マルチメディア通信と分散処理　106－21
コンピュータセキュリティ　16－21
（２００２．２．１５）

セキュアなユーザ応答時間測定機構の詳細設計

谷口幸久† 石原進‡ 西垣正勝‡ 水野忠則‡

†静岡大学大学院情報学研究科 ‡静岡大学情報学部

インターネットにおける端末間の遅延は，その構造上一定ではない。そのためゲームなどの多数同時参加型のネットワークアプリケーションにおいて，サーバが複数のクライアント間で早い者勝ちの論理で勝敗を決定する場合，端末間の遅延較差による不公平が発生する。この問題に対し筆者らは，メンバ間公平性保証方式 ICEGEM （Impartial Communication Environment for GamE Members）を提案しているが，この方式にはクライアントが偽証を行った場合，システムが正常に動作しないという問題が存在する。そこで本論文ではこの問題を解決するため，耐タンパハードウェアとして実現する SIC(Secure ICEGEM Card) をネットワークインターフェイスカードとしてクライアントに装着し，この SIC によってクライアントの応答時間を測定する方式を提案し，実装に向けた具体的設計を示す。

Detailed design of the secure mechanism for measuring user response time

Yukihisa Taniguchi † Susumu Ishihara ‡ Masakatsu Nishigaki ‡ Tadanori Mizuno ‡

†Graduate School of Information, Shizuoka Univercity

‡Faculty of Information, Shizuoka Univercity

The delay between hosts in the Internet is not uniform because of its structure. Therefore, when servers determine victory or defeat on network application like multiplay network games in the logic of first come among some clients, the user's operations are treated unfairly because of the difference of delay between clients and the server. We have proposed a fairness guarantee system ICEGEM (Impartial Communication Environment for GamE Members) to solve this problem. However there are problems that ICEGEM does not work properly when clients send perjued messages. In this paper, in order to solve this problem, we propose a method to prevent purjuries on the ICEGEM and show the detailed design for mounting. In this method, each client installs a special network interface card SIC (Secure ICEGEM Card) which is implemented as a tamper registant hardware.

1 はじめに

インターネット環境におけるホスト間の遅延は，端末へのラスト１ホップの接続速度の違いに加え，ホップ数の違いや遅延の揺らぎなどにより一定ではない。このため，早い者勝ちの論理が働く，対戦ゲームやコンサートチケット予約のアプリケーションを，ネットワーク上で実現する場合，サーバからクライアントへ送られたメッセージの，クライアント側の応答時間と，サーバ側でのクライアントからのメッセージ到着時刻の間にずれが生じる。例えば早押しクイズでは，あるクライアントのユーザが，サーバからの問題提示メッセージに十分早く応答したにもかかわらず，ネットワーク上の遅延により，サーバへ応答メッセージの到着が遅れたために，サーバ側では他のクライアントより遅く応答したとみなされる場合がある。

この不公平性を解消するための機構として，筆者らはメンバ間公平性保証方式（ICEGEM: Impartial Communicatuion Environment for GamE Members）を提案している [1]。この方式では，ユーザの応答時間のみに基づいて順序の制御を行うことにより公平性を保証する。しかしながら，この方式はクライアントプログラム自身がクライアントの応答時間の測定を行うため，クライアントプログラムが虚偽の情報をサーバへ送信 (偽証) した際にシステムが正しく動作しないという問題が存在する。

この問題に対し，筆者らは，クライアントが使用しているプログラムが正規のプログラムであるかどうかを，サーバが鍵付きハッシュ関数を使用して正当性を検証する方法を提案している [3]。しかしながら，この方式においてはクライアントプログラムが自分自身の正当性を証明するための秘密情報を全て知り得てしまう。すなわち，クライアントは改竄したプログラムを使用しているにもかかわらず，正規プログラムのハッシュ値をサーバに送信することにより，サーバによるチェックを回避することが可能である。このことより，クライアント自らが応答時間を測定する方式では偽証の防止は困難であると考える。

筆者らはこれまでに，クライアントではなく，信頼できる耐タンパハードウェアが応答時間の測定を行うことによって，クライアントの偽証を防止する方式を提案している [2]。本稿では応答時間測定の具体的手法と，パケットフォーマット，IPsec 使用時の動作など，実装に向けた詳細設計を示す。

