# 通信品質に動的に対応するストリーム認証方式の提案

上田 真太郎† 川口 信隆† 重野 寛† 岡田 謙一†

† 慶應義塾大学理工学部

**概要:** 本稿ではリアルタイム性を持つストリーム認証方式を提案する．提案方式は，強度の高い公開鍵署名と高速なハッシュを併用し，ある間隔のパケットごとに公開鍵署名を用いる．署名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，リアルタイム性を厳しく要求する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．既存方式と比較評価をし，リアルタイムなストリーム転送の認証を行う際，本方式の有効性を示す．

# Proposal of a Stream Authentication Scheme Dynamically Responsive to Transmission Quality

Shintaro UEDA†, Nobutaka KAWAGUCHI†, Hiroshi SHIGENO†, and Ken-ichi OKADA†

† Faculty of Science and Engineering, Keio University
{ueda,kawaguti,shigeno,okada}@mos.ics.keio.ac.jp

**Abstract:** In this paper we propose a streaming authentication scheme. It uses both digital signatures and hashes. To clear the strict real-time interaction requirements of real-time streaming, 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 and receiver buffer size, and tolerance to packet loss.

## 1 はじめに

DSL 等のブロードバンドアクセス網の進展や定額料金のサービスの普及により，IP ネットワークへの常時接続環境が整備されつつある。その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 [1][2]．しかし，IP 電話は依然データの改ざん，なりすまし，事後否認といった問題を抱えている．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，送受信者が送受信の事実を否定できない，否認防止等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．

ここで認証を行う際に全てのパケットに対して署名を施せば，全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，本稿では強度の高い公開鍵暗号署名とハッシュを併用することで，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おり，IP 電話上のやり取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，第 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，第 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，第 4 章で本提案方式の評価について述べ，第 5 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 [3] 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，全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 [4] では複数のパケットから tree 構造 [5] をつくり，一回の署名認証演算で複数のパケットの認証・署名を行うことにより，認証にかかる時間を短縮する．この方式は非常に効率よ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．

Perrig らによる EMSS 方式 [6] 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[7]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確認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．

## 3 提案

本稿では，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．ここで，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他に，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，公開鍵署名が含まれる．ただし，全てのパケットで公開鍵署名演算を行わず，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．

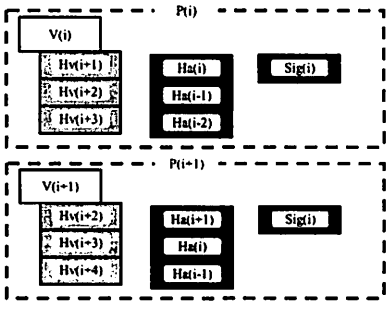


図 1: 署名方式

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 $Ha$，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．例えば，図1で $P(i)$ には $V(i)$ に続く音声データ $V(i+1)$〜$V(i+3)$ に対するハッシュ値 $Hv(i+1)$〜$Hv(i+3)$ 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 $V(i)$ を連結（concatenation）した値のハッシュ値である．

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

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．

$Sig$ 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．例えば，図1で $P(i)$ には $Ha(i)$〜$Ha(i-2)$ の3つの $Ha$ が含まれている．このとき，$Sig(i)$ は，これら3つの $Ha$ 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．

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

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．このように全てのパケットで署名の演算を行わないことによって，演算効率化を図る．以上の $V$，$Hv$，$Ha$，$Sig$ をまとめて，1つのパケット $P$ の中に格納する．

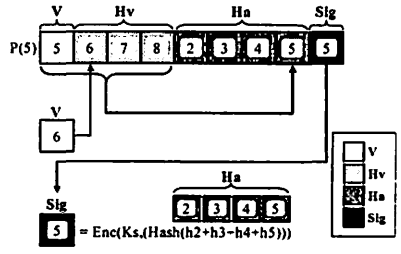


図 2: パケット構成

図2は具体例として，$P(5)$ とそれに格納される $V$，$Hv$，$Ha$，$Sig$ 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 $\delta$ 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 $\delta$ 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．本提案では署名に使用する公開鍵暗号アルゴリズム，鍵長は特定せず，実装によるものとする．

