マルチメディア通信と分散処理　109－11
電子化知的財産・社会基盤　17－11
（２００２．９．１３）

セキュアマルチキャストのためのFECを用いた鍵配送方式

藤井亮治
静岡大学大学院
情報学研究科

佐藤文明
静岡大学情報学部

グループ通信サービスでのプライバシー保護のため，全順序マルチキャストプロトコルに基づいた全メンバでの同期も考慮したグループ鍵更新方法が提案されている．しかし，このフレームワークでは，実際の通信で起こりうるパケットのロスが考慮されていない．鍵配送の際にパケットがロスすれば，メンバからの鍵要求が多くなり，その鍵要求がシステム全体のボトルネックになってしまう．そこで，本稿では鍵配送にFECの機能を追加し，鍵配送の際のパケットロスに対応できるよう改良した．また，全順序マルチキャストプロトコル上での性能を評価するためシミュレーションを行った．

The Session Key Distribution using FEC for Secure Multicast

Ryouji Fujii
Graduate School of Imformation,
Shizuoka University

Fumiaki Sato
Faculty of Information
Shizuoka University

To keep the privacy of the group communication, the group key renewing method based on the Totally Ordered Multicast Protocols has been proposed, which is synchronous among members. However,in this framework, the packet loss in the actual communication is not taken in to consideration. If the packet loss of key delivery increase, key request messages increase and becomes bottle neck of the system. In this paper, we propose the FEC combined method for key delivery and improve the key delivery ratio. We also present the performance evaluation by the simulation with AMP, which is a Totally Ordered Multicast Protocol that we have developed.

1　はじめに

近年，コンピュータネットワークの発達により分散アプリケーションの使用が一般的になってきた．分散アプリケーションの設計において，全順序グループ通信サービスは効果的かつ必要不可欠となっている．また，全順序グループ通信サービスでのプライバシー保護のため，全順序マルチキャストプロトコルに基づいた，全メンバでの同期も考慮したグループ鍵更新方法が提案されている．その方式では，マルチキャストグループは複数のメンバと１つの鍵配送サーバ(KDS)を持つ複数のドメインで構成されており，KDSは，自身の管理するドメイン内のメンバにグループ鍵を配送する責任を持つ．この方式は，いくつかの前提を満たせばどのような種類の全順序マルチキャストプロトコルにも適用することが可能である．

ただし，この方式では，実際の通信で起こりうるパケットのロスを考慮していない．グループ鍵配送の際にパケットがロスすれば，メンバからのグループ鍵要求が多くなる．そして，そのグループ鍵要求がボトルネックになってしまい，システム全体の性能がダウンしてしまう．そこで，本研究では鍵配送にFECの機能を追加し，パケットロスのある環境下で効率的にグループ鍵を配送できるよう，プッシュに基づくグループ鍵配送方式を改良した．また，全順序マルチキャストプロトコル上での性能を評価するためシミュレーションを行った．

2 研究背景

2.1 グループ鍵更新における同期

鍵配送サーバ (Key Distribution Server:KDS) が新しいグループ鍵を生成し，メンバに配送する時には，全てのメンバがグループ鍵の更新を受信し，同期して更新していることに注意しなければならない．なぜなら，グループ鍵の更新ができなかったメンバは，以下のような一時的なセキュリティ違反を発生させる可能性を持つからである．

- グループ鍵の更新に失敗した受信者は，以降の通信を復号化し続けることができなくなる．
- グループ鍵の更新に失敗した受信者は，すでにグループから削除されているメンバからの通信も受け入れてしまう可能性がある．
- グループ鍵の更新に失敗した送信者は，すでに使用されていない，すなわち，他の受信者が復号化することができないようなグループ鍵を用いてメッセージを暗号化し続けてしまう．
- グループ鍵の更新に失敗した送信者が古いグループ鍵で暗号化したメッセージは，すでに脱退したメンバがメッセージを復号化できてしまう．

