IP 電話を想定したリアルタイム性を持つストリーム認証方式

上田 真太郎[†] 江藤 秀一[†] 川口 信隆[†]
宇田 隆哉[††] 重野 寛[†] 岡田 謙一[†]

本論文では IP 電話を想定したリアルタイム性を持つストリーム認証方式を提案する．本方式は，公開鍵署名とハッシュを併用し，ある間隔のパケットごとに公開鍵署名を用い，その間隔を通話品質の変化に応じて動的に変化させる．これにより，リアルタイム性を重視したストリーミング転送時の際，継続的に検証可能で効率的に電子署名を行うことが可能となる．既存の方式と比較し，送信側遅延・受信側遅延とパケットロスの観点から，リアルタイムなストリーム転送の認証における本方式の有効性を示す．

Real-time Stream Authentication Scheme for IP Telephony

Shintaro Ueda,[†] Shyuichi Eto,[†] Nobutaka Kawaguchi,[†]
Ryuya Uda,[††] Hiroshi Shigeno[†] and Ken'ichi Okada[†]

In this paper we propose a streaming authentication scheme for IP telephony. It uses both digital signatures and hashes. To clear the strict real-time interaction requirements of IP telephony, the latency and interval between signatures are changed dynamically according to the transmission quality of the network. This provides efficient signing and continuous authentication during real-time streaming. We show our scheme's advantages over previously proposed schemes by comparing sender buffer size, verification timing and tolerance to packet loss.

1. はじめに

ブロードバンドアクセス網の普及により，IP ネットワークへの常時接続環境が整備されつつある中で，ストリーミング技術を用いる IP 電話が注目されている[3),7)]．しかし，IP 電話は依然データの改ざん，なりすまし，事後否認（しらばくれ問題），いたずら電話といった問題をかかえている．これらの問題を解決する技術として，発信元の正当性の確認，改ざんの有無確認，送信者の事後否認の防止などを行うメッセージ認証技術がある．メッセージ認証には一般的に電子署名が用いられる．現在，電子署名は紙面の文書に対して物理的な印鑑で捺印したものと同等の信頼性を持つことが法的に認められるようになってきている[12)]．

現時点で，安全性が高いといわれている電子署名方式として以下のような方式がある．

- RSA 方式（鍵長が 1,024 bit 以上）
- ESIGN 方式（鍵長が 1,024 bit 以上）
- ECDSA 方式（鍵長が 160 bit 以上）
- DSA 方式（鍵長が 1,024 bit 以上）

これらを用いてストリーミングに署名を行う際に計算負荷を考慮する必要がある．すなわち，すべてのパケットに対して署名を施せば，すべてのパケットに対して認証が可能になるが，強度の高い公開鍵暗号演算は非常に計算負荷が高いため，リアルタイム性を損なう可能性がある．そこで，本論文では強度の高い公開鍵暗号署名とハッシュを併用し，状況に応じて演算負荷を動的に減少することによって，リアルタイムなインタラクションが厳しく要求される用途の際にも使用可能なストリーム認証方式を提案する．本提案は，具体的に IP 電話への適用を想定している．IP 電話上のやりとりに署名を施すことで，信頼性の高い会話もしくは取引ができ，IP 電話がより重要な状況で使用されるようになると考えられる．

以下，2 章では現在提案されているストリーミング認証技術とその問題点について述べ，3 章で IP 電話を想定したリアルタイム性を持つストリーム認証方式を提案し，4 章で本提案方式の評価について述べ，5

† 慶應義塾大学理工学部
Faculty of Science and Technology, Keio University
†† 東京工科大学
Tokyo University of Technology

章を結論とする．

2. 関連研究

ストリーミング転送を行う際に，ストリーム認証を効率化する技術について，現在いくつかの提案がなされている．

Gennaro らの Chain 方式[1]では，各パケットが 1 つ後のパケットのハッシュ値を持ち，最初のパケットのみに署名を施す．よって，この方式では，すべてのパケットが揃わない限り送信側で署名計算が行えないため，リアルタイム送信を行う際には，複数のパケットをブロックに区切って署名を行う必要がある．一般にストリーミング転送はリアルタイム性を重視するため，パケットの再送を行わない UDP を使って送信される．Chain 方式では，パケットロスによって署名が連続しない部分が生じると認証が途切れてしまうため，パケットロスに対する耐性がないことが欠点である．

Wong らの WLtree 方式[8]では複数のパケットから tree 構造[4]をつくり，1 回の署名認証演算で複数のパケットの署名・認証を行うことにより，認証にかかる時間を短縮する．この方式は非常に効率良く署名を行うことができるが，送信時間のパケットバッファリング時間が大きくなり，遅延時間が長くなる．

Golle らの Augmented Chain 方式[2]では，再帰的なハッシュ連鎖を行うことにより，バーストパケットロスに対応する効率的な手法を提案している．しかし，ランダムパケットロスに対応していないという弱点がある．