2 メンバ間公平性保証方式 ICEGEM

2.1 応答時間に基づく順序制御

サーバからのメッセージに対し複数のクライアントが応答するアプリケーションにおいて，各ホスト間の遅延が一定ではないインターネット環境では，各クライアントに与えられる環境は公平ではない。例えばネットワークゲーム等において，サーバへク

ライアントからの反応が到着する順序でサーバが順序判定を行った場合，ホスト間の遅延差により実際の反応順序と到着順序が入れ替わる場合があり，公平性が保たれない。その例を下に示す。

図 1 において，クライアント C1 とサーバ間の遅延はクライアント C2 とサーバ間の遅延より小さい。この遅延差により，C1 は C2 より早くサーバからのパケットが到着する。このため，実際の応答にかかる時間は C2 のほうが小さいにもかかわらず，C1 のパケットは C2 のパケットより早くサーバへ到着する。このときサーバは C1 が C2 より早く反応したとみなすため，C1 と C2 の間には不公平が生じている。

端末間の遅延差を吸収するための仕組みとして，各クライアント間でサーバから受信したデータの出力時刻を同期させる手法 [4] があるが，ネットワーク上の遅延は一定ではないため，出力時刻を完全に同期することは困難である。またこの方式では，データがクライアントに到着した後，そのデータが表示されるまでの遅延をソフトウェアによって制御している。このため，ユーザがクライアントプログラムを改竄した場合，ユーザが不正に早くデータを表示させることが可能になる。

そこで ICEGEM[1] では，各クライアントプログラムが自分自身の応答時間を測定し，応答メッセージに付加する。これによって，サーバがクライアントの応答時間を知ることが可能となり，クライアントの応答時間に基づく順序制御を行うことができる。

2.2 ICEGEM における偽証の問題

ICEGEM は，早いもの勝ちの論理によって勝敗が決まるアプリケーション全般，例えばネットワークゲームやオンラインカジノ，オークションに適用可能である。しかしながら，金銭の授受が行われるアプリケーション，たとえばオンラインチケット予約やオンラインカジノに対して ICEGEM を適用しようとした場合，不正の発生を完全に防止する必要がある。

また，ICEGEM はクライアントが不正を行わないという前提に基づいているため，サーバが受け

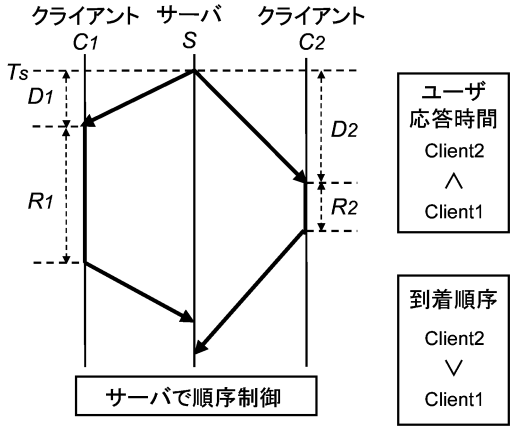
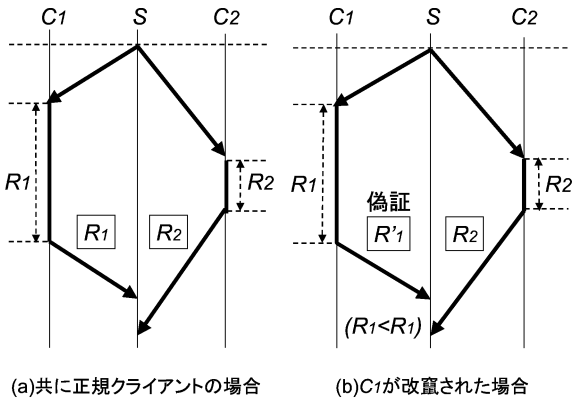


図 1 到着時間とユーザ応答時間の逆転



(a)共に正規クライアントの場合　(b)C1が改竄された場合

図 2 クライアントからのメッセージの改竄

取った応答時間が実際のクライアントの応答時間と異なる場合，システムが正常に動作しない。そこで，本稿においては，クライアントのユーザ自身による応答時間の偽証を防止する方式について述べる。

3 ハードウェアを用いた偽証防止

3.1 ハードウェアを用いる必要性

偽証を防止するためには，クライアントプログラムが正しく動作しているかをソフトウェアによってチェックする方法と，応答時間の測定もしくはそのために必要なデータを信頼できるハードウェアを用いて収集する方法の 2 つが考えられる。

