攻撃者のプロトコルへの攻撃が成功する場合において，メッセージ送信タイミングとメッセージ内容が満たすべき条件を示す．現在，大きく分けて 2 種類の同時実行可能なゼロ知識プロトコルが提案されている．1 つ目は，文献 [1] で提案されたプロトコル実行時間に制限を与える認証プロトコルである．2 つ目は，文献 [3, 4] で提案されたユーザ数に依存したメッセージ送信回数を要求する証明プロトコルである．しかしこれらのプロトコルは，攻撃者が攻撃を成功させるために満たすべき条件が成立しているため，完全なゼロ知識性を持つとは言えない．本研究で提案するプロトコルは，攻撃成功条件を成立させることなく認証可能である．

以降，2 節で基本的用語とゼロ知識プロトコルの定義について述べる．3 節でゼロ知識プロトコルの例を挙げ，ゼロ知識プロトコルの性質を述べる．4 節で同時実行下でのプロトコルへの攻撃と，同時実行可能なゼロ知識プロトコルが満たしてはならない条件を示す．5 節で同時実行可能な認証プロトコルを提案する．6 節でまとめと今後の課題を述べる．

2 準備

本稿で使用する基本的用語を以下に説明する．

認証

ネットワークにおける認証は証明者 P と検証者 V によって行われ，個人認証とメッセージ認証に区別される．

- 個人認証 : P は情報の出所が P であることを V に対して証明し，V はその証明が正しいかどうか検証する．
- メッセージ認証 : P は情報の出所が P であり，かつその内容が m であることを V に対して証明し，V はその証明が正しいかどうか検証する．

本研究で扱う認証はメッセージ認証である．

コミットメントプロトコル $(\boldsymbol{A}, \boldsymbol{B})$[5]

プロトコルは以下に示すコミットステージとオープンステージで構成される．

1. コミットステージ : A は情報 α をコミットメント $C(\alpha)$ として B に教えるが，B は α を知ることができない．

2. オープンステージ : A は $C(\alpha)$ をオープンし，B に α を教える．A は $C(\alpha)$ を α 以外にオープンできない．

コミットメントは鍵付きの箱に例えることができる．A が情報 α の入った箱に鍵をかけて B に渡し，B は後から鍵を受け取り箱の中身 α を知る．B に箱を渡してあるので，A は後から箱の中身を変えることはできない．コミットメントプロトコルは，互いの情報を同時に知りたい情報交換に利用できる．A が α，B が β を相

手に渡したいとき，A は $C(\alpha)$ を B に渡し，B は β を A に渡す．その後 A は $C(\alpha)$ をオープンする．互いに，自分の情報を相手に送信する時には相手の情報を知らないため，同時に情報を交換しているのと同じである．

ゼロ知識性 [1，2]

ゼロ知識プロトコルは，その実行中に情報を漏らさないことを保証している．以下の 2 つが区別できないとき，証明者 P と検証者 V によるプロトコル (P,V) はゼロ知識性を持つ．

(1) : V の視点から見る P,V 間の通信．つまり P の出力であり，プロトコル実行によって，V が P から受け取る情報 (図 1(a)).

(2) : V にアクセス可能な多項式時間シミュレータ S の出力．つまり V が知っている情報のみを用いて，プロトコル実行時の P の出力をシミュレーションした場合の結果 (図 1(b)).

ゼロ知識認証プロトコルを実行して検証者が得られる情報，つまり証明者の証明記録を用いて，証明者が認証しているということを検証者が第三者に対して証明することはできない．

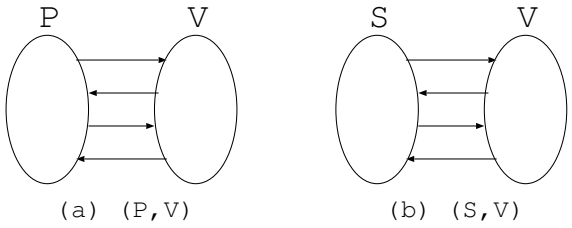

図 1: プロトコル実行とシミュレーション

3 ゼロ知識プロトコル

デジタル署名を用いた認証では，証明者 P が情報 m に自身の個人鍵を使って署名することで検証者 V に対して証明を行う．しかしながらデジタル署名認証では，P は m を自身が望まない相手に対しても認証してしまう．公開鍵認証においては，P しか導出し得ない情報を V が受け取るという証明記録が残ってしまう．このように，デジタル署名認証や公開鍵認証では P が V に渡す証明記録を用いて，P が証明者もしくは認証者であることを V が第三者に証明可能である．そのため，証明記録からは証明者が特定できないよう，つまり，証明者でない参加者が，証明者になりすますことはできないが，やりとりの記録だけは作成可能である証明法が要求される．この証明法としてゼロ知識証明がある．

