平成29年度　修 士 論 文

和文題目

ネットワーク管理者に情報が漏洩しない セキュアグループ通信方式の提案

英文題目

Proposal for Secure Group Communication Method that can Prevent Leakage of User Information to Network Administrator

情報工学専攻　渡邊研究室

(学籍番号: 163430015)

棚田 慎也

提出日: 平成30年1月29日

名城大学大学院理工学研究科

概要

グループ通信のセキュリティを満たすため，グループ鍵を用いて相手認証やコンテンツの暗号化を行う方法が一般的に用いられる．このとき鍵管理要件と呼ばれる条件を満たすことが必須とされている．しかし，EFF（Electric Frontier Foundation）の評価によると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．評価項目の中には，鍵管理要件以外に “管理者が閲覧できないように暗号化されているかどうか” があり，多くのシステムではこの項目を満たしていない．GSAKMP（Group Secure Association Key Management Protocol）においてもサーバ管理者がグループ鍵を生成し管理することから，サーバ管理者が通信内容を閲覧でき，情報が漏洩する恐れがある．そこで本論文では，2 種類の乱数からグループ鍵を生成するグループ通信方式を提案する．これによりネットワーク管理者にも情報が漏洩しないセキュアグループ通信が実現できる．提案方式の一部を実装し，仮想環境において動作検証と計測を行った．この結果から，実用上問題がない時間で実行できることを確認し，セキュアグループ通信が実現できることを示した．

目 次

第1章　序論

ネットワーク技術の急速な発展により，ネットワーク利用者は急激に増加し，情報化社会へと発展した．これはパソコンだけでなく携帯電話やスマートフォン，タブレットといった小型の通信機器の普及による影響が大きい [1]．これらの通信端末の普及により，インターネットを介して情報を共有する機会が増加している．プライベートであれば Twitter などのコミュニケーションネットワークや，LINE，Skype などのチャットアプリケーションが主流であり，ビジネスであれば社内 SNS（Social Networking Service）が導入されている企業も少なくない．その中でもチャットアプリケーションに関して，非営利団体である EFF（Electric Frontier Foundation）[⋆1] によりセキュリティ評価が行われた．その結果，Secure Messaging Scorecard [2] によって評価された現状の主流なチャット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わかった．TextSecure [3] や Signal [4] は EFF の評価項目を全て満たすが，各端末間で共通鍵を共有する 1 対 N のグループ通信が主流であり，グループ鍵を用いた通信を行う場合は管理サーバなしでグループ鍵の共有を行うため，処理が複雑になり鍵管理要件を満たすことができない．また，セキュリティが確保されたグループ通信方式として IPsec を用いたグループ通信が考えられるが，現時点で拠点間接続に対応した技術のみであり，端末間のグループ通信に対応した技術は未だ存在しない．そこで業務でも利用可能で，かつ EFF の項目を満たすセキュリティが確保されたグループ通信方式があると大変有用であり，今後幅広い業界で使用されることが考えられる．

グループにおける鍵管理を実現する技術としては Multimedia security in group communications [5] が検討されてきたが，通信相手の認証やグループ鍵の更新期間中メッセージの送受信が困難であるという課題があった．これらの課題を改善した GSAKMP（Group Secure Association Key Management Protocol）[6] が RFC4535 として標準化されている．GSAKMP は一般的なグループ管理方式に用いられるグループ鍵を用いて認証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では一般的に鍵管理要件を満たす必要があるとされている [7]．鍵管理要件にはグループ管理において鍵の更新処理や外部からの攻撃に対しての安全性や参加や退会時の処理に関する規定が含まれていて，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）だけでなくユーザ端末も公開鍵証明書を所有していることが前提であり，この公開鍵証明書を用いてサーバと端末間の認証を確実に行う．この認証によりサーバからユーザ端末へ安全にグループ鍵を配布することができる．鍵管理要件は十分な条件であると考えられていたが，2013 年にアメリカの国家安全保障局 NSA（Nation Security Agency）と連邦捜査局 FBI（Federal Bureau of Investigation）がアメリカに拠点を置く大手 IT 企業のサーバから直接データを収集していたことが発覚した [8]．ネットワーク管理者経由で情報が漏洩したこ

[⋆1] https://www.eff.org/

とから，管理者もセキュリティホールになることが認識されるきっかけとなった．この報道を受けて，EFF は主流なチャットアプリケーションのセキュリティ評価を行った．その評価項目の中には“管理者が閲覧できないように情報が暗号化されているかどうか” が含まれている．GSAKMP は鍵サーバ GCKS がグループ鍵を生成しユーザへ配布する方式であるため，この条件を満たせていない．鍵サーバ管理者でも通信内容を閲覧できない仕組みが保証されているとユーザが安心して利用でき，大変有用である．

そこで本論文では，鍵管理要件と EFF の評価項目の両者を同時に満たすためのセキュアなグループ鍵共有方式を提案する．提案方式はユーザ端末とグループ管理サーバ GMS（Group Management Server）によって構成され，ユーザ端末と GMS においてそれぞれ乱数を生成する．その 2 種類の乱数をユーザ端末間の経路と GMS とユーザの経路で配布し，2 種類の乱数を用いてグループ鍵 GK を生成する．GMS はユーザ端末間において共有した乱数を取得できないため，GK を知ることができず，通信内容を閲覧することができない．提案方式を一部実装し動作検証と評価を行い，実用性があることを確認した．

以後，2 章では関連技術とその課題について述べる．3 章では提案方式について詳細に説明する．4 章では定量的評価と定性的評価の 2 つの側面から評価を行い，最後に 5 章でまとめる．

第2章　関連技術

本章では，グループ鍵を用いる場合に重要な鍵管理要件を述べる．既存技術として IPsec を用いたグルーピング，RFC4535 として標準化されたグループコミュニケーションシステム GSAKMP（Group Secure Association Key Management Protocol）の概要と課題について述べる．

2.1　鍵管理要件

グループ鍵管理プロトコルの目的は，機密性や認証のため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 [7].

