1N－7

分散マルチメディア環境のための セキュリティ機構†

堀切和典[1‡]　多田征司[2‡]　河内谷清久仁[3‡]　西尾信彦[4]　徳田英幸[4]　斎藤信男[4]

[1] 富士ゼロックス(株) [2] 横河・ヒューレット・パッカード(株) [3] 日本アイ・ビー・エム(株) [4] 慶應義塾大学

1 はじめに

慶應義塾大学と企業10社が共同で行なっているマルチメディア統合環境基盤ソフトウェア(Keio-MMP)プロジェクトでは，Real-Time Machをベースに連続メディア情報を扱うためのカーネルの基本機能，連続メディア情報の制御を行なうメディアサーバ，連続メディア情報を蓄えておくメディアベースなどの研究を行なっている[1].

連続メディアデータの時間制約を満たすためには，メモリ，CPU，ネットワーク，ディスク等の資源に関して，適切な資源配分を行なう機構が必要となる．資源配分を実現するために，それぞれの資源を管理するリソース・マネジャが提案されている．(メモリ: [2]; CPU: [3], [4]; ネットワーク: [5]; ディスク: [6] 等)

これらのリソース・マネジャはプライオリティに応じて資源配分を行なうが，不正な計算主体(プリンシパル)によって資源が使用されることを防止するためには資源を要求しているプリンシパルを認証する必要がある．本稿では，Kerberos認証サービス[8]に基づく，Keio-MMPのセキュリティ機構について概説する．

2 現状と問題点

Machでは，プロセス間通信に，ポートを介したメッセージ伝達機構を用いている．ポートはケイパビリティとしてのポート・アクセス権を通じてのみアクセスすることができる．ポートおよびポート・アクセス権はカーネルによって保護されており，基本的なセキュリティ機構を提供することができる．しかし，ポートにはプリンシパルを同定する機能は提供されておらず，ポート・アクセス権をベースに認証システムを構築する必要がある．

一方，Unix上のKerberosのインプリメンテーションは

- プロセス間通信
- ファイル・アクセス制御

の点でUnixに強く依存しており，必ずしもUnixのOSパーソナリティを利用できないMach環境でセキュリティ・サービスを提供するためには，これらの部分の変更が必要となる．

A Security Mechanism for Distributed Multimedia Environment
Kazunori HORIKIRI [1], Seiji TADA [2], Kiyokuni KAWACHIYA [3], Nobuhiko NISHIO [4], Hideyuki TOKUDA [4], Nobuo SAITO [4],
[1] Fuji Xerox, Systems & Communications Laboratory, 9-14 Naka-cho 4-chome, Atsugi-shi, Kanagawa, 243 Japan [2] Hewlett-Packard Company, Asia Pacific Product Operation, [3] IBM Research, Tokyo Research Laboratory, [4] Keio University,

† この研究は，情報処理振興事業協会(IPA)が実施している開放型基盤ソフトウェア研究開発評価事業「マルチメディア統合環境基盤ソフトウェア」プロジェクトのもとに行なわれた．

‡開放型基盤ソフトウェア湘南藤沢キャンパス研究室の研究員としてIPAに登録されている．

3 セキュリティ・サービスの構成

Keio-MMPのセキュリティ・サービスではUnix等の分散環境で用いられているKerberos認証システムをMach環境に対応させたものをベースに用いる．

Keio-MMPセキュリティ・サービスは，以下のモジュールにより構成される．

- ファイル・アクセス制御機構
- 認証サービス，チケット交付サービス
- 鍵配布サービス
- チケット共有シェル
- セキュア・ネット・メッセージ・サービス(ネット・ネーム・サービス)
- アクセス制御，プライオリティ管理ライブラリ

以下ではこれらのモジュールの機能について述べる．

3.1 ファイル・アクセス制御機構

認証サービスで扱われる秘密鍵やプログラム・コードに代表されるセンシティブな永続データはディスク上に保持管理する必要がある．これらのデータを破壊等の攻撃から守るためには，ディスクに対するアクセス制御の基本機能を用意する必要がある．

Machではデバイスはデバイス・マスター・ポートから取得されるポートとして与えられるため，これらのポートを安全に管理することによりセンシティブな永続データに対する基本的な保護機能が実現できる．

センシティブなリソースに対するポートを管理する方法には，以下に挙げるような方式が考えられる．

- 特定のサーバのみがリソースのポートを保持しリソースに対するアクセスはサーバ経由で行なう(リソース・マネジャ)
- リソースのポートを配布するサーバを用意し，安全であると判断した相手にポートを配布する

センシティブなリソースを操作するポートは特定のタスクのみが保持した方がより安全であるため前者の方式を採用することとする．

3.2 認証サービス，チケット交付サービス

認証サービス，チケット交付サービスはUnix環境のKerberosで用いられるものをMach環境に対応するように，プロセス間通信をソケットからMach IPCに，ファイル・アクセス制御をUnix環境から独立させ前述の方法に変更したものである．