3.1 ゼロ知識証明

図 2 の洞窟は C 地点と D 地点間に魔法の扉があり，証明者 P はその扉を開ける秘密の言葉を知っている．P は自身が証明者であることを洞窟を使って検証者 V に対してのみ証明する．

プロトコル 1 ．ゼロ知識証明プロトコル[2]

Step1 : P,V ともに A 地点に立ち，P は C もしくは D 地点へ行く．

Step2 : V は B 地点へ行き，C 側の道もしくは D 側の道から出てくるよう P に要求する．

Step3 : P は要求に応える．

プロトコル 1 では，秘密の言葉を知らなくても 1/2 の確率で Step3 で要求に応えることができるため，プロトコル実行を k 回繰返すことでなりすまし成功確率を下げる．$k = 16$ の場合，秘密の言葉を知らない参加者の証明者へのなりすまし可能確率は 1/65536 となり，なりすましは不可能である．

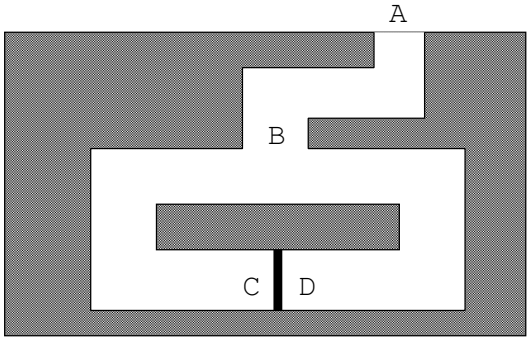

図 2: ゼロ知識洞窟

次に，やりとりの記録が作成可能であることを示す．秘密の言葉を知らない協力者と V は以下の方法でやりとりの記録を作る．

Step1 : 協力者と V は A 地点に立ち，協力者は D 地点へ行く．

Step2 ：V はB地点に行き，C側から出てくるよう協力者に要求する．

やり直し

Step1' ：協力者と V はA地点に立ち，協力者はC地点へ行く．

Step2' ：V はB地点に行き，C側から出てくるよう協力者に要求する．

Step3 ：協力者は秘密の言葉を知らずして要求に応える．

協力者はStep2で V からの要求を知った後で，再びStep1からやり直すことでやりとりの記録を作ることが可能である．

プロトコル1において，秘密の言葉を知らない参加者が証明者になりすますことは不可能であるが，やりとりの記録を作ることは可能である．そのため，プロトコル1によって V が得る証明記録を用いても，第三者に P が証明者であることを納得させることができない．

3.2 ゼロ知識認証プロトコル

ゼロ知識認証プロトコルを用いることで，証明者 P は情報 m を検証者 V に対してのみ認証することができる．V は P の証明記録を用いて，m が P によって認証されていることを第三者に対して証明することができない．そのため，P は m の認証相手を限定することができる．

文献 [1] で提案された逐次実行でゼロ知識性を持つ認証プロトコルを示す．E_P は P の公開鍵暗号，E_K はセッション鍵 K の共通鍵暗号，r は V が決定する n ビット乱数，$m \circ r$ は m と r の連結とする．

プロトコル 2 ．ゼロ知識認証プロトコル

Step1 $V \rightarrow P : E_P(m \circ r)$

Step2 $P \rightarrow V : E_K(r)$

Step3 $V \rightarrow P : r$

Step4 $P \rightarrow V : K$

P はStep1で受信した情報を復号して x を得，自身が認証したい m を用いて，x から m に対応する接頭辞を除いた y を求める．P は y を V に送信し，V は自身の持つ r が y と一致することを確認する．m の出所が証明者であることを V が検証できるため，認証が可能となる．しかしゼロ知識プロトコルとするために，P が r をコミットメント $E_K(r)$ として V に渡すことで，P, V 間で同時に r を交換している．