マルチキャストグループ内でこのような問題が発生したなら，そのグループ内のセキュリティは非常に危うくなる．

このような同期に関する問題に対処するため，2相コミットプロトコル [2] などのプロトコルが考案されている．しかし，このプロトコルはスケーラビリティが低く，動的なメンバーシップを持った大きなグループには適応できない．なぜなら2相コミットプロトコルでは，プロトコルのコーディネータは全てのメンバ間でメッセージを交換し，インタラクションの間，システムを停止させなければならないからである．

2.2 全順序マルチキャストプロトコル

全順序マルチキャストとは，全ての受信者において，同一の順序でメッセージをアプリケーション

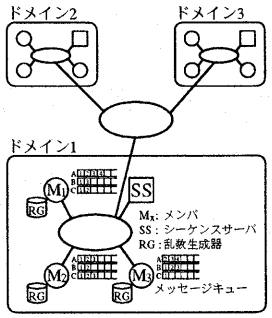

図 1: 適応的マルチキャストプロトコル

へと配送することを保証する方式である．全順序マルチキャスト方式の例をいくつか以下に示す．

シーケンスサーバ方式 [3] は，グループ内にシーケンスサーバを1つ以上設置する．クライアントはメッセージを送信する際，シーケンスサーバにシーケンスナンバーの発行を依頼し，発行されたシーケンスナンバーをマルチキャストメッセージに付与して送信する．メッセージに付与されたシーケンスナンバーでメッセージを順序づけることでマルチキャストグループ内の全順序性を保証する方式である．

適応型全順序マルチキャスト方式 [4] は，全てのクライアントで同一の乱数生成器と重みベクトルを使用して全順序性を保証する方式である．受信者は送信者毎に用意したバッファにメッセージを貯め，乱数と重みベクトルによってメッセージを取り出すバッファを選択する．全メンバで同一の乱数生成器を用いて生成した乱数と重みベクトルを用いることで，全てのクライアントが同じ順序でメッセージをバッファから取り出すことができ，マルチキャストグループ内の全順序性を保つことができる．

従来の全順序マルチキャストのスケーラビリティを高めるために，シーケンスサーバ方式と適応型全順序マルチキャスト方式を組み合わせた方式が提案されている．クライアントを複数のドメインに分け，それぞれのドメインにシーケンスサーバを1つずつ設置する．メッセージ送信時にクライアントは，自分のドメインにあるシーケンスサーバにシーケンスナンバーの発行を依頼し，発行さ

れたのシーケンスナンバーを付与してメッセージを送信する．全てのクライアントはドメイン毎にバッファを用意しておき，シーケンスナンバーと送信元ドメインにしたがって，メッセージをバッファに格納し，並び換える．また，全メンバで同一の乱数生成器を用い，その乱数生成器で生成した乱数と，ドメイン毎の送信頻度を示す重みベクトルを用いることで，全てのクライアントが同じ順序でメッセージをバッファから取り出すことができ，全順序性を保つことができる．これが適応的マルチキャストプロトコル (AMP)[5] である (図 1)．

3 階層的な鍵管理とプッシュに基づくグループ鍵配送

安全なマルチキャストに関するグループ鍵の配送や更新の方式には，同期に注意を払ったプロトコルは少なく，効果的で停止のない更新方式は皆無である．特に，リアルタイム性を重視されるインタラクティブなアプリケーションには，停止のない更新が必要である．

ここでは，高信頼全順序マルチキャストプロトコル (Reliable Totally Ordered Multicast Protocol:RTOMP) に基づいたグループ鍵更新方式と，プッシュに基づいた鍵配送方式 [6] について述べる．

この方式では，グループ鍵の配送コストを減少させるために，Wong[1] らによって提案された階層的な鍵管理を導入しており，1 つのグループ鍵と複数のドメイン鍵を用いる 2 層で鍵を管理する単純な階層的鍵管理方式を用いている．さらに，効率的にグループ鍵を配送するため，この方式では KDS がドメイン内の安全なマルチキャスト通信路を介してドメイン内の全てのメンバにグループ鍵を配送する．この方式のアーキテクチャを図 2，図 3 に示す．