- 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　IPsec を用いたグルーピング

セキュリティが確保されたグループ通信方式として IPsec を用いたグルーピングが考えられる．IPsec とは暗号化システムの技術によりネットワーク層でデータのセキュリティを保護するために使用されるプロトコルである．IPsec によって IP に対して様々なセキュリティを付加することができ，トンネリング機能や暗号化機能，相手認証機能などがあり汎用性が重視されている．また通信は 1 対 1 であり，IPsec を使用するにあたりアドレスタイプやローカルアドレスなど必要な設定項目が多くある．図 1 に IPsec を用いたグルーピングを示す．各端末間の設定を個々に行い，端末ごとに認証と暗号化を行うことにより実質的にセキュアなグループ通信を構築できる．しかし実用性を考えると端末ごとに設定することは困難である．設定を自動化する技術も存在するが [9]，拠点間接続に対応したものであり，ユーザ端末どうしの設定を簡易化する技術は存在しない．NAT を経由する場合 IP ヘッダを認証する際に書き換えられたアドレス情報が書き換えられると不正パ

*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．

ケットとして認証エラーが発生するなど IPsec は NAT との相性が悪いため，NAT を経由するネットワークでは利用できず，利用範囲が限定されている．

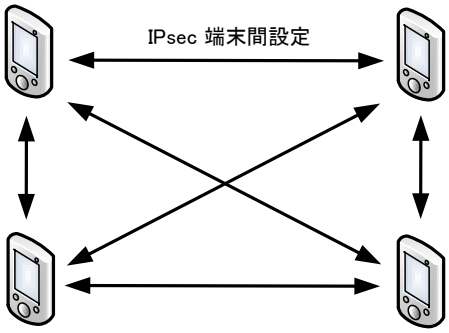

図 1　IPsec を用いたグルーピング

2.3　GSAKMP

2.3.1　GSAKMP の概要

GSAKMP [6] 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あり 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 の用途として様々なアプリケーションと併用が可能であり，メンバー間のマルチキャストやユニキャスト通信を保護する役割を持ち，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 はグループオーナー，鍵サーバ GCKS（Group Controller Key Server），グループメンバの 3 つの主要な要素でグループ管理を分散している．GSAKMP では GCKS だけでなくグループメンバとなるユーザ端末も公開鍵証明書を所有していることが前提であり，これによってサーバと端末間の確実な認証が可能である．グループオーナーは鍵の更新処理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 はグループ鍵の生成や配布および鍵の更新，グループメンバの管理を行う．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必要がある．例えばグループメンバの変更によってグループ鍵が更新された場合，それが適切であるか，セキュリティポリシーに基づいているかどうかを確認しなければならない．

2.3.2　GSAKMP の鍵共有方式

GSAKMP におけるグループ鍵共有シーケンスを図 2 に示す．新たに参加するユーザはグループオーナーから招待を受けていることが前提である．

（1）Request to Join

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，このメッセージの中にはユーザの公開鍵証明書や招待されたときに付与されているグループ ID などが含まれている．

（2）Key Download

参加申請を受け取った GCKS は新たに参加するユーザの公開鍵証明書の検証を行う．この検証が成功し，正しいユーザであることを確認した場合 GCKS からユーザへ 2 つの鍵を配布する．鍵の 1 つは GTPK（Group Traffic Protection Key）でありグループ通信のデータを暗号化するグループ鍵であり，もう 1 つは Rekey Key と呼ばれるグループ鍵を更新する際に使用される要素となる鍵である．また，認証に失敗した場合，Key Download の代わりに Request to Join Error を送信し，認証が失敗したことをユーザへ通知する．このメッセージはオプションであり，送信の有無を設定することができる．

（3）Key Download Ack

2 つの鍵を受け取ったユーザは応答を返す．これらの一連の処理が完了した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 から各グループメンバへグループ鍵更新の通知を送る．通知の中にはグループ鍵更新の要素が含まれており，受け取ったメンバはあらかじめ配布されている Rekey Key と新たな要素を用いて GTPK の更新を行う．グループ鍵の更新を行った各グループメンバは GCKS へ更新を行った旨の通知を送ることで新たな Rekey Key を GCKS から受け取ることが可能である．これにより，ユーザは退会した後の通信内容を閲覧できない，また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ため，前方秘匿性と後方秘匿性の両者を確保することができる．

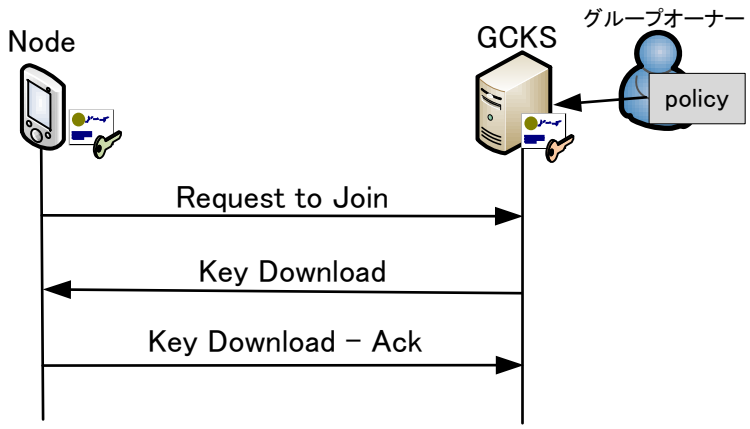

図 2　GSAKMP におけるグループ鍵共有シーケンス

2.3.3 GSAKMP の課題

GSAKMP では GCKS とユーザの両者が公開鍵証明書を所有しているため相互認証を行い，確実にグループ鍵を配布することが可能である点が特徴である．そのため，外部のコミュニケーションサーバを利用したチャットなどの通信においても，コンテンツを暗号化できるため，コミュニケーションサーバから情報が漏洩する恐れはない．しかし，GCKS がグループ鍵 GTPK とその生成要素である Rekey Key を生成しユーザへ配布しているため，EFF の Secure Messaging Scorecard に提示されている “管理者が閲覧できないように暗号化されているかどうか” を満たしていない．また，グループオーナーが鍵の更新処理やメンバの招待方法の設定を行うことができるため，悪意のあるメンバをグループに混入させることも可能である．