Perrig らによる EMSS 方式[5]では，各パケットに過去の複数のパケットのハッシュ値を付与し，ハッシュ連鎖を用いる．また複数のパケットをまとめてブロックとし，ブロックの最後のパケットに署名を施すことにより，Chain 方式のような送信側での遅延は生じないが，検証時に受信側でブロックに含まれるパケットの数分だけ遅延が生じる．

田中らの署名方式[6]では，2 階層のハッシュ連鎖を用いて，パケットを一定数バッファリングして処理することで，バーストパケットロスとランダムパケットロスに対応している．しかし，この方式では，検証時の受信側で遅延が発生する．

よって，IP 電話のようにリアルタイムなストリーミング転送を必要とする際に，署名を行う際の送信側遅延と署名の検証を行う際の受信側遅延を抑えつつ，バーストパケットロスやランダムパケットロスが生じた場合でも，データの認証を可能とする技術が必要である．

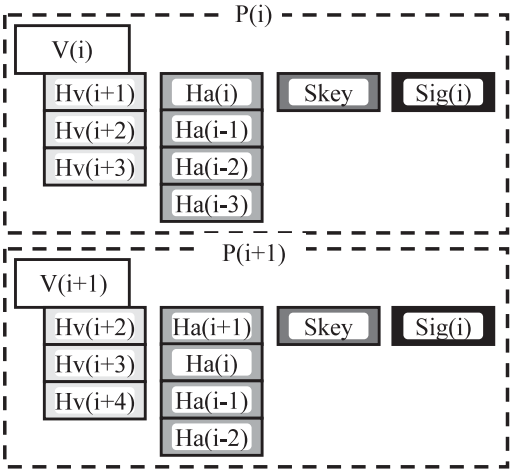

図 1 署名方式
Fig. 1 Signature method.

3. 提　　案

本論文では，リアルタイムなストリーム転送の際に認証を行うストリーム認証方式を提案する．本方式では，公開鍵署名とハッシュを併用する．そして，リアルタイム性を保つため，ある間隔のパケットごとに公開鍵署名を用い，その間隔を通話品質の変化に応じて動的に変化させる．公開鍵署名の間のパケットには高速なハッシュを用いる．

3.1 署名方式

本方式では，パケットにはそのパケットの元々のデータである音声データのほかに，音声データのハッシュ値，音声データと音声データのハッシュのハッシュ値，セッションキー，公開鍵署名が含まれる．ただし，すべてのパケットで公開鍵署名演算を行うのではなく，ある間隔のパケットごとに公開鍵署名演算を行う．この間隔を公開鍵署名間隔（以下署名間隔）と呼び，その間隔に含まれるパケットは同じ公開鍵署名を持つ．

図 1 に本方式のパケットに含まれるデータを示す．パケット P は，音声データ V，音声ハッシュ値 Hv，音声と Hv のハッシュ値 Ha，セッションキー $Skey$，署名 Sig を含む．図中 $P(i)$ は i 番目のパケットであり，公開鍵署名間隔の先頭パケットであるものとする．$V(i)$ は $P(i)$ 中の音声データ，$Hv(i)$ は $V(i)$ のハッシュ値であり，$Ha(i)$ は $V(i)$ と複数の Hv を合わせてハッシュ演算を行った値である．たとえば，図 1 で $P(i)$ には $V(i)$ に続く音声データ $V(i+1)$，$V(i+2)$，$V(i+3)$ に対するハッシュ値 $Hv(i+1)$，$Hv(i+2)$，$Hv(i+3)$ が格納されている．このとき，$Ha(i)$ は，これら 3 つの Hv と音声データ $V(i)$ を連

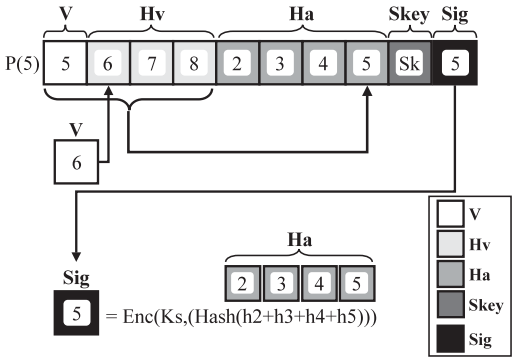

図 2 パケット構成
Fig. 2 Packet architecture.

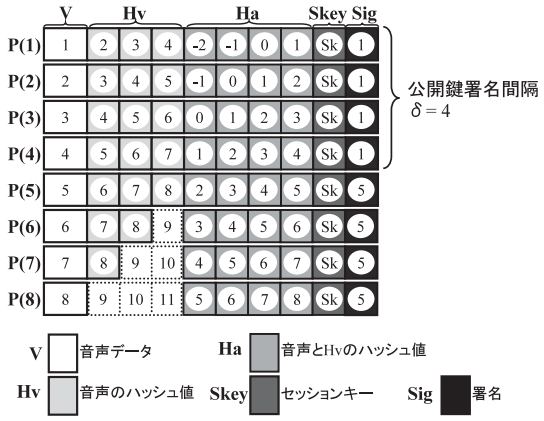

図 3 署名例
Fig. 3 Signature example.