この方式の利点には，スケーラビリティに優れていることや，各 KDS が非同期にグループ鍵を生成するため，グループ鍵の更新時の同期を取る必要がなくなることが挙げられる．また，この方式はいくつかの前提条件さえ満たせばどのような RTOMP にも適用可能である．この方式の欠点は，メンバと KDS は複数のグループ鍵を管理する必要があるということである．グループ鍵は使用されなくなった時点で消去する必要があり，この管理が少々複雑になる．

3.1 前提条件

この方式では以下の条件を前提とする．

- マルチキャストグループは複数のドメインから成り立っている．また，ドメインは複数のメンバと 1 つの信頼できる鍵配送サーバ (KDS) を持っている．
- それぞれのドメインごとにマルチキャスト通信路が存在する．
- KDS 間の通信路として安全な RTOMP 通信路が存在する．これは，KDS の認証のための信頼できるサーバが別に存在し，そのサーバが発行した秘密鍵を使用するなどの方法で確保する．KDS はメンバほど頻繁に参加，脱退しないという前提があるならば，従来のグループ鍵配送方式でも十分安全な RTOMP 通信路を保つことができる．
- それぞれのドメイン内のメンバは 3 種類の鍵，グループ鍵とドメイン鍵，個別鍵を管理している．マルチキャストグループは 1 つのグループ鍵を共有していて，この鍵によって通信を暗号化することでマルチキャストグループ全体への通信を安全にする．ドメイン鍵は，KDS がドメイン内にマルチキャスト通信路を介してグループ鍵を安全に配送するために用いられる．個別鍵はメンバとドメインの KDS との間で共有されている鍵であり，メンバと KDS の間のユニキャスト通信の暗号化に使用される．

プッシュに基づくグループ鍵配送における鍵の配送，更新の手順を以下に示す．

1. KDS は，自身の管理しているドメインのメンバーシップが変更する時に，新しいグループ鍵とドメイン鍵を生成する．
2. KDS は KDS 間の安全な通信路を介して他の全ての KDS にグループ鍵を配送する．

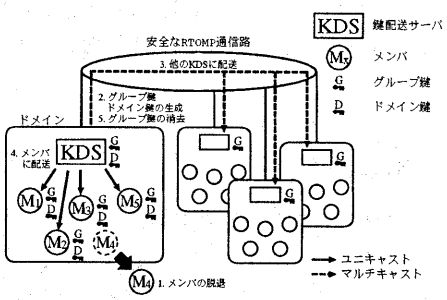

図 2: プッシュに基づくグループ鍵配送 (自ドメインで鍵を生成する場合)

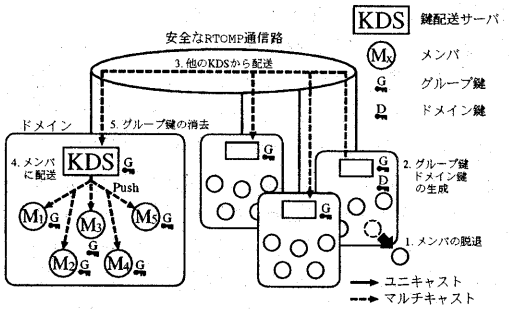

図 3: プッシュに基づくグループ鍵配送 (他ドメインで鍵を生成する場合)

3. 新しいグループ鍵を受け取ったKDSは，ドメイン内のマルチキャスト通信路を介してドメイン内の全てのメンバにグループ鍵を配送する．この時，グループ鍵をドメイン鍵で暗号化することにより，ドメイン内のマルチキャスト通信路の安全性が確保される．

4. 新しい鍵を生成したKDSは個別鍵で暗号化したグループ鍵とドメイン鍵を自身のドメインの全てのメンバにユニキャストで送信する．
以上のように，グループ鍵の生成と配送のためのコストは全てのKDSに分散される．また，全てのグループ鍵の配送はRTOMPによって順序が保たれているため，KDS間でのグループ鍵の更新順序に矛盾が生じることはない．