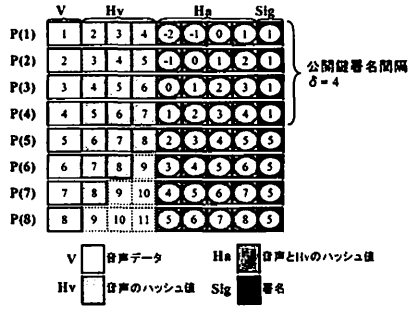


図 3: 署名例

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．例えば，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．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．

$$Ds = \delta - 1 \tag{3}$$

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．

$$Da \geqq Ds + Dr \tag{4}$$

### 3.3 署名確認とパケットロス

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，$Ha(-2)$〜$Ha(1)$ を連結した値に対する公開鍵署名である．ここで，$Ha(1)$ は $V(1)$，$Hv(2)$〜$Hv(4)$ を連結した値のハッシュ値であるので，$V(1)$ の正当性，$Hv(2)$〜$Hv(4)$ の元となる $V(2)$〜$V(4)$ 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正当性が確認できる．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，$P(1)$〜$P(3)$ 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 $Ha(4)$ によって正当性の確認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，確認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 $V(4)$ を確認するためには，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 $Ha$ でその署名を常に確認できる．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 $\delta$ は以下の式で表される．

$$\delta \leqq Ds + Dr \tag{5}$$

### 3.4 パケットロス時の署名確認の具体例

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関わらず，$Hv$ 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 $Hv(4)$ によって確認が取れるが，$V(6)$ については $P(6)$ が持つ $Sig(5)$ は $Ha(2)$〜$Ha(5)$ 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，$Sig(5)$ の確認が行える．これにより，$P(4)$ の $Ha(4)$ の正当性が確認されれば，$P(4)$ 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確認が可能である．

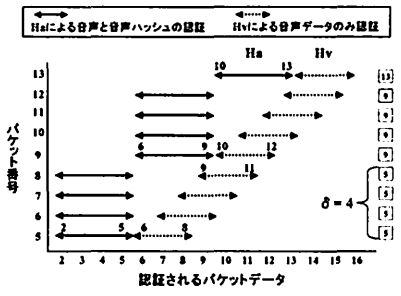


図 4: 各パケットによって認証されるハッシュ

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で図示する．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 $Hv, Ha, \delta$ の関係

本方式では $Hv$，$Ha$，$\delta$ の関係を式 (6) のように設定する．以下，$Hv$，$Ha$，$\delta$ 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．

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．例えば図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 $\delta$ 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 $\delta$ 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 $\delta$ 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 $\delta$ 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 $\delta$ 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 $\delta\ min$，最大の署名間隔 $\delta\ max$ および許容遅延 $Da$ を決定する．ここで，$\delta$ 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 $\delta$ max を決定する．$Da$ 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パケット辺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，$Ds$ が最大値を取るように $Ds$，$Dr$ を決定する．これは $Ds$ を大きく取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 $\delta$ 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，ランダムロスとバースト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 $\delta$ 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 $\delta$ 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 $\delta$ が小さくなることになる．ここで $\delta$ 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 $\delta$ 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伴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．

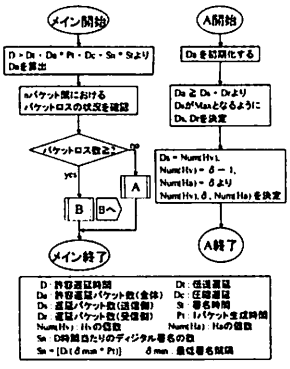


図 5: 署名間隔 $\delta$ シフトアルゴリズム (1)

