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

163430015　棚田 慎也
渡邊研究室

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て情報を共有する機会が増加している．その中でもグループ内での情報共有のためにグループ通信におけるセキュリティが極めて重要な項目として挙げられる．そ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を暗号化する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，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 の評価項目の中には，鍵管理要件以外にグループ鍵が管理者が読め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．そこで本稿では，2 種類の乱数からグループ鍵を生成することにより鍵管理要件と EFF の評価項目の両者を満たしたグループ通信方式を提案する．

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．
- 鍵データは厳重に保管され，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 既存技術 GSAKMP

グループ鍵を用いたグループ通信におけるセキュリティフレームワークとして GSAKMP(Group Secure Association Key Management Protocol[1]:RFC4535) が標準化されている．GSAKMP ではグループオーナー GO，鍵サーバ GCKS，グループメンバ GM の 3 つの主要な要素でグループ管理を分散している．GO によって作成されたセキュリティポリシーを基にアクセス制御を実行する．そのセキュリティポリシーに基づき GCKS は GM の管理やグループ鍵の生成を行う．GSAKMP では GCKS だけでなく GM も公開鍵証明書を所有し GCKS と各 GM 間の相互認証を確実に行う．GSAKMP におけるグループ鍵共有シーケンスを図 1 に示す．新たに参加するユーザは GO から招待されていることが前提である．招待されたユーザは GCKS へ Request to Join を送信する．GCKS はユーザの公開鍵

[1]https://www.eff.org/

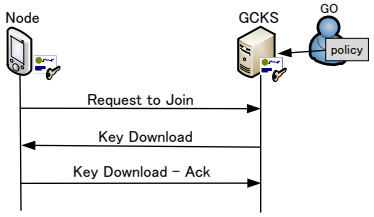


図 1: GSAKMP におけるグループ鍵共有シーケンス

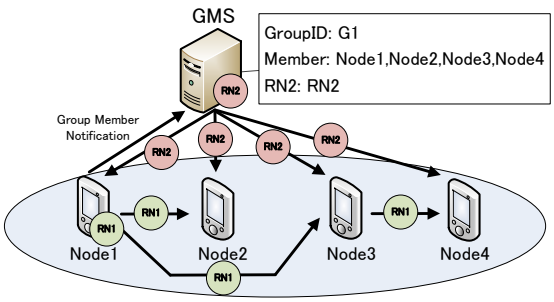


図 2: 提案方式のシステム構成

証明書を確認しグループ鍵を配布する．ユーザは応答を返すことによってこのタイミングで当該グループのメンバとなる．

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．また，GO が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．

3. 提案方式

3.1 提案方式の原理とシステム構成

本提案はサーバ管理者にも情報が漏洩し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，生成元が異なる 2 種類の乱数 RN1,RN2 を異なる配送経路でグループメンバへ配送し，その 2 種類の乱数から新たなグループ鍵 GK を生成する．鍵管理サーバ GMS(Group Management Server) の管理者は GK を知ることができないため，EFF の評価項目を満たすことができる．

図 2 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 はグループ情報のデータベース管理，乱数 RN2 の生成管理および配布を行い，あらかじめ設定してあるタイミングで RN2 の更新を行う．ユーザ端末は乱数 RN1 の生成，共有および GK の生成を行う．また，グループメンバの招待はユーザ端末が実行し，GMS の管理者はグループメンバの管理には関与しない．なお公開鍵

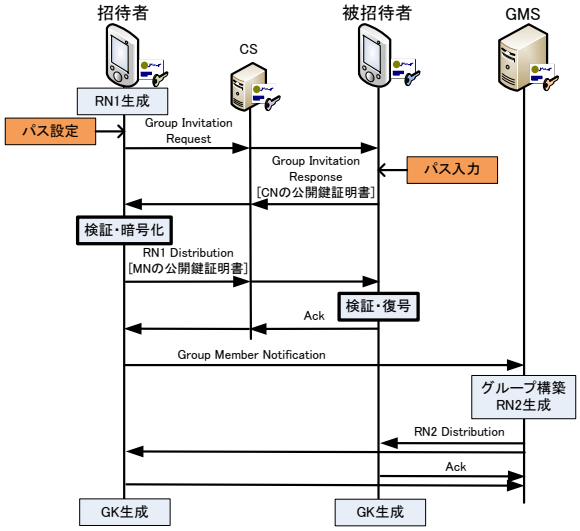


図 3: 公開鍵証明書を用いた鍵共有シーケンス

は RSA(鍵長 1024 ビット以上)，ハッシュ関数は SHA256 を使用し，暗号化アルゴリズム上はセキュリティに課題がないことを前提とする．

3.2 鍵共有方式

提案方式の実現方法として，公開鍵証明書をユーザ端末に所有させる方法と，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の 2 通りが考えられる．公開鍵証明書を用いる場合，乱数配送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が可能なネットワークの場合，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[2] がある．