結（concatenation）した値のハッシュ値であり，

$$Ha(i) = Hash\,(V(i) \parallel Hv(i+1) \parallel Hv(i+2) \parallel Hv(i+3)) \quad (1)$$

となる．

各パケットに付与する Hv の数と Ha の数は一定ではなく，署名間隔に応じて動的に変化する．図 1 は Hv が 3 個，Ha が 4 個とした例である．

$Skey$ は，セッションキーであり，3.7 節で詳細について述べる．

Sig は，パケットの電子署名であり，具体的には $Sig(i)$ は i 番目のパケットに付与する Ha の署名である．たとえば，図 1 で $P(i)$ には $Ha(i)$，$Ha(i-1)$，$Ha(i-2)$，$Ha(i-3)$ の 4 個の Ha が含まれている．このとき，$Sig(i)$ は，これら 4 個の Ha を連結した以下に示すハッシュ値に署名をつけたものとなる．

$$Hash(Ha(i) \parallel Ha(i-1) \parallel Ha(i-2) \parallel Ha(i-3))$$

なお，図 1 において，$P(i+1)$ 中の Sig は $i+1$ 番目のものではなく，i 番目のものとなっている．これは $P(i)$ のパケット中の $Sig(i)$ を複写したものである．このようにすべてのパケットで署名の演算を行わないことによって，演算の効率化を図る．以上の V，Hv，Ha，$Skey$，Sig をまとめて，1 つのパケット P の中に格納する．

図 2 は具体例として，$P(5)$ とそれに格納される V，Hv，Ha，$Skey$，Sig を示す．

3.2 公開鍵署名間隔と遅延

本方式では，署名間隔は，端末の演算負荷を軽減するために動的に変化する．

署名間隔を δ パケットとおくと，本方式では Hv 署名数は $\delta - 1$，Ha 署名数は δ となる（後述）．よって，署名 Sig は以下の式で表される．

$$Sig(i) = Enc(K_S, Hash(\sum_{k=[(i-1)/\delta]*\delta+2-\delta}^{[(i-1)/\delta]*\delta+1} Ha(k))) \quad (2)$$

ここで，$Enc(KEY, DATA)$ は，鍵 KEY を使って $DATA$ に暗号化処理を施すことを示す．Ks は，公開鍵暗号の秘密鍵である．$Enc(Ks, DATA)$ は，公開鍵署名を表す．

図 3 は署名間隔 $\delta = 4$ の場合の各パケットの構成を表す．なお，便宜上，パケットは 1 番から開始するものとして記述してある．各パケットに Hv は $\delta - 1 = 3$ 個，Ha は $\delta = 4$ 個付与している．図 3 は 8 番目までの音声データ V が生成された時点での送信側の状態を示しており，この際計算できない Hv は点線で囲まれた四角で表されている．たとえば，6 番目のパケットに $Hv(9)$ などが存在しないのは，$V(9)$ が未生成で $Hash(V(9))$ がこの時点では計算できないためである．この状態では $Hv(9)$ を必要とする $P(6)$ 以降は送信できない．このように，送信側では音声データ V が生成されても，それより先の V のハッシュ値 Hv が計算されるのを待ってからパケットを送出する．この，署名もしくはハッシュを付与するための遅延を送信側遅延 Ds と呼ぶ．Ds は付与する Hv の数だけ待つため，以下のようになる．

$$Ds = \delta - 1 \quad (3)$$

ここで，リアルタイム性を維持するための許容できる遅延を許容遅延 Da とおく．Da は，本方式に関係するパケットの送受信タイミングを遅らせることのできる限界のパケット数であり，本方式に無関係の暗号化や音声の圧縮，パケット化などの遅延，またネットワーク通過時の遅延は除く．つまり，Da が 0 の状態が一番リアルタイム性に優れた通話品質となる．この

とき，受信側遅延を Dr とすると，以下の関係を満たす必要がある．

$$Da \geq Ds + Dr \quad (4)$$

また，ここで図 3 の $P(1)$ がストリームで最初の先頭パケットとした場合，$P(1)$ に付与される Ha を計算するのに最初の音声データのハッシュ値 $Hv(1)$ より前の Hv が必要になる際のパケットの構成について説明する．$P(1)$ が最初のパケットである場合，$Ha(-2)$，$Ha(-1)$，$Ha(0)$ を計算する際に $Hv(-2)$，$Hv(-1)$，$Hv(0)$ が必要となる．しかし，$Hv(-2)$，$Hv(-1)$，$Hv(0)$ は $V(-2)$，$V(-1)$，$V(0)$ が存在しないため計算できない．そこで $Hv(-2)$，$Hv(-1)$，$Hv(0)$ の元となる $V(-2)$，$V(-1)$，$V(0)$ は，$P(1)$ が作成される際に空パケット（無音データ）として作成する．これはストリームの最初でデフォルトで行われる．これにより，$Hv(-2)$，$Hv(-1)$，$Hv(0)$ が計算でき，$Ha(-2)$，$Ha(-1)$，$Ha(0)$ も計算することが可能となる．また $P(8)$ がストリームで最後のパケットとした場合，同様に Ha を計算する際に必要かつ存在しない音声データに対して，空パケットを作成し，それぞれの Hv と Ha を計算しパケットに付与する．

