マイクロカーネルLavenderにおけるセキュリティ機構の導入

2F−5

寺澤 弘泰† 芝 公仁† 毛利 公一† 斎藤 彰一†† 大久保 英嗣†††
†立命館大学大学院理工学研究科 ††和歌山大学システム工学部情報通信システム学科
†††立命館大学理工学部情報学科

1 はじめに

近年，様々な場面でセキュリティ機能の必要性が増加してきており，OSレベルでのセキュリティ機能も必要となってきている．そこで，我々は，現在開発中のマイクロカーネルLavenderへ暗号化を中心としたセキュリティ機能の導入を考えている．Lavenderにおいては，IPCによりネットワークを介して通信する際，通信内容が盗聴や改鼠されるといった危険性がある．特に，改鼠された場合，システムを破壊する恐れがある．そこで，ネットワークを介して通信する際に暗号化を行う機能を実装している．また，機密ファイルの漏洩といった問題も挙げられる．特に移動体計算機など，計算機自体を紛失した場合，マシンに物理的にアクセスすることで，盗聴される可能性がある．場合によっては，ディスクだけ取り出し，他のマシンにマウントさせることも可能である．そこで，ファイルなどは，アクセス制御だけでなく，暗号化して保存しておく必要がある．

Lavenderでは，階層化インタフェースを用いて，リモートへのIPCが発生した場合に通信内容を暗号化する機能と，ファイルサーバなどの各システムサーバのセキュリティ機能を付加することを考えている．

2 暗号化サーバ

データの暗号化/復号は，暗号化サーバが行う．システムサーバとして独立しているため，Lavenderの各機能は，暗号化サーバに処理を要求することで容易に暗号化/復号機能を使用できる．暗号化サーバによる処理方式については，3章で述べる．

暗号化サーバで用いる暗号化アルゴリズムには，対称鍵暗号としてFEAL-8，非対称鍵暗号系としてRSA，ハッシュ関数としてMD5を用意している[1]．FEAL-8は，64ビットの秘密鍵を用いて，64ビットの平文を64ビットの暗号文に組み立てるブロック暗号化方式を採用している．平文を暗号化すると，平文と等しいサイズの暗号文を生成するため，コスト的に優れている．また高速な暗号化が実現できるため，OSレベルでの暗号化に適しているといえる．

3 セキュリティ機構の処理方式

3.1 IPCにおける暗号化

Lavenderで用いられるIPC[2]のうち，ネットワークを介したIPCに限定して暗号化を適用することを考えている．現在のLavenderのIPCは，受信側プロセスのポート番号，転送メッセージ，メッセージサイズなどの情報を含んでいる．セキュリティの向上のために，さらに署名情報とセキュリティ情報を付加し，以下の区別が可能となるように実装している．

- 暗号化の有無
- ローカル通信かリモート通信か
- IPsec[3]が適用されているか否か

暗号化の有無は，セキュリティ情報のフラグを見て判断する．ローカル通信かリモート通信かに関しては，IPCがリモート通信である場合について暗号化を行う．IPsecは，ネットワークにIPsecが適用されていない場合，暗号化を行う．

Lavenderのネームサーバに登録されているデータは，MMUにより保護されている．このことを利用して，認証の際に用いるポート番号に対応したRSAの公開鍵，秘密鍵(以下，公開鍵，秘密鍵)をネームサーバで管理する．ネームサーバは，要求に応じて必要な鍵をメモリ上から取得し，IPCサーバへ配布する．また，メッセージを暗号化するためのFEAL-8の秘密鍵(以下，対称鍵)は，IPCサーバで保持する．なお，Lavenderを動作させるマシンは，動作させる前にあらかじめ解っているものとし，対称鍵は前もってLavenderを動作させるマシン同士で保持しておく．具体的には，以下に示す手順で処理を行う(図1参照)．

1. 受信側プロセスは，IPCサーバに対してポート名を指定してポートの割り当て要求を出す．
2. 送信側プロセスがメッセージを送信しようとする場合，ポート名を指定することにより，ネームサーバと暗号化サーバに受信側のプロセスのポート番号を問い合わせる．

The Security Mechanism in Lavender Micro Kernel

Hiroyasu Terazawa†, Masahito Shiba†, Koichi Mouri† Shoichi Saito††and Eiji Okubo†††
†Graduate School of Science and Engineering, Ritsumeikan University
††Department of Computer and Communication Sciences, Faculty of Systems Engineering, Wakayama University
†††Department of Computer Science, Faculty of Science and Engineering, Ritsumeikan University

