情報処理学会論文誌 Vol.55 No.5 1518–1529 (May 2014)

FFS認証においてチャレンジの分割により検証に必要な乗算回数を低減させる手法の検討

豊田 健太郎[1,a)] 笹瀬 巌[1,b)]

受付日 2013年5月22日, 採録日 2014年2月14日

概要：Mobile IPv6 およびアドホック・ネットワークの実用化に向けて，第三者による認証機関を利用せず，低演算量で検証可能な端末認証技術が求められている．これらの要件を満たす認証技術としてゼロ知識対話証明があり，その実装例として FFS（Feige-Fiat-Shamir）認証がある．しかし，FFS 認証において正当性を検証する端末は，正当性の証明を要求されるたびにチャレンジを生成し，レスポンスを検証しなければならない．モバイル端末で検証を行う場合，端末の計算能力および電力は限られているため，認証にともなう演算量を低減させる必要がある．そこで本論文では，検証にともなう乗算回数を制御できるようにチャレンジを分割することによって検証にともなう平均乗算回数を低減させる方式を提案する．検証にともなう演算は対応するチャレンジ系列の各要素が 1 である場合に 1,024 ビット長の乗算が発生することが主な原因であるため，各チャレンジにおいて 1 のビットを立てることができる数の上限値を設定することにより，従来方式と比較してレスポンスの検証に必要な平均乗算回数を低減させる．従来と同程度の安全性を達成したうえで，チャレンジの分割数および上限値と平均乗算回数がどの程度低減するかを明らかにする．そして本方式を Android 端末上で実装し，検証にともなう計算時間およびメモリ量について評価を行い，本方式の有効性を示す．

キーワード：ゼロ知識対話証明（ZKP: Zero Knowledge Proof），FFS（Feige-Fiat-Shamir）認証，端末認証，演算量低減

Divided Challenge Method to Reduce the Number of Multiplication in FFS Identification Protocol

Kentaroh Toyoda[1,a)] Iwao Sasase[1,b)]

Received: May 22, 2013, Accepted: February 14, 2014

Abstract: FFS (Feige-Fiat-Shamir) is one of ZKP (Zero Knowledge Proof) which is a light-weight challenge-response authentication protocol. FFS is considered to be used to authenticate mobile nodes under Mobile IPv6 or ad hoc networks environment. However, a verifier must check every response and this could burden him/her. In this paper, we propose a divided challenge method in order to decrease the calculation amount of verification. We point out that the complexity of verification is mainly due to multiplication and that this happens whenever the element of challenge is 1. Thus, we decrease calculation amount by introducing upper bound to restrict the number of element 1 in each divided challenge. We tune two parameters, division number and upper bounds of each challenge, and implements our scheme on an off-the-rack Android device to clarify the efficiency of our method.

Keywords: ZKP (Zero Knowledge Proof), FFS (Feige-Fiat-Shamir) identification protocol, authentication, calculation reduction

1. はじめに

今後実用化が期待されている Mobile IPv6（MIPv6）およびアドホック・ネットワーク，P2P（Peer-to-Peer）な

[1] 慶應義塾大学理工学部情報工学科
Department of ICS, Information and Computer Sciences, Keio University, Yokohama, Kanagawa 223–8522, Japan
[a)] toyoda@sasase.ics.keio.ac.jp
[b)] sasase@ics.keio.ac.jp

どのネットワークにおいて，ネットワークに参加している正当な端末が悪意のある端末になりすまされることを防止するため，端末の認証技術の導入が求められている．これらのネットワークでは，参加する端末数が非常に多く，端末の認証にCA（Certificate Authority）といった第三者による認証機関を利用できない．特にこれらのネットワークでは端末の計算性能および電力が限られているため，低演算量で端末認証が可能な技術が求められている．これらの要件を満たす認証技術としてゼロ知識対話証明（ZKP: Zero Knowledge Proof）がある．ゼロ知識対話証明は，2者間において，一方（証明者）が秘密を持つことをその秘密を明かすことなく他方（検証者）に証明するチャレンジ・レスポンス型の証明技術である．これまでに，MIPv6，アドホック・ネットワークもしくはP2Pの端末認証にゼロ知識対話証明を用いる研究がさかんに行われている [1], [2], [3], [4], [5], [6], [7], [8], [9]．ゼロ知識対話証明の実装例として，主にFFS認証 [10] が用いられる．FFS認証は認証に必要なメモリ量と計算量，なりすましを許容する確率を，鍵のサイズおよび試行回数を変化させることにより柔軟に設定できるため，異なるセキュリティレベルを要求される前述のネットワークにおける端末認証に適している．FFS認証を端末認証に用いる場合，各端末はあらかじめ k 個の秘密鍵と公開鍵の対を用意し，端末の id を公開鍵のハッシュ値と定める．端末はネットワーク参加時もしくは他の端末と通信を行う際に，自身の id を生成するような公開鍵の対となる秘密鍵を所持していることを示す．FFS認証では，通信相手の端末が生成したチャレンジに対して端末がレスポンスを計算し，通信相手が検証を行うチャレンジ・レスポンス方式により証明する．端末は自身の秘密鍵を所持しているという事実のみを他の端末に証明することにより，第三者による認証機関を利用することなく，その端末が他の端末になりすまされていないという正当性を示すことができる．しかし，正当性を検証する端末は，正当性の証明を要求されるたびにチャレンジを生成し，レスポンスを検証しなければならない．モバイル端末で検証を行う場合，端末の計算能力および電力は限られているため，認証にともなう演算量を低減させる必要がある．そこで，検証にともなう演算は検証者が生成したチャレンジ系列の各要素が1である場合に1,024ビット長の乗算を行い，これが検証者にとって負担の大きい演算であることに着目し，本論文ではチャレンジを複数に分割し，各チャレンジ系列において1となる数に上限値を設けることにより平均乗算回数を低減させる方式を提案する．FFS認証では，チャレンジの要素の組合せ数がなりすましを許容する確率に影響するため，単純にチャレンジの1となる要素数に上限を設けるだけでは，安全性が低下する．提案方式ではチャレンジを複数に分割し，すべてのチャレンジに対して正当なレスポンスを返さなければ認証失敗とすることで，従来のなりすまし許容率を満たしたうえで，検証に必要な平均乗算回数を低減させることが可能となる．提案方式および従来方式において，計算時間およびメモリ量をAndroid端末上で評価し，チャレンジを分割した際に不正な端末の検証に必要な計算時間を最高で約73%，正当な端末の検証に必要な演算時間を最低で約34%低減させることができること示す．

以下2章ではFFS認証を用いた既存研究を，3章では従来方式およびその問題点をあげ，4章では提案方式を説明する．5章では特性評価を示し，6章で結論をまとめる．

2. 関連研究

FFS認証はゼロ知識対話証明を用いた認証プロトコルの1つであり，2者間において一方が秘密を保持することを，その秘密を明かすことなく他方に証明する技術である．FFS認証は2者間のみで認証可能なため，第三者機関による認証を用いることができないMIPv6，自律分散制御であるアドホック・ネットワークおよびP2Pネットワークといった環境における端末認証に適しており，これまでに様々な分野への研究がなされている．

KizzaはFFS認証プロトコルを空中ネットワーク（AN: Airborne Network）に適用している [11]．空中ネットワークにおいてはメッセージの伝達に時間的制約があるため，認証時間を短くしなければならない．文献 [11] では，FFS認証において検証者は繰り返し行われるチャレンジを1度に伝達することによって，認証の速度を向上させている．

Leらは MIPv6 における端末認証に FFS 認証を用いている [8]．文献 [8] では端末が使用する IP アドレスを識別するために，IP アドレスの後半 64 ビット部を FFS 認証における公開鍵のハッシュ値から生成し，このような公開鍵に対応する秘密鍵を保持していることを FFS 認証を用いて通信相手に証明する．MIPv6 では IP アドレスの保持証明に第三者による認証を用いることが困難であるが，FFS認証を採用することによりこの課題を解決している．

無線センサネットワーク（WSN: Wireless Sensor Network）ではネットワーク内の正規のノードになりすましてネットワークへの参加を試みる複製攻撃が存在する．そこで Udgata らは WSN において，送信者ノードの正当性を確認するために FFS 認証を用いている [5]．

文献 [1], [3] では P2P ネットワークにおいて，ピア間の相互認証を行うために FFS 認証を用いている．

3. 従来方式

本章では従来方式として FFS 認証の手順および安全性を説明した後，モバイル端末環境において FFS 認証を用いた際の問題点を述べる．

FFS認証を端末認証に用いる場合，各端末はあらかじめ公開鍵および秘密鍵を生成しておく．そして端末は自身の

id を公開鍵から生成し，その公開鍵に対応する秘密鍵を保持していることを通信相手とのチャレンジ・レスポンス認証により証明する．**図 1** に FFS 認証を用いた端末認証のシーケンス図を示す．