第3章　提案方式

本章では，提案するセキュアグループ通信方式について述べる．

3.1　提案方式の概要と目的

本提案はサーバ管理者が通信内容を読み取ることができないセキュアなグループ通信方式を実現することが目的である．この目的を達成するために，ユーザ端末間で共有し，ユーザのみが使用する乱数 RN1 とグループ管理サーバ GMS（Group Management Server）からユーザ端末へ配布する乱数 RN2 を用意し，その 2 種類の乱数からグループ鍵 GK を生成する．ユーザ端末間で共有する乱数は GMS が経路上に存在していたとしても暗号化が行われ共有されているため，GMS は取得できず，GK を所有しているユーザのみが当該グループのメンバとして通信を行うことが可能である．

3.2　システム構成と暗号アルゴリズム

図 3 に乱数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 とユーザ端末の 2 つから成る．GMS はグローバルアドレス空間上に設置され，グループ情報の管理や乱数 RN2 の生成，配布を行う．GMS が保持するグループ情報のデータベースの例を表 1 に示す．グループ ID やユーザのログイン状況を判断するログインステータス，ユーザの識別子（FQDN，IPv6 アドレス，グローバル/プライベート IPv4 アドレス），RN2 バージョンを管理している．ログインステータスは，GMS からグループメンバに乱数 RN2 を配送する際に確認し送信するか否かを決定する．RN2 バージョンはユーザが所持する RN2 と GMS が配布する RN2 のバージョンを確認する際に使用する．乱数 RN2 はユーザからの申請を受けて生成や配布を行い，あらかじめ設定されているタイミングで更新を行う．ユーザ端末は RN1 の生成やユーザ間共有および GK の生成や管理を行う．RN1 は半永久的に使用し，メンバの参加や退会があった場合でも更新は行わない．またグループメンバの変更を行った際に GMS へ Group Member Notification を送信し，グループ情報の更新を通知する．グループメンバの招待はユーザ端末が実行し，GMS 管理者はグループメンバの管理には関与しない．

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），ハッシュ関数は SHA256 を使用することで暗号化アルゴリズム上セキュリティの課題がないことを前提としている．

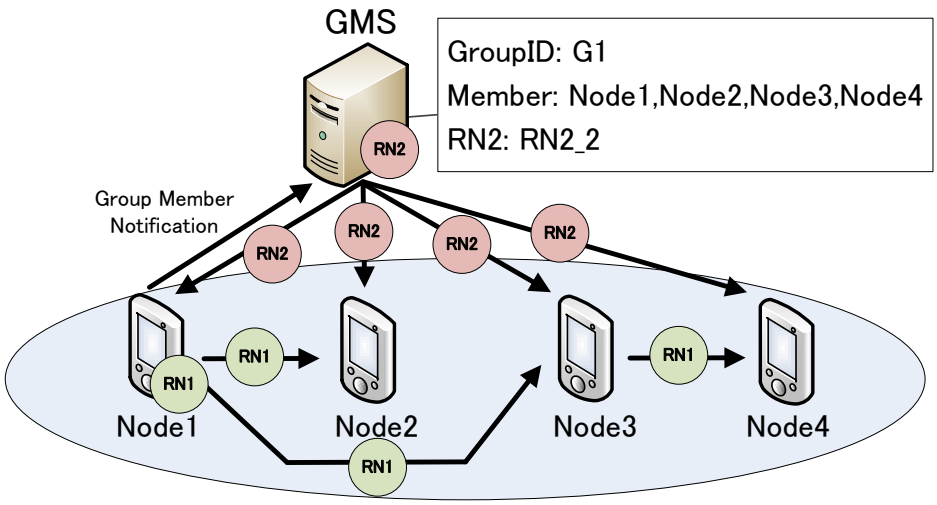

図 3　提案方式のシステム構成

表 1　GMS が所有するグループ情報データベース

Group ID	Login Status	FQDN	IPv6	global IPv4	private IPv4	RN2 version
G1	OFF	node1	—	200.0.10.1	192.168.1.2	1
G1	ON	node2	—	128.0.10.10	—	2
G1	ON	node3	—	201.0.100.10	192.168.10.10	2
G1	ON	node4	—	192.0.2.1	—	2

3.3　鍵共有方式

RN1 をサーバ管理者に分からないようにグループメンバ間で共有する方法として，ユーザ端末に公開鍵証明書を所有させる方法（公開鍵証明書方式）とエンドツーエンド通信が可能なセキュアネットワークを使用する方法（エンドツーエンド通信方式）の 2 通りが考えられる．公開鍵証明書方式の場合，乱数共有に一般的なコミュニケーションサーバを使用することができるが，公開鍵証明書の取得や管理にコストがかかる．一方，エンドツーエンド通信方式の場合，ユーザ端末が公開鍵証明書を取得する必要はないが，エンドツーエンド通信が可能なセキュアネットワークを準備する必要がある．このようなネットワークの例として NTMobile [10] [11] [12] がある．

