IP電話向け音声ストリーム認証手法

宇田隆哉[†] 松下温[†]

本論文ではIP電話向け音声ストリームを認証する手法を提案する．本研究の目的はIPSecのような通信の機密性ではなく，音声データの認証である．話者は自分の発言内容を否認できず，受話者は話者の発言内容を変更できない．電子署名法の施行により，電子データは紙面に署名捺印したものと同等の法的効力を持つようになった．それゆえ，音声認証された通話内容は法的な証拠能力を持ちうる．本提案の方式では，公開鍵署名を施すハッシュダイジェストを複数パケットに分散し，ハッシュダイジェストに冗長性を持たせることにより，パケットの転送遅延によって署名が喪失してしまう認証時の問題を解決している．本方式を用いたIP電話は，サーバを介さないP2P状態で音声データの認証を行うことが可能であり，パケットロスや遅延に耐性を持つ認証を行える．さらに，定常遅延や遅延のゆらぎを考慮しつつ，通信品質に応じて署名間隔を変更し，電話端末の演算負荷を下げることが可能である．本論文では，送信音声および受信音声双方の署名を統合することにより，IP電話の通話内容を完全に認証する方式を提案している．

Voice Stream Authentication Method for IP Telephony

Ryuya Uda[†] and Yutaka Matsushita[†]

A voice stream authentication method for IP telephony is described in this paper. The aim of the study is not communication confidentiality such as provided by IPSec, but authentication of voice data. With the method proposed in this paper, we have realized non-repudiation by a sender and non-falsification by a receiver. A document with a digital signature has become as legally valid as a document with physical sign and seal under the laws of digital signature and authentication technology, therefore recorded voice that is digitally authenticated might be adopted as evidence in court in the future. In our proposed method, validity of a digital signature is maintained by hash digests which are distributed into plural packets that have verbosity, so that the signature can be reconstructed even when some packets are lost or delayed. IP phones implemented with our method can authenticate voice data in P2P communication. Furthermore, for constant and unstable delay, our method adjusts total delay time to a constant length by dynamically changing an interval between signatures in order to decrease the calculation load on a phone device. In summary, we propose a method which integrally handles authentication of voice data both for sending and receiving applications in IP telephony.

1. はじめに

日本ではIP電話の普及がめざましいにもかかわらず，安全面における対策はあまり進んでいない．IP電話におけるサービスは，既存の公衆電話網をIPに置き換えただけのものとなっている．そこで，本論文では，通話音声認証機能を備えたIP電話実現のためのパケット構成手法を提案し，評価を行う．本研究の目的はIP電話の音声認証であるが，これは着信時における通話相手の認証でも音声の暗号化通信でもなく，音声ストリームの即時認証である．電子署名及び認証業務に関する法律（電子署名法）の施行により，電子データは紙面に署名捺印したものと同等の法的効力を持つようになった[1)]．IP電話の通話内容に対して電子署名を施すことにより，通話内容に対する否認や，現在のアナログ音声録音で行われてしまうような通話記録を不当に改竄した詐欺を防止でき，高圧縮で低品質な音声データであったとしてもその証拠価値を法的に十分なものとすることが可能となる．

現在，固定電話の代替としてIP電話が普及してきており，将来的にはモバイル環境での利用も見込まれている．モバイル環境では，携帯性向上のために機器が小型化，省電力化していることから，演算性能の低いデバイスを用いなければならず，バッテリを節約する必要も生じる．加えて，通信環境も，無線LAN，PHS

† 東京工科大学
Tokyo University of Technology

網，第三世代携帯電話網などが混在しており，移動しながら通信した場合，一定の通信品質を得られるとは限らない．本研究は，モバイル環境で使用される IP 電話に特化し，通信遅延やパケットロスを考慮しつつ，電話としての即時応答性を損なわないように演算負荷の軽減を図ることを目的とし，従来研究の弱点であるパケット長の増大を防ぐ手法を提案している．

本論文において，2 章では音声ストリームに対する署名方式とその問題点および関連研究について，3 章では提案方式を説明し，4 章において評価を述べ，5 章を結論とする．

2. 音声認証の問題点

2.1 サーバを介した認証

音声の認証サービスは電話を用いた商取引においてすでに実用化されている[2]．現存するサービスでは，音声データはつねにサーバを介して送受信されており，サーバに蓄積した通話記録から通話内容を証明する仕組みとなっている．しかし，この方式では通話するクライアント数が増加するほどサーバの負荷も増加するため，大規模運用するには高コストとなる．これに対し，本論文で提案する手法では，P2P で直接通話を行い，鍵の確認には CA（認証局）を利用するが，音声データの認証は通話しているクライアント上で行うため，通話時にサーバに負荷がかからないのが特徴である．

2.2 ストリーミングメディアの認証

ストリーミングメディアに対し一般的な署名方法を採用するのは困難である．**図 1** に一般的なデータとストリーミングメディアに対する署名の違いを示す．図 1 の (a) のように，一般的なデータに署名する場合は，データ全体のハッシュを計算し，そのハッシュダイジェストを公開鍵暗号の秘密鍵で暗号化することにより，データ全体の署名とする．ストリーミングメディアも同様の方法で署名することが可能であるが，データ全体に署名が施されているとデータが完全に転送されるまで署名の検証を行うことができず，受信しながら再生するというストリーム再生形態をとることができない．そこで，ストリーミングメディアに対する署名は，図 1 の (b) のように施される．データを特定の単位で分割し，分割されたデータごとに署名を添付する．この場合，データ分割を細かくすればするほど，受信中のデータが再生されるまでの時間が短くなり，リアルタイムでの再生状態に近くなる．しかし，分割数が増加してしまうと署名数も増加してしまい，署名を検証するための演算負荷が分割数に比例して増加してしまう．分割数を少なくすれば署名検証のための演算負荷は減少するが，受信開始から再生までの時間が長くなり，リアルタイムに近い再生は実現しない．また，署名を生成する側でも受信する側と同様の遅延問題が生じる．署名を施すデータが生成されるまでは計算を行うことができず，これが送信側遅延となる．本研究は IP 電話向けの音声ストリーム認証であり，署名のための演算負荷を抑制しつつ認証のための再生遅延を少なくする方法をとっている．

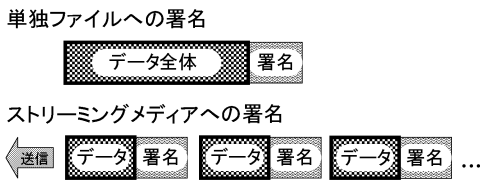

図 1 ストリーミングメディアへの署名
Fig. 1 Signing on streaming media.

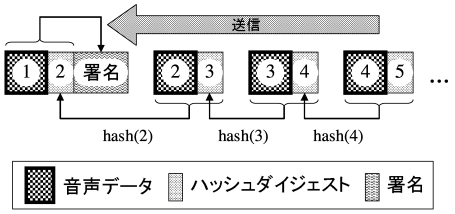

図 2 ハッシュチェーン方式
Fig. 2 Hash chain method.