多項式時間シミュレータ S によるプロトコル2のシミュレーションを以下に示す．

Step1 $V \rightarrow P : E_P(m \circ r)$

Step2 $P \rightarrow V : E_K(?)$

Step3 $V \rightarrow P : r$

巻き戻し ：Step2からやり直す．

Step2' $P \rightarrow V : E_K(r)$

Step3' $V \rightarrow P : r$

Step4 $P \rightarrow V : K$

S はいったん適当なデータ $E_K(?)$ をコミットメントとして送信しておき，r が明らかになった後でシミュレーションを巻き戻しコミットメントを $E_K(r)$ に作り直す．S によりシミュレート可能であるため，プロトコル実行によって V が得る情報，つまり証明記録は S を使って V 自身で作ることができる情報である．

4 同時実行におけるプロトコルのゼロ知識性

プロトコルの逐次実行において，証明者は各検証者1人ずつと順々にプロトコルを実行する．また並列実行において，証明者は複数の検証者とプロトコルを同時に実行し，いずれも各検証者は証明者によって決められたタイミングでプロトコルを実行する．しかしプロトコルの同時実行においては，並列実行と同様に証明者は同時に複数の検証者とプロトコルを実行するが，並列実行と異なり各検証者がプロトコルを開始するタイミングやメッセージを送信するタイミングを証明者が決定することができない．この状況においてゼロ知識プロトコルであるためには，何らかの方法で複数の検証者が協力し，メッセージ送信タイミングの自由な制御を利用してそのプロトコルを攻撃したとしても，プロトコルはゼロ知識性を維持しなければならない．

4.1 同時実行における攻撃

プロトコルの同時実行における攻撃を考えるため，以下に示す攻撃者 A の存在を仮定する．A は n 人の各検証者 $V_i(1 \leq i \leq n)$ に対して，プロトコル中にメッセージを送信するタイミングや送信するメッセージの内容を決定することができる．また検証者間の情報交換のためのパイプ役を果たす．A が存在したとしても，プロトコルがゼロ知識性を維持できるならば，そのプロトコルは同時実行可能なゼロ知識性を持つと考えられる．

プロトコル 2 を n 人の検証者 V_i で同時に実行した場合の A の攻撃例を示す．A は V_i に図 3 のタイミングで Step を実行させる．また A は，V_i が Step1 で使用する r_i を n 入力の一方向関数 h と $V_1, V_2, \cdots, V_{i-1}$ が Step2 で受信するコミットメント $E_{K_1}(r_1), E_{K_2}(r_2), \cdots, E_{K_{i-1}}(r_{i-1})$ により以下の通り決定する．ここで，コミットメント $C_i = E_{K_i}(r_i)$ とし，C_i の解を r_i とする．

$$r_i = h(C_1, C_2, \cdots, C_{i-1}, 0, \cdots, 0)$$

V_i が r_i を決定するとき，$V_i, \cdots, V_n$ に対するコミットメント $C_i, \cdots, C_n$ は分からないため，h の入力の i から n 項までは 0 を入力する．

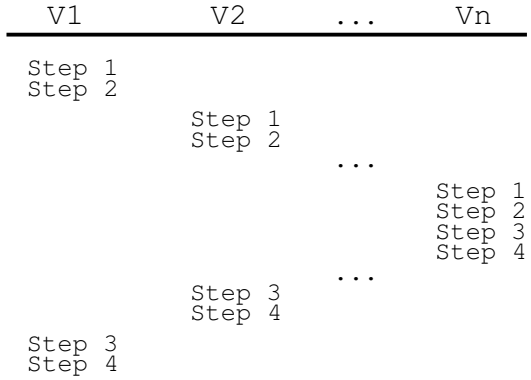

図 3: 攻撃者 A による Step 実行タイミングの決定

$C_1, ..., C_n$ が各解 r_i に合ったコミットメント，つまり $C_i = E_{K_i}(r_i)$ であるかの判定関数 d を用いる．ここで，C_i が解 r_i に合ったコミットメントであるとき，d の出力の i 項目の値が 1 で，そうでないとき 0 である．d は n 個のコミットメントの列から n ビット列への関数であり，シミュレーションを完成するためには，d の出力は全て 1 である n ビット列でなければならない．シミュレーションにおいて，V_i の Step3 に到達したとき，Step2 でのコミットメント C_i が Step3 で V_i が明かす解 r_i に合ったコミットメントでないとき，シミュレーションを巻き戻しコミットメント C_i を作り直す．$n = 3$ の場合の多項式時間シミュレータ S によるシミュレーションを表 1 に示す．