提案する鍵共有方式には 2 つのフェーズがあり，第 1 フェーズは方式ごとに異なる RN1 の共有であり，第 2 フェーズは両方式共通の RN2 の配布と GK の生成である．以下に各方式における RN1 共有方法を述べる．両者の RN1 共有シーケンス，およびユーザの操作には統一性を持たせるようにした．各方式において RN1 の共有が完了した後，RN2 の配布と GK の生成を行う．

3.3.1　RN1 の共有

(1) 公開鍵証明書方式

図 4 に公開鍵証明書を用いた提案方式の RN1 共有シーケンスを示す．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．招待者/被招待者と GMS はあらかじめ公開鍵証明書を用いて認証が完了していることが前提である．RN1 を公開鍵で暗号化して送信することができるため，端末間の通信には一般のコミュニケーションサーバ CS を経由した方法を使用しても構わない．公開鍵証明書方式における RN1 の共有手順は以下の通りである．

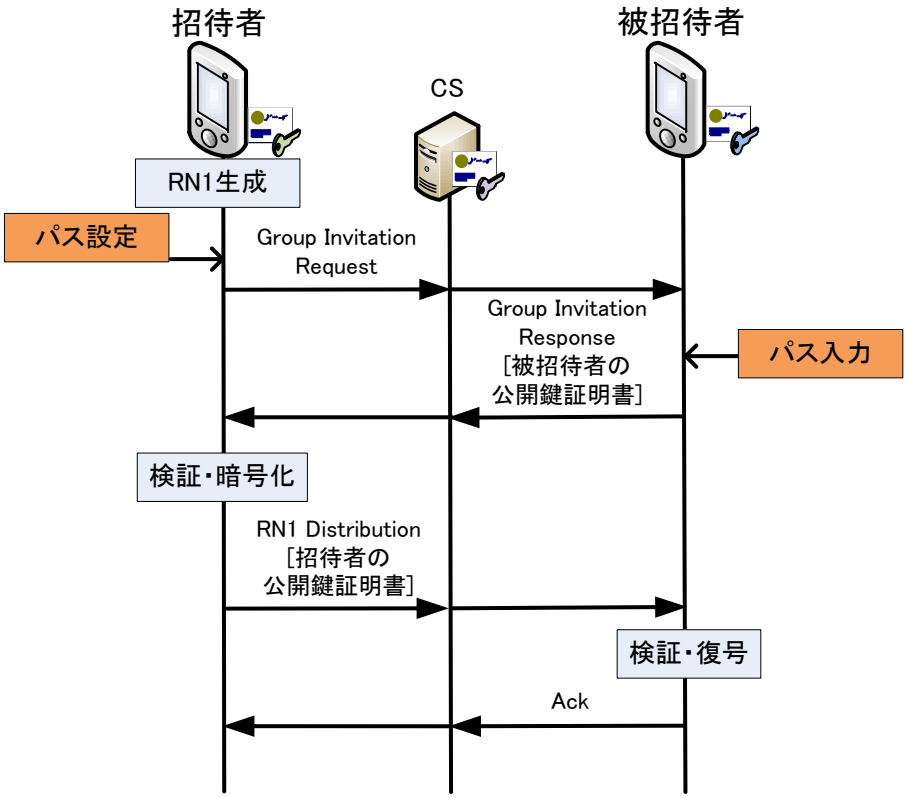

図 4　公開鍵証明書を用いた RN1 共有シーケンス

図 4 において，グループを作成する最初の招待者の端末は，RN1 を生成する．RN1 は新たにメンバが参加した場合でも半永久的に引き継がれる．招待者と被招待者はあらかじめパスワードを共有しておくことが望ましい．このパスワードは操作しているユーザの正当性を確認することが目的である．被招待者は招待者から権限を与えられた場合，新たに招待者となることができる．招待者は被招待者へ Group Invitation Request を送信する．これはグループ招待であり，このメッセージ内には当該グループの Group ID と Group Name が含まれている．この時点では DoS 攻撃の対象となることを防ぐため，招待者側の公開鍵証明書は送付しない．被招待者は事前共有したパスワードを入力し，Group Invitation Response を返す．このメッセージは Request に対する応答であ

り，被招待者の公開鍵証明書が含まれている．招待者は被招待者の公開鍵証明書の検証を行い，正しい場合 RN1 を被招待者の公開鍵で暗号化する．そして招待者の公開鍵証明書とともに暗号化した RN1 を RN1 distribution として被招待者へ送信する．被招待者は招待者の公開鍵証明書の検証を行い，正しい場合 RN1 を自身の秘密鍵で復号し，応答を返す．以上により RN1 の共有が完了する．公開鍵証明書を確認することで通信相手の認証を確実に行うことが可能である．

(2) エンドツーエンド通信方式

エンドツーエンド通信が可能なセキュアネットワークとして NTMobile [10] [11] [12] 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．通信接続性とはユーザ端末がグローバルアドレス/プライベートアドレス空間にいることに関わらず，双方向から通信が開始できる機能であり，移動透過性とは通信中にネットワークの切り替えを行っても通信を継続できる機能である．エンドツーエンドセキュリティであり，NTMobile の詳細に関して本論文の提案内容とは関連しないため省略する．

NTMobile を導入した端末を NTM 端末といい，GMS と NTM 端末間は GMS 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する．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ち，期限が切れると GMS の公開鍵と NTM 端末のパスワードによって認証を再度行い，新たな共通鍵を共有する．

図 5 にエンドツーエンド通信が可能なセキュアネットワークを用いた提案方式の RN1 共有シーケンスを示す．公開鍵証明書方式とは異なりパスワードを招待者と被招待者の両者が設定しあらかじめ共有することが望ましい．エンドツーエンド通信方式の RN1 の共有手順は以下の通りである．