2.3 関 連 研 究

ストリーミングメディアに対する認証方式はすでにいくつか存在する[3)~9)]．最も有名なものは Gennaro らによるハッシュチェーン方式[3] である．Gennaro らのハッシュチェーンは最初に登場した効果的なストリーミングメディア認証方式である．ハッシュチェーン方式では，**図 2** に示すように，次のパケットのハッシュダイジェストが 1 つ前のパケットに付属している．署名は先頭のパケットにしか付随しないためハッシュのみで後続パケットの認証が行え非常に効率が良いのが特徴である．具体的な比較は 4.1 節で述べるが，一般的にハッシュダイジェストは公開鍵署名よりも 100〜1,000 倍高速に計算可能といわれている．

しかし，下記にあげる理由で実時間で会話をする IP 電話にはハッシュチェーン方式の認証は適さない．

- 全データが作成されるまでチェーンが生成できない．
- 1 つでもパケットが欠落すると全後続パケットが認証不能となる．

本論文では UDP を用いた IP 電話を想定している

が，IP 電話は実時間性を優先するため，TCP のような再送要求を返してパケットの欠落を防ぐようなプロトコルは適用されないのが一般的である．また，受信側ではシームレスに音声が再生されるため，ネットワーク遅延のゆらぎによって到着が間に合わなくなってしまったパケットの持つ音声は再生されない．以上の理由からパケットロス耐性のない署名アルゴリズムは IP 電話には使用できない．ゆえに，本研究において想定しているモバイル環境での IP 電話に対して音声署名を実現するには，遅延時間の考慮とパケットロス耐性が重要な要素となる．

パケットロス耐性を考慮したストリーミングメディアの認証方式はいくつか存在する．Wong & Lam 方式[4] ではスター型もしくはツリー型のパケット構造を作成し，ハッシュ関数を用いることで署名回数を削減している．しかし，スターまたはツリーを効率良く形成するには送信時にある程度のパケット数をバッファリングしなければならない．Golle らの方式[5] はパケットの連続欠落であるバーストパケットロスに耐性を持たせているが，ランダムパケットロスに対する耐性はない．Perrig らの方式[6] はパケット認証に MAC（Message Authentication Code）を用いている．MAC はパケットの完全性を示すために用いられ，第三者による改竄を検出できる．しかし，この方式では話者の一方に悪意がある場合に，どちらの話者が不正をしたのか検証することは不可能であり，本論文が目的とする通話記録の内容証明には使用できない．田中らの方式[7] はハッシュチェーンを 2 つ使用し，送信時にバッファリングされるパケット数を減少させつつバースト，ランダムのパケットロスに対する耐性を持たせている．しかし，2 つのハッシュチェーンを確認するために受信側でのバッファリング遅延が大きくなってしまうのが欠点である．

そこで，前述の研究の欠点を補うため，バッファリング遅延を任意に調整可能なリアルタイム音声ストリーム認証方式[8),9)] が提案された．この方式では，ランダムパケットロス耐性を有しつつ，即時性を保つように遅延時間を考慮して署名間隔を任意に変更可能としている．しかし，この方式には IP 電話への適用に際し致命的な欠点がある．まず，この方式は演算効率を上げるために署名間隔を長くするとパケットサイズも線形に増加してしまうため実用には適さない．また，この方式では署名間隔の動的変更と通話内容の双方向認証が行えないという欠点があり，IP 電話には適用できない．本論文では，これらの問題点を解決し，パケットロスに対する耐性を持つとともに，署名間隔に依存せずに送出パケット長を固定でき，署名間隔の動的変更と通話内容の双方向認証に対応した，IP 電話向け署名方式を提案している．

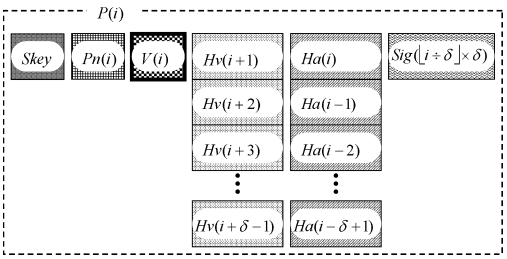

図 3 リアルタイム音声ストリーム認証方式の構造
Fig. 3 Construction of RTVS auth. method.

本論文の新規性を明確にするため，リアルタイム音声ストリーム認証方式について説明する．**図 3** にリアルタイム音声ストリーム認証方式のパケット構成を示す．

P はパケットであり，$P(i)$ は i 番目のパケット構成であることを示している．$Skey$ は再送攻撃対策用に設定されるセッションキーであり，通話ごとに新たな値が設定される．ただし，この $Skey$ には再送攻撃に対する欠陥があり，この問題と本提案における解決策について 4.4.1 項で述べることとする．Pn はパケット番号であり，受信時にパケット順序を保つため連番が振られている．V は発話された内容が録音された音声データである．Hv および Ha は次式によって計算されるハッシュダイジェストであり，δ は公開鍵署名の署名間隔である．なお，本論文では $\bullet$ を連接記号と定義して扱うものとする．連接では数値が和算されず，データ領域が結合される．

$$Hv(i) = \mathrm{Hash}(Skey \bullet Pn(i) \bullet V(i)) \quad (1)$$

$$Ha(i) = \mathrm{Hash}(Hv(i) \bullet Hv(i+1) \bullet \ldots \bullet Hv(i+\delta-1)) \quad (2)$$

Sig は次式によって計算される公開鍵署名である．図 3 中の $\lfloor\rfloor$ は小数点以下を切り捨てるという意味を表すガウス記号である．

$$Sig(i) = \mathrm{Sign}(\mathrm{Hash}(Ha(i-\delta+1) \bullet \ldots \bullet Ha(i-1) \bullet Ha(i))) \quad (3)$$

リアルタイム音声ストリーム認証方式における署名作成までの流れを分かりやすくするため，具体例を用いて解説する．**図 4** は，署名間隔が 4 パケットのときに，8 番目のパケットが完成した様子を示している．

図 3 より，各パケットに Hv は $\delta - 1$ 個，Ha は δ 個付随する必要があることが分かる．公開鍵署名は δ パケットに 1 回作成され，他のパケットではそのコピーが使用される．なお，図 4 において，4 番目以前

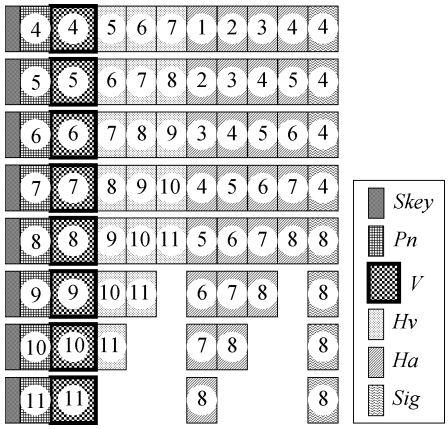

図 4 リアルタイム音声ストリーム認証方式の署名例
Fig. 4 Signing example of RTVS auth. method.

のパケットがすべてロストしてしまうと，5〜7 番目のパケットは 8 番目のパケットが到着するまで署名の検証が行えない．つまり，完全なバーストパケットロス耐性を保つためには $\delta - 1$ パケットの受信遅延が必要となる．さらに，文献 9) にはこの方式において動的な署名間隔変更が可能と記されているが，図 4 を見れば分かるようにすべてのパケットが署名間隔 δ に依存した構成となってしまっているため，一度音声通話を中断しない限り署名間隔の変更が行えない．つまり，シームレスな署名間隔変更は不可能である．

本論文で述べる新たな提案方式は，リアルタイム音声ストリーム認証方式を改良することにより，次の点を克服していることが特徴である．

- パケット長が署名間隔とは独立で固定長．
- 再送攻撃対策．
- パケットロス耐性のために受信時にバッファリングを必要としない．
- シームレスな署名間隔変更．

3. IP 電話向け音声署名の提案

本章では，IP 電話における音声認証を実現する手法を提案し，解説する．まず，本方式で用いられる初期化パラメータの設定方法と意義について説明を行い，提案の基本となる単方向署名確認方式について解説し，その解説をふまえて，相互の発話内容を一度に署名する双方向署名確認方式についての解説を行う．