表 1: プロトコル 2 のシミュレーション $(n = 3)$

巻き戻し回数	作り直す C_i	変化する r_i	$d(C_1, C_2, C_3)$ の出力
1	C_3	なし	$(0, 0, 1)$
2	C_2	r_3	$(0, 1, 0)$
3	C_3	なし	$(0, 1, 1)$
4	C_1	r_2, r_3	$(1, 0, 0)$
5	C_3	なし	$(1, 0, 1)$
6	C_2	r_3	$(1, 1, 0)$
7	C_3	なし	$(1, 1, 1)$

最初の巻き戻しでは，V_3 の Step3 に到達し，Step2 に戻り $C_3 = E_{K_3}(r_3)$ に作り直す．次に V_2 の Step3 に到達し解 r_2 が明らかとなり，Step2 に戻り C_2 を $C_2 = E_{K_2}(r_2)$ に作り直す．このとき，C_2 が作り直しにより変化したため，$r_3 = h(C_1, C_2, 0)$ より，解 r_3 も変化する．そのため，再び V_3 の Step3 に到達したとき，コミットメント C_3 を変化後の解 r_3 に合うように作り直さなければならない．C_3 の作り直し後 V_1 の Step3 に到達し，Step2 に戻り $C_1 = E_{K_1}(r_1)$ に作り直したとき，C_1 が変化し $r_2 = h(C_1, 0, 0)$,$r_3 = h(C_1, C_2, 0)$ より解 r_2, r_3 ともに変化する．変化後の解 r_2, r_3 に合ったコミットメントに C_2, C_3 を再び作り直さなければならない．

ある検証者に対しての巻き戻しが，他の検証者に巻き戻しを要求する攻撃において，C_2 を 1 回作り直すとき，C_3 は 2 回作り直さなければならない．同様に C_1 を 1 回作り直すとき，C_2 は 2 回作り直さなければならない．よって，C_1 を 1 回作り直すとき，C_i は 2^{i-1} 回作り直さなければならず，$C_1, \cdots, C_n$ 全てを作り直すにはシミュレータ S は 2^n 回の作り直しを要求される．指数時間かかるため S ではシミュレートできず，プロトコル 2 は同時実行可能なゼロ知識

性を持たない．

このように攻撃者 A は複数の検証者を操り，証明者 P による証明が多項式時間でシミュレートできない状況，つまり証明者以外がやりとりの記録を作成できない状況を作りだそうとする．多項式時間でシミュレートできないのであれば，証明記録を用いて，情報 m が P によって認証されていることを第三者に証明可能である．

4.2 問題点の解析

シミュレータ S と巻き戻しについて考えるためにプロトコル 2 を一般化する．文献 [1] によると，同時実行可能なゼロ知識プロトコルのメッセージ送信回数の下界が 4 であることが証明されている．また，ゼロ知識プロトコルにはコミットメントプロトコルが用いられることから，プロトコル 2 は以下のように一般化できると考えられる．

Step1 $V_i \rightarrow P : P$ であれば解ける問題 Q_i．Q_i の解は α_i.

Step2 $P \rightarrow V_i : Q_i$ の解のコミットメント $C_i(\alpha_i)$

Step3 $V_i \rightarrow P : Q_i$ の解 α_i

Step4 $P \rightarrow V_i : C_i$ のオープン

巻き戻しが要求されるのは，コミットメントが $C_i(\alpha_i)$ でない場合，つまり，シミュレータ S が問題 Q_i の解 α_i を知らないとき，もしくは問題が Q_i から Q_i' に変わり，解が α_i から α_i' に変わったときである．後者の場合，一度作り直したコミットメント $C_i(\alpha)$ を $C_i(\alpha')$ に作り直さなければならない．シミュレーション中に問題 Q_i を変化させるには，変化する情報を基に問題 Q_i を決定すればよい．コミットメントは，巻き戻しによって作り直される前と作り直した後で変化している．そのため検証者 V_i は，問題 Q_i の決定に他の検証者 V_j へのコミットメント $C_j(\alpha)$ を利用すればコミットメントの変化とともに問題 Q_i を変化させることができる．図 4 の Step タイミングで検証者 V_i, V_j がプロトコルを実行する場合を考える．問題 Q_i の解 α_i が明らかとなりシミュレータ S が検証者 V_i に対するコミットメントを $C_i(\alpha_i)$ に作り直した後，問題 Q_j の解 α_j が明らかとなり S がもう一方の検証者 V_j に対するコミットメントを $C_j(\alpha_j)$ に作り直すと，問題 Q_i が Q_i' に変化し，S は V_i へのコミットメントを $C_i(\alpha')$ に作り直さなければならない．このようにして検証者 V_i に対するコミットメントの作り直し回数，つまりシミュレーションの巻き戻し回数を指数的に増加させたのが 4.1 節での攻撃であり，最も強力な攻撃法であると考えられる．