5. メンバはある間隔毎にKDSに対して自身の持っている最新のグループ鍵識別子を通知する．もしグループ鍵識別子が古ければ，KDSは最新までのグループ鍵をメンバに対して送信する．
この確認の間隔は，例えば，メンバがメッセージを送信する時に毎回と，少なくともT秒毎などカスタマイズすることが可能である．

6. メッセージ送信時に，メンバは最新のグループ鍵があるかどうかをKDSに問い合わせ，あればそのグループ鍵を受信する．メンバは送信するメッセージを最新のグループ鍵で暗号化した後に，そのグループ鍵識別子と自身の所属するドメイン識別子を付与し，グループ全体にマルチキャストする．

7. メンバが自信の持っている鍵で復号化できないメッセージを受信した時，メンバはKDSに必要なグループ鍵の識別子を通知し，必要なグループ鍵を要求する．要求を受け取ったKDSはグループ鍵を返す．

この階層的鍵管理方式やプッシュに基づくグループ鍵配送は，他のドメインに影響を与えないため，ドメインごとに個別に導入することができる．例えば，あるドメインのKDSの性能が悪く，そこがシステム全体のボトルネックになっているような場合には，そのドメインのみ階層的鍵管理方式やプッシュに基づくグループ鍵配送を導入することで，システム全体のボトルネックを改善することができる．

3.2 グループ鍵の管理

KDSは，他の全てのKDSと自ドメインのメンバにグループ鍵を配送し終えたときに，その鍵を消去することができる．言い換えると，KDSは他の全てのKDSと自ドメインのメンバにグループ鍵を配送する義務があるといえる．

また，メンバは全てのドメインからのメッセージが新しいグループ鍵に移行したことを確認する

と，その鍵を消去することができる．これは，全てのドメインからのメッセージに付与されているグループ鍵識別子を記憶しておくことで判断できる．

グループ鍵を消去するタイミングは，メンバが全てのドメインから新しいグループ鍵で暗号化されたメッセージを受信してからT+t秒後である．ここで，Tはメンバがグループ鍵が新しいことを確認する間隔であり，tはネットワークの遅延時間である．

もし送信者がメッセージ送信時に毎回グループ鍵をチェックしているのならば，メンバはt秒後に古いグループ鍵を消去することができる．さらに，もしメンバ間のマルチキャスト通信路が全てのドメインからのメッセージの順序を保っているならば，メンバは全てのドメインから新しいグループ鍵で暗号化されたメッセージを受信したときに古いグループ鍵を消去することができる．

4 鍵配送方式のFECを用いた改良

プッシュを用いたグループ鍵配送方式では，ドメイン毎のマルチキャスト通信路でのパケットロスを考慮していない．鍵配送の際にパケットがロスすれば，メンバからの鍵要求が多くなり，その鍵要求処理がボトルネックになってしまい，システム全体の性能がダウンしてしまう．そこで，実際のネットワーク上においても確実かつ効率的なグループ鍵配送を行なうため，この方式にFECを取り入れる．

この方式はプッシュに基づくグループ鍵配送方式を元にしており，各KDSが非同期でグループ鍵を生成するため，グループ鍵更新時に同期をとる必要はない．また，この方式も他のドメインに影響を与えないため，ドメインごとに個別に導入することができる．例えば，あるドメイン内でパケットロスが頻繁に起こり，そのドメインがシステム全体のボトルネックになっている場合，そのドメインのみにFECを用いた鍵配送を導入することで，システム全体のボトルネックを改善することができる．

4.1 FEC(Forward Error Correction)

FECは本来，不安定なネットワークや，動画や音声のリアルタイム配送のように，再送では対応できない場合に用いられる方法である．今回はこのFECを鍵配送に用いることで，パケットロスのある環境下での鍵配送を確実かつ効率的に行なえるようにする．

FECには大きく分けて，pro-active FECとre-active FECの2つの方式がある．それらの方式について以下で説明する．

- pro-active FEC
  送信者はあらかじめ，修復用のFEC冗長パケットを送信すべきオリジナルパケットに付加して送信する．受信者は全てのパケットを受信できなくても，修復用の冗長パケットから，元のオリジナルパケットを復元できる．