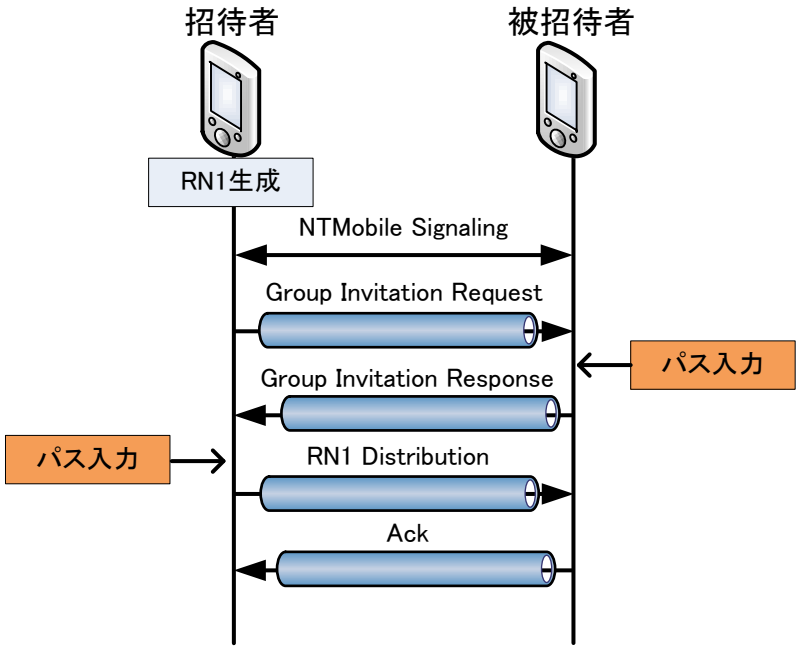

図 5　エンドツーエンド通信を用いた RN1 共有シーケンス

招待者と被招待者は通信に先立ち，エンドツーエンド通信の経路を構築する NTMobile シグナリングにより端末間に暗号化されたトンネル経路を生成する．このトンネル経路を用いることによりセキュアなエンドツーエンドの暗号通信が可能である．トンネル経路を利用し，招待者から被招待者へグループ招待である Group Invitation Request を Group ID や Group Name とともに送信する．被招待者はあらかじめ共有したパスワードを入力し，招待者へ Group Invitation Response を返す．招待者は被招待者が入力したパスワードを確認し，正しい場合 RN1 と被招待者のパスワードを入力し RN1 Distribution を送信する．被招待者は受信したパスワードを確認し，招待者へ Ack を返すことにより RN1 の共有が完了する．

3.3.2 RN2 の配布と GK 生成

図 6 に RN2 配布/GK 生成シーケンスを示す．あらかじめ端末側は公開鍵証明書またはパスワードを用いて GMS と認証を行い，各端末と GMS の共通鍵を共有していることを前提としている．そのため，各端末と GMS 間の通信は共通鍵を用いた暗号化通信である．RN2 配布と GK 生成は共通の処理である．

招待者は GMS へ ID とともに Group Member Notification を送信し，グループ情報の変更を通知する．Group Member Notification のメッセージ内にはグループ ID と参加する端末のユーザ識別子（FQDN，IPv6 アドレス，グローバル/プライベート IPv4 アドレス）が含まれている．この情報を用いて GMS はグループ情報管理用データベースの更新を行い，新たに参加したメンバの情報を加える．また当該グループの RN2 を作成し，グループメンバへそれぞれ RN2 Distribution として配布する．RN2 はあらかじめ設定された更新期間を経過した場合，もしくはグループメンバが新たに参加や退会をした場合にその都度更新を行いグループメンバへ配布される．これにより前方秘匿性や後方秘匿性を確保する．

RN1，RN2 を取得した各ユーザはハッシュ関数を用いて [RN1|RN2|GroupName] のハッシュ値を取り，グループ共通鍵 GK として生成する．この方式によるとサーバ管理者は RN1 や Group Name を取得できないため，グループ鍵を生成できず通信内容を閲覧し漏洩することができない．そのため，同一グループメンバのみが GK を所有することができる．

3.4 RN2 の更新処理

鍵管理要件を満たすため，GMS はあらかじめ設定された更新を経過した場合，もしくはグループメンバが新たに参加や退会した場合に RN2 を更新する．新たにメンバが参加する場合は鍵共有方式と同様であり，GMS において RN2 を更新し，全てのグループメンバへ RN2 を送信する．

図 7 にメンバを退会させる場合における RN2 の更新処理を示す．例としてユーザ 3 が参加/退会の権限を持ち，ユーザ 3 がユーザ 4 を退会させるケースを説明する．まず，ユーザ 3 からユーザ 4 へ退会指示を送ると，ユーザ 4 は強制的に退会させられる．ユーザ 3 から GMS へ Group Member Notification を送りユーザ 4 が退会したことを通知する．通知を受けた GMS は RN2 の更新を行い，新しい RN2_2 を生成し，自身のデータベースにあるメンバの情報を更新する．その後 GMS は新

