1S-06 メンバ間公平性保証方式における偽証防止に関する一検討

谷口幸久† 石川貴士‡ 石原進† 水野忠則†
†静岡大学情報学部 ‡静岡大学理工学研究科

1 はじめに

本稿では、クライアント・サーバシステム全体の動作が複数のクライアントの正常な動作に依存するようなシステムにおいて、各クライアントプログラムの正当性をサーバが検証する手法を検討する。そのようなシステムの一つとして筆者らが提案しているメンバ間公平性保証方式 *ICEGEM*(*I*mpartial *C*ommunicatuion *E*nvironment for *GamE M*embers)[1] を取り上げ、問題点の検討と解決の提案を行う。ただし本稿に示す提案方式は他のシステムにおいても応用が可能である。

2 メンバ間公平性保証方式 *ICEGEM*

2.1 概要

各クライアント・サーバ間の遅延が一定ではないインターネット環境では、サーバ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ける、各クライアントに与えられる環境は平等ではない。例えば、ネットワークゲーム等がサーバにクライアントの反応が到着する順序で順序判定を行った場合、ユーザ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。そこで *ICEGEM* では上記の問題を解決するために、クライアントがサーバのメッセージが到着した時刻から反応を行うまでの時間を測定し、応答メッセージに付加することによって、サーバにおいてクライアントでの反応時間に基づく順序制御を行う。

2.2 問題点

ICEGEM は、クライアント自身が反応時間を測定しサーバに送信するため、クライアントが偽の反応時間を送信した場合（偽証）にそれを検出できず、正しい制御が行われないという問題が存在する。(**図1**)

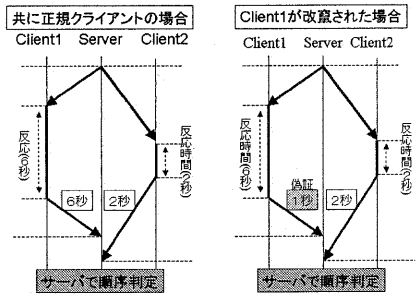

図 1: *ICEGEM* クライアントの改竄

2.3 偽証の発生理由

クライアントによる偽証が発生する理由は、以下の4点が考えられる。

1. 経路上における第三者による情報の改竄
2. あるクライアントの実装ミス
3. クライアントプログラムの破損、もしくは悪意ある第三者による変更
4. ユーザによるクライアントプログラムの意図的な改竄

次章においてはこの4点のうち、2，3について検討する。1については既存の暗号化技術、例えば電子署名や、サーバの公開鍵による暗号化によって防止が可能である。また、4については、3.5節で述べる問題が発生するため、今後の課題とする。

3 クライアントプログラムの正当性の検証

3.1 概要

発生理由 2，3 は、クライアントプログラムの正当性をサーバが検証することにより防止可能である。そこで、クライアント上のプログラムが定期的に計算する自分自身のハッシュ値と、サーバ上にあるクライアントプログラムのハッシュ値を比較することによって、クライアントプログラムの正当性の検証を行う方式を提案する。(**図 2**) ただし、前提として、サーバとユー

Protecting from unreliable programs on *I*mpartial *C*ommunicatuion *E*nvironment for *GamE M*embers (*ICEGEM*).

Yukihisa Taniguchi† Takashi Ishikawa‡ Susumu Ishihara† Tadanori Mizuno†
†Faculty of Information, Shizuoka University
432-8011, Hamamatsu, Japan
‡Science and Engineering, Shizuoka University
432-8011, Hamamatsu, Japan

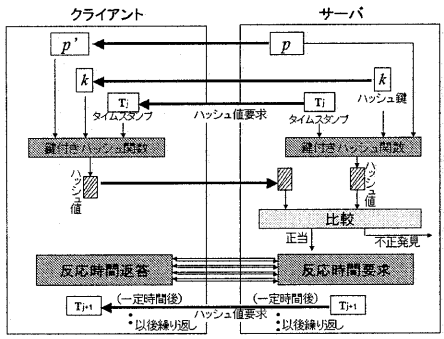

図 2: クライアント・サーバ間の情報の流れ

ザはそれぞれ公開鍵と個人鍵、及び ID を所持しているものとする。また、サーバ・クライアント間の通信はパケットレベルで暗号化されているものとし、第三者よるパケット傍受やなりすましは行われないものとする。

