管理者への情報漏えいを防止する グループ鍵共有方式の提案と実装

173426013　菅沼 良一
渡邊研究室

1. はじめに

ネットワーク技術・インフラの発展により，世界中のあらゆる場所でインターネットを介した情報共有が行われている．インターネットを介した情報共有では，1 対 N や，N 対 N といった，グループ通信を行うことも多い．そのため，グループで行われる通信を実現するための手法は必要不可欠である．このとき，暗号化を行うための鍵は，鍵管理要件と呼ばれる条件を満たすことを要求される．しかし，メッセージアプリケーションのセキュリティ評価を行っている非営利団体 EFF(Electric Frontier Foundation) は，現状の主流なメッセージアプリケーションのセキュリティが極めて脆弱であることを指摘している．EFF の評価項目の中には，グループ鍵を管理者が読めないよう暗号化されているかがあり，多くのシステムはこの項目を満たしていない．

そこで本稿では，2 種類の乱数を別々の経路で配送し，それを基にグループ鍵を生成することで，鍵管理要件を満たし，かつ管理者がグループ鍵を取得できないグループ鍵共有方式を提案する．また，サーバを用いた鍵管理を行う為，実環境において実現し易い．

2. 鍵管理要件と既存技術

2.1 鍵管理要件

グループ鍵管理プロトコルは機密性や認証に必要なデータを最新の暗号化状態でグループメンバに送信することが重要であり，一般的に以下のような鍵管理要件が存在する．

- 鍵はあらかじめ定めた期間で定期的に更新を行う．
- 鍵データは正規ソースからのみ入手可能であり，正しいグループメンバのみに送られる．
- 鍵管理プロトコルはリプレイ攻撃や DoS(Denial of Service) 攻撃に対して安全である．
- 参加や退会が容易に可能であり，新たに参加したメンバがグループに参加する前の鍵データにアクセスできない (後方秘匿性)．また退会したメンバがそれ以降の鍵データにアクセスできない (前方秘匿性)．

2.2 既存のグループ鍵管理技術

管理サーバを用いてグループ鍵管理を実現するプロトコルとして GSAKMP(Group Secure Association Key Management Protocol[1]:RFC4535) が挙げられてる．GSAKMP では鍵サーバ GCKS，グループメンバ GM，グループオーナー GO の 3 つの要素でグループ管理を実現する．

この方式では，GCKS は，GO によって作成されたセキュリティポリシーの基で GM の管理やグループ鍵の生成を行う．GSAKMP では GCKS，GM 双方ともに公開鍵証明書を所有し，GCKS と各 GM 間の相互認証を行う．

また，グループに参加する端末は，GO から招待されている必要がある．参加処理を行う際，端末は GCKS に対し，参加要求メッセージである Request to Join を送信する．それを受信した GCKS は，ユーザの公開鍵証明書を確認しグループ鍵を配布する．最後に，応答を返すことで，端末は当該グループのメンバとなる．

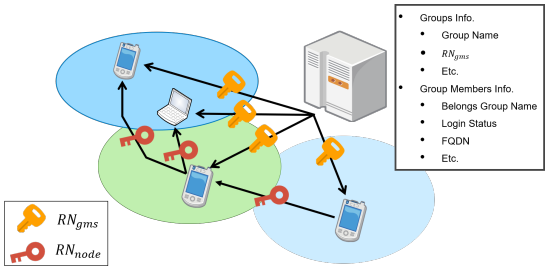

図 1: 提案方式のシステム構成

しかし，この方式は GCKS がグループ鍵を配布しているため，グループ鍵管理者に通信内容を読み取られる恐れがある．

分散型の鍵管理方式として，Simple and Fault-Tolerant Key Agreement for Dynamic Collaborative Groups[2](以下 SFKA-DCG) が提案されている．この方式は，グループ鍵を，2 分木のキーツリーを用いて管理する方式である．グループメンバの数が N の時，各グループメンバは，$N+1$ 個の鍵を所有する．これらの鍵を用いて DH 鍵交換のプロトコルのような冪乗演算を行うことで，全てのグループメンバはグループ鍵となる値を共有可能できる．グループ管理サーバを用いないため，管理者への情報漏洩リスクはない．しかし，鍵の共有や更新を行う際に、全てのグループメンバに対してブロードキャストを行う必要がある．複数の異なるプライベート空間でこれを行う場合，複数の E2E（End-to-End）経路を構築する必要があることから，実環境において動作させることが難しい．

