暗号技術を用いた セキュアグループコミュニケーションの提案

棚田 慎也[1]　鈴木 秀和[1]　内藤 克浩[2]　渡邊 晃[1]

概要：セキュアな情報共有を実現するためにグループメンバ間でグループ鍵と呼ばれる共通鍵を用いて暗号化する方法が一般的である．しかし, 現存する方式ではサーバに通信内容が蓄積されていたりサーバ管理者がグループ鍵を所有したりすることから情報が漏えいする恐れがある．そこで, 本稿では生成元が異なる2つの乱数を異なる配送経路で配布し, その2つの乱数を用いてグループ鍵を生成する．この方法によりグループメンバが安全にグループ鍵を共有できるため, セキュリティを向上したセキュアグループコミュニケーションシステムを実現できる．

Proposal for Secure Group Communication using Encryption Technology

SHINYA TANADA[1]　HIDEKAZU SUZUKI[1]　KATSUHIRO NAITO[2]　AKIRA WATANABE[1]

1. はじめに

ネットワーク技術の発展により, インターネットを介したセキュアな情報共有に関心が高まっている．セキュアなグループコミュニケーションはグループで情報を共有する際に重要な技術である．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)[1] によると,Secure Messaging Scorecard[2] の評価において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が指摘されている．EFF による評価項目として, ユーザのデータが管理者であっても読めないように暗号化されているかという項目がある．このような項目が設けられた背景として 2013 年に米国家安全保障局 NSA が大手 IT 企業のサーバから直接情報を取得していた事件が関連していると考えられる．EFF によるセキュリティ評価項目をすべて満たしているメッセージアプリケーションとして ChatSecure[3] と呼ばれるシステムが存在するが,1 対 1 のメッセージアプリケーションであり, グループコミュニケーションを行うことはできない．企業の業務などでも使用可能でかつ EFF の項目も満たすグループコミュニケーションシステムがあると有用である．

グループコミュニケーションの代表例にチャットアプリケーションがあり, その中でも LINE は日本で最も多く使用されている．LINE は伝送路上において暗号化されているが, セキュリティの配慮が十分とは言えないため, 企業の業務として使われることは少ない．また交換されるメッセージの内容はすべてサーバに平文で蓄積されているため, サーバを経由して第三者に内容が漏れる可能性がある．

セキュリティを考慮したグループコミュニケーションを実現する技術として,MSEC(Multicast Security)Working Group による GKMA(Group Key Management Architecture)[4], およびこれを改良した GSAKMP(Group Secure Association Key Management Protocol)[5] が標準化されている．この技術は, グループ鍵を用いて通信内容やコンテンツの暗号化を行う．グループ鍵を用いるコミュニケーションシステムは所定の鍵管理要件を満たす必要があるとされている．鍵管理要件には, グループ招待方法や鍵の更新期間に関する要件が含まれていて,GSAKMP はこれらの要件をすべて満たしている．GSAKMP では鍵サーバ GCKS(Group Controller Key Server) がグループ鍵の生成

[1] 名城大学
Meijo University
[2] 愛知工業大学
Aichi Insutitute of Technology

や配布および更新を行う. またグループメンバと GCKS はそれぞれ公開鍵証明書を所有することを前提としており, これを用いて相互認証を行うことにより, 安全にグループ鍵を共有することができる. しかしこの方式は EFF の評価項目の一部を満たしていない. すなわち,GCKS がグループ鍵を知っているため, 悪意のあるサーバ管理者が通信内容を閲覧することができる.

本論文では, このような課題を解決するためのグループ鍵共有方式を提案する. 提案システムは, ユーザが使用するエンド端末 ET(End Terminal) とグループ管理サーバ GMS(Group Management Server) によって構成される. ET と GMS においてそれぞれ別の乱数を生成し, その 2 つの乱数を異なる配送経路で ET が共有し,2 つの乱数から新たにグループ鍵 GK を生成する. ET 間で直接交換した乱数は管理者にも把握できないため,EFF の評価項目を満たすことができる. GMS で生成した乱数を更新するタイミングを設定することにより, グループコミュニケーションの鍵管理要件を満たすこともできる.