なお，本論文で使用する演算記号の定義について述べる．● は連接記号と定義して扱うものとする．連接では数値が和算されず，データ領域が結合される．また，$\lfloor\rfloor$ は小数点以下を切り捨てるという意味を表すガウス記号である．さらに，j 番目から k 番目までの a の排他的論理和について，$EXOR(a_j, a_{j+1}, \ldots, a_k)$ と表記しているが，このような演算を次のようにも表現することとする．

$$\mathrm{EXOR}(a_j, a_{j+1}, \ldots, a_k) \equiv \mathrm{EXOR}(a_i)_{i=j}^{k} \quad (4)$$

3.1 初期化パラメータ

本提案方式における初期化パラメータは次のとおりである．初期化パラメータは通話開始前に双方で交換し，初期化される．

Ds：送信署名間隔
Dr：受信署名間隔
$Skey$：パケット番号用疑似乱数列作成初期値

Ds は自分用のものを設定し，Dr および $Skey$ は通信相手側のものを設定する．Ds は送信側の公開鍵暗号署名間隔である．4.1 節で示すように，公開鍵暗号の署名計算は非常に端末に負荷をかけるため，端末の環境や状況を考慮して Ds の値を設定する．デスクトップ PC などの高速な端末を 100%の負荷で使用できるのであれば毎秒 20 パケット（送信側音声バッファ遅延 50 ms）で送出される音声すべてに署名を施すことも可能であると考えられるが，携帯型端末での稼働や他アプリケーションとのマルチタスクを考えると，Ds の値を任意に設定可能なことはモバイル環境での IP 電話にとって重要である．

Dr は通信相手側にとっての Ds となる．Ds は相手側の端末性能に応じて決定されてしまうため，演算性能の低い端末で間隔を短く設定されると問題が生じることも考えられるが，4.1 節の評価より，RSA の特性によって署名よりも署名確認のほうがはるかに高速に演算可能であるため，問題は発生しにくい．また，Ds は動的に変更可能なため，演算負荷が大きすぎて音声の再生が間に合わない（音声が途切れる）場合には，Ds を再調整することで問題を回避できる．

$Skey$ はパケット番号用の疑似乱数列を作成するための初期値であり，プロセスの実行時間などをもとにランダムに作成される．本方式の実装には Visual C++ 6.0 を用いているが，$Skey$ を srand 関数にセットすることにより，rand 関数を用いて 32 ビットの疑似乱数列を得ることができる．本提案方式ではパケットに添付されているパケット番号がシーケンシャルではないが，疑似乱数の初期値を双方で共有することにより，番号の並び順を再現した数列を確保できる．$Skey$ は通話相手側の値を設定する．これは 4.4.1 項で述べる再送攻撃を防ぐためである．これにより，送信側は過去に使用した $Skey$ と同様のものを指定し，事前計算により過去に作成した別の音声と差し替える攻撃が不可能になる．

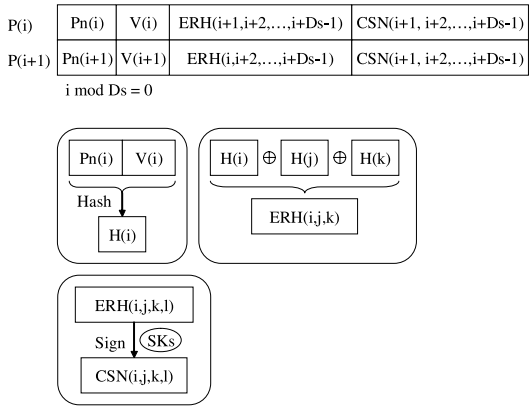

図 5 単方向署名確認方式
Fig. 5 One side verification method for signatures.

3.2 単方向署名確認方式

本方式では，音声ストリームに対して**図 5** の方法で署名を施す．図 5 上部にパケット構成を描き，下部に各項目の構成方法について示した．

i はパケット番号を表す．$i = 1$ であれば，音声が送信される 1 番目のパケットであることを示す．Pn はパケット番号を疑似乱数により付与したものである．論文中の表現が紛らわしくならないよう，パケットの順序を数値としている i をパケット番号と呼称し，Pn はパケットの順序を決めるための疑似乱数列と呼称することにする．本論文では，電話通話における受話器を上げて通話を開始してから終了して受話器を降ろすところまでの通信を 1 セッションとしてとらえ，1 セッションの音声通信に使われるパケット番号を 32 bit の数値とし，32 bit の疑似乱数列を発生させるメソッドに初期値としてセッション用の鍵 $Skey$ を使用する．この鍵は通話開始前に事前に共有されるものである．$Pn(i)$ をシーケンシャルなものにせず，疑似乱数列とするのは，4.4.2 項で述べる既知平文攻撃の対策である．Pn の初期値は通信開始時につねに受信者側によって決定される．V は，各パケットに含まれる音声データ部である．$V(i)$ は i 番目のパケットに含まれる音声データであることを示す．$H(i)$ は，次式のように $Pn(i)$ と $V(i)$ を連接し，ハッシュダイジェストを計算したものである．

$$H(i) = \mathrm{Hash}(Pn(i) \bullet V(i)) \tag{5}$$

音声データのみのハッシュダイジェストを使用しないのは，4.4 節で述べる既知平文攻撃対策のためである．ERH は引数すべての排他的論理和（EXOR）である．Ds は送信側遅延（単位はパケット数）であり，図 5 ではパケット番号 i が Ds で割り切れる場合のものとして描いているが，一般的な i に対して ERH は次式のように表現される．

$$\mathrm{ERH} = \mathrm{EXOR}(\mathrm{Hash}(Pn(k) \bullet V(k)))_{k=\lfloor i/Ds \rfloor \times Ds}^{k=\lfloor i/Ds \rfloor \times Ds + Ds - 1 (k \neq i)} \tag{6}$$

ERH の計算において $(k \neq i)$ となっているのは，i 番目のパケットに含まれる ERH にはそのパケットの音声データのハッシュダイジェストが含まれないためである．これは本提案方式における工夫の 1 つであり，その理由については理解しやすいように 3.4 節に具体例を示しながら解説した．CSN は公開鍵暗号による電子署名である．CSN は各パケットに付与されるが，署名時に公開鍵暗号の演算を行うのは，Ds 回に 1 回でよい．図 5 に示すように，パケット番号 i $(imodDs = 0)$ から $i + Ds - 1$ までは同一の CSN が付与されるため，1 回演算されたもののコピーが各パケットに含まれている．$\mathrm{Sign}(KEY, DATA)$ を $DATA$ を秘密鍵 KEY で署名したものとし，送信者側の（公開鍵暗号における）秘密鍵を SKs とすると，$CSN(i)$ は一般的な i に対して次式のように表現される．

$$\mathrm{CSN} = \mathrm{Sign}(SKs, \mathrm{EXOR}(\mathrm{Hash}(Pn(k) \bullet V(k)))_{k=\lfloor i/Ds \rfloor \times Ds}^{k=\lfloor i/Ds \rfloor \times Ds + Ds - 1}) \tag{7}$$

以上のように，本提案方式では，すべてのパケットで署名計算を行わず，送信側署名間隔 Ds パケットに 1 回だけ署名計算を行うことにより，演算負荷が軽減されている．通信時には，Pn，V，ERH，CSN がまとめて 1 つの UDP パケットに格納されて送信される．なお，Pn，V，ERH，CSN をまとめて 1 つの音声パケットとして偽装し，録音時のマイクと再生時のスピーカとの間に本当の音声を分離するフィルタを挿入できるインタフェースを持つ通話装置であれば，本方式は別の通信プロトコルを用いた通話にも利用可能である．