3. 提案方式

3.1 提案方式の原理とシステム構成

提案方式では，サーバ管理者にグループ鍵を取得されないよう，サーバ側の乱数 (RN_{gms}) と端末側の乱数 (RN_{node}) を別々に生成し，グループメンバ全体で共有させる．各端末は，共有された 2 つの乱数を用いてグループ鍵 (GK) を生成する．そのため，管理サーバは，グループ全体で共有される GK を知ることができない．また，GMS は，グループメンバの参加，退会時や，あらかじめ決められたタイミングで，RN_{gms} の更新と再配布を行う．これにより，前方・後方秘匿性を満たすことができる．図 1 に提案方式のシステム構成を示す．提案方式は GMS とユーザ端末から成る．GMS は RN_{gms} の更新，配送，グループやグループメンバに関する情報の管理を行う．ユーザ端末は，RN_{node} の生成，配送，GK 生成，グループへの招待を行う．

3.2 鍵共有方式

提案方式の実現方法として，ユーザ端末に公開鍵証明書を所有させる方式と，E2E(End-to-End) 通信が可能なセキュアネットワークを使用する方法の 2 通りが考えられる．公開鍵証明書方式の場合，RN_{node} の配送に，一般的な CS(Communication Server) を使用できるが，公開鍵証

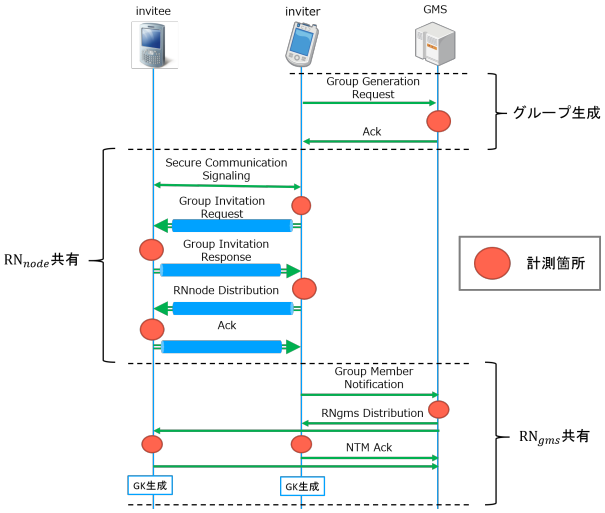

図 2: E2E 通信を用いた鍵共有シーケンス

明書の取得や管理にコストが発生する．一方，E2E 通信を用いる方式では，ユーザ端末が公開鍵証明書を取得する必要はないが，E2E 通信が可能なセキュアネットワークを準備しなければならない．このようなセキュアネットワークの例として NTMobile[3] がある．

図 2 に E2E 通信を用いた鍵共有シーケンス及び性能評価を行った際の計測ヶ所を示す．招待を行う端末を招待者，招待される端末を被招待者とする．本提案方式では，グループの招待権限を持っているユーザのみが招待を行うことができる．ここで，招待者と被招待者は，GMS と安全なコネクションが張られており，招待者と被招待者間では事前にパスワード共有が完了しているものとする．