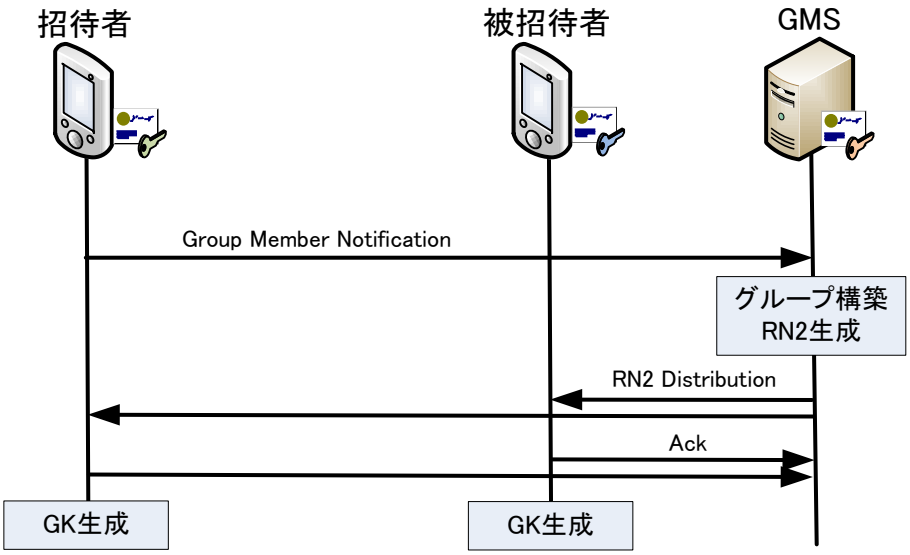

図 6　RN2 配布/GK 生成シーケンス

しい RN2_2 を更新されたグループメンバへ配布する．各ユーザ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秘匿性を確保することができる．

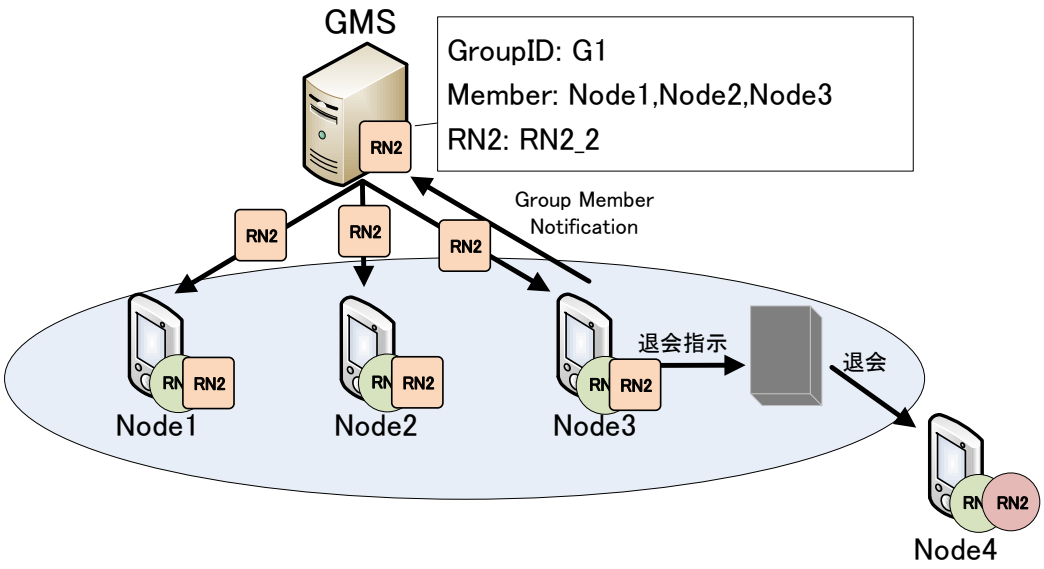

図 7　メンバ退会時の RN2 更新処理

3.5 RN2 のバージョン管理機能

提案方式における RN2 の更新において，新しく生成した RN2 がすべてのメンバに確実に届くかどうか保証できない．ユーザ端末の電源がオフの状態の時にメンバの変更などがあると RN2 を更新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 RN2 のバージョンが異なるため，GK を共有できず，暗号化通信を行えない．

そのため，RN2 のバージョン管理機能は必須である．図 8 にバージョン管理機能を用いた RN2 の更新処理を示す．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．グループメンバ更新により招待者から GMS へ Group Member Notification が送信されると，GMS において RN2 の更新を行い，新たな RN2 を GMS から各グループメンバへ配布する．しかし，電源がオフである Node1 には RN2 を配布することができない．ユーザのタイミングで Node1 の電源をオンにし，Node1 から GMS へ RN2 Inquiry を送信する．このメッセージは RN2 の問い合わせであり，受け取った GMS はユーザ認証を行うとともに，ユーザが所有している RN2 のバージョンを確認する．バージョンが異なっていた場合，GMS は新しい RN2 を RN2 Distribution として送信する．新しい RN2 を受け取った Node1 は GMS へ応答を返し，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．

他にも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る状況や電波が届かない状況などが考えられる．そのためメッセージフォーマット内に RN2 version 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 version を確認する．メッセージ受信側が古い場合は，受信側が GMS へ RN2 Inquiry を行う．メッセージ送信側が古い場合は，受信側から送信側へ通知を送信し，送信側から GMS へ RN2 Inquiry を行う．