3.3 双方向署名確認方式

音声ストリームに対する効率的な署名方式[3)~6)] や，リアルタイム音声ストリーム認証方式[8),9)] では，単方向の音声認証しか行えない．二者間で通話するとき，お互いはお互いの発話内容に対して署名を施して送信するが，受信者がその署名付き音声を保持していたとしても，何という発言に対して何という返答があったのかは相手の音声だけでは判断できない．もちろん，送信側の音声は自所に保存可能なため，合わせて提示することは可能であるが，自分の音声は後日の捏造が容易であるため意味をなさない．

そこで本論文では，3.2 節の単方向署名確認方式を改良し，双方向の音声通話を署名する方式を提案する．

図 6 に示すように，各パケットには図 5 と比較し

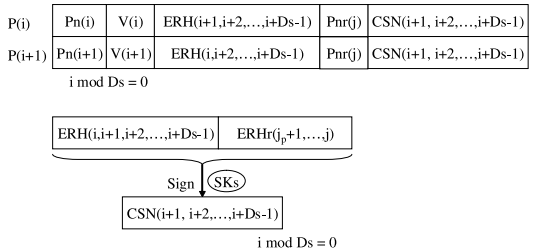

図 6 双方向署名確認方式
Fig. 6 Mutual verification method for signatures.

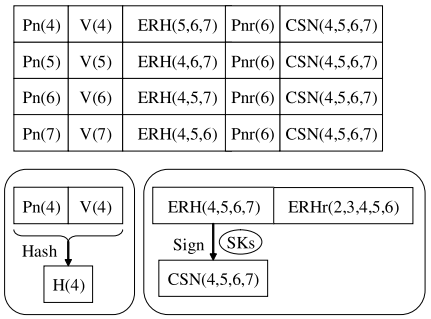

図 7 署名作成例
Fig. 7 Example of making signatures.

て $Pnr(j)$ が付与されている．Pnr は，相手側のパケットを j 番目まで受信した（正確には，署名が正しいことを確認して再生バッファに格納した）ことを示す．この双方向方式では，署名 CSN の作成法が図 5 のものとは異なり，図 6 のように示される．

図 6 において，$ERHr$ は受信したパケットの音声ハッシュ値に対する排他的論理和である．j を今回受信完了したパケット番号，j_p を前回の j（j が更新される前の値），Pnr を受信したパケットの Pn（パケットの順序を決めるための疑似乱数列），Vr を受信したパケットの V（音声）とすると，$ERHr$ は次式で計算される．

$$ERHr = \mathrm{EXOR}(\mathrm{Hash}(Pnr(k) \bullet Vr(k)))_{k=j_p}^{k=j} \tag{8}$$

よって，$\mathrm{Sign}(KEY, DATA)$ を $DATA$ を秘密鍵 KEY で署名したものとし，送信者側の（公開鍵暗号における）秘密鍵を SKs とすると，$CSN(i)$ は一般的な i と j に対して次式のように表現される．

$$\begin{aligned}\mathrm{CSN} = \mathrm{Sign}(SKs, \mathrm{EXOR}(\mathrm{Hash}(Pn(k) \\ \bullet V(k)))_{k=\lfloor i/Ds \rfloor \times Ds}^{k=\lfloor i/Ds \rfloor \times Ds+Ds-1} \\ + \mathrm{EXOR}(\mathrm{Hash}(Pnr(k) \bullet Vr(k)))_{k=j_p}^{k=j}\end{aligned} \tag{9}$$

なお，受信したパケットに欠落があった場合であるが，たとえば k 番目のパケットが欠落したとすると，$Pnr(k)$ と $Vr(k)$ が失われるため，$Hash(Pnr(k) + Vr(k))$ の演算が不可能になる．パケットロス時の対処方法については 3.5 節で述べる．

3.4 署名検証とパケットロス

本提案方式の双方向署名検証について，パケットロスが起きた場合もふまえて解説する．なお，一般式では理解が困難と思われるため，具体的に番号を振った**図 7** を使用する．

図 7 では，4 番目のパケットから 7 番目のパケットが示されており，送信署名間隔は $Ds = 4$ 受信署名間隔は $Dr = 4$ である．署名間隔がともに 4 であるのに，$ERHr$ の引数が 5 個であるのは，送受信のゆらぎによって毎回同じ数のパケットを処理できるとは限らないから（図 7 ではたまたま 5 個受信した）である．音声バッファの詳細は 4.3 節で述べる．本方式では，転送速度のゆらぎによって，受信する音声パケットの受信速度に誤差が生じても，署名検証には影響を与えない方式となっている．

パケットロスがない場合，署名検証は以下の手順で行われる．4 番目のパケットの場合，次式で $ERH(4,5,6,7)$ が求まる．

$$\begin{aligned}&ERH(4,5,6,7) \\ &\quad = ERH((Pn(4) \bullet V(4)), 5, 6, 7) \\ &\quad = ERH((Pn(4) \bullet V(4)), ERH(5,6,7))\end{aligned} \tag{10}$$

$ERHr(2,3,4,5,6)$ は，こちらから送出されたパケットのハッシュ値より作成されるため，$ERH(4,5,6,7)+ERHr(2,3,4,5,6)$ を求めることができる．この値が，$CSN(4,5,6,7)$ を相手の公開鍵で復号した結果と一致すれば署名が正しく施されていると確認できる．5 番目，6 番目，7 番目のパケットも同様の方法で署名を確認できる．ただし，4 番目のパケットを確認した時点で $CSN(4,5,6,7)$ を相手の公開鍵で復号しているので，その結果を利用すれば，5，6，7 番目のパケットでは公開鍵署名確認の演算を省略することができる．以上より，公開鍵署名計演算は Ds 回に 1 回，署名確認演算は Dr 回に 1 回行えばよく，演算の効率化を図ることができる．

次に，パケットロスが起きた場合について解説する．4 番目のパケットが不着の場合，$Pn(4)$ と $V(4)$ が入手できないため，そのハッシュダイジェストである $H(4)$ が算出できない．しかし，$ERH(4,6,7)$ が 5 番目のパケットに付属しており，$H(6)$，$H(7)$ は Pn と V より算出できるため，$H(4)$ は次式で算出可能である．

$$H(4) = ERH(ERH(4,6,7), H(6), H(7)) \quad (11)$$

また，他のパケットに付属する ERH を用いて，次式のようにも算出できる．

$$H(4) = ERH(ERH(4,5,7), H(5), H(7)) \quad (12)$$
$$= ERH(ERH(4,5,6), H(5), H(6)) \quad (13)$$

これは排他的論理和が同演算により逆算可能な法則に基づいている．これらのハッシュダイジェストを一致させた状態で，4 番目のパケットの音声データを後から任意なものに変更することは，送信者，受信者ともに非常に困難であり，1 方向であるハッシュダイジェストの逆算は莫大な演算コストを必要とするため現実的に不可能である．次に，4 番目のパケットと 5 番目のパケットが不着の場合について考える．6 番目と 7 番目のパケットから，$ERH(4,5)$ は次式で求まる．

$$ERH(4,5) = ERH(ERH(4,5,7), H(7)) \quad (14)$$
$$= ERH(ERH(4,5,6), H(6)) \quad (15)$$

これらのハッシュダイジェストを一致させた状態で，4 番目と 5 番目のパケットの音声データを後から任意なものに変更することは，送信者，受信者ともに非常に困難であり，ハッシュダイジェストの逆算は莫大な演算コストを必要とするため現実的に不可能である．