EFF の評価項目と鍵管理要件の項目により提案方式と既存技術との比較評価を行い, 提案方式が有用であることを示した.

以降,2 章では既存技術とその課題を説明する. 3 章で提案方式についての詳細を述べる. 4 章で既存技術と比較評価を行い,5 章でまとめる.

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の目標は, 機密性や認証のための必要なデータを最新の暗号化状態でグループメンバに提供することであり, 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.
- 鍵データは厳重に保管され正規ソースからのみ入手可能で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 [*1] 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はグループに参加する前の鍵データへアクセスできない (後方秘匿性). また退会したメンバはそれ以降の鍵データへアクセスすることができない (前方秘匿性).

2.2 LINE

グループコミュニケーションとして日本で最も普及しているチャットアプリケーションの LINE の動作とそのセキュリティについて述べる.

[*1] ユーザのログイン時や参加申請時にネットワークに流れるデータを盗聴し, そのデータを認証サーバへ送ることで不正な通信をする行為.

LINE チャットはユーザが使用するエンド端末とチャットに用いられるチャットサーバによって構成される. ユーザの生成するメッセージはすべてサーバに送信され, サーバからグループメンバに同じメッセージが配送される. 各ユーザは招待したいユーザを自由に自らのグループに勧誘し, グループを拡大していくことができる. グループ招待のメッセージもすべてチャットサーバを経由する. LINE にはグループ鍵のようなものは存在せず, 通信データの暗号化のみが行われている.

グループ鍵が存在しないため, 鍵管理要件を満たすことはない. またメッセージを複数のメンバに配送するため, サーバ上では平文で蓄積される. このように LINE はセキュリティがきわめて脆弱である.

2.3 GSAKMP

GSAKMP はグループコミュニケーションにおけるセキュリティフレームワークで RFC4535 として標準化されている. グループのセキュリティポリシーを提供し, アクセス制御のルールによりユーザ認証を行いグループの確立を行う. GSAKMP の使用例として IETF 参加者のためのグループコミュニケーションが挙げられている.

GSAKMP では 3 つの主要な要素でグループ管理を分散している. その 3 つとは, グループオーナー, 鍵サーバ GCKS, グループメンバである. GSAKMP では GCKS だけでなくグループメンバも公開鍵証明書を所有していて,GCKS と各グループメンバ間における相互認証を確実に行うことができる. グループオーナーは鍵の更新期間やメンバの招待方法などを含むセキュリティポリシーを作成し提供する役割を担っている. このセキュリティポリシーに基づき GCKS やグループメンバはグループ招待やグループ鍵の更新など, グループ生成やセキュリティ確保に関係する動作を実行することができる. GCKS はセキュリティポリシーに基づきグループ鍵の生成や配布および鍵更新, グループメンバの管理を行うサーバである. グループメンバはグループ鍵が更新された場合, それが適切であるか, またセキュリティポリシーに基づいているか確認しなければならない.

GSAKMP における鍵共有シーケンスを図 **1** に示す. GSAKMP の鍵共有方式は GCKS とユーザによって実行される. グループオーナーはセキュリティポリシーを決定するが, 鍵共有の際には直接関わらないため図には示されていない. 新たに参加するユーザはグループオーナーまたはすでにグループに参加しているメンバから招待されていることが前提である. 図中の番号は以下の説明に対応している.

(1) 招待されたユーザは GCKS へ Request to Join を送信する. これは参加申請であり, このメッセージには自身の公開鍵証明書やメンバから招待された時に付与されているグループ ID などが含まれている.