電子透かしを使用することによりファイルの改竄を防止する研究 [5][6] がされている。これらの研究では，ユーザが自分の所有するファイルの改竄を検知することは可能であるが，サーバがクライアントの改竄を知ることは困難である。

ネットワーク上のタイムスタンプサーバ [7] を利用することにより，公式なタイムスタンプを記録する方式も提案されており，この公式なタイムスタンプを利用して応答時間を測定することも考えられる。しかしながら，各クライアントとタイムスタンプサーバ間の遅延が一定はないため，遅延差による不公平が生じる。また，各応答がクライアントから，タイムスタンプサーバを経由してサーバへ送信される必要があるため，即時性が必要なアプリケーションにおいて適用が困難である。

そこで，筆者らは [3] において鍵付きハッシュ関数の使用することにより，クライアントプログラムの正当性をサーバがチェックする方法を提案している。しかしながら，ソフトウェアによってチェックを行う場合，チェック用ルーチンがクライアント側で改竄される可能性がある。これに加え，チェック用ルーチンで改竄防止のために何らかの秘密情報を用いたとしても，クライアント側ではクライアントの計算機内の全ての秘密情報を知ることが可能である。そのため，ソフトウェアを使用する場合にクライアント上で動作するチェック用ルーチンによって完全な偽証防止を行うことは困難である。

筆者らは，このようなクライアントによる偽証を防止するために，図 3 のようにクライアントマシン

のネットワークインターフェースに，耐タンパ性を備えた応答時間測定用のハードウェア SIC（Secure ICEGEM Card) を取り付け，クライアントの応答時間をクライアントプログラムに代わって測定する手法を提案している [2]。以下 [2] で提案した手法の具体的設計について検討する。耐タンパハードウェアは不正な使用ができず，無理な変更を行うと正常に動作しないハードウェアである。

3.2 応答時間測定方法

3.2.1 ICEGEM のメッセージ

ICEGEM はサーバからの全てのクライアントへのメッセージと，それに対する全てのクライアントからの返答の組を 1 ターンとして，ターン毎にクライアントの順序制御を行う [1]。以下，応答時間と，測定するべきサーバからのメッセージ，及びそれに対するクライアントからサーバへのメッセージをまとめて ICEGEM メッセージと呼ぶ。応答時間の計算には，サーバからの ICEGEM メッセージがクライアントに到着した時刻 T_S と，それに対する応答メッセージがクライアントから送信された時刻 T_C が必要となる。この 2 つを対応付けるため，ICEGEM のメッセージ ID が使用される。

各メッセージは 1 つの UDP パケットで送られるものとする。サーバは，各ターンの ICEGEM メッセージに対して一意のメッセージ ID を割り当てる。このメッセージ ID はクライアントとサーバそれぞれの IP アドレスと，ターン開始時刻のハッシュ値から算出される。サーバプログラムはこの ID を元に順序制御を行うため，クライアントのメッセージ ID が改竄された場合，そのクライアントからのメッセージは無効となる。

3.2.2 2 つの応答時間測定方法

T_S と T_C から応答時間を求める方法には，SIC で行う動作の違いにより，図 4 に示すように 2 つの実現方式が考えられる。1 つは応答時間を SIC が測定する方式（時間測定型 SIC) であり，もう一つは SIC はパケットが SIC を通過した時刻をサーバへと報告し，サーバが応答時間を求める方式（時刻報告型 SIC）である。

■時間測定型 SIC

時間測定型 SIC は，SIC のみでクライアントの応答時間を測定する。時間測定型 SIC は，T_S の保存方法によって以下の 2 つが考えられる。

(a) 記憶式測定型： 記憶式測定型の SIC は，SIC 自身が，通過したデータの T_S を保持するテーブルを持ち，T_S を記憶する。T_S と T_C を対応付けるため，SIC は T_S と共にメッセージ ID を保持する。クライアントからサーバへの返信メッセージが SIC を通過する時，SIC は T_C を記録し，メッセージ ID に合致する T_S を取り出して，クライアントユーザの応答時間 $T_C - T_S$ を計算する。

