平成27年度　卒 業 論 文

和文題目

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

英文題目

Proposal of Secure Group Communication using Encryption Technology

情報工学科　渡邊研究室

(学籍番号: 120430063)

棚田　慎也

提出日: 平成 28 年 2 月 10 日

名城大学理工学部

概要

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. これを実現するためにグループでの情報共有の際には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管理者やグループ退会者がグループ鍵を所有していることから情報が漏えいする恐れがある. そこで, 本論文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共有するグループ鍵を用いて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する.

目 次

第1章　序論

ネットワーク技術の発展により, ネットワーク利用者が急激に増加している. これはパソコンだけでなく携帯電話やスマートフォンといった小型の通信機器の普及による影響が大きい [1–3]. これらの通信機器の普及により, インターネットを介した情報共有に関心がさらに高まっている. チャットアプリケーションは情報共有のための強力なコミュニケーションツールの 1 つであり,LINE や Skype といったチャットアプリケーションが普及している. しかし, メッセージアプリケーションにおけるセキュリティ評価を行っている非営利団体の EFF (Electronic Frontier Foundation) [4] によると評価項目 Secure Messaging Scorecard [5] において現状のメッセージアプリケーションのセキュリティが極めてぜい弱であると評価されている. EFF によるセキュリティ評価項目を全て満たしているメッセージアプリケーションとして ChatSecure [6] があるが 1 対 1 のアプリケーションであり, グループチャットを行うことはできない. そこで業務などでも使用可能なセキュリティが万全なグループチャットシステムがあると有用である.

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, 既存技術を改良し鍵管理を可能とした提案 [7] があるが, 通信相手の確実な認証が行えない点と鍵の更新期間のメッセージの送受信が困難であるという問題を抱えている. 認証や鍵の更新を含む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) による GKMA (Group Key Management Architecture) [8] を改良した GSAKMP (Group Secure Association Key Management Protocol) [9] がある. この技術は, グループ鍵を用いて暗号化を行う方法が用いられている. グループ鍵を用いるコミュニケーションシステムには一般的に鍵管理要件を考慮する必要がある. その鍵管理要件には, 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 はこれらを満たしている. GSAKMP ではグループ鍵を新たに設置する鍵サーバ GCKS (Group Controller Key Server) により生成や管理, 配布および更新を行う. またグループメンバと鍵サーバはそれぞれ公開鍵証明書を所有することを前提としていて, その公開鍵証明書を用いて相互認証を行うことができ, 認証が成功するとその公開鍵を用いてグループ鍵を安全に共有することができる. この方式では鍵サーバからグループ鍵を生成するために必要な要素をグループメンバへ配布している. そのため悪意のある鍵サーバ管理者がグループ鍵を使用して通信内容を閲覧することができるという課題がある. この課題に関する管理者が読めないように暗号化されているかという EFF による評価項目があるが, この項目の背景として 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データを取得しているというニュース [10] がある. 業務で用いる場合, グループメンバ以外の誰にも通信内容を閲覧されたくはないため鍵サーバ管理者でも通信内容を閲覧することができないセキュリティを実現できると有用である.

本稿では, このような課題を解決するためのグループ鍵生成方式を提案する. エンド端末とグルー

プ管理サーバにおいてそれぞれ生成された 2 つの乱数をグループ管理サーバ経由とエンド端末間による異なる配送経路で配布し, その 2 つの乱数からグループ鍵を生成する. 提案方式はエンド端末とグループ管理サーバ GMS (Group Management Server) によって構成する. エンド端末と GMS に公開鍵証明書を持たせ, 装置間で双方向認証を確実に行う. グループ管理サーバ管理者は自身が生成していない乱数を取得できないためグループ鍵を生成することができず通信内容を閲覧することができない. これにより, グループメンバのみによるセキュアチャットコミュニケーションを可能にする. また, グループ管理サーバ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更なるセキュリティの向上を図った. EFF の評価項目 [5] と独自に追加した項目により既存技術との比較評価を行い, 提案方式のセキュリティ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概要と構成要素および鍵共有方式について詳細に述べる. 4 章で既存技術と比較し提案方式の評価を行い,5 章でまとめる.