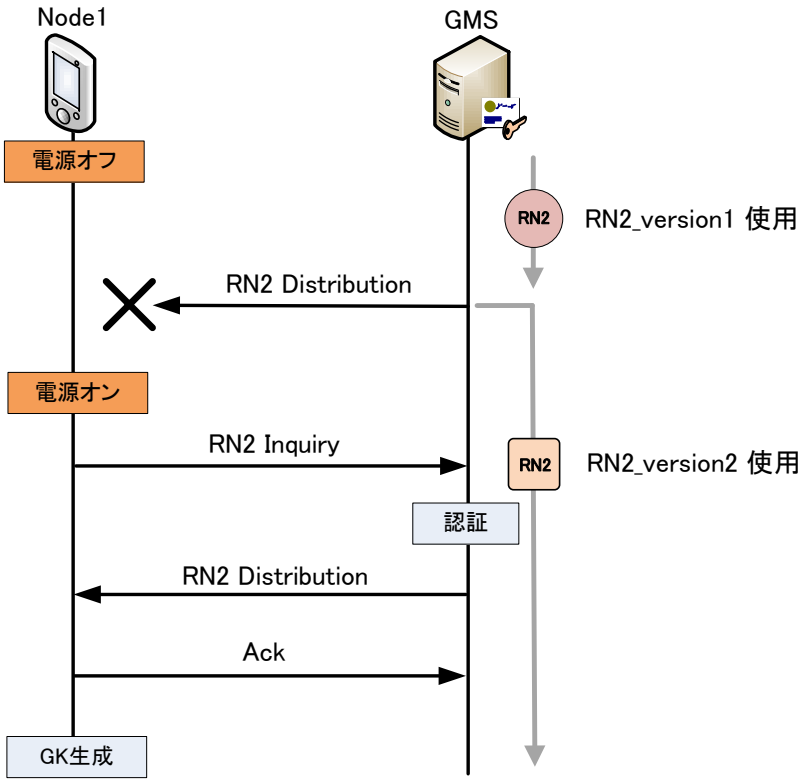

図 **8**　バージョン管理機能を用いた **RN2** の更新処理

第4章　実装と評価

本章では，提案方式の一部分を実装したためその動作検証，性能評価，および既存技術との比較について述べる．

4.1　乱数共有部の実装と計測

本論文にて提案する方式の 1 つである公開鍵証明書方式の RN1 共有部を実装し仮想環境において動作検証を行った．

4.1.1　テスト環境と実装箇所

表 2 にホスト PC の構成，表 3 に仮想環境の構成を示す．1 台のホスト PC 上に仮想マシン VMware Player[⋆1] を用いて Node1（招待者）, Node2（被招待者）, 認証局 CA（Certificate Authority）の 3 台を構築した．認証局 CA は通常階層構造であるが，今回はルート CA1 台のみとし，自己署名を行っている．また 2 台の端末 Node1（招待者）, Node2（被招待者）の公開鍵証明書を生成し配布を行った．公開鍵のアルゴリズムは RSA であり，鍵長 1024 ビットである．

実装箇所は図 9 の通りであり，被招待者からメッセージを送信し，招待者側で公開鍵証明書の検証，公開鍵の取り出し，RN1 の暗号化を行う．そして招待者はメッセージを送信し，被招待者側で公開鍵証明書の検証，RN1 の復号を行う一連の動作である．openssl コマンドを用いて C 言語により，各公開鍵証明書のチェーン検証や証明書から公開鍵を取りだし暗号化/復号する処理を実装した．

表 2　ホスト PC の構成

	ホスト PC
OS	Windows7 64bit
CPU	Intel Core i7-2600 3.40GHz
Memory	8.00GB

4.1.2　動作検証と性能評価

招待者側では被招待者の公開鍵証明書の検証と公開鍵証明書から公開鍵の取り出し，RN1 の暗号化，被招待者側では招待者の公開鍵証明書の検証と RN1 の復号を行う．各端末において正しく

[⋆1] https://www.vmware.com/jp

表 3　仮想環境の構成

	Node1（招待者）, Node2（被招待者）, CA
OS	Ubuntu 14.04
Kernel version	3.13.0-24-generic
CPU	Intel Core i7-2600 3.40GHz
Memory	各 1.00GB

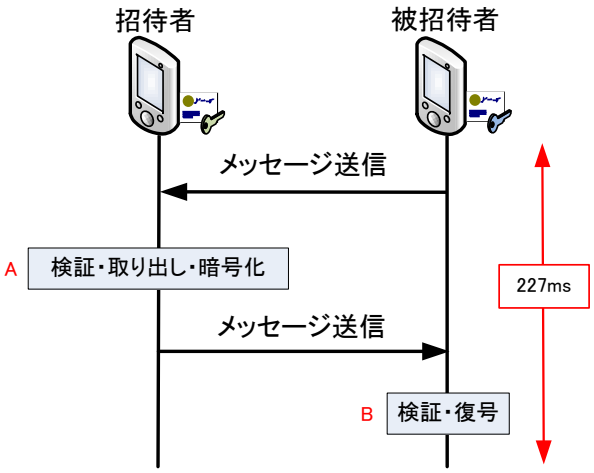

図 9　実装部と一連の動作の計測結果

検証が行えていること，RN1 の復号が行えていることを確認した．また，これらの一連の動作と各処理にかかった時間を計測し，性能評価を行った．各処理における測定には C 言語で準備されている clock_gettime() を用いて，各試行回数 10 回の平均時間を取得した．表 4，5 にその結果を示す．表 4 は招待者側の測定結果であり，A は図 9 における A である．表 5 は被招待者側の測定結果であり，B は図 9 における B である．

この測定結果より，メッセージ送信処理にかかる時間に比べ，公開鍵証明書における検証や公開鍵の取り出し，公開鍵を用いた RN1 の暗号化・復号において多くの時間を要するが，この処理はメンバの招待時のみであるため実用上問題がないと考えられる．

また，実環境を想定すると，スマートフォンやタブレット端末による実行が考えられる．今回使用した PC とスマートフォンにおいてベンチマークテストを行い，処理時間を計測した結果，スマートフォンは PC に比べ約 2 倍の処理時間がかかり，通信時間は約 5 倍であった．そのため，通信効率が悪い場合であっても全体の処理時間は 2 倍から 3 倍程度であることが考えられ，実用上問題がないと考えられる．

表 4　招待者側における実装部測定結果

招待者		[ms]	[ms]
A	証明書検証	46	139
	公開鍵取り出し	45	
	RN1 暗号化	48	
送信処理			0.4

表 5　被招待者側における実装部測定結果

被招待者		[ms]	[ms]
B	証明書検証	46	86
	RN1 復号	40	
送信処理			0.4

4.2　関連技術との比較

表 6 にグループ鍵の共有に着目した関連技術との比較を示す．項目は以下の 2 つであり，比較対象は GSAKMP 及び TextSecure とした．

（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.