テーブル内の T_S は，参照されるか，もしくは十分な時間経過後に消去される。

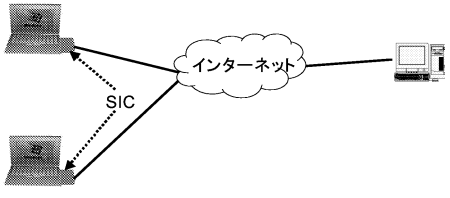


図 3 SIC 接続方法

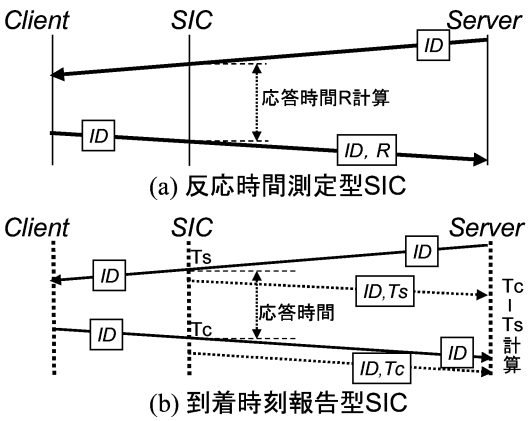


図 4 SIC2 方式動作比較

(b) 記録式測定型： 記録式測定型ではの SIC は，T_S を自分自身では記憶しない。SIC はサーバから到着した UDP パケットに T_S を書き込み，クライアントへ送信する。このとき，クライアントによる T_S の改竄を防止するため，SIC は電子署名用の鍵を持ち T_S のメッセージ確認コード (MAC:Message Authentication Code) を計算し，T_S と共にクライアントへ送信する。クライアントは応答内容と共に T_S と MAC をサーバへ送信する。クライアントからサーバへの返信メッセージが SIC を通過する時，SIC は T_C を記録し，受信した T_S から MAC を再計算し，受信した MAC と比較する。T_S が改竄されていなければ，クライアントユーザの応答時間 $T_C - T_S$ を計算する。

■時刻報告型 SIC

時刻報告型 SIC における SIC の動作は，ICEGEM のデータが通過した際に T_S または T_C を，データの ID と共にサーバへ送信するのみである。ユーザ応答時間 $T_S - T_C$ の計算はサーバが行う。各 T_C と T_S は，測定型 SIC の場合と同様にメッセージ ID によって対応付けられる。

サーバはクライアントへデータ送信後，T_S と T_C が共に SIC から到着するまで，すでに到着している通過時刻と ID を保持するテーブルを持つ。ただし，サーバプログラムがクライアントをタイムアウトと判断し T_C が到着する前に順序制御を終えた場合，テーブル内の T_S は消去される。

3.3 第三者機関による認証

SIC を用いることを前提としたシステムでも，クライアントが実際には SIC を用いず，ソフトウェアによって SIC を不正にエミュレートすることで，ICEGEM による応答時間の公平性保証が壊される

場合が考えられる。例えば，あるクライアントが不正な処理を行う SIC のエミュレータを使用し，常に実際に応答に要した時間の半分を応答時間としてサーバに送信した場合，他の正しい SIC を使用しているクライアントが不利になってしまう。

この他にも，あるサーバが特定のクライアントと結託を行ったり，クライアントのプライバシを漏洩する可能性も考えられる。

このような不正を防止するため，SIC とサーバの正当性を検証する第三者機関を設け，全ての SIC とサーバを管理する。以下に詳細を示す。

SIC は署名のための秘密鍵を持ち，SIC を利用するクライアントは第三者機関が発行する SIC 証明書を持つ。SIC 証明書は第三者機関が SIC 製造時に正しい SIC に対してのみ発行し，SIC を一意に特定する ID と公開鍵，第三者機関による署名から成る。同様にサーバは，第三者機関が発行するサーバ証明書を持つ。サーバ証明書は，クライアントのプライバシの扱い等についての一定の規定に従うサーバにのみ与えられ，サーバの IP アドレスとポートと，第三者機関による署名を含む。

クライアントはまずセッションを行いたいサーバに対し，自分の SIC 証明書を送信する，サーバは SIC 証明書から公開鍵を取り出し，自身の証明書をクライアントに送信する。クライアントはサーバ証明書を確認することにより，サーバが信頼できることを知る。

SIC は，サーバへ応答時間を送信する時，応答時間のハッシュ値を秘密鍵で暗号化することによって電子署名を作成し，応答時間と共にサーバへ送信する。サーバはこの署名によって，応答時間の改竄を検知すると共に，正当な SIC からのメッセージであるか否かを判定する。