3.5 パケットロス時の受信音声ハッシュダイジェスト作成

図 7 で，受信側が $ERHr(2,3,4,5,6)$ を計算するとき，パケットロスが起きた場合について解説する．4 番目のパケットが欠落すると，5 番目と 6 番目のパケットが到着しても，$ERH(4,6,7)$ と $ERH(4,5,7)$ というように 7 番目のパケットが来ない限り $H(5)$ と $H(6)$ を ERH から分離できない．つまり，共通の CSN を持つブロック（範囲）でパケットの欠落が起きると，ハッシュダイジェストを確認できるのはその直前のブロックの最後のパケットまでということになる．この場合は，添付する Pnr を $Pnr(3)$ にして返信し，7 番目のパケットの到着を待って，次の CSN 作成のタイミングで 7 番目まで確認したことを返信する．

逆に，受信パケットが失われた場合，次の CSN に与える影響であるが，これは次の手順で問題を回避できる．図 7 で $ERHr(2,3,4,5,6)$ が正しく作成された場合，これは 4 番目と 5 番目と 6 番目のパケットはすべて正しく届いているということである．ここで 7 番目のパケットが不着だとしても，$H(7)$ は以下の式により計算可能である．

$$H(7) = ERH(ERH(5,6,7), H(5), H(6) \quad (16)$$
$$= ERH(ERH(4,6,7), H(4), H(6) \quad (17)$$
$$= ERH(ERH(4,5,7), H(4), H(5) \quad (18)$$

よって，次回の CSN 計算時に，たとえば $ERHr(7, 8, 9)$ のような値は作成可能となる．

本提案の手法では，IP 電話での双方向通話において，相手の音声パケットの署名を確認できたところまでのパケット番号で任意に応答でき，途中で通話が切れた場合にも確認できたところまでの会話内容に署名が施されているという点で非常に重要である．これは，署名確認が毎回ブロック単位（同じ CSN の値を持つ範囲）である場合に，効果を発揮する．送信側が Ds を故意に長く設定するような場合（$Ds > Dr \times 2$ など）に，途中で会話が切れても送信音声だけ利用されないために必須の条件であるといえる．

3.6 署名取得不可能なパケット群を挟む応答

図 7 において，4 番目から 7 番目のパケットがすべて消失した場合を考える．この場合は $CSN(4,5,6,7)$ の取得が不可能となり，このブロック（CSN が等しい値を持つ範囲）を含む応答が不可能である．この場合は，$ERH(4,5,6,7)$ の計算時に 0 を代入し，$ERHr$ を作成する．受け取った $ERHr$ が $ERHr(j_p + 1, ..., j)$ である（今回署名確認されたパケットが j 番までで，前回署名確認されたパケットが j_p 番までである）とき，$ERHr$ が一致せずなおかつ $j_p + 1 \sim j$ の範囲にブロックが 1 つ以上含まれている場合は，そのブロックの ERH の値を 0 として再計算する．これで署名が一致すれば，相手側が受け取ったパケットには，1 ブロックすべて不着のパケット群が存在することが確認できる．署名が一致しない場合には何らかの改竄が行われていることになる．

1 つのブロック内の ERH が偶然 0 になる場合があると，相手側がそのブロックのパケットをすべて受信し損ねたのか，正しく受信できたのかの区別がつかなくなる．そのため，送信側でパケット作成時に偶然の 0 が発生しないように値を調整する必要がある．図 7 を例にとると，$ERH(4,5,6,7)$ の値が 0 になってはならない．$ERH(4,5,6,7)$ は $V(7)$ を作成した時点で確定されるため，この値が 0 となった場合には $V(7)$ の末尾の 1 ビットを反転させて対処する．$V(7)$ は音声データであるため，ビット操作を行うと録音された音声と比較して音がひずんでしまうが，提案の手法が想定している環境ではその影響は些細であり，人間の耳には感知不能と考えられる．現在，日本の家庭で最も普及している G.711 準拠の IP 電話などでは，非圧縮の 8 bit 8 kHz PCM が使用されている．振幅を対数で表現する 8 bit のうち，最後の 1 ビットが 1 カ所だけ変化しても，これは誤差の範囲といえる．さらに，ハッシュダイジェストが 160 ビットであれば，同一ブ

表 1 比較評価
Table 1 Evaluation in comparison.

Method Type	Signature [number]	Hash [number]	OH ave. [bytes]	OH max. [bytes]	Wait(s) [packet]	Wait(r) [packet]
Hash Chain	1	16	43	100	16	0
W & L tree	1	21	160	160	16	0
KDDI chain	1	16	39	280	16	0
Voice stream (D=1)	16	16	20(+4)	20(+4)	1	0
Voice stream (D=4)	4	32	140(+4)	140(+4)	4	3
Voice stream (D=8)	2	32	300(+4)	300(+4)	8	7
Proposed (Ds=1)	16	16(32)	20(+8)	20(+8)	1	0
Proposed (Ds=4)	4	16(32)	20(+8)	20(+8)	4	0
Proposed (Ds=8)	2	16(32)	20(+8)	20(+8)	8	0

ロック内の $ERH(i, ..., i + Ds - 1)$ が偶然 0 になる確率は $1/2^{160}$ であり，通常使用の範囲ではまず発生しない．

4. 評　　価

4.1 比較評価

評価を行う前に，ハッシュ関数と公開鍵暗号における演算速度の違いについて述べておく．ハッシュダイジェストは公開鍵署名よりもはるかに高速に計算可能である．Pentium 4 2.5 GHz，Windows XP の PC で，1 秒間あたりの演算回数を測定したところ，ハッシュ関数の MD5 で 341,000 回，SHA-1 で 114,131 回，公開鍵暗号の RSA 署名で 46 回，RSA 署名確認で 1,548 回であった．なお，公開鍵署名は鍵長および暗号化できるデータサイズが決まっているため演算回数を測定しているが，MD5 および SHA-1 のハッシュアルゴリズムは 200 バイトあたりの計算回数に換算している．これは，本論文の 4.3 節で想定している音声データ（3 章の V）の 1 パケットあたりのサイズが 200 バイトとなっているためである．つまり，本提案や関連研究ではハッシュ関数による計算を大量に行っているが，ハッシュダイジェストを数十回計算したとしても，公開鍵署名を 1 回計算するよりもずっと演算効率が良いことが分かる．

本提案方式を他方式と比較した評価を**表 1** ☆に示す．

比較対象はハッシュチェーン[3]，Wong & Lam 方式[4]，KDDI の田中らの方式[7]，リアルタイム音声ストリーム認証方式[8),9)] である．リアルタイム音声ストリーム認証方式に関しては，本提案方式と同様に送信側の署名間隔を任意に設定可能なため，署名間隔を 1，4，8 の場合に対して比較を行った．また，ハッシュダイジェストによるパケット増分のオーバヘッドを求める際には，本方式の実装に使用した SHA-1 に統一し，1 ハッシュダイジェストあたり 20 バイトとしている．表 1 の Signature と Hash は 16 パケットあたりの公開鍵暗号電子署名およびハッシュダイジェストの計算回数である．これは Wong & Lam 方式が 2 の乗数でないと署名ツリーを完成できないため，16 パケットあたりのものを比較対象としている．Wait(s) と Wait(r) は送信時および受信時のバッファリングパケット数であり，リアルタイム音声ストリーム認証方式以外は，2.3 節で述べたように，送信時に署名構築パケット数と同数の待ち時間が必要であり，効率化を図るにはバッファリングパケット数を多くとる必要が生じ，総務省がその通信遅延に応じて IP 電話のクラスを区別している[10] ように，即時性が必要な通信には不向きであることが分かる．OH ave. および OH max. は平均および最大オーバヘッド（電子署名を除く）である．括弧書き部分はパケット番号もオーバヘッドとして数えた場合に追加されるバイト数である．リアルタイム音声ストリーム認証方式では，各パケットに 32 ビット（4 バイト）のパケット番号が付属するため，その分が増加する．本提案方式では，パケットの順序を決めるための疑似乱数列 Pn および Pnr を用いており，それぞれ 32 ビット（4 バイト）であるので，合計 8 バイトずつ増加する．本提案方式のパケット番号によるオーバヘッドが大きいのは，送信パケットだけでなく受信パケットも認証可能な双方向署名確認方式をとっているためである．