3.3 署名検証とパケットロス

本方式では，音声データの認証方式として Hv による認証と Ha による認証の 2 つの方法を併用する．

1 つ目は Hv による認証である．図 3 で，1 番目のパケット $P(1)$ に着目する．1 番目のパケットに付与する $Sig(1)$ は，$Ha(-2)$，$Ha(-1)$，$Ha(0)$，$Ha(1)$ を連結した値のハッシュ値に対する公開鍵署名である．ここで，$Ha(1)$ は $V(1)$，$Hv(2)$，$Hv(3)$，$Hv(4)$ を連結した値のハッシュ値であるので，$V(1)$ の正当性はむろん，$Hv(2)$，$Hv(3)$，$Hv(4)$ の元となる $V(2)$，$V(3)$，$V(4)$ に関しても，受信時にそのハッシュ値を計算するだけで，正当性の有無を $Sig(1)$ の公開鍵署名によって検証することができる．このように，送信側でパケットを遅延させることによって，受信時にハッシュ値から音声データの検証ができる．この付与する Hv 分だけの遅延が送信側遅延 Ds である．

2 つ目の方法は，Ha による認証である．図 3 で，4 番目のパケット $P(4)$ に着目すると音声データ $V(4)$ は，$P(1)$，$P(2)$，$P(3)$ が持つ $Hv(4)$ によっても認証可能であるが，これらのパケットが受信時に失われていた場合にも，$P(5)$ が持つ $Ha(4)$ によって検証を行うことができる．$P(5)$ では $Ha(4)$ に対して $Sig(5)$ の公開鍵署名が付与しているため，検証時の暗号強度は公開鍵暗号のものと同等である．ただし，$P(5)$ を待って $V(4)$ を検証するためには，受信側で 1 パケット待つ必要が生じる．このような遅延が受信側遅延 Dr である．

以上のように，本提案では各パケット中の音声データ V については，送信側遅延 Ds を許すことによって Hv が，または受信側遅延 Dr を許すことによって Ha でその署名をつねに検証できる．各パケットは Hv，Ha のいずれかによって署名の検証が行われる必要があるので，署名間隔 δ は以下の式で表される．

$$\delta \leqq Ds + Dr \quad (5)$$

3.4 パケットロス時の署名検証の具体例

前節で述べたように，本方式における音声データの認証方法は Hv による認証方法と Ha による認証方法の 2 つを併用する．署名間隔の先頭パケットが正しく受信されている場合，その間のパケットについては，パケットロスのあるなしにかかわらず，Hv によって認証が可能である．パケットロスによって署名間隔の先頭パケットを失ってしまった場合，失われたパケットの前後のパケットに付与された Ha ハッシュを利用して，先頭パケットに付与されている公開鍵署名を検証することにより，継続的な認証を実現する．

以下では，図 3 で示された $P(1)$〜$P(8)$ の転送中に署名間隔の先頭パケットである $P(5)$ が失われた場合について述べる．このとき $V(4)$ については $P(1)$ が持つ $Hv(4)$ によって検証できるが，$V(6)$ については $P(6)$ が持つ $Sig(5)$ は $Ha(2)$，$Ha(3)$，$Ha(4)$，$Ha(5)$ の署名である．ここで，$Ha(3)$〜$Ha(5)$ は $P(6)$ 自身が保持するが，$Ha(2)$ は保持しない．しかし，$Ha(2)$ はすでに認証された $P(4)$ が保持しているため，$Ha(2)$〜$Ha(5)$ がそろい，$Sig(5)$ の検証が行える．これにより，$P(4)$ の $Ha(4)$ が検証されれば，$P(4)$ 中の $Hv(6)$ の正当性も検証可能なため，受信側で継続的な署名の検証が可能である．

各パケットに含まれる情報によって，どのように認証が行われるかを**図 4** に示す．縦はパケット番号，横はそのパケットが認証されることによって得られるハッシュデータの番号を表している．そして，ここで破線矢印は，音声データのみを認証できること，直線矢印は音声データと音声データのハッシュを認証できることを表している．

3.5 Hv，Ha，δ の関係

本方式では Hv，Ha，δ の関係を式 (6) のように設定する．以下，Hv，Ha，δ をこのような関係に設定した理由について述べる．

$$Hv \text{ 署名数} = \delta - 1, \quad Ha \text{ 署名数} = \delta \quad (6)$$

これらの値を決める際には許容遅延とパケットロスが大きく関係する．

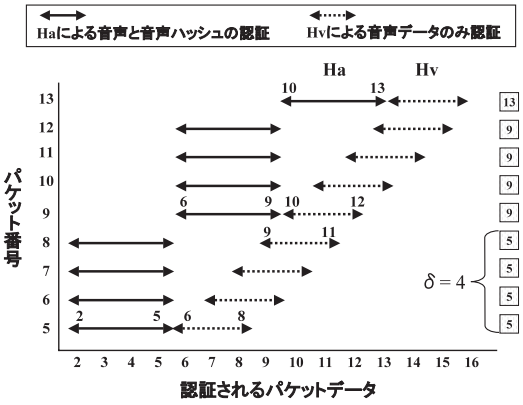

図 4 各パケットによって認証されるハッシュ
Fig. 4 Hashes authenticated by each packet.