- re-active FEC
  送信者は，オリジナルパケットをそのまま送信する．受信者は足りないパケット数をNACKとして送信者に返信する．送信者はそれらのNACKから必要な数だけ修復パケットを生成し，送信する．

re-active FECでは，オリジナルパケットを全て受信できなかった受信者全員がKDSへ再送要求を送信するので，KDSに再送要求が集中してしまい，スケーラビリティに欠ける．したがって，あらかじめ冗長パケットを付与し，再送要求を少なくすることのできるpro-active FECをプッシュによるグループ鍵配送に導入する．

4.2 FECを用いた鍵配送

プッシュによるグループ鍵配送方式におけるパケットロスに対応するため，オリジナルパケットに冗長パケットを付加して送信する．これにより，メンバは全てのパケットを受信できなくても，オリジナルパケットを復元することができ，パケットロスのある環境下でも確実かつ効率的にグループ鍵を配送することができる．

FECを用いた鍵配送の手順を以下に示す(図4)．

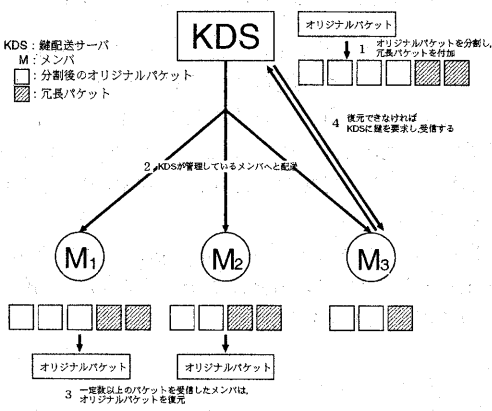

図 4: FEC を用いた鍵配送

1. 自身の管理するドメインのメンバーシップの変更によって，KDS は新しいグループ鍵とドメイン鍵を生成する．

2. 新しいグループ鍵を生成した KDS は，KDS 間の安全な通信路を介して他の KDS にグループ鍵を配送する．

3. 新しいグループ鍵を受け取った KDS は，ドメイン内のマルチキャスト通信路を介して，そのドメインで用いられているドメイン鍵で暗号化されたグループ鍵をドメイン内の全てのメンバに配送する．このとき，KDS は送信パケットを複数に分割し，分割したパケットから冗長パケットを生成し，それら全てを送信する．メンバはそれらのパケットのうち，一定数のパケットを受信できれば元のパケットを修復できる．一定時間以内に元のパケットに修復できなかったメンバは KDS に鍵要求を送信する．

4. 新しい鍵を生成した KDS は，新しく生成したグループ鍵とドメイン鍵を，個別鍵で暗号化して自身のドメインの全てのメンバにユニキャストで送信する．

5. メンバはある間隔毎に KDS に対して自身の持っている最新のグループ鍵識別子を通知する．もし古ければ，KDS は最新までのグループ鍵をメンバに対して送信する．

6. メッセージ送信時に，メンバは最新のグループ鍵があるかどうかを KDS に問い合わせ，あればそのグループ鍵を受信する．メンバは送信するメッセージを最新のグループ鍵で暗号化した後に，そのグループ鍵識別子と自身の所属するドメイン識別子を付与し，グループ全体にマルチキャストする．

7. メンバが自身の持っている鍵で復号化できないメッセージを受信した時，メンバは KDS に必要なグループ鍵の識別子を通知し，必要なグループ鍵を要求する．鍵要求を受け取った KDS はグループ鍵を返す．

5 シミュレーション

5.1 シミュレーションモデル

全順序マルチキャストプロトコルにおいて，この方式の性能を評価するため，シミュレーションを行った．パラメータを以下のように設定する．

- ドメイン数 4
- オリジナルパケットのブロック数 4
- メッセージ送信間隔 1s
- グループ鍵更新間隔 1s
- ユニキャストの通信遅延 100ms
- マルチキャストの通信遅延 300ms
- 暗号化/復号化にかかる時間 50ms
- FEC のエンコード/デコードにかかる時間 50ms