第2章　既存技術

本章では, チャットシステムとして普及している LINE と, セキュリティを考慮したグループコミュニケーションシステムの既存技術として GSAKMP (Group Secure Association Key Management Protocol) について述べる.

2.1　LINE

2.1.1　LINE の通信方式

LINE の通信方式を図 1 に示す. ユーザが私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. 各ユーザは招待したいユーザをグループ勧誘し, グループを生成することができる.

各エンド端末とチャットサーバ間ではそれぞれ異なる共通鍵を用いて通信経路において暗号化を行う. 共通鍵はユーザの新規登録時に生成が行われ, チャットサーバの公開鍵を用いて暗号化しチャットサーバへ送信する. この暗号化された共通鍵を自身の秘密鍵で復号することでエンド端末とチャットサーバ間で共有が行われる. メッセージ送信時には, チャットサーバの公開鍵を用いて暗号化を行い, チャットサーバへ送信する. チャットサーバにおいて暗号化されたメッセージを自身の秘密鍵を用いて平文の状態に復号しメッセージの情報を蓄積する. そして送信先のエンド端末との共通鍵を用いて暗号化を行い送信する.

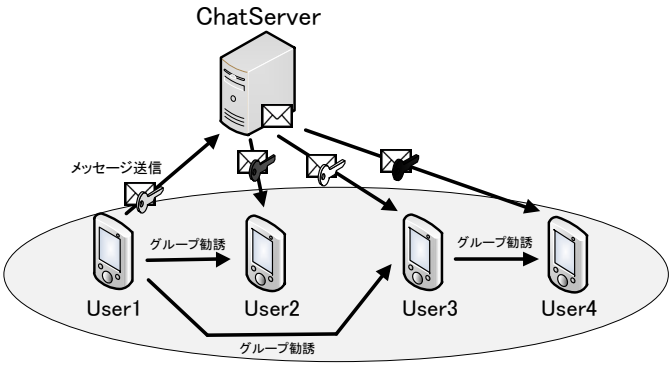


図 1　LINE における通信方式

2.1.2 LINE の課題

この通信方式ではエンド端末とチャットサーバ間の共通鍵を用いて通信経路における暗号化を行っている. しかし, 通信内容が平文の状態でチャットサーバに蓄積されていることにより悪意のあるサーバ管理者に通信内容が閲覧される可能性がある課題がある. また使用するアカウントのパスワードが漏えいすると別端末からも通信履歴を読める課題がある.

2.2 GSAKMP

2.2.1 GSAKMP の概要

GSAKMP はネットワーク上の暗号化グループを生成し管理するためのセキュリティフレームワークである. 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ピングが挙げられている.

グループ鍵管理プロトコルの目標は最新の暗号化状態で, 機密性や認証のための必要なデータを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あるため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 やサービス妨害攻撃 *2 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 (後方安全性). 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 (前方安全性).

GSAKMP は 3 つの主要な役割にグループ管理の責任を分散し構成されている. その 3 つとは, グループオーナー, 鍵サーバ GCKS (Group Controller Key Server) およびグループメンバである. この方式では GCKS だけでなくユーザも公開鍵証明書を所有していて, 認証を確実に行えることが前提である. グループオーナーはグループのための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グループ招待やグループ鍵の更新などグループにおけるセキュリティ関連の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機関である. グループメンバはセキュリティポリシーに基づき適切にグループ鍵を使用する. 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.

*1 ユーザがログインするときにネットワークに流れるデータを盗聴しコピーし, そのデータを認証サーバへ送ることで不正ログインする行為

*2 大量のデータや不正データを送りつけ相手のシステムを正常に稼働できない状態に追い込む行為

2.2.2 GSAKMP の鍵共有方式

GSAKMP における鍵共有シーケンスを図 2 に示す. GSAKMP の鍵共有方式は GCKS とユーザによって構成されている.

グループオーナーは鍵共有方式の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。新たに参加するメンバ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）招待されたメンバは GCKS へ Request to Join を送信する. これは参加申請であり, この参加申請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.

（2）Request to Join を受け取った 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 により新たなメンバへ 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ある. もう 1 つは Rekey Key と呼ばれる GTPK を更新するための鍵である.

（3）認証を失敗した場合,Request to Join Error のメッセージをメンバへ返し, 認証が失敗したことを通知する. このメッセージの通知はオプションであり, 送信するかどうかの設定をすることができる.