グループを生成したい端末は RN_{node} を生成し，GMS に対し，作成したグループ名 (ID) と自身のアドレス情報を Group Generation Request として送信する．それを受信した GMS は，最初のメンバ登録とグループ生成を行い，Ack を返すことでグループ生成が完了する．次に，招待者は被招待者に対して E2E 接続のためのシグナリングを行い，グループ名とともに Group Invitation Request を送信する．被招待者はパスワードを入力し，Group Invitation Response を送信する．招待者は受信したパスワードを検証し，成功した場合，被招待者に対して，RN_{node} とグループ招待権限の有無を送信し，被招待者は応答を返す．被招待者の招待を終えると，招待者は GMS に対し，Group Member Notification を送信する．この通知により，GMS はグループに関するデータベースを更新及び RN_{gms} の生成 (更新) を行い，RN_{gms} Distribution を送信することで，RN_{gms} の配送を行う．RN_{gms} を受信したグループメンバは，GMS に対して応答を返したのち，$[RN_{node} \mid RN_{gms} \mid \text{GroupName}]$ のハッシュ値を取ることで GK を生成する．これにより，グループ鍵の共有が行われる．

4. 実装と評価

4.1 動作検証と性能評価

提案方式の一部を実装し，動作検証及び処理時間の計測を行った．仮想環境において図 2 に相当するシステムを実装し，処理時間を計測した．測定箇所は図 2 のグループ生成，RN_{node} 共有，RN_{gms} 共有の 3 つのシーケンスであり，各試行回数 10 回の平均時間を取得した．測定結果を表 1 に示す．RN_{gms} を共有する際の GMS の処理時間が他と比べ大きくなっているが，これは，GMS がグループ

表 1: 提案方式の各シーケンスでの処理時間計測結果

	invitee[ms]	inviter[ms]	GMS[ms]
グループ生成	-	-	3.01
RN_{node}共有	3.72	3.25	-
RN_{gms}共有	0.393	0.374	7.17

表 2: 関連技術と提案方式の比較

	項目 1	項目 2	項目 3	項目 4
GSAKMP	×	○	○	×
SFKA-DCG	○	△	×	△
提案方式	○	○	○	△

メンバ情報を登録する処理と，グループに所属しているすべてのメンバ情報を取得する処理が DB を介して行われているためである．これは，GMS の性能を上げることなどで対応できる．また，inviter，invitee の，RN_{node} 共有と RN_{gms} 共有にかかる合計処理時間は，RTT25ms を仮定しても合計で $100ms$ 未満であることから実用上問題無いと考えられる．

4.2 関連技術との比較

表 2 に関連技術との比較を示す．項目は以下の 3 つである．比較対象は GSAKMP，SFKA-DCG である．

1. 管理者が読めないように暗号化されているか．
2. 鍵管理要件を満たしているか．
3. 鍵管理が容易か．(実環境での実行を考慮する)
4. グループメンバの公開鍵証明書が不要か．

GSAKMP は鍵管理要件を満たしているが，サーバがグループ鍵配布を行っているため，項目 1 を満たしていない．また，公開鍵証明書を必要とする為，項目 4 も満たしていない．SFKA-DCG は項目 1 を満たしているが，鍵管理のリプレイ攻撃や DoS 攻撃対策が提案されていない．またこの方式も公開鍵証明書を必要とする．一方提案方式では項目 1，2，3 を満たしており，セキュアな E2E 経路が確保できれば，公開鍵証明書最をグループメンバが所有する必要もないことから，最も安全で，様々なケースに適応できる．

5. まとめ

2 種類の乱数を別々の経路で配送することでグループ鍵生成を行うグループ鍵共有方式を提案し，サーバ管理者がグループ情報を読み取ることができないグループ通信が可能であることを示した．提案方式の一部を実装し，動作検証と計測の結果，実用上問題がない時間で実行できることを確認した．

参考文献

[1] H. Harne, U. Meth, A. Colegrove, G. Gross: GSAKMP: Group Secure Association Key Management Protocol, RFC4535, IETF (2006).

[2] Y. Kim , A. Perrig , G. Tsudik: Simple and fault-tolerant key agreement for dynamic collaborative groups, Proceedings of the 7th ACM conference on Computer and communications security, p.235-244, November 01-04, 2000, Athens, Greece

[3] 上醉尾一真，鈴木秀和，内藤克浩，渡邊晃:IPv4/IPv6 混在環境で移動透過性を実現する NTMobile の実装と評価情報処理学会論文誌，Vol.54，No.10，pp.2288–2299 (2013).