端末は公開鍵および秘密鍵を生成する準備として，ブラム数 [12] n を 4 で割ると 3 余る素数 p および q から，

$$n = pq, \tag{1}$$

とする．ただし，n から素因数 p および q を算出することが困難となるように，n のビット数 $|n|$ を $|n|$=1024 bit 程度とする．秘密鍵 $\boldsymbol{s}$ を

$$\boldsymbol{s} = (s_1, s_2, \cdots, s_k), \tag{2}$$

ただし，k を鍵の要素数として $k = \mathcal{O}(\log |n|)$ となるようにあらかじめ用意しておいたうえで，s_i を $1 < s_i < n-1$ を満たすように無作為に k 個選択する．また

$$\boldsymbol{b} = (b_1, b_2, \cdots, b_k), \tag{3}$$

を無作為に $b_i = \{0,1\}$（$1 \leq i \leq k$）として選択し，公開鍵 $\boldsymbol{v}$ を

$$\boldsymbol{v} = (v_1, v_2, \cdots, v_k), \tag{4}$$

ただし

$$v_i = (-1)^{b_i}(s_i^2)^{-1} (\mathrm{mod} n), \tag{5}$$

として算出する．この公開鍵 $\boldsymbol{v}$ と一方向性ハッシュ関数 $f(\cdot)$ を用いて id を式 (6) のように計算する．

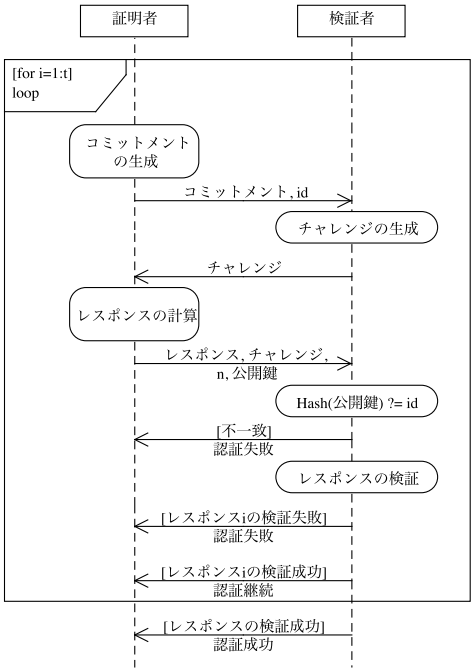

図 1 FFS 認証を用いた端末認証のシーケンス図
Fig. 1 Flow chart of FFS identification protocol.

$$id = f(\boldsymbol{v}). \tag{6}$$

端末はネットワークへの参加時もしくは通信開始時に，自身の正当性を通信相手に証明する．まず自身の id と認証要求としてコミットメント x を，

$$x = (-1)^b r^2 (\mathrm{mod} n), \tag{7}$$

として算出し，通信相手に送信する．ただし，$r \in [0, n)$ および $b = \{0,1\}$ を無作為に選び，それぞれ秘密にする．

通信相手は端末からコミットメントを受け取ると，チャレンジ $\boldsymbol{e}$ を，

$$\boldsymbol{e} = (e_1, \cdots, e_k), \tag{8}$$

として生成し，端末に送信する．ただし，$e_i = \{0,1\}$ とし，少なくとも 1 つの要素は 1 となるように無作為に選択する．

端末はチャレンジを受け取ると，レスポンス y を

$$y = r \prod_{i=1}^{k} s_i^{e_i} (\mathrm{mod} n), \tag{9}$$

として生成し，チャレンジ $\boldsymbol{e}$，公開鍵 $\boldsymbol{v}$ および n とともに通信相手に送信する．

通信相手はレスポンス y，チャレンジ $\boldsymbol{e}$，公開鍵 $\boldsymbol{v}$ および n を受け取ると，まず公開鍵 $\boldsymbol{v}$ に対してハッシュ値を算出し，端末の id に一致するかを確認する．一致しない場合，認証失敗となる．一方，一致した場合はチャレンジ $\boldsymbol{e}$ に対するレスポンス y を検証するため，検証式 z を，

$$z = y^2 \prod_{i=1}^{k} v_i^{e_i} (\mathrm{mod} n). \tag{10}$$

として計算する．$z \neq \pm x$ または $z = 0$ の場合，通信相手は端末が主張する公開鍵 $\boldsymbol{v}$ に対応する秘密鍵 $\boldsymbol{s}$ を保持していないと見なし，該当のチャレンジを破棄したうえで端末に認証失敗を返す．以後このチャレンジに対するレスポンスを受信しても，検証を行わずに破棄する．一方 $z = \pm x$ かつ $z \neq 0$ の場合，通信相手は端末が主張する公開鍵 $\boldsymbol{v}$ に対応する秘密鍵 $\boldsymbol{s}$ を保持している可能性があるため，端末に認証継続を通知する．その後，コミットメントの送信，チャレンジの送信，レスポンスの送信，レスポンスの検証という一連の動作を 1 ラウンドとし，t ラウンド終了時点までに認証が失敗していないならば端末に認証成功を通知する．以上のように複数回のラウンドを繰り返すプロトコルの実行形態をシーケンシャル実行と呼ぶ．

3.1 従来方式の安全性

次に従来方式の安全性を確認する．従来方式の安全性証明は文献 [13] で示されているため，以下では文献 [13] の安全性証明を基に記述する．ゼロ知識対話証明は，完全性，健全性，ゼロ知識性の 3 つを有することを証明すること

で安全性が確認される．ここで，完全性とは正しい証明者（端末）が圧倒的に高い確率で検証者（通信相手）を納得させることができるという性質，健全性は秘密を持たない悪意のある証明者が秘密を保持していると検証者を納得させようとしても無視できる程度に低い確率でしか納得させることができないという性質，ゼロ知識性とは，悪意のある検証者は証明者とのメッセージのやりとりによって得られるいかなる情報を利用しても秘密鍵 s_i に関する情報を得ることができないという性質を指す．本論文における「圧倒的に高い確率」および「無視できる程度に低い確率」を従来方式である文献 [10] 内の 2 章，p.82 の Soundness（健全性）にある overwhelming probability および negligible probability と同一の表現として定義する．すなわち「圧倒的に高い確率」とは鍵長 $|n|$ および任意の整数 b に対して $1-1/|n|^b$ 以上となる確率を，「無視できる程度に低い確率」は任意の整数 a に対して $1/|n|^a$ 以下となる確率を示す．ここで $|n|$ は上述のとおり，セキュリティパラメータ n のビット数を表す．

3.1.1 完全性

Lemma 3.1. 正しい証明者は必ず検証者を納得させることができる．

Proof. 検証者が検証する z は以下の等式を満たす．ただし，定義より $s_i^2 v_i = \pm 1 (\mathrm{mod} n)$ の関係を用いる．

$$\begin{aligned} z = y^2 \prod_{i=1}^{k} v_i^{e_i} (\mathrm{mod} n) &= \left(r \prod_{i=1}^{k} s_i^{e_i} \right)^2 \cdot \prod_{i=1}^{k} v_i^{e_i} \\ &= r^2 \prod_{i=1}^{k} (s_i^2 v_i)^{e_i} \\ &= \pm r^2 = \pm x (\mathrm{mod} n). \end{aligned}$$

□

3.1.2 健全性

Lemma 3.2. $\prod_{i=1}^{k} v_i^{c_i} (\mathrm{mod} n)$（$c_i = -1, 0, 1$，ただしすべて 0 となる場合は除く）の任意の積の平方根を多項式時間で計算できず，かつ秘密 s_i を知らない証明者を想定した場合，プロトコルに対して正しく振る舞う検証者および任意の多項式時間の計算が可能な証明者に対して，検証者が証明者を受理する確率はたかだか 2^{-kt} である．