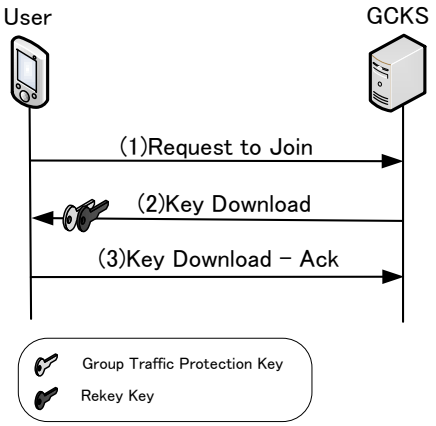

図 1　GSAKMP におけるグループ鍵共有シーケンス

(2) GCKS は新たに参加するユーザの公開鍵証明書を確認し認証が成功した場合,Key Download により 2 つの鍵を配布する．1 つは GTPK(Group Traffic Protection Key) と呼ばれるグループデータを暗号化する鍵で, もう 1 つは Rekey Key と呼ばれる GTPK を更新するための要素となる鍵である．

(3) ユーザは Key Download -Ack を応答する. このタイミングで招待されたユーザは当該グループのグループメンバとなる．

グループ鍵の更新を行う際には GCKS からグループメンバへグループ鍵更新の通知を送り, 通知を受け取ったメンバはあらかじめ配布されている Rekey Key を用いて GTPK の更新を行う. 更新を行った後 GCKS へ更新したことを通知することにより次回の Rekey Key を GCKS から受け取ることができる．グループ鍵はユーザが新たに参加したり, 退会したりするたびに更新されるため, 前方秘匿性や後方秘匿性を確保することができる．GSAKMP は鍵管理要件をすべて満たしている．なお, 公開鍵証明書による認証は初回のみであり, 以後は共有した Rekey Key と GTPK により認証と鍵の更新を行う.

GSAKMP を利用すれば, サーバを利用したチャットであっても, コンテンツを暗号化できるため, チャットサーバから情報が漏えいすることはない．しかし GCKS が GTPK と Rekey Key のどちらも生成と配布を行っているため,EFF が提示した管理者であってもユーザのデータが読めないように暗号化されているかという評価項目を満たしていない.

さらに GSAKMP では公開鍵証明書を各メンバが所有しなければならず, 公開鍵証明書の取得費用や管理コストがかかるという課題がある.

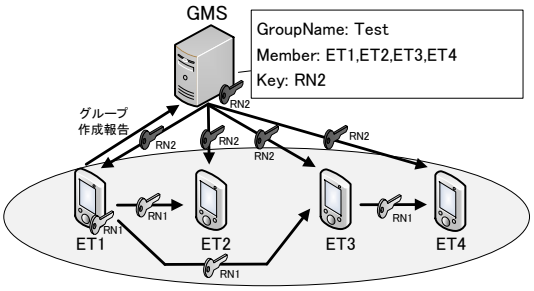

図 2　提案方式のシステム構成

3. 提案方式

セキュアなグループコミュニケーションを実現するために, 生成元が異なる 2 つの乱数を 2 通りの異なる配送経路でグループメンバに配布し, その 2 つの乱数から新たにグループ鍵 GK を生成する．提案方式は鍵管理要件を満たし, かつ EFF の要求も満たすことができる.

3.1 システム構成と方式の概要

図 2 に鍵共有部分に着目した提案方式のシステム構成を示す．提案方式のシステム構成は, グループ管理サーバ GMS とエンド端末 ET から成る．グループの定義は管理者が行ってもよいし, ユーザが主体になってもよい．GMS はグループ名やメンバの管理を行う．また RN2 の生成を行い, 配布や管理およびあらかじめ設定してあるタイミングで RN2 の更新を行う．ET は,GMS へのグループメンバ報告を行う．また乱数 RN1 を生成し, メンバ間で直接共有する．ET は RN1 と RN2 を取得後 GK を生成する.

3.2 鍵共有方式

提案方式の特徴は RN1 をグループメンバ間で管理者にわからないように共有する点にある．実現方法として以下の 2 通りが考えられる．1 つは GSAKMP と同様に ET に公開鍵証明書を所有させる方法である．もう 1 つは, エンドツーエンド通信が可能なネットワークを使用する方法である.

公開鍵証明書を用いた場合では, 鍵配送にコミュニケーションサーバを利用することができるが, 公開鍵証明書の取得費用や管理コストがかかる．一方, エンドツーエンド通信が可能なネットワークの場合,ET が公開鍵証明書を所有する必要はないが, エンドツーエンドネットワークを準備する必要がある.