このような場合について，ロス率を 5，7%，ドメイン毎のクライアント数を 1 から 200 に変化させ，その時のヒット率と，KDS の平均鍵リスト長を計測した．いずれも FEC を用いない場合と，1，2，3 個の冗長パケットを付与した FEC を用いた場合について計測している．

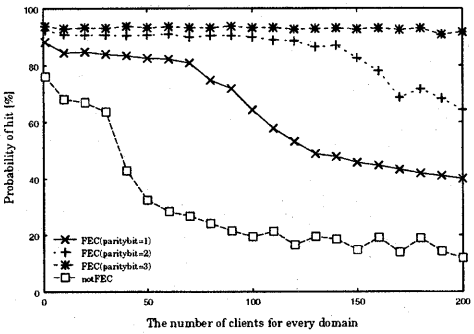

図 5: メンバの平均ヒット率 (ロス率 5%)

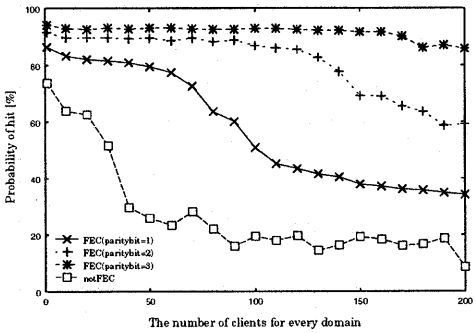

図 6: メンバの平均ヒット率 (ロス率 7%)

FEC のコード化は，コード化された n ブロックを作るため，ソースデータの k ブロックがコード化されるような (n,k) コードと呼ばれるコード化 [7] を用いるとする．このコード化では，n のコード化ブロックの集合のうち，n-k のロスから受信者が回復できる．

5.2 結果

図 5，図 6 にメンバの平均ヒット率を示す．これはメンバがメッセージを復号化するとき，グループ鍵を要求せずに，持っているグループ鍵で復号化できる割合を示している．FEC を使わない場合には，プッシュによるグループ鍵配送で鍵を受信できないメンバが多く，ドメイン毎のメンバ数がさほど多くなくても，ヒット率は低い値になってしまう．これに対して FEC を使った場合には，パリティパケットの数が多い程，メンバの増加による影響を受けにくくなり，ヒット率は高い値を維持できるようになる．すなわち，FEC を使えばほとんどのグループ鍵配送がプッシュによる配送で完了するため，効率良く確実にグループ鍵が配送されていると言える．

また，メンバはメッセージを復号化できないとき，KDS へグループ鍵要求を送信し，必要とするグループ鍵を受信するまで配送をストップさせる必要がある．すなわち，ヒット率が高いということは，メンバがマルチキャストメッセージを復号化する際，グループ鍵要求の応答を待つことが少なくなるので，グループ鍵の配送に FEC を用いれば，メンバは効率よくメッセージを復号化できているとも言える．

図 7，図 8 に KDS の平均グループ鍵リスト長を示す．これは，ドメイン内のメンバがまだグループ鍵を受信できていないため，KDS が蓄えておかなければならないグループ鍵の個数である．FEC を用いなければ，メンバがプッシュによる配送でグループ鍵を受信できないため，少ないメンバ数でもグラフは上昇してしまう．しかし FEC を用いた場合は，パリティパケットの数が多い程，メンバ数が多い場合にも少ない値を維持できる．

また，メンバ数が増加すると，プッシュによるグループ鍵配送で鍵を受信できないメンバが増加し，単純にグループ鍵要求の数も増加する．そのため，FEC を用いた場合でもメンバ数の増加と共にグラフは上昇してしまう．そして，メンバ数がある値を越えると，KDS の処理性能がボトルネックとなってグループ鍵要求を処理できなくなり，グラフが急激に上昇してしまう．