Proof. （*Sketch*）悪意のある証明者が，秘密鍵を保持しないにもかかわらず検証者に自身が秘密鍵を持つことを納得してもらうには，通信相手の計算する検証に合格するように x および y を送信する方法が考えられる．そのため，悪意のある端末は，あらかじめチャレンジ $\boldsymbol{e}$ を推測しておく．公開鍵 $\boldsymbol{v}$ は容易に取得できるため，この推測に基づき，$\prod_{i=1}^{k} v_i^{e_i} (\mathrm{mod} n)$ の項を計算する．したがって，悪意のある端末は前述の推測が正しいと仮定し，$z = \pm x$ かつ $z \neq 0$ が成立するように x および y の値を無作為に選ばれた r を用いて以下のように計算する．

$$x = \pm r^2 \prod_{i=1}^{k} v_i^{e_i} (\mathrm{mod} n), \tag{11}$$

$$y = r. \tag{12}$$

悪意のある端末は前述のチャレンジ $\boldsymbol{e}$ の推測が正しければ，なりすましに成功する．しかし x はチャレンジ $\boldsymbol{e}$ を検証者から受信する前に送信しなければならないため，悪意のある端末のなりすましが成功する確率 p は，(チャレンジ $\boldsymbol{e}$ の組合せ数)$^{-1}$ と表せる．ここで，チャレンジ $\boldsymbol{e}$ の組合せ数は，k 個の要素のうち，最低 1 個，最大 k 個までビットを立ててよい組合せの数の和で表せ，$2^k - 1 \approx 2^k$ である．したがって，チャレンジ $\boldsymbol{e}$ の推測が正しい確率，すなわち，秘密を保持しない証明者が，この方法で検証者に自身が秘密を持つことを納得してもらえる確率 p は $p = 2^{-k}$ である．t ラウンドの FFS 認証においては，すべてのラウンドに対するチャレンジの推測が正しかった確率となるため，全体として秘密を持たない証明者が認証に合格する確率は 2^{-kt} となる．ここで悪意のある証明者がこの合格率を増大させるために，1 つのコミットメント x に対して異なる複数のチャレンジ $\boldsymbol{e}$ に合格できるようなレスポンス y を計算しておく手法を考える．ここでは複数のチャレンジ $\boldsymbol{e}$ およびレスポンス y のうち任意の 2 つをそれぞれ $(\boldsymbol{e}', \boldsymbol{e}'')$，$(y', y'')$ と表すことにする．ただし，y は非 0 であり，コミットメント x はチャレンジを受け取る前に検証者に送信しなければならないことから $x = y'^2 \prod_{i=1}^{k} v_i^{e_i'} (\mathrm{mod} n) = y''^2 \prod_{i=1}^{k} v_i^{e_i''} (\mathrm{mod} n)$ となる．しかしここで正しい証明者ならば v_i に対する s_i を知っているため $\left(\frac{y'}{y''}\right)^2 = \frac{\prod_{i=1}^{k} v_i^{e_i'}}{\prod_{i=1}^{k} v_i^{e_i''}} (\mathrm{mod} n) = \prod_{i=1}^{k} v_i^{c_i} (\mathrm{mod} n)$（$c_i = -1, 0, 1$，ただしすべて 0 となる場合は除く）の平方根を計算できるはずであるが，この計算には $\prod_{i=1}^{k} v_i^{c_i} (\mathrm{mod}\, n)$ の平方根を求める計算を含むため，秘密 s_i を知らない証明者の能力では計算できない．したがって，秘密を保持しない証明者は任意のチャレンジに合格できるようなレスポンスを用意できず，確率 p を 2^{-kt} より大きくすることができない．よって検証者に秘密を持つことを納得してもらえる確率 p はたかだか $p = 2^{-kt}$ となる．ここで $k = \mathcal{O}(\log |n|)$ および $t = \Theta(|n|)$ であることから p の大きさは $\mathcal{O}(1/(|n| \cdot 2^{|n|}))$ となるため，任意の整数 a に対して $1/|n|^a > 2^{-kt}$ を満たし，秘密を知らない証明者が検証に合格できる確率は無視できる程度に低い確率となる． □

3.1.3 ゼロ知識性

Lemma 3.3. $k = \mathcal{O}(\log |n|)$ および $t = \Theta(|n|)$ に対して，本プロトコルをシーケンシャル実行した場合，ゼロ知識性を満たす．

Proof. （*Sketch*）本プロトコルをシーケンシャル実行した際にゼロ知識性を満たすという直観を得るためには，悪

意のある検証者が正当な証明者とのやりとりにおいて得られる情報は，正当な証明者と対話することなく，悪意のある検証者が自分自身で得られる情報と同じであることを示す必要がある．ここで正当な証明者を $\bar{A}$，正当な検証者を $\bar{B}$，$\bar{A}$ から秘密を知ろうとする検証者を $\widetilde{B}$ とし，以下のシミュレータ $M_{\widetilde{B}}$ を考える．

(1) 以下のステップ (2) から (6) を t 回繰り返す．
(2) チャレンジ $\boldsymbol{e}'$ を推測する．
(3) コミットメント $x = \pm r^2 \prod_{i=1}^{k} v_i^{e'_i} (\mathrm{mod} n)$ およびレスポンス $y = r$ を計算する．
(4) 検証者 $\widetilde{B}$ にコミットメント x を送り，チャレンジ $\boldsymbol{e}$ を受け取る．
(5) 推測したチャレンジ $\boldsymbol{e}'$ と受け取ったチャレンジ $\boldsymbol{e}$ が異なる場合，(2) からやり直す．
(6) 推測したチャレンジ $\boldsymbol{e}'$ と受け取ったチャレンジ $\boldsymbol{e}$ が等しい場合，レスポンス r を $M_{\widetilde{B}}$ に送る．

FFS 認証をシーケンシャル実行した場合，このシミュレータによって得られる系列 $\{x, y, \boldsymbol{e}\}$ は秘密 s_i を知らなくても検証に合格できる系列であり，正当な証明者 $\bar{A}$ との対話によって得られる系列 $\{x, y, \boldsymbol{e}\}$ とまったく同じものとなる．シミュレータ $M_{\widetilde{B}}$ は平均 $\mathcal{O}(t \cdot 2^k)$ 回であり，k の大きさが $\mathcal{O}(\log |n|)$，t の大きさが $\Theta(|n|)$ であることから，この計算は $|n|$ の多項式時間で計算可能である．したがって，悪意のある検証者 $\widetilde{B}$ はシミュレータ $M_{\widetilde{B}}$ を用いたとしても，秘密 s_i をまったく知らなくても自分自身で正当な証明者 $\bar{A}$ とのやりとりにおいて得られる情報しか得ることができないため，ゼロ知識性を満たす． □

3.2 従来方式の問題点

以上のように，FFS 認証では端末間で認証を行うことが可能なため，第三者の認証機関を用いることが困難なアドホック・ネットワーク，P2P ネットワーク，MIPv6 といった環境に適している．しかしながら，第三者による認証が不要な代わりに通信相手が正当性の検証作業を行わなければならない．またこれらのネットワークのノードはモバイル端末が想定されるため，限られた計算資源およびエネルギー資源をなるべく浪費させないようにする必要がある．そこで CPU 資源を浪費させる要因は 1,024 bit 程度の乗算演算であることに着目する．通信相手はコミットメントを受け取ると，移動端末に対してチャレンジ $\boldsymbol{e} = (e_1, \cdots, e_k)$（ただし $e_i = \{0, 1\}$）を出題する．その後，式 (10) において移動端末から受け取るレスポンス y から z を算出する．したがって，通信相手自身がチャレンジを生成する際に $e_i = 1$ としたビットに対し，v_i の乗算を行い，$e_i = 0$ とした v_i に対しては乗算を行わないことが分かる．

4. 提案方式

そこで本論文では，チャレンジを複数に分割し，各チャレンジにおいて 1 のビットを立てることができる数の上限値を設定することにより，従来方式と比較してレスポンスの検証に必要な平均乗算回数を低減させる方式を提案する．従来，通信相手は一度に検証式を計算していたのに対し，本方式は検証フェーズを分割し，各チャレンジにおいてビットを立てることが可能な数に上限を設けることで，秘密鍵を知らないなりすましを試みる端末の検証を早期に終了させることを可能とする．またすべてのチャレンジに通過しないと認証が成功しないようにすることにより，チャレンジ生成時にビットを立てることができる数に制限を設けたとしても，なりすましを行う確率を十分に小さくすることを可能とする．

図 2 にシーケンシャル実行版 FFS 認証においてチャレンジを複数に分割した際の端末認証のシーケンス図を示す．端末は従来方式と同じ手順であらかじめ公開鍵 $\boldsymbol{v}$，秘密鍵 $\boldsymbol{s}$，n を生成し，id を公開鍵 $\boldsymbol{v}$ のハッシュ値として設定しておく．ネットワークへの参加時もしくは通信開始時に，端末は自身の正当性を通信相手に証明する．d を分割するチャレンジ数とし，ラウンド数を j（$1 \leq j \leq d$）とする．まず公開鍵 $\boldsymbol{v}$ と認証要求としてコミットメント x_j を，

$$x_j = (-1)^{b_j} {r_j}^2 (\mathrm{mod} n), \tag{13}$$

として算出し，通信相手に送信する．$r_j \in [0, n)$ および $b_j = \{0, 1\}$ を無作為に選択し，それぞれ秘密にする．

通信相手は端末からコミットメント x_j を受け取ると，以下の式よりチャレンジ $\boldsymbol{e}_j$ を生成し，端末に送信する．

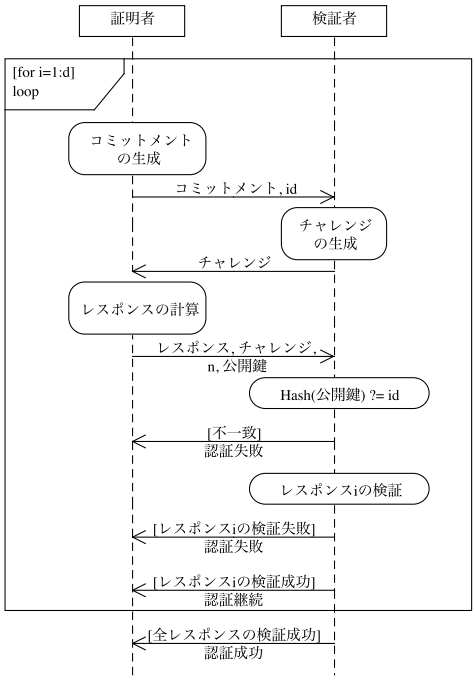

図 2 FFS 認証においてチャレンジを複数に分割した際の端末認証のシーケンス図

Fig. 2 Flow chart of FFS identification protocol with divided challenges.