3.2 サーバによるクライアントプログラムの送信

ユーザはクライアントプログラム使用に先立ち、サーバへ ID の送信とクライアントプログラムの要求を行う。ユーザは要求を受けたサーバからのクライアントプログラムの送信後に、クライアントプログラムの使用が可能となる。またサーバは、ユーザ ID に基づいてそのユーザ専用のハッシュ関数用の秘密鍵 k を作成する。以下において、サーバ上におけるクライアントプログラムを p、ユーザが受信したクライアントプログラムを p' とする。

3.3 サーバによるプログラムチェック

サーバは、まず k を p' へ送信する。その後、p' へハッシュ値の要求を行い、タイムスタンプ T_j を送信する。p' は、T_j を加味した自分自身のハッシュ値 h_c を HMAC-MD5[2] 等の鍵付きハッシュ関数の鍵として k を用いることよって作成し、サーバへ送信する。一方でサーバも同様に p と k と T_j からハッシュ値 h_s を求め、h_c と比較を行い、等しければ p' を正当なプログラムであるとみなす。

3.4 ハッシュ値の改竄対策

鍵付きハッシュ関数の利用により、k に使用期限を設けることで h_c の偽造防止が可能である。また、ハッシュ値は t_j により毎回異なった値となるため、再送攻撃も防止できる。

3.5 問題点

提案方式の問題点は大きく分けて 2 つ存在する。1 つ目は、提案方式では h_c をクライアントが作成するため、ユーザによる h_c の改竄を防止する機構が必要であることである。特に悪意あるユーザが使用する際には以下のようなシナリオが考えられる。

i. ユーザは正規のプログラム p' と、偽証を行うプログラム p'' を持つ。
ii. 通常の反応時間を含む応答メッセージ送信においては、p'' が偽証を行う。
iii. プログラム正当性の検証プロセスにおいては、p'' は p' を用いることによりハッシュ値 h'_c を作成する。
iv. サーバは h_c' と h_s を比較し結果が等しいため p'' を正当であるみなす

これを防止するために、p'' が p' のハッシュ値を求められないようにする必要がある。

2 つ目は、ハッシュ計算に伴う負荷の発生である。特にサーバにおいては、ユーザ数の増加に伴い多数の負荷が集中する。またクライアントにおいても、*ICEGEM* を 3D ゲームなど複雑な処理を必要とするアプリケーションに適用する際に、負荷によってアプリケーションの動作に支障をきたさないよう適切に制御する必要がある。この対策として、例えばユーザの反応頻度や一度のセッション時間、ユーザ数などによって、それぞれ要求頻度を変化させる方式が必要である。

4 まとめ

クライアントプログラムが自分自身のハッシュ値を作成し、サーバに送信することによってサーバがクライアントプログラムの正当性を検証する方式を提案した。提案方式はサーバによって他の端末のプログラムの正当性を検証する方式であるため、本稿で取り上げた *ICEGEM* の例に限らず、分散システムにおけるプログラムの正当性の検証に広く応用が可能であると考える。今後の課題としては、提案方式の実装と評価、及び更新頻度の検討、サーバ、クライアント双方の負荷と正当性の測定が挙げられる。また、ユーザが意図的にハッシュ値の改竄を行った際の検出方法の検討と、提案方式の他の分野への応用の検討も行う予定である。

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, Vo.2000, No.15, pp.81-88, (2000.12).]

[2] Cheryl Madson Cisco Systems, Inc. Rob Glenn NIST:The Use of HMAC-MD5-96 within ESP and AH, rfc-2403, (1998.11)