RN1 と RN2 の共有後, 各 ET では [RN1|RN2|GroupName] のハッシュ値をとり, そのハッシュ値をグループの暗号鍵 GK とする．以上の方式によると GMS 管理者は GK の内容を知ることができない.

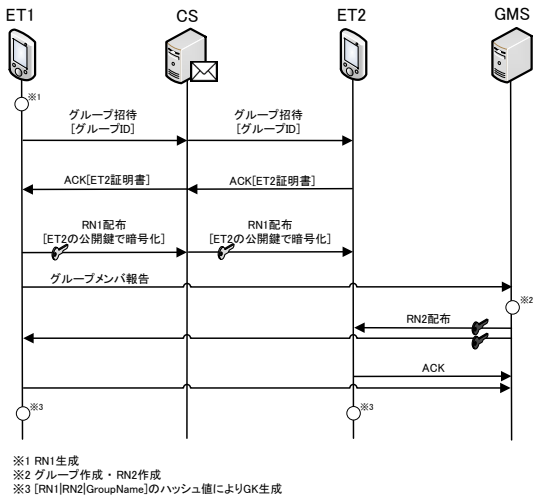

図 3 公開鍵証明書を用いた提案方式のグループ作成シーケンス

同一の GK を所有しているメンバのみが正式メンバとなり以後の相互認証と暗号化に利用することができる. RN1 の更新期間は長くてもよいが,RN2 は鍵管理要件を満たすために頻繁に更新する. 以下に RN1 の共有方法で分類したグループ鍵の共有方法を述べる.

3.2.1 RN1 を公開鍵証明書を用いて共有する方式

図 3 に公開鍵証明書を用いた提案方式のグループ作成シーケンスを示す. 招待を行うユーザを招待者, 招待されたユーザを被招待者とする. ユーザ間の通信にはコミュニケーションサーバ CS を経由したクライアントサーバモデルを利用する. グループを作成する際に最初のメンバ ET1 において RN1 を生成する. RN1 を生成したメンバは招待者になることができる. 被招待者がさらに新たにメンバを招待できるかどうかは招待者からの指示によるものとする. メンバの招待時に招待者がグループ ID とともにグループ招待メッセージを被招待者に送る. 被招待者は自身の公開鍵証明書を添付した応答を返す. 招待者は,RN1 を被招待者の公開鍵で暗号化して送信する. 被招待者は自身の秘密鍵で RN1 を復号し,ET 間による鍵共有が終了する. RN1 の更新周期は十分長く取るものとする. そのため, 各 ET において厳重に保管する必要がある. 公開鍵による暗号化が施されているため,CS の管理者であっても RN1 を知ることはできない.

次に ET1 は GMS へグループメンバ報告を送信する. 報告を受け取った GMS はグループ管理用のテーブルを作成する. また当該グループ用の RN2 を作成し, 新たなグループメンバへそれぞれ配布する. RN2 は一定の更新期間を設け,GMS が定期的に生成しメンバに配布する. また参加していたユーザが退会する場合や新たにユーザを追加した場合にも RN2 の更新を行う. これにより前方秘匿性や後方秘匿性を確保する.

3.2.2 RN1 をエンドツーエンド通信ネットワークを用いて共有する方式

エンドツーエンド通信が可能なネットワークとして NTMobile[6][7] が提案されている. NTMobile は通信接続性と移動透過性を同時に実現する技術である. ここで通信接続性とは, エンド端末がグローバルアドレス/プライベートアドレスに関わらず, 双方向からの通信が開始できる機能を示す. 移動透過性とは, 通信中にネットワークを切り替えても通信を継続できる機能を示す. NTMobile は,NTM 端末とグローバルアドレス空間上に設置する DC(Direction Coodinator) から構成される. NTM 端末は NTMobile を導入したエンド端末である. DC は NTM 端末の位置とネットワークの構成から, 最適な経路を決定し NTM 端末にトンネル経路の構築を指示する. この処理を以後 NTMobile シグナリングと呼ぶ. シグナリング後の通信は NTM 端末間でトンネル経路を通してエンドツーエンドの暗号通信が実現される. GMS と NTM 端末間は DC の公開鍵証明書と NTM 端末に設定したパスワードを用いてあらかじめ共通鍵を共有しておく. GMS と NTM 端末間で共有している共通鍵は長期の有効期限を持つ. 期限が切れると,GMS の公開鍵と NTM 端末のパスワードで認証をやり直し, 新たな共通鍵を共有する.