$$\boldsymbol{e}_j = (e_{1j}, \cdots, e_{kj}). \tag{14}$$

ただし，$e_{ij} = \{0, 1\}$ であり，$e_{ij} = 1$ としてよい上限値を $u_j, (u_j \in [1, k))$ として生成する．

端末はチャレンジを受け取ると，レスポンス y_j を，

$$y_j = r_j \prod_{i=1}^{k} s_i^{e_{ij}} (\mathrm{mod} n), \tag{15}$$

として生成し，公開鍵 $\boldsymbol{v}$ および n とともに通信相手に送信する．

通信相手はレスポンス y_j，チャレンジ $\boldsymbol{e}$，公開鍵 $\boldsymbol{v}$ および n を受け取ると，まず公開鍵 $\boldsymbol{v}$ に対してハッシュ値を算出し，端末の id に一致するかを確認する．一致しない場合，認証失敗となる．一方，一致した場合は従来方式と同様にチャレンジ $\boldsymbol{e}_j$ に対するレスポンス y_j を以下の式を用いて検証を行う．

$$z_j = y_j^2 \prod_{i=1}^{k} v_i^{e_{ij}} (\mathrm{mod} n). \tag{16}$$

$z_j \neq \pm x_j$ または $z_j = 0$ の場合，通信相手は端末が主張する公開鍵 $\boldsymbol{v}$ に対応する秘密鍵 $\boldsymbol{s}$ を保持していないと見なし，該当のチャレンジを破棄したうえで端末に認証失敗を返す．以後このチャレンジに対するレスポンスを受信しても，検証を行わずに破棄する．一方，$j < d$ かつ $z_j = \pm x_j$ かつ $z_j \neq 0$ の場合，端末に認証継続を通知し，$j = d$ ラウンド目が終了した時点までに認証が失敗していないならば，端末に認証成功を通知する．

4.1 提案方式の安全性

本方式は従来の FFS 認証に対してチャレンジの選択方法に制限を加えると同時に複数のチャレンジに返答するため，安全性を確認する必要がある．本節では，FFS 認証の安全性証明に則り [10], [13], [14]，本方式が完全性，健全性およびゼロ知識性を有するプロトコルであることを示す．

4.1.1 完全性

Lemma 4.1. 正しい証明者は必ず検証者を納得させることができる．

Proof. j に関して一般性を失わず，検証者が検証する z_j は以下の等式を満たす．ただし，定義より $s_i^2 v_i = \pm 1 (\mathrm{mod}\, n)$ の関係を用いる．

$$\begin{aligned} z_j = y_j^2 \prod_{i=1}^{k} v_i^{e_{ij}} (\mathrm{mod} n) &= \left(r_j \prod_{i=1}^{k} s_i^{e_{ij}} \right)^2 \cdot \prod_{i=1}^{k} v_i^{e_{ij}} \\ &= r_j^2 \prod_{i=1}^{k} (s_i^2 v_i)^{e_{ij}} \\ &= \pm r_j^2 = \pm x_j (\mathrm{mod} n). \end{aligned}$$

□

4.1.2 健全性

Lemma 4.2. $\prod_{i=1}^{k} v_i^{c_{ij}} (\mathrm{mod}\, n)$ ($c_{ij} = -1, 0, 1$，ただしすべて 0 となる場合は除く）の任意の積の平方根を多項式時間で計算できず，かつ秘密 s_i を知らない証明者を想定した場合，プロトコルに対して正しく振る舞う検証者および任意の多項式時間の計算が可能な証明者に対して，検証者が証明者を受理する確率 p は $p < 2^{-d\bar{u}}$ である．ただし，$d = \Theta(|n|)$，$\bar{u}$ は各チャレンジの上限値 u_j の $1 \leq j \leq d$ に対する平均値であり，$\bar{u} = \mathcal{O}(\log |n|)$ とする．

Proof. （*Sketch*）悪意のある証明者が，秘密鍵を保持しないにもかかわらず検証者に自身が秘密鍵を持つことを納得してもらうには，通信相手の計算する検証に合格するように x_j および y_j を送信する方法が考えられる．そのため，悪意のある端末は，あらかじめ $1 \leq j \leq d$ に対するすべてのチャレンジ $\boldsymbol{e}_j$ を推測しておく．公開鍵 $\boldsymbol{v}$ は容易に取得できるため，この推測に基づき，$\prod_{i=1}^{k} v_i^{e_{ij}} (\mathrm{mod} n)$ の項を計算する．したがって，悪意のある端末は前述の推測が正しいと仮定し，$z_j = \pm x_j$ かつ $z_j \neq 0$ が成立するように x_j および y_j の値を無作為に選ばれた r_j を用いて以下のように計算する．

$$x_j = \pm r_j^2 \prod_{i=1}^{k} v_i^{e_{ij}} (\mathrm{mod} n), \tag{17}$$

$$y_j = r_j. \tag{18}$$

悪意のある端末は前述のチャレンジ $\boldsymbol{e}_j$ の推測が $1 \leq j \leq d$ に対してすべて正しければ，なりすましに成功する．しかし x_j はチャレンジ $\boldsymbol{e}_j$ を検証者から受信する前に送信しなければならないため，悪意のある端末のなりすましが成功する確率 p は，(すべてのチャレンジ $\boldsymbol{e}_j$ の組合せ数)$^{-1}$ と表せる．ここで，あるチャレンジ $\boldsymbol{e}_j$ においてビットを立ててよい上限値 u_j を用いると，あるチャレンジ $\boldsymbol{e}_j$ の組合せ数は，k 個の要素のうち，最低 1 個，最大 u_j 個までビットを立ててよい組合せの数の和で表せる．したがって，あるチャレンジ $\boldsymbol{e}_j$ の推測が正しい確率 p_j は，

$$\begin{aligned} p_j &= \left\{ \binom{k}{1} + \binom{k}{2} + \cdots + \binom{k}{u_j} \right\}^{-1} \\ &= \left\{ \sum_{i=1}^{u_j} \binom{k}{i} \right\}^{-1}, \end{aligned} \tag{19}$$

となる．よって 1 から d までのすべてのチャレンジの推測が正しい確率 p は p_j の積で表せるため，

$$\begin{aligned} p &= p_1 p_2 \cdots p_d \\ &= \left\{ \sum_{i=1}^{u_1} \binom{k}{i} \right\}^{-1} \left\{ \sum_{i=1}^{u_2} \binom{k}{i} \right\}^{-1} \cdots \left\{ \sum_{i=1}^{u_d} \binom{k}{i} \right\}^{-1} \\ &= \left\{ \prod_{j=1}^{d} \sum_{i=1}^{u_j} \binom{k}{i} \right\}^{-1}, \end{aligned} \tag{20}$$

となる．ところで $u_j < k$ であることから，

$$\sum_{i=1}^{u_j}\binom{k}{i} - 2^{u_j} = \sum_{i=1}^{u_j}\binom{k}{i} - \sum_{i=0}^{u_j}\binom{u_j}{i} > 0,$$
$$\Leftrightarrow \left\{\sum_{i=1}^{u_j}\binom{k}{i}\right\}^{-1} < 2^{-u_j}, \tag{21}$$

であり，これは $1 \le j \le d$ に対して成り立つ．したがって，

$$p = \left\{\prod_{j=1}^{d}\sum_{i=1}^{u_j}\binom{k}{i}\right\}^{-1} < 2^{-(u_1+\cdots+u_d)} \tag{22}$$