（4）Key Download の応答としてユーザは Key Download -Ack/Failure を送信して完了となる. これによって配布されたグループ鍵を用いてグループメンバによる安全な通信を行うことができる.

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で次回の Rekey Key を GCKS から受け取ることができる. そのため, ユーザが退会した後の通信内容を閲覧できないようにするための前方安全性や, 新たに参加したユーザが参加する前の通信内容を閲覧できないようにするための広報安全性を確保することができる.

2.2.3 GSAKMP の課題

GSAKMP によるグループコミュニケーションでは公開鍵証明書を用いて相互認証を確実に行い, グループ鍵を用いてグループメンバによる通信の暗号化を行っている. しかし GCKS がグループ鍵と鍵更新鍵のどちらも生成と配布を行っている. そのため悪意のあるサーバ管理者がグループ鍵を用いて通信内容を読み取り, 重要な情報が漏えいする恐れがある.

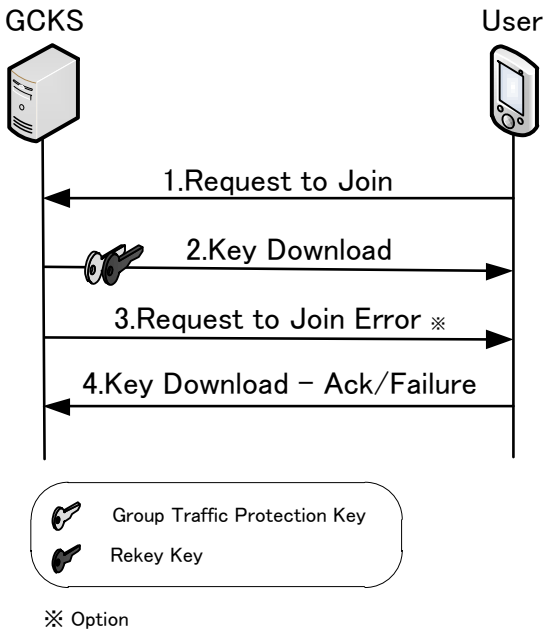


図 2　GSAKMP におけるグループ鍵共有シーケンス

第3章　提案方式

本章では, 提案するグループ鍵の共有方式について述べる. この目的を達成するために生成元が異なる 2 つの乱数 RN1,RN2 を GMS 経由とエンド端末間の 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. そのためグループ管理サーバ GMS (Group Management Server) の管理者には GK を生成できない. この GK を用いて同一の GK を所有しているユーザのみが正式メンバとして相互通信を行うことができる.

3.1　構成

図 3 に提案方式のシステム構成を示す. この図では鍵共有のみに着目した図であり, チャットサーバは図には示されていない. 提案方式のシステム構成はグループ管理サーバ GMS (Group Management Server) とエンド端末の 2 つから成る. これらの装置はいずれも公開鍵証明書を持つことを前提とする. これにより GMS と各エンド端末間やエンド端末間の認証を確実に実行することができる. 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. また乱数 RN2 の生成機関としてユーザからの申請に応じて 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. エンド端末では,GMS へのグループ作成報告やメンバ変更の報告を行う. また RN1 の生成や共有および GK の生成と管理を行う. RN1 や GK はグループメンバのみが所有する要素であるため厳重な保管が必要となる. なお, 公開鍵は RSA(鍵長 1024 ビット以上), 共通鍵は AES(鍵長 128 ビット以上) を使用することで暗号化アルゴリズム上はセキュリティの課題がないことを前提する.