3.4 ICEGEM メッセージの処理

SIC は ICEGEM メッセージを含む UDP パケットを，宛先と送信先の IP アドレス，ポート番号，およびデータ部の ICEGEM メッセージ ID の有無によって判別する。サーバの IP アドレス及びポート番号は，あらかじめクライアントユーザによって SIC に登録される。またメッセージ ID は UDP パケットのデータ部に含まれるため，対象とする UDP パケットが IP 層でフラグメント化されている場合，全てのパケットの到着を待ち，結合してからデータ部分を参照する。

パケットフォーマットは，図 5 に示す 5 通りが考えられる。

(a) は，時刻報告型 SIC を使用するときや，記憶式測定型のサーバからクライアントへの送信時のように，SIC がパケットを読むだけで何も書き込まない場合に使用する。

(b) は，記録式測定型や記憶式測定型のクライアントからサーバへの送信時のように，経路上で SIC が何らかの情報をパケットに書き込む場合に使用する。あらかじめ SIC が書き込む領域をパディングすることによって，IP パケットの新たなフラグメントを防止する。

(c) は，記録式測定型 SIC とクライアント間で使用され，サーバからのパケットの到着時刻 T_S とその MAC(Message Authentication Code) が含まれる。

(d) は，時間測定型 SIC からサーバへ送信される

図 5 ICEGEM パケットフォーマット

時に使用され，クライアントの応答時間 $T_C - T_S$ と，それに対する SIC による電子署名が含まれる。

(e) は，時刻報告型 SIC において，パケットが通過した時サーバに送信されるパケットである。通過したパケットのメッセージ ID と報告する通過時刻が T_S か T_C のどちらかを判別するための送信者フラグ，通過時刻 (T_S または T_C) と，それらに対する SIC による電子署名が含まれる。

(c) と (d) において，測定型 SIC がデータの書き込みを行うと，パケット全体のチェックサムを再計算する必要がある。そのため SIC は，情報をパケットに書き込んだあと，IP ヘッダと TCP ヘッダのチェックサムを新たに計算しなおし，書き換える。

3.5 SIC の動作例

SIC を使用した場合の，サーバ，クライアント，SIC，第三者機関の動作例を以下に示す。前提条件として，第三者機関はあらかじめサーバを信頼できると評価しているものとする。

3.5.1 時間測定型 SIC

① クライアントはセッションを行うサーバを決め，SIC にサーバの IP アドレス，及び使用ポートを登録する。

② クライアントはサーバに対し，所有する SIC の SIC 証明書を送信する。

③ サーバは，クライアントから受信した SIC 証明書の正当性を検証する。

④ 証明書が正しいものであることを確認すると，サーバは SIC の鍵を取り出し，クライアントへサーバ証明書を送信する。

⑤ サーバがクライアントへ，ICEGEM メッセージを送信する。

⑥ SIC は通過する全パケットの中から ICEGEM メッセージを含む UDP パケット P を探し出す。

⑦ SIC は，記録式なら T_S の MAC を計算し，P のデータ部分に T_S と MAC を記録し，IP ヘッダのチェックサムを書き換える。記憶式なら，到着時

刻をメッセージ ID と共に SIC 内のテーブルに記憶する。

⑧ SIC はクライアントへ P を送信する。

⑨ クライアントは P に対し応答を行い，応答メッセージを含む UDP パケット P' をサーバへ送信する。

⑩ SIC は通過する全パケットの中から P' を探し出し，T_C を測定する。

⑪ SIC は，記録式なら MAC を用いて T_S の改竄の有無を検証する。記憶式ならメッセージ ID から T_S を取り出す。

⑫ SIC は，T_S と T_C の差からユーザ応答時間を計算する。

⑬ SIC は計算したユーザ応答時間の電子署名を作成し，応答時間と電子署名を P' に書き込む。

⑭ SIC は P' の IP ヘッダのチェックサムを書き換え，サーバへパケットを送信する。

⑮ サーバは P' を受信後，署名を用いてユーザ応答時間の改竄の有無および SIC の正当性を検証する。これが確かめられた後，得られた応答時間を用いてクライアントの順序制御を行う。