3.3 鍵配布サービス

Unix環境においてサーバ・アプリケーションが秘密鍵を保持するための機構の実現例は以下のようになっている．

ホスト毎にファイルを1つ用意しサーバ・アプリケーションの鍵のリストを保持する．Unixのファイル・ア

クセス制御機構によりファイルをroot権限のプロセスにのみ公開する．サーバ・プロセスをroot権限で動作させる．

以上の機構はUnixのユーザとファイル・アクセス制御機構に依存しており，Unix環境を必ずしも仮定できないMach環境では同じ手法を用いることはできない．

このため，鍵配布サーバを用意しアプリケーション・サーバに安全に鍵を持たせるために用いる．

1. マスタ・ユーザが正当であると判断したアプリケーション・サーバ・プログラムのパス名，プリンシパル名，鍵を管理表に登録する．
   管理表およびアプリケーション・プログラム・コードは，ファイル・サーバによって安全に管理される．
2. アプリケーション・サーバの起動は鍵配布サーバが行なう
3. 各サーバに対して専用の鍵配布ポートを作りサーバにwell-known-portとして付加する
4. サーバが鍵を必要とする場合はwell-known-portを通じて鍵配布サーバに要求する
5. 鍵配布サーバはマスター・パスワードを用いて鍵ファイルを復号化し，配布する

3.4 チケット共有シェル

Kerberosではあるプリンシパルの権限で動作するクライアントはチケットを共有し認証に必要なインタラクションを低減することが可能である．あるプリンシパルの権限で動作するタスク間でチケットを共有するための機構としてチケット共有シェルを用いる

1. チケット共有シェルは起動時に秘密鍵をプリンシパルから貰う(パスワードの入力)
2. チケット共有シェルはチケット交付チケットを取得する
3. チケット共有シェルから起動するタスクにはチケットをやりとりするためのwell-known-portを付加する

3.5 セキュア・ネット・メッセージ・サービス

標準のMach環境で用いられているネット・メッセージ・サービスは以下の2つの点に関して必ずしも安全とはいえない．

- ネーム・スペースの管理
- ネットを跨るIPCのセキュリティ

3.5.1 ネーム・スペース

標準のMach環境で用いられているネーム・サービスでは名前が既に登録されていない限り自由に登録を行なうことができる．この方式ではwell-known-port-nameに偽りの登録が行なわれると，その名前で提供されるべき本来のサービスを受けることができなくなる．セキュア・ネット・メッセージ・サービスではマスタ・ユーザによって予約された名前を管理し，予約された名前に関しては特定のプリンシパルのみが登録を行なうことができる方式を用いる．

3.5.2 ネットを跨るIPCのセキュリティ

前述のようにホストにローカルなIPCはカーネルによって保護されるため安全であるが，ネットワークを跨るIPCに関しては，盗聴あるいは成りすましを行なうことが可能である．これを防止するためにはネット・メッセージ・サーバ間で秘密鍵を共有し内容を暗号化することが必要となる．

このためセキュア・ネット・メッセージ・サービスではメッセージ・サーバ間で認証を行ない，この時に共有される秘密鍵(セッション・キー)を用いてメッセージを暗号化する機構を設ける．

ただし，常にメッセージの暗号化を行なうとオーバヘッドになるため，

- 認証を用いない，内容を暗号化しない
- メッセージ・サーバ間のみで認証を行ない，ネットを跨る通信の内容を暗号化する
- end-to-endのクライアント/サーバがお互いを認証する
- end-to-endでお互いを認証し，通信の内容を暗号化する

の4つのセキュリティ・レベルを用意する．

3.6 アクセス制御，プライオリティ管理ライブラリ

各種のリソース・マネジャやサーバで用いられるアクセス制御機構やプライオリティ管理機構をライブラリとして提供する．

4 おわりに

本稿ではReal-Time Machをベースとした分散連続メディア環境で資源を適切に配分するためのベースとなるセキュリティ機構について述べた．

現在，本稿で述べた設計に基づき実装を行なっている．

参考文献

[1] 徳田，他: “分散マルチメディア統合環境Keio-MMPプロジェクトにおける連続メディア処理のためのソフトウェアアーキテクチャ”，第49回情処全大論文集，7R-1，pp. 3-313–3-314 (1994).

[2] Moriai, S., Kihara, S. and Nambu, A: “Memory Management Mechanism and External Resource Manager Interface for Continuous Media Objects”, in *Mach/Chorus Workshop, OSDI 1994 Winter* (1994).

[3] 追川修一，徳田英幸: “外部スケジューラに基づくマイクロカーネルの構成”，コンピュータソフトウェア，Vol.11，No.5，pp.31–43 (1994)

[4] 河内谷，他: “連続メディアのQOS制御のためのOSサポート”，第6回コンピュータシステム・シンポジウム論文集，pp. 119–126 (1994).

[5] 南部，他: “Keio-MMPにおけるプロトコルアーキテクチャ”，第49回情処全大論文集，6R-3，pp. 2-417–2-418 (1994).

[6] 多田，他: “実時間カーネルを用いた連続メディアベースの設計”，第5回コンピュータシステム・シンポジウム論文集，pp. 33–40 (1993).

[7] Tezuka, H and Nakajima, T: “Design and Implementation of a Continuous Media Storage System on Real-Time Mach”, *JAIST Research Report, IS-RR-94-15S*, JAIST (1994).

[8] Steiner, J. G., Neuman, C. and Schiller, J. I.: “Kerberos: An Authentication Service for Open Network Systems”, in *Conf. Proc. USENIX 1988 Winter* (1988).