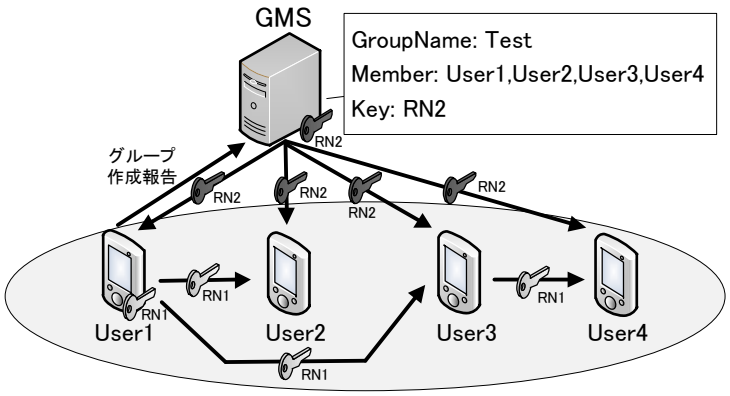


図 3　提案方式のシステム構成

3.2 鍵共有方式

図 4 にグループ作成時の鍵共有方式を示す. 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チャットサーバを用いて通信を補助する方式が考えられるが図では省略している. ユーザがグループを作成する際に最初のエンド端末において RN1 を生成する. RN1 を生成したユーザは招待したいメンバをグループに招待することができる. また, 被招待者もさらに新たにメンバを招待することができる. メンバの招待時には招待者が自身の公開鍵証明書を付与しグループ招待を送る. 被招待者は招待者の認証を行い, 自身の公開鍵証明書を添付した応答を返す. 応答を受け取った招待者は認証を行い,RN1 を被招待者の公開鍵で暗号化を行い送信する. RN1 を受け取った被招待者は自身の秘密鍵で復号し, 応答を返す. RN1 は更新を行わず, 当該グループの GK において使うため, 各エンド端末において厳重に保管する必要がある.

図 4 では, ユーザ 1 がユーザ 2 を招待したため, 招待者であるユーザ 1 から GMS へグループ作成報告を送信し, その報告を受け取った GMS はグループの作成と当該グループの RN2 を作成し, グループデータを自身のデータベースに書き込む. その後 GMS は生成した RN2 を招待者と被招待者へ配布する. 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. これにより前方安全性や後方安全性を確保することができる.

RN2 を取得したユーザから GMS へ応答を返し, 各エンド端末において [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 として生成する. 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相互通信を行うことができる.

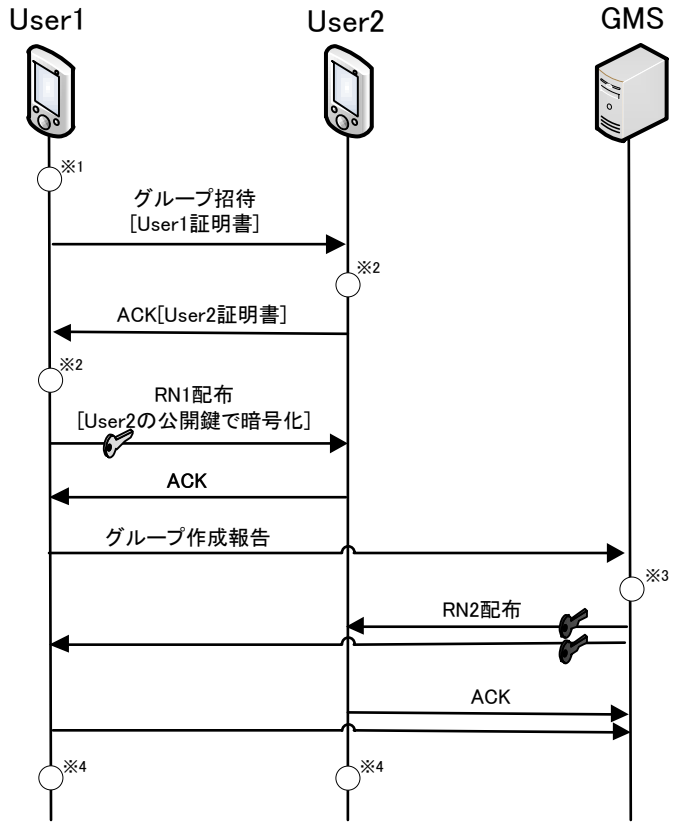
3.3 鍵の更新処理

セキュリティ機能向上のため, グループ鍵はあらかじめ定めた期間で定期的に更新を行う必要がある. そのため提案方式では一定の更新期間でグループ鍵を更新する. また, 前方安全性や後方安全性を考慮し, メンバが退会した場合と新たにメンバを追加した場合でもグループ鍵の更新を行う. ここでは, メンバが退会した場合と新たにメンバを追加した場合について記述する.