許容遅延と Hv の関係では，前述したように送信時にパケットに付与した Hv 分だけ送信側遅延 Ds が生じる．Ds を減らすのに，パケットに付与する Hv を減らすと，署名で認証できる音声データの範囲が狭くなる．たとえば図 3 で，$P(1)$ は $Hv(2)$，$Hv(3)$，$Hv(4)$ の 3 個の Hv を持っている．よって，$Sig(1)$ により $P(4)$ までの音声データを検証することができる．しかし，Hv を 2 個に減らした場合，署名間隔 δ の先頭パケットから署名が認証できる音声データが 2 個に減少する．よって，Hv の数は最低でも署名の最長到達地点までの数と一致する必要がある．逆に，それ以上に Hv を設定すると Ds が増えるので，$Hv = \delta - 1$ が最適となる．

パケットロスと Ha の関係では，パケットロスが起こった場合に Ha を用いて認証を行うため，パケットに付与する Ha を減らすとパケットロスへの耐性が弱くなる．

また Ha を署名間隔 δ より大きい値にすると，冗長なデータが増えることになるため，Ha は δ と同じ値に設定する．よって，$Ha = \delta$ となる．

3.6 署名間隔の動的な変化

パケットロスが少ない場合は署名間隔 δ を大きく設定し，公開鍵署名演算による負荷を軽減する．以下，署名間隔 δ を動的に変化させるアルゴリズムについて述べ，それを**図 5** と**図 6** に示す．

ここでは，任意のパケット間のパケットロス率によって署名間隔を動的に変化させる．まず，署名間隔の初期値を決定するため，ネットワークの遅延，最大パケットサイズ，端末の演算性能から最低の署名間隔 δmin，最大の署名間隔 δmax および許容遅延 Da を決定する．ここで，δ が大きくなると Hv，Ha も増加するため，パケット分割が起こらない限界値で δmax を

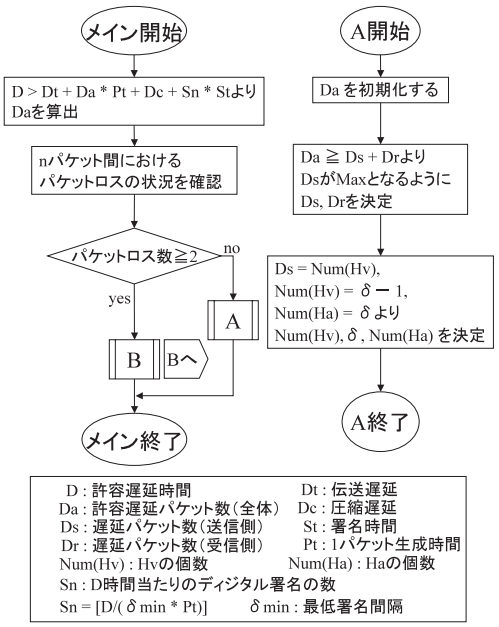

図 5 署名間隔 δ シフトアルゴリズム（1）
Fig. 5 Signature interval δ shifting algorithm (1).

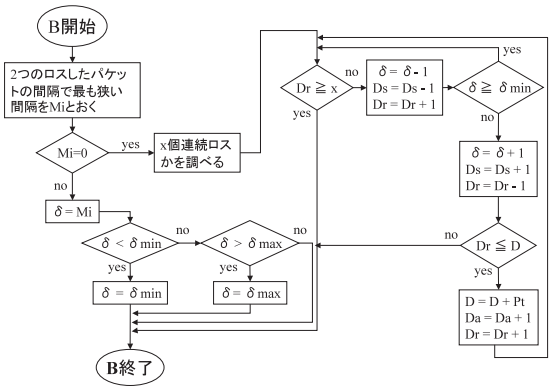

図 6 署名間隔 δ シフトアルゴリズム（2）
Fig. 6 Signature interval δ shifting algorithm (2).

決定する．Da は，希望する遅延時間 D から，ネットワークによる伝送遅延時間，音声のパケット化による遅延時間，コーデックによる音声圧縮時間，署名などの演算時間を差し引き，決定する．そして残りの時間を，1 パケットあたりに格納される音声データの時間で割り，Ds と Dr の和が最大何パケットまで許容されるのかを算出する．

次に，任意の n パケット間（$n = 16$〜20 を想定）で，許容遅延時間内でのパケットロス状況を確認する．ここで，パケットロスが 0 個か 1 個の場合と 2 個以上の場合について述べる．

まず，n パケット間でのパケットロスが 0 個か 1 個

の場合，式 (4) より，Ds が最大値をとるように Ds，Dr を決定する．これは Ds を大きくとった方が公開鍵署名 Sig の間隔を開けることができ，演算効率が上がり，署名時間による遅延時間を短縮できるためである．Ds が決まると，式 (3) により δ が，式 (6) により Hv，Ha が決定する．