シミュレーション結果から，FEC を用いたグループ鍵配送方式では，FEC を用いない場合と比べ，明らかにスケーラビリティが向上していることが分かる．しかし，ドメイン内のメンバ数が非常に多くなる場合には，KDS の処理性能がボトルネックとなりシステム全体の性能を落してしまう．この問題は，大きなドメインを小さなものに分割したり，複数のグループ鍵更新を一度にまとめて，グループ鍵の更新頻度を下げることで改善できる．

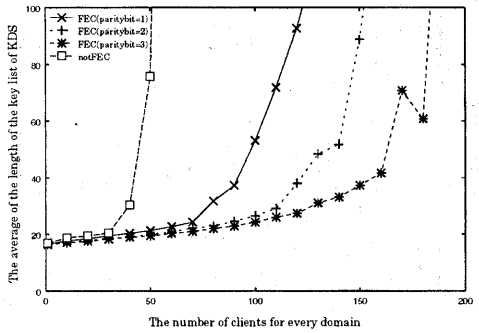

図 7: サーバの平均鍵リスト長 (ロス率 5%)

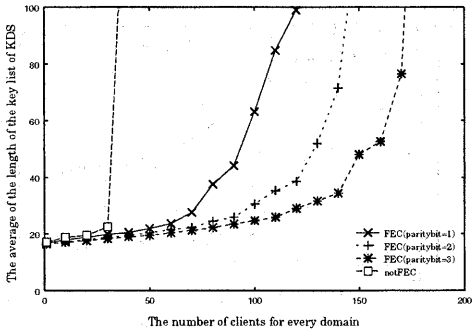

図 8: サーバの平均鍵リスト長 (ロス率 7%)

6 おわりに

本稿では，マルチキャスト通信におけるセキュリティを確保するための全順序マルチキャストプロトコルに基づいた非同期のグループ鍵更新方式にFECの機能を追加することで，グループ鍵配送の際のパケットロスに対応できるような拡張を行い，シミュレーションにより性能の評価を行なった．

この方式では，グループ鍵をメンバに配送する時，元のオリジナルパケットに冗長パケットを付与して送信する．グループ鍵を受信するメンバは，全てのパケットを受信できなくても，冗長パケットを用いることでオリジナルパケットを復元でき，受信できなかったメンバも改めてグループ鍵を要求することで最新のグループ鍵を得ることができる．

今回のシミュレーションにおいて，パケットのロス率は一定で行なった．しかし，実際のネットワーク上でのロス率は変化する．ロス率の低い環境で冗長パケットをたくさん付与すれば帯域を圧迫しまう．また，ロス率が高い環境で冗長パケットが少なければほとんどのメンバがオリジナルパケットを復元できなくなる．このようなロス率の変化に対応できるよう，動的に冗長パケットを変化させる方法を考慮することが今後の課題である．

参考文献

[1] C. K. Wong, M. Gouda, and S. S. Lam, "Secure Group Communications Using Key Graphs," IEEE/ACM Trans. on Networking, vol.8, no.1, pp.16-30, 2000.

[2] 朴 美娘, 渡辺 晃, 岡崎 直宣, 井手口 哲夫, "セキュアマルチキャスト通信方式に関する検討," マルチメディア, 分散, 協調とモバイル (DICOMO) シンポジウム, 1998.

[3] 南端邦彦, 佐藤文明, 水野忠則, "共有仮想環境のための高信頼マルチキャスト方式の提案と評価", 情報処理学会論文誌 Vol.41,No.2,pp.254-261, (2000).

[4] G.V.Chockler, N.Huleihwl, D.Dolev, An Adaptive Totally Ordered Multicast Protocol that Tolerates Partitions, The 17th ACM Annual Symposium on Principles of Distributed Computing(1998).

[5] 細谷 篤, 佐藤 文明, 水野 忠則, "適応的全順序マルチキャストの拡張," 情報処理学会論文誌, Vol. 42, No. 2, pp.138-146, 2001.

[6] 田中 慎也, "セキュリティ・プロトコルの設計と検証に関する研究" 静岡大学大学院理工学研究科修士論文, Mar. 2001

[7] L.Rizzo , "Effective erasure codes for reliable computer communication protocols," Computer Communication Review, Apr. 1997