3.3.1 メンバが退会した場合

（1）他のメンバを退会させる場合

例としてユーザ 3 がユーザ 4 を退会させるケースを図 5 に示す. まずユーザ 3 からユーザ 4 へ退会指示を送る. 退会指示を受け取ったユーザ 4 は強制的に退会させられ, そのタイミングでユーザ 3 から GMS へユーザ 4 の退会通知を送る. 退会通知を受け取った 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. 各エンド端末において新しい RN2’ を用いて GK を生成することにより前方安全性を確保することができる.



※1 RN1生成
※2 認証
※3 グループ作成・RN2作成
※4 [RN1|RN2|GroupName]のハッシュ値によりGK生成

図 **4**　提案方式のグループ作成シーケンス

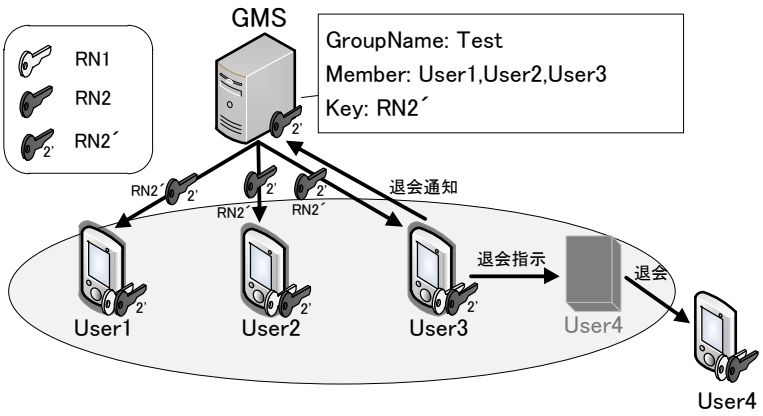


図 **5**　他のメンバを退会させる場合における鍵の更新処理

（2）自らグループを退会する場合

まず退会するユーザから GMS へ退会通知を送る. 退会通知を送ったユーザは自ら退会を行う. 退会通知を受け取ったタイミングで 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. その後は（1）と同様に GMS からメンバへ RN2’ を配布しエンド端末で新しい GK を生成する.

3.3.2 メンバを新たに追加した場合

グループにいるメンバが新たにユーザを招待することができる. そのときの鍵の更新処理の例としてユーザ 3 がユーザ 4 を招待するケースを図 6 に示す. まずユーザ 3 からユーザ 4 に招待通知を送る. 招待通知を受け取ったユーザ 4 が参加申請を送り, ユーザ 3 とユーザ 4 の間で認証し成功した時にユーザ 3 から RN1 をユーザ 4 へ配布する. 認証が成功したタイミングでユーザ 3 から GMS へユーザ 4 の参加通知を送る. 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 から全てのグループメンバへ RN2’ を配布する. 各エンド端末において, ハッシュ値を用いて GK を生成することにより後方安全性を確保することができる.

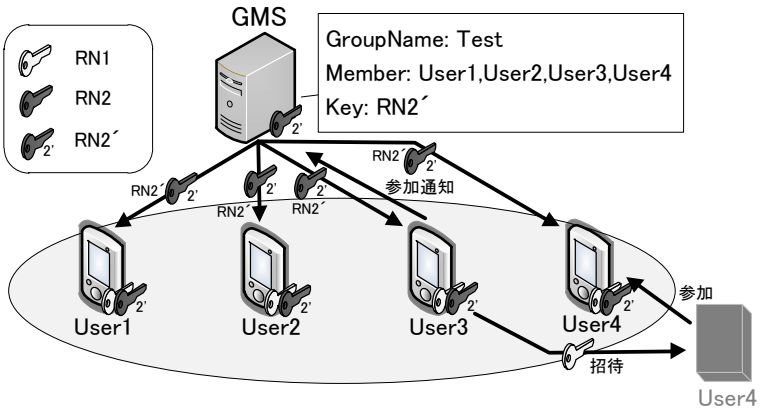


図 6　メンバを新たに追加する場合における鍵の更新処理

3.3.3 RN2 バージョン機能

提案方式における RN2 の更新において新しく生成した RN2 が確実にメンバ全員に届くかどうかという問題がある. ユーザのエンド端末の電源がオフの状態である時に RN2 の更新を行い,GMS からそのユーザへ RN2 を送ることができないというケースが考えられる. ユーザが電源を入れたときに新しい RN2 を所有していないため, グループメンバによる暗号化通信を行えない可能性がある.