2.3 節より，リアルタイム音声ストリーム認証方式では 16 パケットあたりの署名演算回数は 16 ÷ 署名間隔であるため，署名間隔が 1，4，8 のとき，それぞれ 16，4，2 となってる．これは提案方式も同様である．また，ハッシュの計算回数は，リアルタイム音声ストリーム認証方式では署名間隔が 1 のときはすべてのパケットが署名されているため Hv の計算が不要となり，Ha のための計算で各パケット 1 回ずつ，

☆ Hash Chain, W&L tree, KDDI chain の値は論文 9) の p.611 表 1 より

つまり 16 パケットで 16 回となるが，それ以外では Hv と Ha の計算が各パケットで必要となるため 2 倍の 32 回となる．一方，本提案方式では ERH の計算で各パケット 1 回，つまり 16 パケットで 16 回となる．括弧書きの数値は双方向署名確認方式をとるために $ERHr$ も含めた場合である．ERH と $ERHr$ で計算回数は 2 倍となっている．

リアルタイム音声ストリーム認証方式も本提案方式も，パケットごとのオーバヘッド差はないため，OH ave. と OH max. の値はつねに等しい．リアルタイム音声ストリーム認証方式では，2.3 節より，各パケットに Hv が $\delta - 1$ 個，Ha が δ 個付随するため，合計で $(2\delta - 1) \times 20$ バイトのオーバヘッドが生じる．本提案方式では，3.2 節および 3.3 節より，このオーバヘッドはつねに ERH のみで固定であり，1 つ分のハッシュダイジェストサイズであるため 20 バイトである．ただし，双方向署名確認方式で $ERHr$ も含めると倍の 40 バイトとなる．

表 1 の OH 値より，リアルタイム音声ストリーム認証方式では，演算の効率化を図ろうとして署名間隔を大きくとると，添付するハッシュダイジェストの数が増加してしまうため，それにほぼ比例してオーバヘッドが増大していることが分かる．一方，本提案方式はパケット長を固定にしつつ音声ストリームを即時認証可能な，IP 電話用途に適した署名方式であるといえる．

また，表 1 の Signature の値は，署名回数の比較であるが，Hash Chain と KDDI chain ではパケットグループ全体で 1 つ署名を送信すればよいのに対し，本提案方式を含むそれ以外の方式ではすべてのパケットに署名データが付属することになる．今回の実装では RSA 1,024 ビットを用いたため，署名 1 つあたり 128 バイトのデータサイズとなる．4.1 節でも述べたように，本提案方式の実装においては 1 パケットあたり 200 バイトの音声データを想定しており，また，SHA-1 のハッシュダイジェストが 20 バイトであることを考慮すると，各パケットに署名データを持たせることはデータ量の観点から考えても望ましくないことは明白である．この仕様はパケットロス耐性とも密接な関係があり，任意パケットのランダムロスとバーストロスの両方に対応するためには各パケットに署名データを持たせることが必須である．このため Hash Chain はランダムロスには対応しておらず，KDDI chain も一見するとランダムロスに対応しているように見えるが，署名がついているパケットが失われた場合はすべてのパケットの認証が無効となってしまうため，完全に任意パケットのロスに対応するためには他方式と同じようにすべてのパケットに署名データを持たせる必要がある．本提案の実験ではフレームサイズが十分に大きなネットワークを使用したため，署名によるデータ量の増加が大きくても動作に支障は生じなかったが，モバイル環境でフレームサイズの小さなネットワークを使用した場合，1 パケットあたりのデータ量が大きいと問題が生じる可能性がある．そのため，今後は，1,024 ビットの RSA 署名と同等の暗号強度を 160 ビットの署名サイズで実現可能な ECDSA の使用を検討している．ECDSA を用いれば SHA-1 のハッシュダイジェストと同じサイズで各パケットに署名を持たせることが可能である．

Ds = 4

パケット K-4〜k-1	Pn(k-4)	V(k-4)	ERH(k-3,k-2,k-1)	Pnr(j_p)	CSN(k-4,k-3,k-2,k-1)
	Pn(k-3)	V(k-3)	ERH(k-4,k-2,k-1)	Pnr(j_p)	CSN(k-4,k-3,k-2,k-1)
	Pn(k-2)	V(k-2)	ERH(k-4,k-3,k-1)	Pnr(j_p)	CSN(k-4,k-3,k-2,k-1)
	Pn(k-1)	V(k-1)	ERH(k-4,k-3,k-2)	Pnr(j_p)	CSN(k-4,k-3,k-2,k-1)
パケット k〜k+3	Pn(k)	V(k)	ERH(k+1,k+2,k+3)	Pnr(j)	CSN(k,k+1,k+2,k+3)
	Pn(k+1)	V(k+1)	ERH(k,k+2,k+3)	Pnr(j)	CSN(k,k+1,k+2,k+3)
	Pn(k+2)	V(k+2)	ERH(k,k+1,k+3)	Pnr(j)	CSN(k,k+1,k+2,k+3)
	Pn(k+3)	V(k+3)	ERH(k,k+1,k+2)	Pnr(j)	CSN(k,k+1,k+2,k+3)
パケット K+4〜k+7	Pn(k+4)	V(k+4)	ERH(k+5,k+6,k+7)	Pnr(j_f)	CSN(k+4,k+5,k+6,k+7)
	Pn(k+5)	V(k+5)	ERH(k+4,k+6,k+7)	Pnr(j_f)	CSN(k+4,k+5,k+6,k+7)
	Pn(k+6)	V(k+6)	ERH(k+4,k+5,k+7)	Pnr(j_f)	CSN(k+4,k+5,k+6,k+7)
	Pn(k+7)	V(k+7)	ERH(k+4,k+5,k+6)	Pnr(j_f)	CSN(k+4,k+5,k+6,k+7)

図 8 署名グループ
Fig. 8 Groups of signatures.

4.2 シームレスな署名間隔変更

IP 電話はつねに固定された環境や条件で使用されるとは限らない．これからのユビキタス時代においては無線を用いて移動しながら使用するスタイルが一般的になるであろうし，演算性能の低い携帯型デバイスにおいてもマルチタスクで複数のアプリケーションを同時稼働させることは珍しくなくなってきている．

3 章で解説した本提案手法による署名は，通話中に署名間隔を変更できるように設計されている．**図 8** に $Ds = 4$ 時のパケット群構成を示した．

パケット番号が $k - 4 \sim k - 1$，$k \sim k + 3$，$k + 4 \sim k + 7$ がそれぞれ同一の Pnr および CSN を持つグループであることが分かる．ERH はパケットごとに値が異なるが，これらのパケットグループ内で引数となるパケット番号が閉じているため，パケットの構成内容はパケットグループごとに独立である．よって，本提案方式では前述のパケットグループの区切りのパケット番号の位置で，送信側署名間隔 Ds を変更してもパケット構成に影響は出ない．なお，IP 電話向けに提案されているリアルタイム音声ストリーム

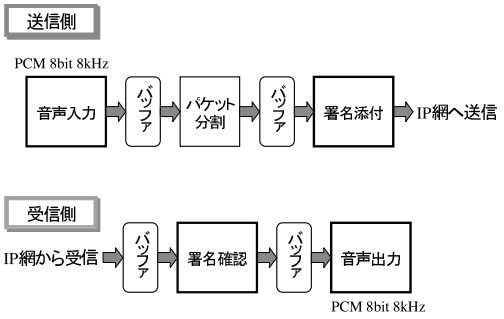

図 9 音声入出力部分の実装
Fig. 9 Implementation of voice I/O part.