次に，n パケット間でのパケットロスが 2 個以上の場合について，バーストロスの場合と，ランダムロスの場合にわけて考える．ランダムロスの場合は署名間隔 δ を小さくして対応する．ロスパケット間の距離が最も短いものの値に δ を指定するが，$\delta\min$ および $\delta\max$ の範囲を超えない値とする．バーストロスの場合は，最大何個連続してパケットが抜けたかを調べ，Dr を連続パケットロスの値まで引き上げる．このとき，$Dr + Ds$ は一定のため，署名間隔 δ が小さくなることになる．ここで δ が $\delta\min$ よりも小さくなる場合は，初期値で設定した通信遅延品質を維持することができなくなるため，通信遅延品質を落として署名を継続させる必要がある．

以上の手法を繰り返すことにより，本提案の認証方式では，署名に対する演算負荷を動的に軽減することが可能となる．動的に変化させるタイミングは，受信側からの応答を待つため，最短で，生成された音声データが署名間隔 δ 分の受信を終えるまでの間隔となる．本提案は，パケットロスが少ない場合には公開鍵署名を可能な限り減らすことにより端末の処理負荷を下げ，パケットロスの増加にともない動的に公開鍵署名頻度を増加させ，継続的な署名付きストリーミングの再生を可能とする．

3.7 安 全 性

本方式では，次のような攻撃に対して以下のようにして耐性を持たせる．

- 再送攻撃

同じ内容のパケットを繰り返し送信することにより会話の内容を改ざんする再送攻撃に耐性を持たせるため，各パケットにシーケンス番号を付与する．これにより，同じシーケンス番号が付与されたパケットを受信した場合，そのパケットが不正なパケットであることが分かる．

- 選択文書攻撃

攻撃者が任意に選択した暗号文とそれに対応した平文が利用できる場合の選択文書攻撃に耐性を持たせるため次のような手法をとる．

毎セッションに異なるセッションキーを作成し，パケットに音声データと認証データとともにセッションキーを付与したものを署名し，それを送信する．セッションキーはセッションの開始時に送信者の秘密鍵で暗号化し相手に送信する．これにより，以前の情報から平文に対応した暗号文を持って，正規の送信者をなりすまして攻撃することを防ぐ．このセッションキーが図 1，図 2，図 3 の $Skey$ である．

4. 評 価

本章では，提案した音声ストリーム認証方式と既存の方式について，比較評価を行う．比較項目は署名処理回数，ハッシュ処理回数，1 パケットあたりに付加される認証情報のオーバヘッドの平均，1 パケットあたりに付加される最大オーバヘッド，送信側のパケットバッファサイズ，検証タイミングである．送信側のパケットバッファサイズと受信側の検証タイミングはそれぞれ署名に必要となる送信側遅延 Ds と署名の検証に必要な受信側遅延 Dr に相当する．ここでは連続する 16 パケット列からなる 1 ブロックを対象とし，ハッシュ値の長さは 20 byte，公開鍵署名長は 40 byte を想定する．認証方式の評価を**表 1** に示す．

4.1 リアルタイム性の評価

まず，リアルタイム性に関しての評価について述べる．表 1 の送信側のパケットバッファサイズに注目すると，本方式は他方式に比べ，小さい値になっていることが分かる．これは，リアルタイム性を保つのに，最も重要である．また，送信側のパケットバッファサイズだけを見ると，Augmented Chains 方式や田中らの方式も比較的小さいが，検証のタイミングを見ると，これらの方式では署名を検証するために受信側で 16 パケット分待たなければならず，全体として大きな遅延を招いている．提案方式では受信側での遅延を最小限に抑えている．

IP 電話の品質クラスは，ITU-T，ETSI の TIPHON および TIA において規格化されており，クラス A では遅延が 100 ミリ秒以内，クラス B では遅延が 150 ミリ秒以内，最低のクラス C でも遅延が 400 ミリ秒以内と定められている[11]．

ここで，実装においてリアルタイム性を確保する際，適用する音声符号化方式は重要なパラメータになる．よって，**表 2** に G.711，G.726，G.728，G.729，G.723.1 それぞれの音声符号化方式を用いた際の各認証方式の合計遅延時間を示す．ここでいう合計遅延時間は，パケットの生成にかかる時間，パケットに署名を施す演算時間，伝送時間，検証を行う演算時間を合わせた値である．この合計遅延時間から本提案の実用性について述べる．また，音声パケットは 100 byte，署名長は 40 byte の ECDSA，伝送速度は 1 Mbps を想

表 1 認証方式の評価
Table 1 Signature method results.

Scheme	Signatures	hash	Overhead average (bytes)	Overhead max (bytes)	Sender packet buffer size (number of packets)	Verification Timing
Augmented Chains	1	16	43	100	5	16
Tanaka	1	16	39	280	7	16
WL star	1	17	340	340	16	1
WL tree	1	21	160	160	16	1
WL tree full	1	31	120	120	16	1
Proposed $\delta = 1$	16	16	60	60	0	1
Proposed $\delta = 2$	8	40	100	100	1	1
Proposed $\delta = 3$	5	37	140	140	2	1
Proposed $\delta = 4$	4	36	180	180	3	1
Proposed $\delta = 5$	3	35	220	220	4	1

表 2 認証方式の合計遅延時間
Table 2 Total delay time.