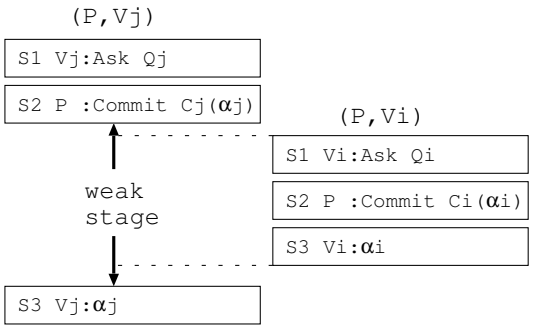

図 4: 巻き戻しを要求される Step 実行

この攻撃が可能となるのは，以下の 2 つの条件が成立することが原因である．

1. ウィークステージの存在 ：
証明者 P が検証者 V_j に対して $C_j(\alpha_j)$ をコミットしてから V_j が解 α_j を送信するまでの間 (ウィークステージ) に，他の検証者 V_i が問題 Q_i を作成し，P が V_i に対して $C_i(\alpha_i)$ をコミット，V_i が解 α_i を送信するまでを実行できる (図 4).

2. ウィークステージへの攻撃手段の存在 ：
検証者 V_i が任意に決定可能な問題 Q_i 中に含まれる情報 (プロトコル 2 では r_i) が，シミュレーションにおいてコミットメント $C_i(\alpha)$ を決定する．つまり，V_i が解 α_i を決定可能である．

上記の条件が同時実行において，攻撃者の攻撃が成功する条件であり，同時実行可能なゼロ知識プロトコルにおいては成立してはならない条件となる．

現在，大きく分けて 2 種類の同時実行可能なゼロ知識プロトコルが提案されている．1 つ目は，時間制約を用いることでウィークステージ間に実行可能な Step 数を制限し巻き戻し回数を減らしている [1]．2 つ目は，V_i に対して複

数のダミーコミットメントを用意し，攻撃者が攻撃対象として選択したコミットメントが，証明に使用される確率を下げることで巻き戻し回数を減らしている [3, 4]．しかし，これらのプロトコルは上記の 2 つの条件が成立するため完全な同時実行可能なゼロ知識性を持っているとは言えない．

次節で示す提案プロトコルでは，条件 2 が成立しない．提案プロトコルでは証明者 P と検証者 V_i 間で共通鍵 s_i を共有し，V_i に対するコミットメントは s_i によって決定される．

5 提案プロトコル

証明者 P と検証者 V_i 間で共通鍵 s_i を 1 つ共有する．提案プロトコルを用いることにより，s_i を知る V_i に対してのみ P は m を認証する．認証中に P から受け取る情報を用いて，V_i は第三者に対して，P が m を認証していることを証明できない．提案プロトコルを以下に示す．E_{s_i} は s_i を鍵とする共通鍵暗号，$r1_i, r2_i$ はそれぞれ V_i, P が決定する n ビット乱数とする．s_i は $k \cdot \log\ n$ ビットであるが，s_i は暗号鍵であるためサイズは E_{s_i} に依存する (k は定数)．

プロトコル 3 ．提案プロトコル 1

Step1 $V_i \rightarrow P : E_P(m \circ s_i \circ r1_i)$

Step2 $P \rightarrow V_i : E_{K_i}(E_{s_i}(r2_i)), r2_i$

Step3 $V_i \rightarrow P : E_{s_i}(r2_i), r1_i$

Step4 $P \rightarrow V_i : K_i$

証明者であるならば $E_P(m \circ s_i \circ r1_i)$ を復号でき，認証する内容が m ならば $m \circ s_i \circ r1_i$ から s_i を得ることができる．そのため，Step2 でのコミットメントの中身が $E_{s_i}(r2_i)$ であるとき，認証が可能となる．