という関係が成り立つ．ここで $\bar{u} = \frac{1}{d}\sum_{j=1}^{d} u_j$ とすると p は以下の関係を満たす．

$$p < 2^{-d\bar{u}}. \tag{23}$$

したがって，秘密を保持しない証明者が，この方法で検証者に自身が秘密を持つことを納得してもらえる確率 p は $p < 2^{-d\bar{u}}$ である．ここで従来方式と同様に，悪意のある証明者がこの合格率を増大させるためにコミットメント x_j $(1 \le j \le d)$ に対して異なる複数のチャレンジ $\boldsymbol{e}_j$ に合格できるような複数のレスポンス y_j を計算しておく手法を考える．ここでは複数のチャレンジ $\boldsymbol{e}_j$ およびレスポンス y_j のうち任意の 2 つをそれぞれ $(\boldsymbol{e}'_j, \boldsymbol{e}''_j)$, (y'_j, y''_j) と表すことにする．ただし，y_j は非 0 であり，コミットメント x_j はチャレンジを受け取る前に検証者に送信しなければならないことから $x_j = y'^2_j \prod_{i=1}^{k} v_i^{e'_{ij}} (\mathrm{mod} n) = y''^2 \prod_{i=1}^{k} v_i^{e''_{ij}} (\mathrm{mod} n)$ となる．しかしここで正しい証明者は v_i に対する平方根 s_i を知っているため $\left(\frac{y'_j}{y''_j}\right)^2 = \frac{\prod_{i=1}^{k} v_i^{e'_{ij}}}{\prod_{i=1}^{k} v_i^{e''_{ij}}} (\mathrm{mod} n) = \prod_{i=1}^{k} v_i^{c_{ij}} (\mathrm{mod} n)$ の平方根を計算できるはずであるが，この計算には $\prod_{i=1}^{k} v_{ij}^{c_{ij}} (\mathrm{mod} n)$（$c_{ij} = -1, 0, 1$，ただしすべて 0 となる場合は除く）の平方根の計算を含むため，秘密 s_i を知らない証明者の能力では計算できない．したがって，秘密を保持しない証明者は複数のコミットメントとレスポンスを用意できず，確率 p を $2^{-d\bar{u}}$ 以上に大きくすることができない．よって検証者に秘密を持つことを納得してもらえる確率 p は $p < 2^{-d\bar{u}}$ となる．$d = \Theta(|n|)$ および $\bar{u} = \mathcal{O}(\log |n|)$ の定義より，p の大きさは $\mathcal{O}(1/(|n| \cdot 2^{|n|}))$ となり，任意の整数 a に対して $1/|n|^a > 2^{-d\bar{u}}$ を満たすため，秘密を知らない証明者が検証に合格できる確率は無視できる程度に低い確率となる． □

4.1.3 ゼロ知識性

Lemma 4.3. $d = \Theta(|n|), u_j = \mathcal{O}(\log |n|)$ に対して，本プロトコルをシーケンシャル実行した場合，ゼロ知識性を満たす．

Proof. （*Sketch*）本プロトコルが完全ゼロ知識性を証明するため，悪意のある検証者が正当な証明者とのやりとりにおいて得られる情報は，正当な証明者と対話することなく，悪意のある検証者が自分自身で得られる情報と同じであることを示す．ここで正当な証明者を $\bar{A}$，正当な検証者を $\bar{B}$，$\bar{A}$ から秘密を知ろうとする検証者を $\widetilde{B}$ とする．ここで以下のシミュレータ $M_{prop\widetilde{B}}$ を考える．

(1) 以下のステップ (2) から (6) を j（$1 \le j \le d$）に対して順々に行う．
(2) チャレンジ $\boldsymbol{e}'_j$ を推測する．
(3) コミットメント $x_j = \pm r_j^2 \prod_{i=1}^{k} v_i^{e'_{ij}} (\mathrm{mod} n)$ およびレスポンス $y_j = r_j$ を計算する．
(4) 検証者 $\widetilde{B}$ にコミットメント x_j を送り，チャレンジ $\boldsymbol{e}_j$ を受け取る．
(5) 推測したチャレンジ $\boldsymbol{e}'_j$ と受け取ったチャレンジ $\boldsymbol{e}_j$ が異なる場合，(2) からやり直す．
(6) 推測したチャレンジ $\boldsymbol{e}'_j$ と受け取ったチャレンジ $\boldsymbol{e}_j$ が等しい場合，レスポンス r_j を $M_{prop\widetilde{B}}$ に送る．

このようなシミュレータ $M_{prop\widetilde{B}}$ によって得られる系列 $\{\boldsymbol{x} = \{x_j\}, \boldsymbol{y} = \{y_j\}, E = \{\boldsymbol{e}_j\}\}$ は秘密 s_i を知らなくても検証に合格できる系列であり，正当な証明者 $\bar{A}$ との対話によって得られる系列 $\{\boldsymbol{x} = \{x_j\}, \boldsymbol{y} = \{y_j\}, E = \{\boldsymbol{e}_j\}\}$ とまったく同じとなる．

このシミュレータは各 j（$1 \le j \le d$）に対して平均 $2^{u_j} = \mathcal{O}(|n|)$ の試行で実行可能である．また $d = \Theta(|n|)$ であることから，全体としての計算量 $2^{u_1} + 2^{u_2} + \cdots + 2^{u_d}$ は $|n|$ の多項式時間で計算可能となる．よって本プロトコルをシーケンシャル実行した際に検証者が正しい証明者とのやりとりにおいて得る知識は，検証者自身がシミュレータを起動することにより得られることが分かり，ゼロ知識性を満たすといえる． □

4.2 平均乗算回数が低減される条件

提案方式において，レスポンスの検証に必要な平均乗算回数を低減させるには，チャレンジ数 d および各チャレンジにおいてビットを立ててよい上限値 $\boldsymbol{u} = \{u_1, u_2, \cdots, u_d\}$ を適切に定める必要がある．そのために，なるべく値が小さな要素 u_j からなる $\boldsymbol{u}$ を定めればよい．しかしながら，あまりに小さい値からなる $\boldsymbol{u}$ を選ぶと検証に必要な乗算回数を低減させることができるが，従来方式と同等のなりすまし許容率を達成することができない．そこで本節では，従来方式と同等のなりすまし許容率を達成する条件をパラメータ k, d, $\boldsymbol{u}$ を用いて示す．そして従来方式および提案方式における 1 回の検証に必要な平均乗算回数を示したうえで，検証に必要な平均乗算回数を低減できるための条件をパラメータ k, d, $\boldsymbol{u}$ を用いて示す．

まず従来方式と同等のなりすましの許容率を達成する条件をパラメータ k, d, $\boldsymbol{u}$ を用いて示す．3.1.2 項で示したとおり，なりすましの許容率はチャレンジの組合せ数の逆数で表される．そこでなりすましの許容率をチャレンジの組

合せ数で議論する．従来方式におけるチャレンジの組合せ数は 2^k と表される．一方，提案方式におけるチャレンジの組合せ数は式 (20) より $\prod_{j=1}^{d}\sum_{i=1}^{u_j}\binom{k}{i}$ である．したがって従来方式と同等のなりすまし許容率を達成する条件は，

$$\prod_{j=1}^{d}\sum_{i=1}^{u_j}\binom{k}{i} \geq 2^k, \tag{24}$$

となる．

次に従来方式および提案方式における 1 回の検証に必要な平均乗算回数を示したうえで，検証に必要な平均乗算回数を低減させることができるための条件をパラメータ k，d，$\boldsymbol{u}$ を用いて示す．従来方式においてレスポンスの検証に必要な平均乗算回数 E_{conv} を求める．通信相手はコミットメントを受け取ると，移動端末に対してチャレンジ $\boldsymbol{e}=(e_1,\cdots,e_k)$（ただし $e_i=\{0,1\}$）を出題する（式 (8)）．このとき，$e_i=\{0,1\}$ は無作為に選択され，少なくとも 1 つの要素は 1，多くとも k 個の要素が 1 となること，レスポンスの検証式 z（式 (10)）を計算する際にはじめに y を掛けることから，従来方式において通信相手が検証時に行う 1 ラウンドあたりの平均乗算回数は $\frac{k+1}{2}+1$ となる．正当な端末は t ラウンドのすべてのチャレンジに対して正しいレスポンスを返答できるため，通信相手はすべてのレスポンスを検証する．したがって従来方式において正当な端末を検証に必要な平均乗算回数 E_{convN} は，t 回分のレスポンスを検証するため，

$$E_{convN}=t\left(\frac{k+1}{2}+1\right), \tag{25}$$

となる．式 (25) は $t\geq 1$ に対して成り立つ．次に，従来方式において悪意のある端末の検証に必要な平均乗算回数 E_{convA} を求める．ここで，あるチャレンジ e_i の推測が正しい場合のみ，次のチャレンジ e_{i+1} に対するレスポンスが計算されるため，2 問目以降のチャレンジに対するレスポンスが計算されるかは，その直前のチャレンジの推測が当たる確率 (2^{-k}) に依存する．そして各レスポンスの検証には平均 $\frac{k+1}{2}+1$ 回だけ乗算を行うため，E_{convA} は，

$$\begin{aligned} E_{convA} &= \left(1+2^{-k}+\cdots+2^{-k(t-1)}\right)\left(\frac{k+1}{2}+1\right), \\ &\approx \frac{1}{1-2^{-k}}\left(\frac{k+1}{2}+1\right), \end{aligned} \tag{26}$$