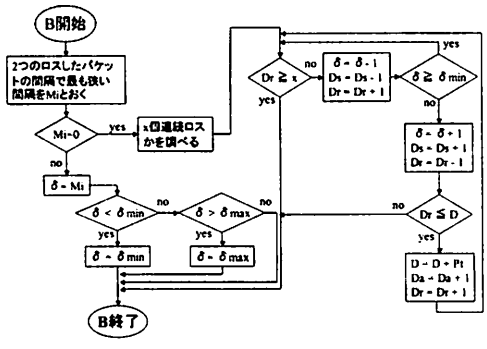


図 6: 署名間隔 $\delta$ シフトアルゴリズム (2)

# 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ーヘッドの平均，1パケットあたりに付加される最大オーバーヘッド，送信側のパケットバッファ量，受信側のパケットバッファ量である．送信側と受信側のパケットバッファ量はそれぞれ署名に必要となる送信側遅延 $Ds$ と署名の確認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バイト，公開鍵署名長は40バイトを想定する．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，少ない値になっていることがわかる．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，田中らの方式も比較的少ないが，受信側パケットバッファサイズを見ると，これらの方式では署名を確認するために受信側で16パケット分待たねばならず，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．

G.723.1 を例にとると，30ミリ秒毎（秒間33パケット）にパケットが送出されるとき，16パケットの遅延は480ミリ秒となる．IP 電話の品質クラスは，ITU-T，ETSI の TIPHON 及び TIA において規格化されており，クラス A では100 ミリ秒以内，クラス B では150 ミリ秒以内，最低のクラス C でも400ミリ秒以内と定められている [8]．従って，認証だけで480ミリ秒の遅延が生じることは，これらの規格を満たすことができない．

署名間隔 $\delta$ を変化させた際の1パケットあたりのオーバーヘッドサイズと遅延への影響について述べる．例えば，署名間隔 $\delta$ 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ーヘッドの平均と1パケットあたりに付加される最大オーバーヘッドが増える．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ることができる．

## 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．

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．又、バーストロスが起こった際にも，そのバース

表 1: 署名方式の評価

| Scheme | Signatures | hash | Overhead average (bytes) | Overhead max (bytes) | Sender packet buffer size (number of packets) | Reciever packet buffer size (number of packets) |
|---|---|---|---|---|---|---|
| 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 |
| $\delta = 2$ | 8 | 40 | 100 | 100 | 1 | 1 |
| $\delta = 3$ | 5 | 37 | 140 | 140 | 2 | 1 |
| $\delta = 4$ | 4 | 36 | 180 | 180 | 3 | 1 |
| $\delta = 5$ | 3 | 35 | 220 | 220 | 4 | 1 |

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，ある特殊の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．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は，そのパケットの認証は不可となる．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，このような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 結論

本稿では、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た結果，比較として署名に必要となる送信側遅延と署名の確認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] Thomas J. Kostas, Micheal S. Borella, Ikhlaq Sidhu, Guido M. Schuster, Jacek Grabiec, Jerry Mahler, "Real-Time Voice Over Packet-Switched Networks", IEEE Network, January/February, pp.18-27, 1998.

[2] Upkar Varshney, Andy Snow, Matt McGivern, Christi Howard, "Voice Over IP", Communications of the ACM, Vol.45, No.1, pp.89-96, January 2002.

[3] Rosario Gennaro, Pankaj Rohatgi, "How to Sign Digital Streams", CRYPTO 1997, LNCS1294, pp.180-197, 1997.

[4] Chung Kei Wong, Simon S. Lam, "Digital Signature for Flows and Multicasts", IEEE/ACM Transactions on Networkng, Vol.7, No.4, pp.502-513, 1999.

[5] R. Merkle, "A Certified Digital Signature", Proceedings of the Conference on Advances in Cryptology, pp.218-238, 1989.

[6] Adrian Perrig, Ran Canetti, J. D. Tygar, Dawn Song, "Efficient Authentication and Signing of Multicast Streams over Lossy Channels", Proceedings of the IEEE Symposium of Research in Security and Privacy, pp.56-73, 2000.

[7] 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.

[8] "IP ネットワーク技術に関する研究報告書", 総務省, 2002.