3. ネームサーバは，データベースを検索してポート番号，対称鍵，そのポートの秘密鍵を送信側プロセスに返す．
4. メッセージを送信する場合，受信側プロセスのポート番号，メッセージ，サイズを指定してメッセージ送信のシステムコールを発行する．
5. 暗号化サーバは，送信メッセージをMD5にかける．さらに，得られたハッシュ値を，そのポートの秘密鍵を用いてRSAで暗号化し，その結果を署名情報とする．また，送信メッセージを対称鍵を用いてFEAL-8で暗号化する．
6. IPCサーバがメッセージ送信の要求を受け取ると，メッセージはIPCサーバ内のバッファにコピーされ，デバイスドライバを経由してネットワークを通過し，受信側のIPCサーバに到着する．
7. 受信側プロセスは，メッセージの到着通知を受け取る．メッセージの受信を行う場合は，ポート番号を指定して，メッセージ受信のシステムコールを発行する．
8. IPCサーバがメッセージ受信の要求を受け取ると，暗号化サーバは，暗号化されたメッセージを対称鍵で復号する．また，ポート番号に対応した公開鍵で署名情報を復号し，復号したメッセージをMD5にかけて得られたハッシュ値と比較する．一致する場合，改鼠されてないことが分かる．さらに，メッセージを受信側プロセスのバッファにコピーし，アドレスを返す．
9. プロセスが通信を終了する場合，ポート番号を指定し，ポートの解放要求を出す．

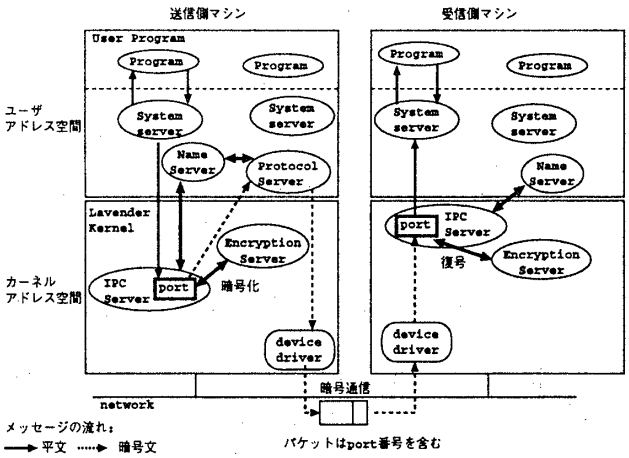

図 1: ネットワークを介したプロセス間通信

3.2 ファイルシステムにおける暗号化

ファイルの暗号化/復号処理は，ハードディスク装置のレベルで行う．この方式では，マシンとディスクとの間で交換されるデータに対して，デバイスドライバが透過的に暗号化/復号処理を行うことでデータを保護する．具体的には，以下に示す手順で処理を行う．

1. デバイスをオープンする際，暗号化サーバは，対称鍵を配布する．
2. デバイスドライバは，暗号化サーバに問い合わせ，マシンとディスク間でデータがwriteされるとき暗号化し，readされるとき復号する．
3. デバイスをクローズする際，対称鍵は暗号化サーバに保存される．

この方式では，デバイスをオープンする際に，暗号化サーバが一回だけ対称鍵を供給すれば，その後は，ユーザがこれらの処理を意識する必要がない．また，対称鍵は正規のユーザにのみ配布されるので，物理的なアクセスを加えられてもファイルが暗号化されているため盗聴できない．

3.3 システムサーバにおけるセキュリティ機構

暗号化サーバは，システムサーバとして独立しているため，IPCだけでなくその他のシステムサーバへも容易に適用できる．例えば，ネームサーバでは，アクセス制御と認証に関して導入を検討している．アクセス制御では，名前の登録の際に，許可されたアクセスのみのサービスを提供する機能である．認証では，名前の変更や問い合わせを依頼したクライアントやサーバが正規のものであるか否かを暗号化サーバに問い合わせることにより実現する予定である．

4 おわりに

マイクロカーネルLavenderにおけるセキュリティ機構の設計を行った．この機構を導入することによって，ネットワーク通信の際のデータの改鼠の防止や機密情報の保護など，セキュリティ強化を図ることが可能となる．また，暗号化サーバで暗号化/復号サービスを提供するため，暗号化機能を利用するシステムサーバは，暗号化サーバへ問い合わせるだけで容易に利用でき，汎用性も高い．

今後の予定として，リカバリ機能の導入を考えている．リカバリ機能では，リカバリ鍵によって，ネームサーバが鍵を紛失した場合や，暗号化ファイルシステムで鍵を使用しなくなった場合に，暗号化されているファイルを復号し，見ることが可能となる．また，強度的な面や，用途に応じた暗号化を行うため，複数の対称鍵暗号アルゴリズムの実装と，暗号化機能を用いることによって，すべての暗号化にかかる時間が実用に耐え得るか否かに関する考察と実験を行う予定である．

参考文献

[1] 辻井重男, 笠原正雄: “暗号と情報セキュリティ，” 昭晃堂 (1990).

[2] 豊岡明, 芝公仁, 佐脇秀登, 毛利公一, 大久保英嗣: “マイクロカーネルLavenderにおけるIPC機構とデバイスドライバの構成，” 情報処理学会研究報告 97-OS-76, Vol.97, No.77, pp. 49-54 (1997).

[3] R. Atkinson: “Security Architecture for the Internet Protocol,” RFC1825(1995).