と表すことができる．したがって，悪意のある端末がネットワークに存在する割合を $\alpha\in[0,1]$ とすると，従来方式において検証に必要な平均乗算回数 E_{conv} は

$$\begin{aligned} E_{conv} &= \alpha E_{convA}+(1-\alpha)E_{convN}, \\ &\approx \{t+\alpha(1-t)\}\left(\frac{k+1}{2}+1\right), \end{aligned} \tag{27}$$

と表すことができる．次に提案方式におけるレスポンスの検証に必要な平均乗算回数 E_{prop} を求める．各チャレンジ $\boldsymbol{e}_j$ において $e_i=\{0,1\}$ は無作為に選択され，少なくとも 1 つの要素は 1，多くとも u_j 個の要素が 1 となること，各レスポンスの検証式 z_j（式 (16)）を計算する際にはじめに y_j を掛けることから，各チャレンジ $\boldsymbol{e}_j$ に対する平均乗算回数 M_j は，

$$M_j=\frac{u_j+1}{2}+1, \tag{28}$$

となる．提案方式では，正当な端末および正当な端末になりすましを試みる端末では，以下に示すように平均乗算回数が異なる．正当な端末はすべてのチャレンジに対して正しいレスポンスを返答できるため，通信相手はすべてのレスポンスを検証する．したがって正当な端末を検証に必要な平均乗算回数 E_{propN} は，$1\leq j\leq d$ に対する M_j の和で表され，

$$\begin{aligned} E_{propN} &= M_1+M_2+\cdots+M_d \\ &= \sum_{j=1}^{d}M_j=\sum_{j=1}^{d}\left(\frac{u_j+1}{2}+1\right), \end{aligned} \tag{29}$$

となる．一方，なりすましを試みる端末は 4.1.2 項で述べたとおり，各チャレンジを推測する．ここで，あるチャレンジ e_i の推測が正しい場合のみ，次のチャレンジ e_{i+1} に対するレスポンスが計算されるため，2 問目以降のチャレンジに対するレスポンスが計算されるかは，その直前のチャレンジの推測が当たる確率によって決まる．したがって，提案方式では，なりすましを試みる端末を検証する際に必要な平均乗算回数 E_{propA} は，式 (19) および式 (28) を用いて期待値として求められ，

$$\begin{aligned} E_{propA} &= M_1+p_1M_2+p_1p_2M_3+\cdots+\left(\prod_{j=1}^{d-1}p_j\right)M_d \\ &= M_1+\sum_{l=2}^{d}\left(\prod_{j=1}^{l-1}p_j\right)M_l \\ &= \left(\frac{u_1+1}{2}+1\right)+\sum_{l=2}^{d}\left(\prod_{j=1}^{l-1}p_j\right)\left(\frac{u_l+1}{2}+1\right), \end{aligned} \tag{30}$$

となる．よって，提案方式における検証に必要な平均乗算回数はそのネットワークにどの程度悪意のある端末（なりすましを試みる端末）が存在するかによって変化する．そこで悪意のある端末がネットワークに存在する割合を $\alpha\in[0,1]$ とすると，提案方式における検証に必要な平均乗算回数 E_{prop} は

$$\begin{aligned} E_{prop} &= \alpha E_{propA}+(1-\alpha)E_{propN} \\ &= \alpha\left\{M_1+\sum_{l=2}^{d}\left(\prod_{j=1}^{l-1}p_j\right)M_l\right\}+(1-\alpha)\sum_{j=1}^{d}M_j \\ &= M_1+\sum_{l=2}^{d}\left\{1-\alpha\left(1-\prod_{j=1}^{l-1}p_j\right)\right\}M_l \\ &= \frac{u_1+1}{2}+1 \end{aligned} \tag{31}$$

$$+\sum_{l=2}^{d}\left\{1-\alpha\left(1-\prod_{j=1}^{l-1}p_j\right)\right\}\left(\frac{u_l+1}{2}+1\right),$$

となる．提案方式によって検証にともなう平均乗算回数が低減される条件を求めるため，E_{conv} と E_{prop} を比較すると，

$$\begin{aligned}&E_{conv}-E_{prop}\\&=\{t+\alpha(1-t)\}\left(\frac{k+1}{2}+1\right)-\frac{u_1+1}{2}-1\\&\quad-\sum_{l=2}^{d}\left\{1-\alpha\left(1-\prod_{j=1}^{l-1}p_j\right)\right\}\left(\frac{u_l+1}{2}+1\right),\end{aligned}\tag{32}$$

となる．したがって $E_{prop}<E_{conv}$ となる条件は，$t\geq 1$，$0\leq\alpha\leq 1$ より $t+\alpha(1-t)>0$ であることに注意すると以下で求められる．

$$\begin{aligned}&E_{conv}-E_{prop}>0\\&\Leftrightarrow\frac{(u_l+3)P+u_1+3}{t+\alpha(1-t)}-3<k,\end{aligned}\tag{33}$$

ただし，$P=\sum_{l=2}^{d}\left\{1-\alpha\left(1-\prod_{j=1}^{l-1}p_j\right)\right\}$ とする．

4.3 パラメータの決定

本節では所望するなりすましの許容率を達成し，かつレスポンスの検証に必要な平均乗算回数 E_{prop} を最も低減させるチャレンジ数 d および上限値 $\boldsymbol{u}=(u_1,u_2,\cdots,u_d)$ を求める．そのうえでまず k を固定する必要があるが，ここでは文献 [8] に示されているとおり，従来方式における悪意のある端末が検証に合格する確率が $2^{-20}\approx 9.5*10^{-7}$ となる $k=20$ とする．また提案方式では，式 (31) より，なりすましを試みる端末がどの程度存在するかを表す α によって通信相手が検証にともなう平均乗算回数が決まる．そこで，E_{prop} を α に関して平均化した $\bar{E}_{prop}$ を，

$$\bar{E}_{prop}=\int_0^1 E_{prop}d\alpha,\tag{34}$$

とし，式 (24) を満たしたうえで，$\bar{E}_{prop}$ を最も低減させる d および $\boldsymbol{u}$ を，コンピュータによる数値計算を用いて求める．

まず $d=2$ の場合を考える．**表 1** に $d=2$ における $\boldsymbol{u}=(u_1,u_2)$ の候補，なりすましを許容する確率 p，1 回の検証に必要な平均乗算回数 $\bar{E}_{prop}$ を示す．また**図 3** に α を変化させた場合の平均乗算回数 E を示す．ここで E は E_{conv} および E_{prop} を含めた表現とする．図 3 において，Proposed. $(1,6)$ および Proposed. $(2,4)$ などは $\boldsymbol{u}=(u_1,u_2)$ を，Conventional. $(k=20,t=1)$ は鍵の数 $k=20$ およびラウンド数 $t=1$ の場合の従来方式をそれぞれ示す．ただし，1 つのグラフにすべての候補を載せることはできず，また検証に必要な乗算回数を低減させることが目的であることから，ここでは比較のため，そのうち最も低減させる候補を 6 つ載せる．図 3 より，$d=2$ の場合 $(1,6)$ もしくは $(2,4)$ とした場合に検証に必要な乗算回数を最も低減させる候補となることが分かる．また表 1 より，$(1,6)$ もしくは $(2,4)$ とした場合になりすまし許容率はそれぞれ約 8.3×10^{-7}，約 7.7×10^{-7} であることから，いずれも従来方式における $2^{-20}\approx 9.5\times10^{-7}$ よりも小さいため，所望のなりすまし許容率を達成していることが分かる．そこで $\bar{E}_{prop}$ を比較する．表 1 より $(1,6)$ の場合 $\bar{E}_{prop}=4.36$，$(2,4)$ の場合 $\bar{E}_{prop}=4.26$ となる．したがって $d=2$ では $(2,4)$ が所望するなりすましの許容率を達成し，かつレスポンスの検証に必要な平均乗算回数 E_{prop} を最も低減させる上限値の組合せとなる．また，図 3 より，従来方式 Conventional. $(k=20,t=1)$ および Conventional. $(k=10,t=2)$ に対して，提案方式における検証 1 回あたりの平均乗算回数はそれぞれ $\alpha=0$ のとき約 48%，54%，$\alpha=1$ のとき約 78%，61%だけ低減させることができることが分かる．

同様にしてコンピュータによる数値計算を用いて $d=3,4,5$ を求める．**表 2** に $d=3,4,5$ に対する $\boldsymbol{u}$，p および $\bar{E}_{prop}$ を示す．そして**図 4** に α に対して，提案方式において $d=2,3,4,5$ とした場合と従来方式においてラウンド数 $t=1,2,4,5$ とした場合に検証に必要な平均乗算回

表 1 $d=2$ における $\boldsymbol{u}$，p，$\bar{E}_{prop}$
Table 1 $\boldsymbol{u}$, p $(d=2)$, and $\bar{E}_{prop}$.