そこで,GMS のデータベースに RN2 のバージョン機能を付与する. バージョン機能を用いた鍵の更新処理を図 7 に示す. ユーザが電源をオフにしている間にグループメンバの更新が生じ GMS に

おいて RN2 を更新するケースを想定する. グループメンバの更新が生じたため RN2 を更新し, 新しい RN2 を GMS から各ユーザへ配布する. しかし, 電源がオフであるユーザと通信が行えず RN2 を配布することができない. その後ユーザは電源をオンにする. そのタイミングでユーザから GMS へ RN2 のバージョン問い合わせを自身の RN2 のバージョンを添付し送る. GMS はユーザの認証を行い, ユーザが所有している RN2 と現在のバージョンと異なっていた場合,GMS は新しい RN2 を送信する. 新しい RN2 を受け取ったユーザは GMS に応答を返し, エンド端末において新しい GK を生成しグループメンバ間の暗号化通信が可能となる.

また, エンド端末の電源がオンであるが, 鍵配布時にネットワークが輻輳していたり, 電波が届かない状態であったりする場合も考えられる. そのため, メッセージフォーマット内に RN2 バージョンを付加し, 通信時に受信側がメッセージフォーマット内にある RN2 バージョンを確認する. メッセージ受信側のバージョンが古い場合は, 受信側が GMS へ最新の RN2 配布要求を行う. メッセージ送信側のバージョンが古い場合は, 受信側から送信側へ通知を送信し, 送信側から GMS へ配布要求を行う.

これら以外にもバージョンを付加する利点として, 過去の通信内容を閲覧する際に通信内容にアクセスし使用している RN2 のバージョンを参照することで使用する鍵を明確にすることができることや GMS が RN2 の管理を容易にできるメリットがある.

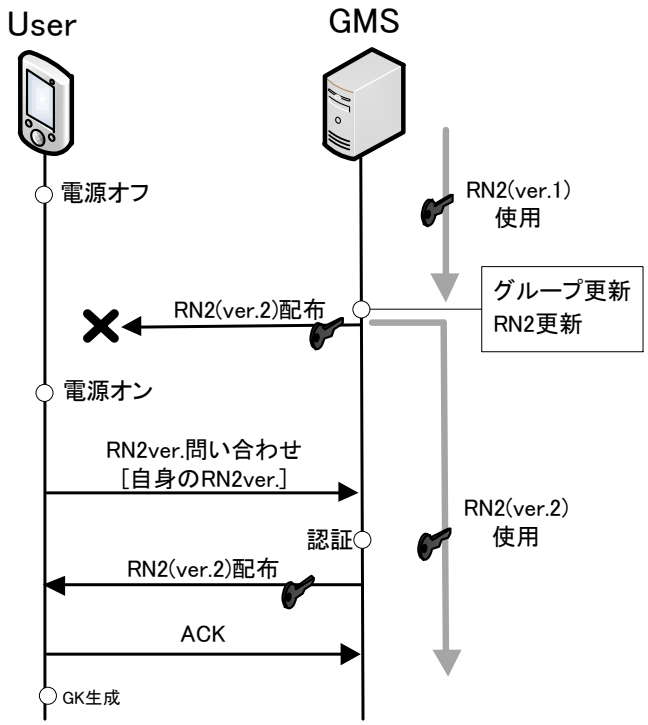


図 7　バージョン機能を用いた鍵の更新処理

第4章　評価

表 1 に類似技術の比較を示す. 項目 (1),(2) は EFF により提示されたチャットアプリケーションのセキュリティ評価項目の一部を使用した. 項目 (3),(4) は独自に追加した評価項目である. 評価項目の内容は以下の通りである.

（1）通信経路が暗号化されている.
（2）管理者が読めないように暗号化されている.
（3）前方安全性と後方安全性を満たしている.
（4）グループ通信を行える.