Scheme	G.711 64 kbps	G.726 32 kbps	G.728 16 kbps	G.729 8 kbps	G.723.1 6.3 kbps
Augmented Chains	133 ms	195 ms	320 ms	570 ms	705 ms
Tanaka	158 ms	245 ms	420 ms	770 ms	959 ms
WL tree full	270 ms	470 ms	870 ms	1670 ms	2100 ms
Proposed $\delta = 1$	82 ms	95 ms	120 ms	170 ms	197 ms
Proposed $\delta = 2$	95 ms	120 ms	170 ms	270 ms	324 ms
Proposed $\delta = 3$	107 ms	145 ms	220 ms	370 ms	451 ms
Proposed $\delta = 4$	120 ms	170 ms	270 ms	470 ms	578 ms
Proposed $\delta = 5$	132 ms	195 ms	320 ms	570 ms	705 ms

定する．Intel 社の CPU pentium 4 2.80 GHz，RAM 2.00 GB を使用した際，40 byte の ECDSA の署名・検証の合計演算時間は 70 ミリ秒である．

WL tree full 方式は署名を行う際に 16 パケット待つ必要があり，G.711 以外の音声符号化方式を使用した際，表 2 より，合計遅延時間がクラス C の 400 ミリ秒以上生じることが分かる．よって，WL tree full は G.711 以外の符号化方式を使用することは困難である．本方式と Augmented Chains 方式では G.711, G.726, G.728 の音声符号化方式を使用した際, IP 電話の規格を満たすことができるが，高圧縮である G.729, G.723.1 を使用すると遅延時間が大きくなるため，使用することができないことが分かる．

本方式はリアルタイム性を保つために最も重要である，署名を行う際に必要とする送信側でのパケットバッファサイズを抑えることにより，合計遅延時間が他方式に比べ最も短い．

ここで，音声符号化方式とパケットロスが起きた際の音声品質の影響について考慮する．パケットロスが起きた際，G.723.1 のような高圧縮方式を用いた場合に比べ，無圧縮の G.711 や低圧縮である G.726 などを用いた場合がより音質劣化は少ない．また，本提案は IP 電話の高音質が求められ，会話に数字を多く含む E-Commerce や行政情報システムのような，より重要な状況での使用を想定しているため，G.711，G.726, G.728 の音声符号化方式を用いることが望ましい．

次に，署名間隔 δ を変化させた際の 1 パケットあたりのオーバヘッドサイズと遅延への影響について述べる．たとえば，署名間隔 δ を 5 に広げた場合，公開鍵署名の処理回数は減少するが，ハッシュの処理回数は増え，1 パケットあたりに付加されるオーバヘッドの平均と 1 パケットあたりに付加される最大オーバヘッドが増える．しかし，ハッシュ計算時間は公開鍵署名の演算に比べてはるかに高速なので，演算時間の観点からはこの増加は問題はないと考えることができる．また，送信側遅延は増えるが，受信側遅延が最小限に抑えられているため，送信側遅延と受信側遅延の両方を足しても，リアルタイムなストリーム転送が可能な値に抑えられていることが分かる．

4.2 パケットロスの評価

次にパケットロスに関しての評価を述べる．本提案は，他の提案同様，ランダムロスとバーストロスに対応している（ただし，Augmented Chains 方式はバーストロスのみにしか対応してない）．

本方式では公開鍵署名演算が施されている公開鍵署名間隔の先頭パケットがランダムロスにより失われても，受信側で継続的な認証が可能である．また，バーストロスが起こった際にも，そのバーストロスの前後に受信したパケットは認証が可能である．

ただし，本方式においても，ある特殊なパターンのパケットロスが発生した場合のみに受信したパケットが認証不可となる．そのパターンとは，あるパケットが署名間隔の先頭パケットではなく，そのパケットを中心として前方の Hv 個以上かつ後方 Ha 個以上のバーストロスが発生した場合である．ただし，実際にIP 電話に本方式を適用することを考えると，署名間隔 $\delta = 4$ の場合，上記の状況は連続 8 パケットのバーストロスに相当し，このよう状況では音声の再生ができないので，パケットの認証ができなくても支障がないと考えられる．

5. おわりに

本論文では，主に IP 電話に使用することを想定し，リアルタイム性を重視したストリーミング転送時に継続的に検証可能で効率的に電子署名を行うストリーム認証方式を提案した．本方式では公開鍵署名とハッシュを併用し，署名間隔ごとに公開鍵署名を用い，公開鍵署名の間のパケットには高速なハッシュを用いた．

既存の方式と比較評価して，署名に必要となる送信側遅延と署名の検証に必要な受信側遅延を抑えられることを示した．これはリアルタイム性を維持するのに最も重要である．また，パケットロスに関しても本提案はランダムロスとバーストロスに対応している．