認証方式[9]に，シームレスな動的署名間隔変更が行えるという記述が見られるが，これは誤りである．この方式ではすべてのパケットが署名間隔に応じた未来のパケット（そのパケットの後に生成されるパケット）に依存したハッシュダイジェストを保持しなければならないため，署名間隔を変更するとパケット構成が破壊されてしまう．

本提案方式においては，到着したパケットの確認を行い，確認されたパケット番号までのハッシュダイジェストを CSN の計算時に用い，Pnr として応答しているが，3.3 節で示したように，Pnr の値は Ds に対して独立であるため，送信側署名間隔変更の影響を受けることはない．

4.3 入出力時の音声バッファ処理

本提案の手法では，実際の IP 電話における音声処理に即した環境での実行を想定している．録音および再生される際，音声は**図 9** のようにバッファを介して処理される．

音声録音時は録音バッファに音声が格納され，必要なデータ量だけを順次切り出してパケットにする．理論的には，録音音声をバッファする速度とパケット化する速度は一定のため，遅延はいっさい生じないように思えるが，バッファ処理にかかる時間には当然ゆらぎが発生するため，実際にはパケット化しようとしてバッファにアクセスした際に，バッファ内に音声が入力されていないという事態も発生する．そのため，ある時間待って再度バッファにアクセスするか，最初からゆらぎ時間を考慮してバッファにアクセスするタイミングを少し遅らせておく必要がある．

受信した音声が再生バッファに格納される処理は，送信時と比較してゆらぎがかなり大きくなる．これは，バッファ処理の遅延だけでなく，ネットワークによる受信遅延が含まれるためである．スピーカから音声を再生しようとしたとき，バッファに次の音声が入力されていないと，再生時に空白時間が生じ，これが音声の途切れとなって現れる．また逆に，遅延した音声が連続して届きバッファに一度に大量の音声が蓄積されるとバッファ容量を超えてしまうこともありうる．そのときはバッファ内の一番古いものを消して上書きすることになり，再生音声に途切れが生じる．バッファが溢れないようにサンプリングデータを詰めて再生することも可能であるが，やはり原音とは異なる状態で再生される．

本提案で想定している IP 電話の音声は 8 bit，8 kHz の PCM である．ITU-T による標準化により一般的に使用される音声コーデックのうち，低いビットレートで使用される G.729[11] と G.723.1[12] については，音声の圧縮率が高く，大きな遅延を許可する環境下での使用を想定しており，即時認証を目的とする本論文の主旨には沿わないため，8 kHz の PCM である G.711[13] に準拠するように音声データを選んでいる．現在，日本の家庭で普及している IP 電話も，その大半がこの仕様に準拠しているものである．なお，G.711 の規定では対数量子化技術に μ-law と A-law の 2 種類があるが，日本および米国で使用されているのは μ-law 規格である．G.711 標準規格は，8 bit の対数の表示に 13 bit の A-law か 14 bit の μ-law で線形 PCM サンプルを圧縮する．実際にはデバイスから 14 ビットの Sign-Magnitude が出力され，MSB は符号（0：正，1：負）であり，残りのビットが振幅である．μ-law では残りの 13 bit がデータの大きさを表し，A-law では 13 bit のうち上位 12 bit でそのデータの大きさを表している．

4.4 攻撃対策

本提案方式の攻撃対策について説明する．

4.4.1 再送攻撃

リアルタイム音声ストリーム認証方式を提案する論文 9) の 3.7 節では，パケット番号を付与することにより再送攻撃を防ぐとしているが，これは同一通話内における再送攻撃対策であり，過去の通話時に録音された通話内容を再送された場合はパケット番号付与では防ぐことはできない．上記論文の手法では，選択文書攻撃への対応策として，セッションキー $Skey$ が組み込まれており[9]，$Skey$ によって過去の通話時の録音データを用いた再送攻撃を防ぐことは可能である．

本論文の 3.2 節で示した単方向署名確認方式および，3.3 節で示した双方向署名確認方式では，パケット番号が疑似乱数列となっているため，3.1 節で述べたように，通話開始時に疑似乱数のシードである $Skey$ を交換すれば，各パケットに無用な情報を付加することなく，再送攻撃対策が可能となっている．本提案では

疑似乱数列に 32 bit（unsigned long）を想定しているため，過去に記録した同一の Pn を持つパケットの入手自体が困難であり，仮に同一の Pn を持つパケットが再送できたとしても，そのパケットに含まれる音声が不正者の希望する内容となっている可能性は皆無に等しい．さらに，4.3 節で述べた本提案が想定する G.711 相当の環境では，1 パケットあたりに含まれる音声の長さが 25 ミリ秒程度であるため，会話内容が意味を持つように希望する音声データを連続して揃えることは非常に困難である．たとえば，1 秒間の音声を不正に作成しようとすると，25 ミリ秒の音声が 40 パケット分必要となるため，32 bit 分すべての番号のパケットを持っていたとしても，希望の連続 40 パケットを組み立てられる確率は $(1/2^{32})^{40}$ であり，さらにそれが希望する会話内容であり，なおかつハッシュダイジェストが正当なものと一致している確率は皆無に等しい．

疑似乱数列の初期値である $Skey$ を送信側で決定できるようにしてしまうと，故意に同一の $Skey$ を同一相手との通話に対して生成し，過去の会話内容と新たな会話内容を差し替えて使用する可能性が生じる．そのため，3.1 節で示したように，$Skey$ はつねに通話相手によって毎回決定される．$Skey$ に，過去に使用したものと同一のものが出現してしまうと，同一のパケット番号列が生成されてしまい，再送攻撃が可能となってしまう．このため，疑似乱数列の初期値を生成するための疑似乱数列を用意し，初期値を秘密鍵の先頭 32 bit とする．この方法を用いれば，何回初期値を生成したかというカウント値 32 bit のみを電話内に記録すればよく，署名用鍵交換のタイミングでカウンタをリセットすればよい．ただし，3.3 節では，署名 CSN に相手側の音声のハッシュダイジェストも含まれているため，$Skey$ が再利用されたとしても，完全に一致した CSN を入手することは不可能である．

4.4.2 既知平文攻撃

既知平文攻撃への対策は，再送攻撃への対策と同様となる．既知平文を組み合わせて署名が有効となるパケット群を作成可能な確率は $1/2^{32}$ であり，本提案手法の想定する IP 電話環境においては，1 パケットあたりの録音時間は $25 \times Ds$ ミリ秒となるため，希望する一連の音声を数秒にわたって偽造するのは事実上不可能である．

4.4.3 選択的文書攻撃

リアルタイム音声ストリーム認証方式を提案する論文 9) の 3.7 節において，選択的文書攻撃への対応策が述べられているが，セッションキーは既知平文攻撃への対策である．RSA 署名における選択文書攻撃では，RSA 署名の乗法性により $\mu(M)$ が $\mu(M_i)$ に素因数分解できる場合，署名も $\Pi\mu(M_i)^d \bmod n$ となり，選択的文書攻撃により入手した多数の署名を利用して M_i に対する署名を得ることができるというものである[14]．攻撃者は M_i の署名を組み合わせ，μ が素数の積となる任意のメッセージの署名を偽造可能となるため，セッションキーを挟んでも偽造に対する耐性は増加しない．セッションキーは選択的平文攻撃対策として用いるものである．RSA 署名への選択文書攻撃に対しては，ISO9796[15] や ISO9796-2[16] が提案されているが，新たな攻撃方法が提案されており，数学的にセキュリティの根拠が確認できる方式の必要性が認識されるようになってきている[14]．RSA 関数は，逆関数を計算することが計算量的に困難であるという仮定が数学的に証明されてはいないが，逆変換が困難であることが RSA 関数解読の困難性に還元できる[14] ため，上述の μ は適切に選択されていることがいえる．