評価項目 (1) は EFF により提示されたセキュリティ評価項目の一部であり，評価項目 (2) は一般的にグループ鍵を使用する際に必要となる項目である．TextSecure は EFF によるセキュリティ項目を全て満たしている数少ないメッセージアプリケーションであるが，グループ鍵を用いる場合，グループ鍵更新処理が複雑であり前方秘匿性を確保できない.

表 6　関連技術との比較

	項目 1	項目 2
GSAKMP	×	○
TextSecure	○	×
提案方式	○	○

GSAKMP はグループ鍵の生成や各種攻撃に対する対策が定義されており，更新期間も設定されているため前方秘匿性や後方秘匿性を満たすため鍵管理要件を満たしている．しかし，鍵サーバ GCKS においてグループ鍵 GTPK と更新鍵 Rekey Key のどちらも生成しているため，サーバ管理者が通信内容を読み取れる恐れがあり評価項目 (1) を満たしていない．TextSecure は管理者がユーザの通信内容を読めないように暗号化が行われているが，グループ鍵を用いてグループ通信を行う場合，管理サーバなしでグループ鍵の更新を行うことにより処理が複雑であり前方秘匿性を満

たすことができず，項目 (2) を満たしていない．

提案方式では，RN1 をユーザ間で安全に共有しグループ鍵を生成するのでサーバの管理者がグループ鍵を生成することができず，管理者が読めないようにコンテンツの暗号化を行うことができる．評価項目 (2) において 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や退会ごとに RN2 の更新を行っているため鍵管理要件を満たしている．

第5章　結論

既存のグループ管理技術では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漏洩する恐れがあった．そこで本論文では，2 種類の乱数からグループ鍵を生成するセキュアグループ通信方式を提案した．その 2 種類の乱数をグループ管理サーバ GMS とユーザ端末の経由とユーザ端末間の経路によって配布を行う．端末間の経路において，公開鍵証明書方式とエンドツーエンド通信方式の 2 通りを提案し，各方式において安全に共有でき，サーバ管理者に情報が漏洩しないセキュアなグループ通信が可能であることを示した．また，グループメンバの参加や退会時にグループ鍵を更新することにより，前方秘匿性と後方秘匿性を満たし，RN2 のバージョン管理機能により通信できない状態である場合でも，確実にグループ鍵の更新を行うことができる．

提案方式の評価として定量的評価と定性的評価を行った．定量的評価では，仮想環境において提案方式の一部を実装し，各端末での公開鍵証明書の検証や乱数の暗号化/復号の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．定性的評価では，関連する技術と比較を行い，提案方式がよりセキュアな通信が可能であることを示した．

謝辞

本研究を進めるにあたり，多大なる御指導と御教授を賜りました，指導教官である名城大学大学院理工学研究科　渡邊晃教授に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，名城大学大学院理工学研究科　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，ご意見並びに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　内藤克浩准教授に感謝致します.
本論文を作成するにあたり，快く副査を引き受けていただきました名城大学大学院理工学研究科　柳田康幸教授に心より感謝致します.
最後に，本研究を進めるにあたり，数々の有益なご助言を賜りました，渡邊研究室および鈴木研究室の諸氏に感謝致します.

参考文献

[1] 総務省｜平成 28 年通信利用動向調査の結果，
http://www.soumu.go.jp/menu_news/s-news/01tsushin02_02000112.html (2018 年 1 月 12 日アクセス).

[2] Electronic Frontier Foundation：Secure Messaging Scorecard.
https://www.eff.org/secure-messaging-scorecard (2018 年 1 月 12 日アクセス).

[3] Signal Private Group Messaging.
https://signal.org/blog/private-groups/ (2018 年 1 月 29 日アクセス).

[4] Paul Rosler, Christian Mainka, Jorg Schwenk: More is Less: On the End-to-End Security of Group Chats in Signal, WhatsApp, and Threema, 3rd IEEE European Symposium on Security and Privacy (EuroS & P 2018).

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

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

[7] M. Baugher, R. Canetti, L. Dondeti, F. Lindholm: Multicast Security (MSEC) Group Key Management Architecture,RFC 4046,IETF (2005).

[8] CNET Japan
https://japan.cnet.com/article/35033099/ (2018 年 1 月 12 日アクセス).

[9] ITpro Special 広域ネットワークで注目の SD-WAN
http://special.nikkeibp.co.jp/atcl/ITP/17/nttpc0314/

[10] 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).

[11] 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 (2013).

[12] 納堂博史，杉原史人，鈴木秀和，内藤克浩，渡邊　晃：NTMobile の実用化に向けた統合的枠組の検討，情報処理学会研究報告モバイルコンピューティングとユビキタス通信研究会（MBL），Vol. 2015 MBL 77, No. 20, pp. 1–8 (2015).

研究業績

国際会議（査読あり）

（1）S. Tanada, H. Suzuki, K. Naito and A. Watanabe: Proposal for Secure Group Communication using Encryption Technology, *The Ninth International Conference on Mobile Computing and Ubiquitous Networking (ICMU 2016)*, Kaiserslautern, Germany, Oct.2016.

国内会議（査読あり）

（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コミュニケーションの提案，マルチメディア，分散，協調とモバイル（DICOMO2016）シンポジウム論文集，pp. 366 –371，Jul. 2016.

（2）菅沼良一，納堂博史，棚田慎也，鈴木秀和，内藤克浩，渡邊 晃:リング状経路を用いたアプリケーションレイヤマルチキャストの提案，マルチメディア，分散，協調とモバイル (DICOMO2017) シンポジウム論文集，pp. 1715 –1720，Jun. 2017.

研究会・大会等（査読なし）

（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， Sep. 2015.

（2）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コミュニケーションの提案，情報処理学会第 78 回全国大会講演論文集，Mar.2016.

（3）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なグループ管理方式の提案，情報学ワークショップ 2016（WiNF2016）論文集，Nov.2016.