暗号化ファイル共有システムのセキュリティ機構

1D−2

新保 淳　室田 真男　高橋 俊成

(株)東芝　研究開発センター

1　はじめに

文献 [1] にて概要を報告する非同期型のファイル共有システムでは、暗号技術の導入によりファイルサーバに対する内容保護を実現しながら、しかもファイルアクセス時にロックをかけない非同期型の同時編集を可能にする。技術的な特徴は、ファイル内容を認識できないサーバが複数のクライアントによる編集結果のマージを行なう“暗号化マージ処理”を実現できる点にある。

本文では、この機能の実現に関わる共有ファイルのデータ構造、暗号方式、暗号化マージ処理手続きを報告し、さらに、認証メカニズムについても説明する。

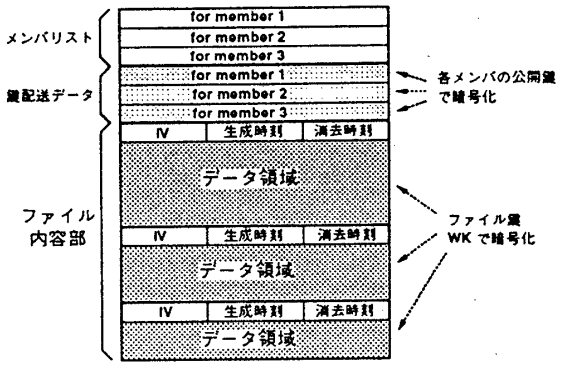

図 1: 共有ファイルの暗号化に関わるデータ構造

2　暗号方式とデータ構造

本システムのセキュリティ上の特徴は、クライアント・サーバ型のファイル・システムにおいてサーバに対する内容保護を実現し、そのうえサーバにファイルの保管作業以上の処理(暗号化マージ処理)を担当させることにある。

内容保護は暗号化によって行なうが、本システムでは“ファイルの構造は見せるが内容は見せない”方針により共有ファイルを暗号化する。

共有ファイルのデータ構造を図1に示す。

ファイル内容部は、複数のブロックから成り、マージ処理と履歴管理に適したデータ構造 [1] を構成している。先の暗号化方針により、個々のブロックをブロック単位に個別に暗号化する一方、生成時刻や消去時刻を表すタグ領域は平文のままとする。この方式の利点は次の通りである。

- 暗号化マージ処理(後述)をサーバが実行できる
- 所望のバージョンに相当するブロックのみを通信すればよく、効率化が図れる

マージ処理 [1] では既存ブロックに対し、その任意の位置での既存ブロックの分割、任意の位置への新規ブロックの挿入を行なう必要がある。こうした処理を暗号化された状態でサーバが実行可能とするためには、平文の1キャラクタが暗号文の1キャラクタに対応し、しかも分割した断片のブロックがほぼそのまま復号可能とならなければならない。さらに、暗号強度を保証する必要もある。このような性質を満たす暗号方式にストリーム暗号の一種である“ブロック暗号の8ビットCFBモード”があり、本システムではファイル内容部の暗号方式としてこれを採用する。このモードの詳細は例えば文献 [2] を参照されたい。この暗号方式ではパラメータとして暗号化初期値IV(Initial Value)が必要となるため、これをブロックごとに設定する。なお、ブロック暗号としてはDES(Data Encryption Standard)などが利用できる。

ファイル内容部は、ファイルごとに固有の鍵(ファイル鍵WK)で暗号化される。ファイル内容の読み取りを許可されたユーザ(メンバ)のみがWKを取得可能とするため、公開鍵暗号による鍵配送メカニズムを次のように導入する。

ユーザU_iには、個別の秘密鍵SK_iと公開鍵PK_i(これらは公開鍵暗号の鍵)が割り当てられている。共有ファイルには個々のメンバに対する鍵配送データの領域が存在し、U_i用の領域には、ファイル鍵WKを公開鍵PK_iで暗号化したデータが記録される。U_iはこれにアクセスし、復号することでファイル鍵WKを得られる。

鍵配送データの作成およびファイルの暗号化は以下のようにして行なわれる。まず、最初に共有ファイルを生成したユーザがメンバ登録(アクセス可能なユーザの定義)、ファイル鍵の生成、全メンバの鍵配送データの生成、初期バージョンのデータ生成／暗号化を実行する。さらに、メンバの追加や削除の権限を持つ“管理者”をメンバの中に設ける。例えば、メンバ全員が管理者でもよい。管理者はメンバの追加・

Security Mechanism of Privacy Enhanced File Sharing System
Atsushi SHIMBO, Masao MUROTA, Toshinari TAKAHASHI
Toshiba Corporation, Communication and Information Systems Research Labs.
1, Komukai-Toshiba-cho, Saiwai-ku, Kawasaki 210, Japan