3.5.2 応答型 SIC

①から⑥までは測定型と同じ

⑦ SIC は，P の通過時刻 T_S の電子署名を作成し，メッセージ ID と T_S，電子署名をサーバの時刻受付用ポートへ送信する。

⑧ SIC はクライアントへ P を送信する。サーバは SIC から P の T_S とメッセージ ID，電子署名を受信し，改竄の有無を検証する。

⑨ クライアントは P に対し応答を行い，応答メッセージを含むパケット P' をサーバへ送信する。

⑩ SIC は全パケットの中から P' を探し出す。

⑪ SIC は P' の通過時刻 T_C の電子署名を作成し，メッセージ ID と通過時刻，電子署名をサーバへ送信する。

⑫ サーバは P' と T_C，電子署名を受信し，改竄の有無を検証する。

⑬ サーバは，受信した T_S と T_C が改竄されていなければ，クライアントの応答時間を計算し，順序制御を行う。

4 検討

4.1 各方式の比較

3.2.2，で示した各方式の特徴を，セキュリティ強度，コスト，発生するトラフィック，信頼性の 4 点から比較し，表 **1** に示す。各評価は相対的なものであり，A がもっとも優秀，C がもっとも劣るものとする。

	記憶式測定型	記録式測定型	報告型
セキュリティ	A	C	A
コスト	C	B	A
トラフィック	A	A	C
信頼性	A	A	C

表 1 方式比較

測定型は，SIC 自身が T_S の保存と応答時間の計算を行うため，SIC にある程度高い処理能力が必要となるため，コストが増す。また，サーバとクライアントがやり取りするパケット以外を必要としない。

記憶式測定型は，SIC 自身が到着時刻を記憶する場合，SIC に記憶するためのメモリを必要するためさらにコストが高くなる。しかしながら，クライアントに応答時間に関する情報を渡すことなく実現が可能であるため，セキュリティは高くなる。

記録式測定型は，到着時刻を記憶するメモリを必要としないため，メモリのコストを必要としない。しかしながら，クライアントが到着時刻を暗号化したデータを参照することが可能であるため，クライアントがこの暗号の復号化に成功した場合，到着時刻を改竄することにより偽証が可能となる。この問題は，内部時計の仕様を秘密にしたり，MAC を計算する代わりに強固な暗号鍵で到着時刻を暗号化することにより，セキュリティ強度を上げることが可能であると考える。しかしながら，暗号化を強固にした場合，記憶式よりコストが高くなることが考えられる。

時刻報告型は，SIC は単純な動作を行うだけなので，コストは低くなる。クライアントには応答時間の測定に必要なデータを送信しないため，セキュリティは高くなる。しかしながら，経路上のトラフィックが大きい場合，SIC が送信する通過時刻を含む UDP パケットが経路上で失われる可能性が高い。この場合サーバは応答時間を計算できないため，混雑したネットワークに属するクライアントが不利になるという問題点がある。また，時刻報告型では，サーバが応答時間を計算するため，特にクライアント数が増加した際にサーバに大きな負荷がかかる。

4.2 再送処理

ICEGEM は UDP を使用するため，トランスポート層プロトコルにおける再送は発生しない。しかしながら，信頼性が必要とされるアプリケーションでは，サーバがクライアントに対してアプリケーションレベルで再送を行うことが考えられる。このとき，クライアントの応答メッセージにたいして SIC がクライアントの応答時間をどのように計算するかが問題となる。

図 **6**(a) のようにメッセージがクライアントへ到着する前にパケットロスが発生した場合，クライアントはまだ応答を行っていないため，SIC が応答時間を測定することでクライアントの正しい応答時間をサーバが受信できる。

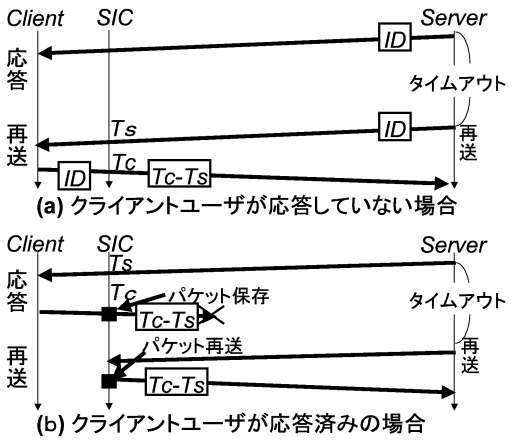


図 6 パケットの再送

一方，クライアントユーザがすでに応答を行っている場合，クライアントに重複したパケットが到着することになる。そこで SIC は，図 **6**(b) に示すように，クライアントにパケットを送ることなくサーバへの返答パケットを送信する。クライアントの応答メッセージをサーバへ送信する際にパケットを記憶する。その後同じメッセージ ID のパケットが到着した場合，サーバが再送したメッセージとみなし，クライアントが以前サーバへ送信したパケットをサーバに対して送信する。