表 1　類似技術の比較

	項目 (1)	項目 (2)	項目 (3)	項目 (4)
LINE	○	×	×	○
Skype	○	×	×	○
GSAKMP	○	×	○	○
ChatSecure	○	○	-	×
提案方式	○	○	○	○

比較対象は既存技術で例に挙げた LINE と Skype および GSAKMP,ChatSecure の 4 つである. LINE や Skype は通信経路で暗号化されているが, サーバには情報が平文で蓄積されるため, 管理者が情報を取得できる. また, パスワードが漏えいすると他端末から情報が盗まれるなどセキュリティがぜい弱である. GSAKMP におけるコミュニケーションシステムでは使用する鍵を全て鍵サーバ GCKS から配布しているためサーバ管理者が通信内容を読み取れる恐れがある. 一方,ChatSecure は,EFF による評価項目を全て満たしているが 1 対 1 のチャットであるため, 項目 (3) は評価できない. 項目 (4) についてはグループコミュニケーションを行うことはできない.

提案方式について考察を行う. 項目 (1) は, 通信経路において GK を用いて暗号化通信が行われている. 項目 (2) においてエンド端末間で RN1 の共有を行い, その RN1 を用いて GK の生成を行っている. この RN1 は GMS を通らない経路で配送を行っているため,GMS が RN1 を所有することはできない. そのため,GMS の管理者は GK を生成することができない. 項目 (3) において RN2 をあらかじめ定めた期間により更新を行っている. また前方安全性と後方安全性を考慮し, メンバが退会するタイミングと新たにメンバを追加するタイミングで RN2 の更新を行うため, グループ鍵が盗まれたとしてもそのグループ鍵を使用している期間内の通信内容は閲覧されるが, その期間以外の通信内容を読み取られることはなく安全である. また, 退会したメンバが新しい GK を生成す

ることはできず通信内容を読み取ることはできない. 新たに参加したメンバも参加する前の RN2 を取得することはできず参加する以前の通信内容を読み取ることはできない. ただし, 退会したメンバであってもそのメンバがグループに在籍していた期間の通信内容は各エンド端末で生成した GK を使用することで閲覧することが可能である. 項目 (4) についてはこの提案の対象がグループコミュニケーションである.

第5章　結論

既存のチャットシステムでは, グループ鍵を用いる通信を行う際に鍵サーバにグループ鍵が残ることにより悪意のある鍵サーバ管理者が通信内容を閲覧できることが課題であった. そこで本論文では, 通信端末に公開鍵証明書を付与し, 生成元が異なる 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セキュアチャットが可能であることを提案した. またグループ鍵を適宜更新することやバージョンを追加することでセキュリティの向上とともにユーザビリティにおける検討を行った. 今後は鍵更新期間の検討を行い, 提案方式を実装し性能評価を行う予定である.

謝辞

本研究にあたり，多大なるご指導とご鞭撻を賜りました, 指導教官である名城大学理工学部情報工学科 渡邊晃教授には心から感謝致します.

本研究を進めるにあたり，様々なご指導を頂きました，名城大学理工学部情報工学科　鈴木秀和准教授に深く感謝致します.

本研究を進めるにあたり，ご意見及びご助言を賜りました，愛知工業大学情報科学部情報科学科　内藤克浩准教授に深く感謝致します.

最後に, 本研究を進めるにあたり, 多くの討論の場においてご意見を賜りました, 渡邊研究室及び鈴木研究室の先輩方, 同期の皆様に感謝心から感謝致します.

参考文献

[1] 総務省-情報通信の現況・政策の動向 (2014).
http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h26/html/nc253120.html

[2] 総務省-情報通信の現況・政策の動向 (2013).
http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h25/html/nc243120.html

[3] 総務省-情報通信の現況と政策動向 (2012).
http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h24/html/nc243120.html

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

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

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

[7] M.Eskicioglu, A.: Multimedia security in group communications: recent progress in key management, autentication, and watermarking, Multimedia Systems Springer -Verlag 2003, pp.239−248 (2003).

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

[9] GSAKMP:Group Secure Association Key Management Protocol,RFC 4535,IETF (2006).

[10] CNET Japan http://japan.cnet.com/news/service/35033099/

研究業績

研究会・大会等

（1）棚田慎也，鈴木秀和，内藤克浩，渡邊 晃：暗号技術を用いたセキュアグループチャットの提案，平成 27 年度電気・電子・情報関係学会東海支部連合大会論文集，講演番号 D2-4(2015).