図 4 にエンドツーエンド通信を用いた提案方式のグループ作成シーケンスを示す. ET は通信に先立ち,NTMobile のシグナリングにより ET 間で暗号化されたトンネル経路を生成する. この経路を利用し,RN1 を送付することにより, 共有を実現する. RN1 を共有後,ET1 から GMS へグループメンバ報告を送る. 公開鍵証明書を用いた方法と同様に GMS はグループ管理用のテーブルを作成する. また RN2 を生成し, 各メンバへ配布する.

3.3 鍵の更新処理

鍵管理要件を満たすため,GMS は一定の更新期間で RN2 を更新する. また, 前方秘匿性や後方秘匿性を考慮し, メンバが退会した場合と新たにメンバを追加した場合も RN2 の更新を行う. 以下にメンバを退会させる場合と新たにメンバを追加する場合について記述する.

3.3.1 メンバを退会させる場合

図 5 にメンバを退会させる場合における鍵の更新処理を示す. 例として ET3 は退会指示の権限を持ち,ET3 が ET4 を退会させるケースを説明する. ET3 から ET4 へ退会指示を送ると,ET4 は強制的に退会させられる. ET3 から GMS へグループメンバ報告を送る. GMS は新しい RN2' を生成し, 自身のデータベースにある当該グループのメンバと RN2 の情報を更新する. その後 GMS は新しい RN2' を更新されたグループメンバへ配布する. 各 ET において新しい RN2' を用いて GK を生成することにより前方秘匿性を確保することができる.

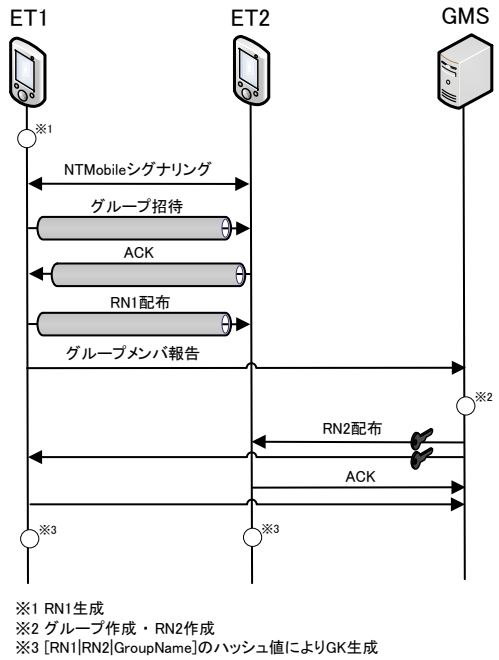

図 4 エンドツーエンド通信を用いた提案方式のグループ作成シーケンス

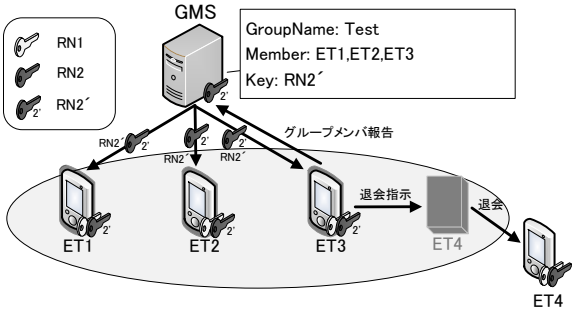

図 5 メンバを退会させる場合における鍵の更新処理

3.3.2 メンバを新たに追加する場合

権限を持つメンバは新たにユーザを招待することができる. **図 6** にメンバを新たに追加する場合における鍵の更新処理を示す. 例として ET3 が追加する権限を持ち,ET4 を招待するケースを示す. ET3 から ET4 にグループ招待を送る. グループ招待を受け取った ET4 が ACK を送り,ET3 から ET4 へ RN1 を配布する. ET3 から GMS へ ET4 のグループメンバ報告を送る. その通知を受け取った GMS は自身のデータベースにある当該グループのメンバと RN2 の情報を更新する. その後,GMS からすべてのグループメンバへ RN2' を配布する. 各 ET において, ハッシュ値を用いて GK を生成することにより後方秘匿性を確保することができる.