よって，アプリケーションによるメッセージの再送を行う場合には，SIC はパケットを記憶するためのバッファと，再送メッセージを発見する処理が必要となり，コストが増大する。

4.3 SIC へのアクセス権の限定

クライアントは SIC に対してあらかじめサーバの IP アドレス及び使用するポート番号を登録する。しかしながら，クライアントが SIC の動作に関わる部分にアクセスしてしまうと SIC の動作が改竄される可能性がある。このため，サーバの IP アドレス登録時には IP アドレス登録用のメモリにのみアクセスを可能とする必要がある。

4.4 IPsec と SIC の共存

IPsec によるパケットの暗号化が行われている場合，トランスポートプロトコルの種類やポート番号，データ本体は経路上のノードから隠蔽される。そのため，SIC による ICEGEM メッセージを含むパケットのフィルタリング処理や，時間測定型におけるデータ部分への書き込みをおこなうことができない。

そのため，IPsec と SIC を共存させる場合，以下のような方法を使用して，SIC に平文が通過するようにする必要がある。

設定による非暗号化 クライアントが，ICEGEM によるセッションを行うサーバとポートに対しては IPsec を使用しない設定を行うことにより，SIC には暗号化されていないデータが通過するようにするようにすることで，SIC を用いることが可能となる。この場合，経路上を平文が流れることになるため，SIC 自身が何らかの暗号化処理を行う必要がある。

SIC への IPsec モジュールの追加 IPsec はすべてのパケットに対して IPsec ポリシチェック，IPsec パケット処理，暗号化/認証値計算処理を行うため，計算量が増大する。そのため，IPsec の処理をハードウェアである IPsec ボードで行う研究が行われている [8]。

そのため，IPsec ボードの機能を含むモジュールを SIC に搭載し，応答時間の測定と共に IPsec の暗号化/複号化を行うことにより，SIC を用いることが可能となる。

5 まとめ

メンバ間公平性保証方式において，クライアントの偽証を防止するために，耐タンパハードウェアである SIC を使用する方式を提案し，動作の詳細とデータ形式の検討を行った。また，IPsec との共存について検討した。今後の課題としては，Linux におけるカーネルレベルでの実装と，各方式の比較評価である。

参考文献

[1] 石川貴士, 石原進, 井手口哲夫, 水野忠則：メンバ間公平性保証方式の同期機構の特性評価，情報処理学会研究報告，モバイルコンピューティングとワイヤレス通信, Vol. 2000, No. 15, pp. 81–88 (2000. 12)

[2] 谷口幸久, 石原進, 西垣正勝, 水野忠則: 遅延較差調停処理のための偽証防止機構の設計, コンピュータセキュリティシンポジウム (CSS2001) シンポジウム論文集, pp. 85-90 (2001. 10)

[3] 谷口幸久, 石原進, 水野忠則: メンバ間公平性保証方式におけるハードウェアを用いた偽証防止，マルチメディア，分散，協調とモバイル (DICOMO 2001) シンポジウム論文集, pp. 771–776 (2001. 6)

[4] Y. Ishibashi, Y. Tachibana and S. Tasaka：media synchronization scheme with causality control in network environments, Proc. IEEE LCN'99, pp. 232–241 (1999. 10)

[5] 村瀬一郎, 牧野京子, 赤井健一郎, 松本勉:FinPri.txt における改竄検出法，情報処理学会研究報告，コンピュータセキュリティ，Vol. 2000，No. 68，pp. 49–56 (2000. 7)

[6] 中村直己，西垣正勝，曽我正和，田窪昭夫：プログラムの冗長化に関する検討，情報処理学会研究報告，コンピュータセキュリティ，Vol. 2000，No. 68，pp. 41–48 (2000. 7)

[7] 宇根 正志, 松浦 幹太, 田倉 昭: デジタルタイムスタンプ技術の現状と課題, 金融研究第 19 巻別冊第 1 号, (http://www.boj.or.jp/ronbun/mes00.htm)

[8] 笠井真理子, 渡辺義則, 中野善之：IPSec 処理の高速化方式の検討, 情報処理学会研究報告，コンピュータセキュリティ，Vol. 2001，No. 12，pp. 67–72 (2001. 2)