5.1 同時実行可能なゼロ知識性

まず，プロトコル 3 の逐次実行を多項式時間シミュレータ S を用いて以下の通りシミュレートする．

Step1 $V_i \rightarrow S : E_P(m \circ s_i \circ r1_i)$

Step2 $S \rightarrow V_i : E_{K_i}(?), r2_i$

Step3 $V_i \rightarrow S : E_{s_i}(r2_i), r1_i$

巻き戻し ：Step2 からやり直す．

Step2' $S \rightarrow V_i : E_{K_i}(E_{s_i}(r2_i)), r2_i$

Step3' $V_i \rightarrow S : E_{s_i}(r2_i), r1_i$

Step4 $P \rightarrow V_i : K_i$

S はいったん適当な値をコミットし，Step3 で $E_{s_i}(r2_i)$ を受信した後でシミュレーションを巻き戻し，$C_i = E_{K_i}(E_{s_i}(r2_i))$ に作り直す．

次に同時実行時のシミュレーションを考える．4.1 節同様 A は V_i に対して，図 3 のタイミングで Step を実行させ，n 入力一方向関数 h と $V_1, \cdots, V_{i-1}$ が Step2 で受信する $E_{K_1}(E_{s_1}(r2_1))$, $\cdots, E_{K_{i-1}}(r2_{i-1})$ を用いて値 $r1_i$ を決定させる．ここで，コミットメント $C_i = E_{K_i}(E_{s_i}(r2_i))$ とする．4.1 節同様，h の入力の i から n 項までは 0 を入力する．

$$r1_i = h(C_1, \cdots, C_{i-1}, 0, \cdots, 0)$$

表 2: プロトコル 2 のシミュレーション ($n = 3$)

巻き戻し回数	作り直す C_i	変化する $r1_i$	$d(C_1, C_2, C_3)$ の出力
1	C_3	なし	$(0, 0, 1)$
2	C_2	$r1_3$	$(0, 1, 1)$
3	C_1	$r1_2, r1_3$	$(1, 1, 1)$

V_3 の Step3 に到達し，コミットメント C_3 を $C_3 = E_{K_3}(E_{s_3}(r2_3))$ に作り直す．

次に V_2 の Step3 に到達し，$C_2 = E_{K_2}(E_{s_2}(r2_2))$ に作り直す．このとき，コミットメント C_2 が変化し，値 $r1_3 = H(C_1, C_2, 0)$ より $r1_3$ が変化する．しかし，4.1 節の場合とは異なり，再び C_3 を作り直す必要はない．なぜなら，$C_3 = E_{K_3}(E_{s_3}(r2_3))$ であり，コミットメント C_3 は値 $r1_3$ によって決定しないからである．シミュレーションにおいて，値 $r2_3$ はシミュレータ S が決定する値であり，鍵 s_3 は検証者 V_3 に対して一意に定まる．そのため，コミットメント C_3 の解 $E_{K_3}(E_{s_3}(r2_3))$ は，値 $r1_3$ によって影響されないため一度作り直したコミットメント C_3 を再び作り直す必要がない．同様に，3 回目の巻き戻しでコミットメント C_1 を作り直しても C_2, C_3 を作り直す必要がないため，3 回の巻き戻しでシミュレーション可能である．

提案プロトコルでは，コミットメント C_i が検証者 V_i の作る情報から影響を及ぼされないため，上記のようにシミュレーションにおける各コミットメント C_i の作り直しが 1 回ずつで

済む．シミュレータ S による n 人に対するシミュレーションが必ず n 回の巻き戻しで可能なため，提案プロトコルは同時実行可能なゼロ知識性を持つと言える．

5.2 提案プロトコルの改良

プロトコル 3 において，Step2 で証明者 P が検証者 V_i に対して送信するコミットメントが $E_{K_i}(E_{s_i}(r2_i))$ であるとき，P は m を認証している．しかし，このコミットメントからは P が認証したい情報が m であることが分からない．P, V_i 間で，m と m' に対する認証プロトコルが同時に実行されている場合を考える．以下のように，両プロトコルとも Step2 まで実行されているとする．

m に対する認証	m' に対する認証
S1 : $E_P(m \circ s_i \circ r1_i)$	
	S1 : $E_P(m' \circ s_i \circ r1'_i)$
S2 : $E_{K_i}(E_{s_i}(r2_i)), r2_i$	
	S2 : $E_{K'_i}(E_{s_i}(r2'_i)), r2'_i$