本提案により IP 電話を用いてお互いの会話内容を公開鍵署名により認証することが可能となる．IP 電話が E-Commerce や行政情報システムのような，より重要な場面で使われる際に，本提案は欠かせない技術であると考えられる．

参考文献

1) Gennaro, R. and Rohatgi, P.: How to Sign Digital Streams, *CRYPTO 1997*, LNCS1294, pp.180–197 (1997).
2) Golle, P. and Modadugu, N.: Authenticating streamed data in the presence of random packet loss, *ECC'99* (1999).
3) Kostas, T.J., Borella, M.S., Sidhu, I., Schuster, G.M., Grabiec, J. and Mahler, J.: Real-Time Voice Over Packet-Switched Networks, *IEEE Network*, January/February, pp.18–27 (1998).
4) Merkle, R.: A Certified Digital Signature, *Proc. Conference on Advances in Cryptology*, pp.218–238 (1989).
5) Perrig, A., Canetti, R., Tygar, J.D. and Song, D.: Efficient Authentication and Signing of Multicast Streams over Lossy Channels, *Proc. IEEE Symposium of Research in Security and Privacy*, pp.56–73 (2000).
6) 田中俊昭，中尾康二，清本晋作：ストリーミング転送における効率的なメッセージ認証方式の検討，第 14 回 CSEC 研究発表会 No.014-003，pp.15–22 (2001).
7) Varshney, U., Snow, A., McGivern, M. and Howard, C.: Voice Over IP, *Comm. ACM*, Vol.45, No.1, pp.89–96 (2002).
8) Wong, C.K. and Lam, S.S.: Digital Signature for Flows and Multicasts, *IEEE/ACM Trans. Networkng*, Vol.7, No.4, pp.502–513 (1999).
9) ITU-T Recommendation G.723.1, Dual Rate Speech Coder for Multimedia Communications Transmitting at 5.3 and 6.3kbit/s (March 1996).
10) ITU-T Recommendation G.729, Coding of speech at 8kbit/s using conjugate-structure algebraic-code-excited linear prediction(CS-ACELP) (March 1996).
11) IP ネットワーク技術に関する研究報告書，総務省 (2002).
12) 電子署名及び認証業務に関する法律，総務省情報通信政策局 (2001).
13) タイムスタンプ付与型音声記録公証システム，ログイット株式会社ニュースリリース．http://www.logit.co.jp/news/release_20021129.

(平成 15 年 5 月 15 日受付)
(平成 15 年 12 月 2 日採録)

上田真太郎（学生会員）

2002 年慶應義塾大学理工学部情報工学科卒業．現在，同大学大学院理工学研究科開放環境科学専攻情報通信メディア工学専修修士課程に在学中．ネットワークセキュリティの研究に従事．

江藤 秀一（学生会員）

2001 年慶應義塾大学理工学部情報工学科卒業．2003 年同大学大学院理工学研究科開放環境科学専攻情報通信メディア工学専攻修了．株式会社フジテレビジョン勤務．

川口 信隆（学生会員）

2003 年慶應義塾大学理工学部情報工学科卒業．現在，同大学大学院理工学研究科開放環境科学専攻情報通信メディア工学専修修士課程に在学中．ネットワークセキュリティの研究に従事．

宇田 隆哉（学生会員）

1998 年慶應義塾大学理工学部計測工学科卒業．2000 年同大学大学院理工学研究科計測工学専攻前期博士課程修了．2002 年同大学院理工学研究科開放環境科学専攻後期博士修了．2003 年 4 月より東京工科大学コンピュータサイエンス学部専任講師．博士（工学）．ネットワークセキュリティの研究に従事．2002 年 IFIP/SEC 2002 Best Student Paper Award 受賞．電子情報通信学会会員．

重野 寛（正会員）

1990 年慶應義塾大学理工学部計測工学科卒業．1997 年同大学大学院理工学研究科博士課程修了．1998 年同大学理工学部情報工学科助手（有期）．現在，同大学理工学部情報工学科助教授．博士（工学）．計算機ネットワーク・プロトコル，モバイル・コンピューティング，マルチメディア・アプリケーション等の研究に従事．情報処理学会マルチメディア通信と分散処理研究会幹事．著書『～ネットワーク・ユーザのための～無線 LAN 技術講座』（ソフト・リサーチ・センター），『コンピュータネットワーク』（オーム社）等．電子情報通信学会，IEEE，ACM 各会員．

岡田 謙一（フェロー）

慶應義塾大学理工学部情報工学科教授，工学博士．専門は，CSCW，グループウェア，コンピュータ・ヒューマン・インタラクション．『ヒューマンコンピュータインタラクション』（オーム社），『コラボレーションとコミュニケーション』（共立出版）をはじめ著書多数．情報処理学会誌編集主査，論文誌編集主査，GW 研究会主査等を歴任．現在，情報処理学会 GN 研究会運営委員，BCC 研究グループ幹事，日本 VR 学会仮想都市研究会副委員長．IEEE，ACM，電子情報通信学会，人工知能学会各会員．1995 年度情報処理学会論文賞，情報処理学会 40 周年記念論文賞，2000 年度情報処理学会論文賞受賞．