$\boldsymbol{u}$	p	$\bar{E}_{prop}$
$(1,6)$	$8.3*10^{-7}$	4.36
$(2,4)$	$7.7*10^{-7}$	4.26
$(3,3)$	$5.5*10^{-7}$	4.50
$(4,2)$	$7.7*10^{-7}$	4.75
$(5,2)$	$2.2*10^{-7}$	5.25
$(6,1)$	$8.3*10^{-7}$	5.50

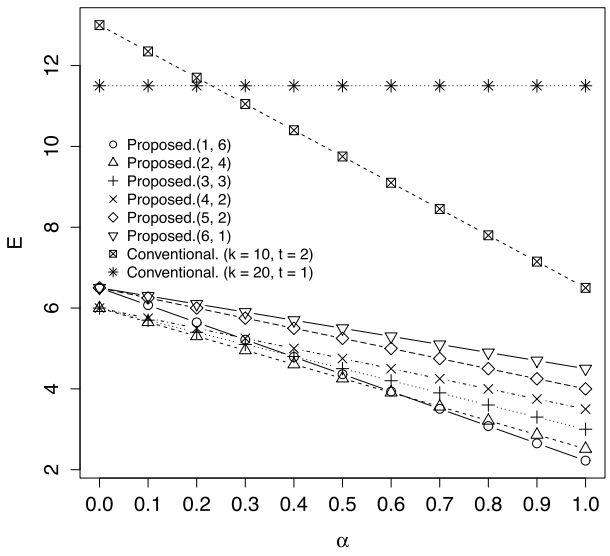

図 3 $d=2$ において α に対する検証に必要な平均乗算回数 E
Fig. 3 Average # of multiplication E versus α $(d=2)$.

表 2 $d = 3, 4, 5$ に対する $\boldsymbol{u}$, p, $\bar{E}_{prop}$
Table 2 $\boldsymbol{u}$, p, and $\bar{E}_{prop}$ ($d \in [3, 5]$).

d	$\boldsymbol{u}$	p	$\bar{E}_{prop}$
3	$(1, 1, 4)$	$4.0 * 10^{-7}$	4.80
4	$(1, 1, 1, 2)$	$6.0 * 10^{-7}$	5.30
5	$(1, 1, 1, 1, 1)$	$3.1 * 10^{-7}$	6.05

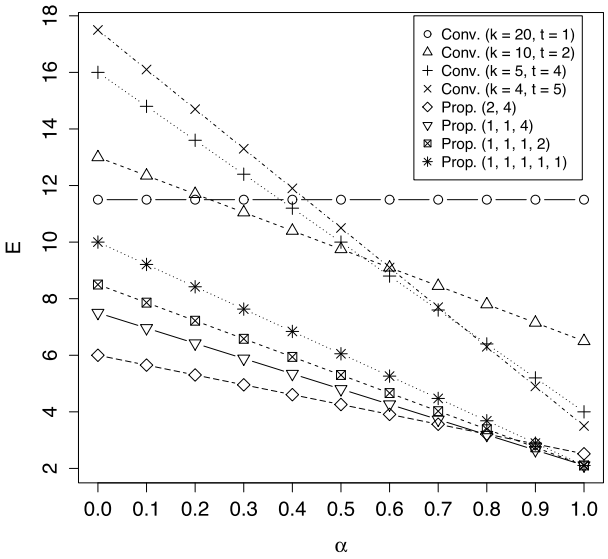

図 4 α に対する検証に必要な平均乗算回数 E
Fig. 4 Average # of multiplication E versus α.

数 E を示す．図 4 より提案方式のいずれの d においても，従来方式よりも平均乗算回数が低減することが分かる．さらに図 4 および表 2 よりチャレンジ数を多くすると平均乗算回数が大きくなることが分かる．これは，各チャレンジにおいて最低 1 つは乗算を行い，また y_j を掛けなければならないため，チャレンジを多くすると分割により得られる平均乗算回数の低減効果は低くなるためである．

以上より，チャレンジを分割し，各チャレンジにおいてビットを立てることができる数に制限を設け，適切に d および $\boldsymbol{u}$ を設定することにより，従来と同程度のなりすましの許容率を達成しつつ，検証に必要な平均乗算回数を低減させることができる．

5. 特性評価

前節で求めた 1 回の検証に必要な平均乗算回数 $\bar{E}_{prop}$ を最も低減させる組合せ $d = 2$ と $\boldsymbol{u} = (2, 4)$ を用いて，検証に必要な計算時間およびメモリ量の評価を行う．本方式がモバイル端末上で用いられることを想定し，1 GHz の CPU，512 MB のメモリを持つ Android 端末上で検証にともなう計算時間を評価する．本実験において取得した計算時間は，検証者がチャレンジに対するレスポンスを貰った後，検証にかかった時間のみである．これは上述のとおり，提案方式はモバイル端末上での検証に必要な計算量を低減させることを目的としているため，提案方式の有効性を確

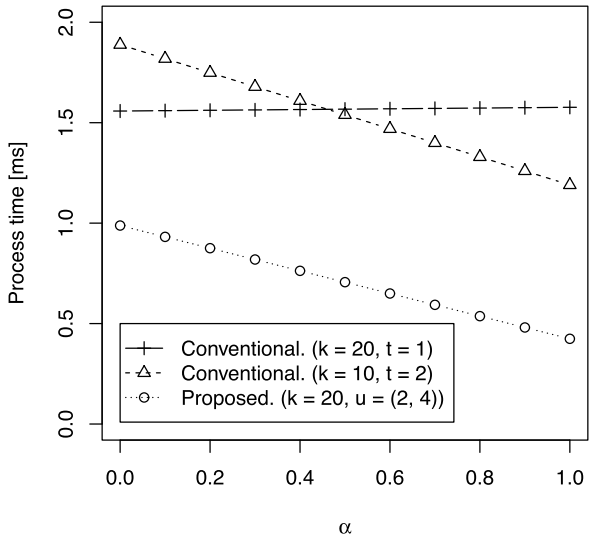

図 5 α に対するレスポンスを検証するのに必要な 1 回あたりの計算時間
Fig. 5 Process time to verify a response versus α.

認するためにはレスポンスの検証にかかった時間のみを評価する必要があると考えたためである．さらに提案方式では，従来方式と比較して検証者および証明者が保持するメモリ量がどの程度増大するかを明らかにする．そして，本方式がモバイル通信をはじめとした遅延の大きいネットワーク上での運用を想定しているため，認証に必要な時間を考えた場合に通信環境は重要な要素となる．そこで本方式の適用先として比較的低速な ZigBee を想定した場合に，提案方式と従来方式でどの程度のコミットメント，チャレンジ，レスポンスといった認証にともなうデータの伝送にどの程度の時間が必要になるかを遅延時間として定義し，評価する．

5.1 検証に必要な計算時間

図 5 に α に対する 1 つのレスポンスを検証するのに必要な計算時間を示す．図 5 より，従来方式 $(k = 20, t = 1)$，従来方式 $(k = 10, t = 2)$ と比較して正当な端末から通信相手が受け取るレスポンスを検証する計算時間は，それぞれ約 37%，約 48%低減されていることが分かる（$\alpha = 0$）．同様に，従来方式 $(k = 20, t = 1)$，従来方式 $(k = 10, t = 2)$ と比較して悪意のある端末から通信相手が受け取るレスポンスを検証する計算時間はそれぞれ約 73%，約 64%低減されていることが分かる（$\alpha = 1$）．しかしながら，提案方式において，理論的に求めた平均乗算回数の低減率ほどは計算時間が減少しないことが分かる．これは提案方式によってチャレンジを複数にしたことにより，チャレンジを複数生成する，もしくはレスポンスを検証する際に 1 問ごとに合否の判定を行う処理によるものと考えられる．

5.2 検証に必要なメモリ量

次に，従来方式および提案方式に必要となるメモリ量を

比較する．従来方式では，1 回の認証に，t 個のコミットメント x，チャレンジ $\boldsymbol{e}$，レスポンス y，および証明者の公開鍵 $\boldsymbol{v}$ と n が必要であるため，従来方式で必要なメモリ量 M_{conv} は，

$$\begin{aligned} M_{conv} &= t|n| + tk + t|n| + k|n| + |n| \\ &= (k + 2t + 1)|n| + tk, \end{aligned} \tag{35}$$

となり，$|n| = 1{,}024\,\text{bits}$ であるため，$k = 20$，$t = 1$ の場合 $M_{conv} = 2{,}964\,\text{Bytes}$，$k = 10$，$t = 2$ の場合，$M_{conv} = 1{,}923\,\text{Bytes}$ となる．一方で，提案方式で必要なメモリ量 M_{prop} も従来方式のそれと同様に求めることができ，

$$\begin{aligned} M_{prop} &= d|n| + dk + d|n| + k|n| + |n| \\ &= (k + 2d + 1)|n| + dk, \end{aligned} \tag{36}$$