4.5 受信証明の意義

本提案の手法では，関連研究の手法にはないものとして，相手の発話内容に対する受信証明を持たせている．パケットロスが起こる環境では，それが偶発的なものか相手の故意によるものかの区別をつけることは不可能である．本提案における趣旨は，パケットロスが偶発的か故意かを調査するのではなく，どちらの原因であってもパケットロスの発生を検知可能とすることにある．商取引においては，相手にこちらの声が届いていなくても，届いたと思って話が進んでしまうことが問題である．本手法を使えばリアルタイムにパケットロス状況をモニタリング可能とため，瞬間的なロス率を画面表示してもよいし，パケットロス率が一定値以上になっているときにはランプが点灯したり警告音が鳴ったりするような仕組みを導入することもできる．大切な会話の部分でパケットロス率が高くなっていることが分かれば，その部分だけを相手に復唱してもらうこともできる．会話の肝心な部分だけ故意にパケットロスを引き起こす相手の場合でも，いつまでも取引が成立しないだけで，詐欺による搾取は不可能である．

5. おわりに

本論文では，IP 電話に使用することを想定し，送受信遅延を最小限に抑えることを重視しつつ，ストリーミングメディアの転送時に電子署名法で定められた強度を持つ公開鍵署名を効率良く施す手法を提案した．

これにより，IP 電話を用いてお互いの会話内容を公開鍵による署名が施された信頼度で認証し合うことが可能となり，電話での商品注文など会話を介した金銭の取引が安全に行えるようになる．本提案の手法は，従来の電話音声の認証方法のように認証サーバを介した音声認証を行う必要がないため，維持コストを非常に安価に抑えることが可能である．このため，商用で大規模な電子商取引のみではなく，一般の利用者が安価な製品の購入に利用する用途にも使用でき，わが国の高度情報通信社会にふさわしい安全で便利な生活を提供する技術の一端を担えるのではないかと考えられる．

将来，IP 電話が広く普及すれば，通話を実行する端末は低消費電力で演算性能の低いモバイル端末に波及することは明白であり，最終的には携帯電話自体が IP 電話の技術を取り入れたものになってゆくのではないかと想像される．

技術の進歩により通信速度が向上し，通信料金がさらに安価になったとしても，端末の小型化はさらに進み，無線通信が利用できる範囲もさらに拡張されてゆくため，一定ではない通信環境によりもたらされる送受信遅延の揺らぎやマルチタスクによる端末の演算性能に関する問題は今後も残っていくと推測される．

本提案は，双方向の音声署名を実現するとともに，固定長パケットのまま通信品質に応じて署名演算の負荷を変更することが可能であり，パケットロスに対する署名の耐性も有するため，これからのユビキタス社会における様々な状況に対応できる署名方式であると考えられる．

参 考 文 献

1) 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).
2) Carleton, J.: Addressing the Regulatory Gap in Financial Services: Reducing Liability and Government Oversight in an Era of Increasing Regulation, *A Frost & Sullivan White Paper Sponsored by NICE Systems and EMC* (2004).
3) Gennaro, R. and Rohatgi, P.: How to sign digital streams, *CRYPTO 1997*, LNCS 1294, pp.180–197 (1997).
4) Wong, C.K. and Lam, S.S.: Digital Signatures for Flows and Multicasts, *IEEE/ACM Trans. Networking*, Vol.7, No.4, pp.502–513 (1999).
5) Golle, P. and Modadugu, N.: Authenticating streamed data in the presence of random packet loss, Extended Abstract (2001).
6) Perrig, A., Canetti, R., Song, D. and Tygar, J.D.: Efficient and Secure Source Authentication for Multicast, *NDSS'01*, pp.35–46 (2001).
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) 宇田隆哉, 江藤秀一, 上田真太郎, 川口信隆, 伊藤雅仁，市村　哲，田胡和哉，松下　温：リアルタイム性を持つストリーミングへの署名方式の提案，情報処理学会分散システム/インターネット運用技術シンポジウム 2003，pp.81–86 (2003).
9) 上田真太郎, 江藤秀一, 川口信隆, 宇田隆哉, 重野寛，岡田謙一：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，情報処理学会論文誌，Vol.45, No.2, pp.605–613 (2004).
10) IP ネットワーク技術に関する研究会報告書，総務省 (2002).
11) Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction (CS-ACELP), ITUT Recommendation G. 729 (Mar. 1996).
12) Dual Rate Speech Coder for Multimedia Communications Transmitting at 5.3 and 6.3 kbit/s, ITU-T Recommendation G. 723. 1 (Mar. 1996).
13) Pulse Code Modulation (PCM) of Voice Frequencies, ITU-T Recommendation G. 711 (1988).
14) 前田　司：RSA 署名の技術動向，JNSA press 第 2 号 (2001). http://www.jnsa.org/active/topics/rsa_doc.pdf
15) ISO/IEC 9796: Information technology — Security techniques — Digital signature scheme giving message recovery, ISO/IEC (1991).
16) ISO/IEC 9796-2: Information technology — Security techniques — Digital signature scheme giving message recovery, Part 2: Integer factorisation based mechanisms, ISO/IEC (2002).

(平成 17 年 11 月 24 日受付)
(平成 18 年 6 月 1 日採録)

宇田　隆哉（正会員）

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課程修了．博士（工学）．現在，東京工科大学コンピュータサイエンス学部講師．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．

松下　温（フェロー）

1963 年慶應義塾大学工学部電気工学科卒業．1968 年イリノイ大学大学院コンピュータサイエンス専攻修了．工学博士．1989〜2002 年慶應義塾大学理工学部教授，2002 年より東京工科大学教授，2003〜2005 年東京工科大学コンピュータサイエンス学部長．マルチメディア通信，コンピュータネットワーク，グループウェア等の研究に従事．情報処理学会理事，同学会副会長，マルチメディア通信と分散処理研究会委員長，グループウェア研究会委員長，電子情報通信学会情報ネットワーク研究会委員長，MIS 研究会委員長，バーチャルリアリティ学会サイバースペースと仮想都市研究会委員長，情報処理学会 ITS 研究会委員長等を歴任．郵政省，通産省，建設省，農水省，都市基盤整備公団，行政情報システム研究所等の委員長，座長，委員を多数歴任．特に国土交通省，住宅情報化標準策定委員会委員長，経済産業省総合エネルギー調査会電子計算機と磁気ディスク委員会委員長，経済産業省総合エネルギー調査会ルータ装置基準委員会委員長，最高裁判所専門委員を務める．『やさしい LAN の知識』（オーム社），『201x 年の世界』（共立出版）等著書多数．1993 年情報処理学会ベストオーサ賞，1995 年および 2000 年情報処理学会論文賞，2000 年情報処理学会 40 周年記念 90 年代学会誌論文賞，2000 年バーチャルリアリティ学会サイバースペース研究賞，2001 年情報処理学会功績賞受賞．情報処理学会フェロー，電子情報通信学会フェロー，人工知能学会，ファジイ学会，IEEE，ACM 各会員．