このとき，両プロトコルの Step2 で V_i が受信するメッセージ中の値 $r2_i, r2'_i$ が，m と m' どちらの認証に対して送信された情報であるかを V_i は区別できない．そのため，$r2_i$ と $r1_i$，$r2'_i$ と $r1'_i$ を対応づけることができず，Step3 において V_i が送信するメッセージの組み合わせが，$E_{s_i}(r2_i), r1'_i$ や $E_{s_i}(r2'_i), r1_i$ となってしまう可能性がある．

プロトコル 3 の Step2 で送信するメッセージを $E_{K_i}(E_{s_i}(r2_i))$ と $m \circ r2_i$ にすることで，この問題を解決できる．

プロトコル 4 ．提案プロトコル 2

Step1 $V_i \rightarrow P : E_P(m \circ s_i \circ r1_i)$

Step2 $P \rightarrow V_i : E_{K_i}(E_{s_i}(r2_i)), m \circ r2_i$

Step3 $V_i \rightarrow P : E_{s_i}(r2_i), r1_i$

Step4 $P \rightarrow V_i : K_i$

プロトコル 4 では V_i は Step2 で受信した $m \circ r2_i$ から $r2_i$(定数ビット分) と対応する接尾辞を除くことで m を求める．Step1 で送信したメッセージから m と $r1_i$ の対応が分かるため，V_i は Step3 で $E_{s_i}(r2_i)$,$r1_i$ を送信する．

プロトコル 4 もプロトコル 3 と同様，P のコミットメントが V_i の決定可能な情報によって決まらない．シミュレーションにおいて，各検証者に対するコミットメントの作り直しがそれぞれ 1 回で済み，合計で n 回の作り直しで済む．そのため，プロトコル 4 も同時実行可能なゼロ知識認証プロトコルである．

6 おわりに

本研究では，プロトコルの同時実行におけるゼロ知識性を考察し，同時実行可能なゼロ知識認証プロトコルを提案した．

また，本研究では，プロトコルへの攻撃が成功するために成立しなければならない，メッセージ送信タイミングとメッセージ内容の条件を示した．提案プロトコルでは，証明者 P と各検証者 V_i は共通鍵を共有し，証明者 P は共通鍵を持つ検証者 V_i に対してのみ情報 m を認証する．提案プロトコルにおいて，送信されるメッセージは攻撃が成功するためのメッセージ内容の条件を満たさないため，攻撃が成功しない．そのため，提案プロトコルは同時実行可能なゼロ知識認証プロトコルである．

今後の課題は，攻撃の成功条件の十分性を示すこと，提案プロトコルのメッセージ作成における複雑さを減らすことなどが挙げられる．

参考文献

[1] C. Dwork, M. Naor and A. Sahai, “Concurrent zero-knowledge,” Journal of the ACM, Vol.51, No.6, pp.851–898 (2004)

[2] B. Schneier, “Applied Cryptography,” John Wiley & Sons Inc (1996)

[3] R. Richardson and J. Kilian, “On the concurrent composition of zero-knowledge proofs,” Proc. on EUROCRYPT, pp.415–431 (1999)

[4] M. Prabhakaran, A. Rosen and A. Sahai, “Concurrent zero knowledge with logarithmic round-complexity,” Proc. on FOCS, pp.366–375 (2002)

[5] D. Dolev, C. Dwork and M. Naor, “Nommalleable cryptography,” SIAM J. Comput, Vol.30, No.2, pp.391–437 (2000)

[6] S. Goldwasser, S. Micali and C. Rackoff, “The knowledge complexity of interactive proof system,” SIAM J. Comput, Vol.18, No.1, pp.186–208 (1989)

[7] T. Cao, D. Lin and R. Xue, “An efficient ID-based deniable authentication protocol from pairings,” Proc. on AINA, pp.388–391 (2005)

[8] M. D. Raimondo and R. Gennaro, “New approaches for deniable authentication,” Proc. on CCS, pp.112–121 (2005)