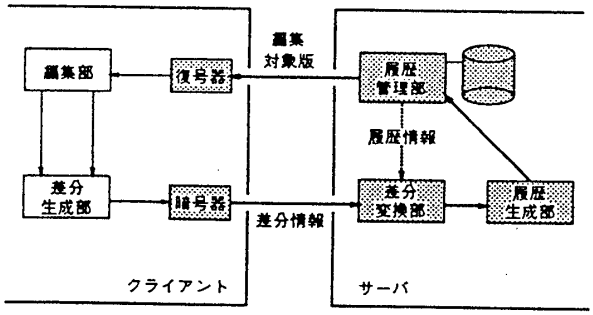

図 2: 暗号化マージ処理の実装

削除を任意の時点で実行でき、以下のようにする。

メンバの追加 メンバリストに追加メンバのIDを加える処理と追加メンバ用の鍵配送データを作成する処理を実行。

メンバの削除 メンバリストと鍵配送データから該メンバ部分を消去する。さらに安全性を高めるためにはファイル鍵の更新、更新されたファイル鍵によるファイル内容部の暗号化、鍵配送データの更新の処理が必要となる。

3 暗号化マージ処理

暗号化マージ処理は図2に示した諸機能により実現される。

履歴管理部 履歴管理されたファイルを保管し、要求に応じて所望のバージョンの提供を行なう機能部

編集部 編集対象のバージョンを取得し、それに対する編集行為を行ない編集結果のバージョンを生成する機能部

差分生成部 編集部出力から、編集行為を表す差分情報(diff)を生成する機能部

差分変換部 差分情報を入力とし、履歴管理部の履歴情報を参照して共有ファイルの現行版に対する差分情報に変換する機能部

履歴生成部 差分情報を入力とし、履歴管理方式に適合した履歴データを生成し、現行版を更新する機能部

図2においてハッチングされたモジュールの部分は暗号化された状態で実行される。

クライアントからサーバに送る差分情報は、版ID, 挿入データ, 削除データの組である。挿入データは挿入位置と挿入ブロックから成るが、挿入ブロックは図1におけるファイル内容部の1ブロックそのものであり、クライアント内の暗号器によりデータ領域は暗号化され、IVも設定される。また、削除データは削除開始位置と削除終了位置から成る。挿入位置や削除範囲は暗号化されず、挿入ブロックの内容だけが暗号化されていることに注意。

マージ処理に必要な差分情報の挿入位置、削除開始位置、削除終了位置を現行版に対する適切な位置に変換する処理は、先に説明したように暗号状態で実行可能である。従って、サーバは以下の処理を実行すればよい。

- 挿入データの処理
  挿入位置の変換後、挿入位置で既存暗号ブロックの切断処理を行ない、新規挿入ブロックを入れる。生成タグに時刻を記録する。
- 削除データの処理
  削除範囲の変換後、削除開始位置と削除終了位置でブロックを分割する。削除タグに時刻を記入する。

4 認証機構

ユーザU_iには個別の秘密鍵SK_iが発行されているため、その秘密鍵の有無を確認する方法によりユーザ認証を行なえる。

本システムでは鍵配送とメッセージ認証を次のように組み合わせ、コマンド単位に認証する方式を用いる。

- クライアントU_iから共有ファイル参照の開始コマンドを受けたサーバは、セッション内で有効な認証鍵を生成し、クライアントの公開鍵PK_iで暗号化して鍵配送する。
- クライアントでは認証鍵を取得し、セッション内のコマンドにはメッセージ認証コード(MAC)を付ける。例えばハッシュ関数により認証鍵とコマンドの引数を圧縮したデータをMACとする。
- サーバは認証鍵によりメッセージ認証コードを検査し、検査に通ればコマンドを受理する。
- セッション終了時にはサーバ側で認証鍵が消去される。

この方式では、最初の鍵配送の処理量が多いものの、メッセージ認証コードは高速に生成できるため、コマンド単位でのオーバヘッドは小さい。

5 おわりに

暗号によるファイルデータの保護機能と非同期型の共有ファイルを融合させ、暗号化マージ処理という新たな特徴を有するシステムのセキュリティ機構を示した。現在試作を進めている段階である。実装上の課題として、暗号機能(特に公開鍵暗号部分)のオーバヘッドをさらに軽減する必要性が生じる可能性がある。

参考文献

[1] 高橋, 新保, 室田, "暗号化ファイル共有システムの概要", 情報処理学会第49回大会、1D-3、1994 (本大会にて発表の予定).

[2] 池野, 小山, "現代暗号理論", 電子情報通信学会.