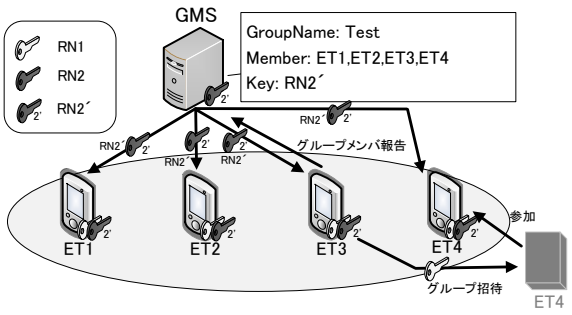

図 6 メンバを新たに追加する場合における鍵の更新処理

表 1 類似技術の比較

	項目 (1)	項目 (2)	項目 (3)
LINE	×	×	○
GSAKMP	○	×	○
ChatSecure	×	○	×
提案方式	○	○	○

4. 評価

表 1 に類似技術の比較を示す. 評価項目の内容は以下の通りである.

(1) 鍵管理要件を満たしている.
(2) 管理者が読めないように暗号化されている.
(3) グループ通信を行える.

比較対象は既存技術で例に挙げた LINE と GSAKMP および ChatSecure の 3 つである. LINE は通信経路で暗号化されているが, グループ鍵の概念がないので, 鍵管理要件を満たしていない. また, サーバには情報が平文で蓄積されるため, 管理者に情報を閲覧される恐れがある. GSAKMP では公開鍵証明書を用いた相互認証やグループ鍵を更新することにより鍵管理要件をすべて満たしている. しかし, グループ鍵をすべて鍵サーバ GCKS から配布しているためサーバ管理者にユーザのデータ内容を読み取られる恐れがある. ChatSecure は, 管理者がユーザのデータを読めないように暗号化されているが,1 対 1 のチャットであるため, グループコミュニケーションを行うことはできない. そのためグループ鍵管理の鍵管理要件を満たさない.

提案方式は,RN2 をあらかじめ定めた期間により更新を行い, かつ新たなメンバの参加, 退会ごとに RN2 を更新するため鍵管理要件を満たしている. また GMS の管理者は GK を生成することができないため, ユーザのメッセージの内容を知ることはできない.

5. まとめ

既存のグループコミュニケーションでは, グループ鍵を用いる通信を行う際に鍵管理サーバからグループ鍵を配布するため悪意のあるサーバ管理者に通信内容を閲覧される

恐れがあった．そこで本稿では，グループ管理サーバとエンド端末が異なる 2 つの乱数を異なる配送経路で共有し，その 2 つの乱数を用いてグループ鍵を生成することで管理者でもユーザデータの内容を知ることができないセキュアグループコミュニケーションが可能であることを示した．またグループ管理サーバが乱数を適宜更新することにより鍵管理要件を満たすことを示した．今後は提案方式を実装し，動作検証と性能評価を行う予定である．

参考文献

[1] Electronic Frontier Foundation https://www.eff.org/

[2] Electronic Frontier Foundation : Secure Messaging Scorecard. https://www.eff.org/secure-messaging-scorecard .

[3] ChatSecure -Encrypted Messenger for iOS and Android. https://chatsecure.org/

[4] Multicast Security(MSEC) Group Key Management Architecture,RFC 4046,IETF (2005).

[5] GSAKMP: Group Secure Association Key Management Protocol, RFC4535, IETF (2006).

[6] 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，Oct.2013.

[7] H. Suzuki, K. Naito, K. Kamienoo, T. Hirose and A. Watanabe NTMobile: New End-to-End Communication Architecture in IPv4 and IPv6 Networks Proceedings of the 19th Anuual International Conference on Mobile Computing and Networking (Mobicom2013)，pp.171-174，Oct.2013.