となる．ここで $|n| = 1{,}024\,\text{bits}$ であるため，$k = 20$，$d = 2$ の場合 3,240 Bytes となる．提案方式では k と d の一般的な関係を求めるのが困難なため，5.1 節と同様に具体的な k，t，d を用いてメモリの増加量を評価する．提案方式 $(k = 20, d = 2)$ および従来方式 $(k = 20, t = 1)$ として k を揃えて比較した場合，$\frac{3240-2964}{2964} \approx 9\%$だけ必要となるメモリ量が増大する．一方で提案方式 $(k = 20, d = 2)$ および従来方式 $(k = 10, t = 2)$ として分割数 d とラウンド数 t を揃えた場合，$\frac{3240-1923}{1923} \approx 68\%$だけ必要となるメモリ量が増大する．

5.3 認証に必要なデータ伝送による遅延時間

本方式はモバイル通信などの遅延の大きいネットワーク上での運用を想定しているため，認証に必要な時間を考えた場合，通信環境は重要な要素となる．そこでコミットメント，チャレンジ，レスポンスといった認証にともなうデータの伝送にどの程度の時間が必要になるかを遅延時間と定義し，従来方式と提案方式でこの遅延時間がどの程度になるかを評価する．ここでの遅延時間には証明者のコミットメントおよびレスポンスの生成，検証者のチャレンジの生成およびレスポンスの検証にかかる時間を含めていないが，これは 5.1 節ですでに評価しているためである．提案方式と従来方式を比較すると，証明者と検証者において送受信されるデータ量が異なり，これが遅延時間に影響する．このデータ量は 5.2 節で示した検証に必要なメモリ量より概算することができ，従来方式 $(k = 20, t = 1)$ の場合 2.964 KBytes，従来方式 $(k = 10, t = 2)$ の場合 1.923 KB，提案方式 $(k = 20, d = 2)$ の場合 3.240 KBytes となる．例として本方式の適用先として比較的低速な ZigBee 規格（250 kbps）を想定した場合，従来方式 $(k = 20, t = 1)$ と提案方式 $(k = 20, d = 2)$ の遅延時間の差は $(3.240\,\text{KBytes} - 2.964\,\text{KBytes}) \times 8\,\text{bits/Byte} / 250\,\text{kbps} = 8.9\,\text{ms}$ となる．また従来方式 $(k = 10, t = 2)$ と提案方式 $(k = 20, d = 2)$ の遅延時間の差は $(3.240\,\text{KBytes} - 1.923\,\text{KBytes}) \times 8\,\text{bits/Byte} / 250\,\text{kbps} = 42\,\text{ms}$ となる．

6. 結論

MIPv6，アドホック・ネットワークおよび P2P といったネットワークにおいて，第三者による認証機関を利用せず，低演算量で検証可能な端末認証技術として FFS 認証があった．しかし，FFS 認証においては通信相手がレスポンスの検証を行うため，モバイル環境を考慮した場合，検証に必要な平均乗算回数を低減させる必要があった．そこで本論文ではチャレンジを複数に分割し，また各チャレンジにおいてビットを立ててよい数に上限値を設けることにより，本来の目的であるなりすましの許容率を維持しつつ，検証に必要な演算量を低減させる方式を提案した．チャレンジを分割することにより，秘密を知らない悪意のある端末の検証を早期に終了させるだけでなく，各チャレンジにおいてビットを立てる上限値を小さくしたとしても所望のチャレンジの組合せ数を得られるため正当な端末の検証に必要な平均乗算回数も低減させることができることを示した．FFS 認証がモバイル端末で用いられることを考慮し，Android 端末を用いて検証に必要な計算時間を評価し，従来方式と比較して，正当な端末になりすました端末からのレスポンスの検証に必要な演算量を最高で約 73%，正当な端末のそれを最低で約 37%低減させることができることを示した．

謝辞 本研究の一部は，「科研費 基盤研究（C）23560465 高効率セキュアアドホックネットワークに関する研究」の助成により行われた．関係者各位に深謝する．

参考文献

[1] Lu, L., Han, J., Liu, Y., Hu, L., Huai, J., Ni, L. and Ma, J.: Pseudo Trust: Zero-knowledge Authentication in Anonymous P2Ps, *IEEE Trans. Parallel and Distributed Systems*, Vol.19, No.10, pp.1325–1337 (2008).

[2] Huang, Q., Jao, D. and Wang, H.: Applications of Secure Electronic Voting to Automated Privacy-preserving Troubleshooting, *Proc. 12th ACM Conference on Computer and Communications Security*, pp.68–80 (2005).

[3] Lai-Cheng, C.: Heightening Security of P2P Networks by Neighborhood Key Method, *ICINIS'08, 1st International Conference on Intelligent Networks and Intelligent Systems*, pp.201–204 (2008).

[4] Anshul, D. and Roy, S.: A ZKP-based Identification Scheme for Base Nodes in Wireless Sensor Networks, *Proc. 2005 ACM Symposium on Applied Computing*, pp.319–323 (2005).

[5] Udgata, S., Mubeen, A. and Sabat, S.: Wireless Sensor Network Security Model Using Zero Knowledge Protocol, *2011 IEEE International Conference on Communications (ICC)*, pp.1–5 (online), DOI: 10.1109/icc.2011.5963368 (2011).

[6] Hashim, M., Santhosh Kumar, G. and Sreekumar, A.: Authentication in Wireless Sensor Networks Using Zero Knowledge Protocol, *Proc. Computer Networks and Intelligent Computing: 5th International Conference on*

Information Processing, ICIP 2011, Bangalore, India, Vol.157, p.416 (2011).
[7] Chatzigiannakis, I., Pyrgelis, A., Spirakis, P. and Stamatiou, Y.: Elliptic Curve Based Zero Knowledge Proofs and Their Applicability on Resource Constrained Devices, *2011 IEEE 8th International Conference on Mobile Adhoc and Sensor Systems* (*MASS*), pp.715–720 (2011).
[8] Le, F. and Faccin, S.M.: IPv6 Address Ownership Solution Based on Zero-knowledge Identification Protocols or Based on One Time Password (2009).
[9] Toyoda, K., Kamiguchi, Y., Inoue, S. and Sasase, I.: Efficient Solution to Decrease the Effect of DoS Attack against IP Address Ownership Proof in Mobile IPv6, *IEEE 22nd International Symposium on Personal Indoor and Mobile Radio Communications* (*PIMRC*), pp.1223–1227 (2011).
[10] Feige, U., Fiat, A. and Shamir, A.: Zero-knowledge proofs of identity, *Journal of Cryptology*, Vol.1, No.2, pp.77–94 (1988).
[11] Kizza, J.: Feige-Fiat-Shamir ZKP Scheme Revisited, *International Journal of Computing and ICT Research*, Vol.4, No.1, pp.9–19 (2010).
[12] Blum, M.: Coin Flipping by Telephone a Protocol for Solving Impossible Problems, *ACM SIGACT News*, Vol.15, No.1, pp.23–27 (1983).
[13] Fiat, A. and Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems, *Proc. Advances in Cryptology – CRYPTO '86*, pp.186–194, Springer-Verlag (1987).
[14] 情報理論とその応用学会：暗号と認証，培風館 (1996).

豊田 健太郎 （学生会員）

平成 23 年慶應義塾大学理工学部情報工学科卒業．平成 25 年同大学大学院理工学研究科修了，同年同大学院理工学研究科助教（有期・研究奨励），同大学院博士課程在学中．セキュリティ&プライバシに関する研究に従事．平成 23 年度電子情報通信学会通信方式研究会奨励賞受賞．IEEE，電子情報通信学会各会員．

笹瀬 巌 （正会員）

昭和 54 年慶應義塾大学工学部電気工学科卒業．昭和 59 年同大学大学院博士課程修了，同年オタワ大学工学部電気・ポストドクトラルフェロー，昭和 60 年同大学講師，昭和 61 年慶應義塾大学理工学部電気工学科助手，昭和 63 年同大学専任講師，平成 4 年同大学助教授，平成 11 年同大学理工学部情報工学科教授，現在に至る．主として，ディジタル通信，通信ネットワーク，光通信理論，マイクロ波通信，非線形通信システム，通信理論，符号理論に関する研究に従事し，これまで原著論文 269 編，国際会議論文 395 編を発表．工学博士．昭和 59 年度 IEEE ComSoc 学生論文賞，昭和 62 年第 3 回井上研究奨励賞受賞．昭和 63 年第 1 回安藤博記学術奨励賞，昭和 63 年篠原記念学術奨励賞，平成 8 年度本会交換システム研究会優秀論文賞受賞．現在 IEEE ComSoc Board of Governors（Member-at-Large.），IEEE ComSoc Tokyo Chapter Chair，電子情報通信学会通信ソサイエティ会長を務める．これまで，2004～2005 年 IEEE ComSoc Asia Pacific Director，2000～2002 年 IEEE ComSoc Satellite and Space Technical Committee Chair，2004～2006 年本会通信ソサイエティ副会長，2004～2006 年ネットワークシステム研究専門委員長，2002～2004 年通信方式研究専門委員長等を歴任．IEEE SeniorMember．