図 3 に公開鍵証明書を用いた鍵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．RN1 を公開鍵で暗号化できるため，ユーザ間の通信には一般のコミュニケーションサーバ CS を経由した方法であっても構わない．図 3 において，最初の招待者の端末は RN1 を生成する．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．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グループ ID とともに Group Invitation Request を送信する．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，被招待者の公開鍵証明書とともに Group Invitation Response として応答を返す．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 Distribtion 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，Ack を返す．以上により RN1 の共有が完了する．RN1 は新たなメンバが加わった場合も半永久的に引き継がれる．

次に招待者は GMS へ Group Member Notification を送信し，グループ情報の変更を通知する．GMS はグループ管理用のテーブルの更新を行う．また当該グループの

表 1: 提案方式の証明書検証と暗号化/復号の計測結果

	招待者側 [ms]	被招待者側 [ms]
証明書検証	67.24	67.83
暗号化/復号	91.02	48.82

表 2: 関連技術と提案方式の比較

	項目 1	項目 2	項目 3
GSAKMP	×	○	○
ChatSecure	○	○	×
提案方式	○	○	○

RN2 を作成し，グループメンバへそれぞれ配布する (RN2 Distribution)．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．

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 を生成する．これにより同一グループメンバのみが GK を所有し，この内容は管理者にもわからない．

4. 実装と評価

4.1 実装と性能評価

提案方式の一部を実装し，処理時間の計測を行った．仮想環境において図 3 に相当するシステムを構築し，GMS とユーザ端末に公開鍵証明書を配布した．測定箇所は図 3 の検証・暗号化と検証・復号の 4 箇所であり，各試行回数 10 回の平均時間を取得した．その結果を表 1 に示す．証明書検証と公開鍵による暗号化/復号に多くの時間を要するが招待時のみであり，実用上問題がないと言える．

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，ChatSecure である．ここで ChatSecure は EFF の評価項目を全て満たすメッセージアプリケーションであるが 1 対 1 通信しか行えない．

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. グループ通信が可能であるか.

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．ChatSecure は 1 対 1 通信のみであり項目 3 を満たせない．提案方式では全項目を満たしており，最も安全である．

5. まとめ

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，サーバ管理者にも情報が漏洩し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] 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).

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

情報工学専攻 渡邊研究室
163430015

棚田　慎也

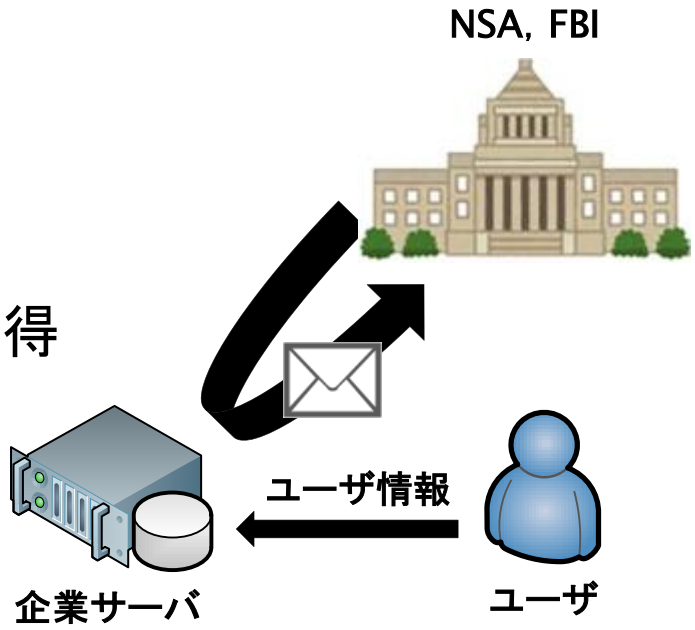
Watanabe Lab.

研究背景

グループコミュニケーションの有用性
- 暗号化通信
- セキュリティ：鍵管理要件

米国NSA，FBI
- 大手IT企業のサーバからユーザ情報取得

EFF※のセキュリティ評価
- 「管理者が読めないように
  暗号化されているかどうか」



管理者に情報漏洩しないセキュアグループ通信方式の提案

※ Electronic Frontier Foundation
Secure Messaging Scorecard <https://www.eff.org/secure-messaging-scorecard>

鍵管理要件

機密性や認証のために必要な情報を最新の暗号化状態で提供

- あらかじめ定めた期間で定期的に更新
- 鍵データは正規ソースからのみ入手可能
- 鍵管理プロトコルはリプレイ攻撃やDoS攻撃に対して安全
- 参加前の通信内容を閲覧できない（後方秘匿性）
  退会後の通信内容を閲覧できない（前方秘匿性）

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

既存技術 –GSAKMP–

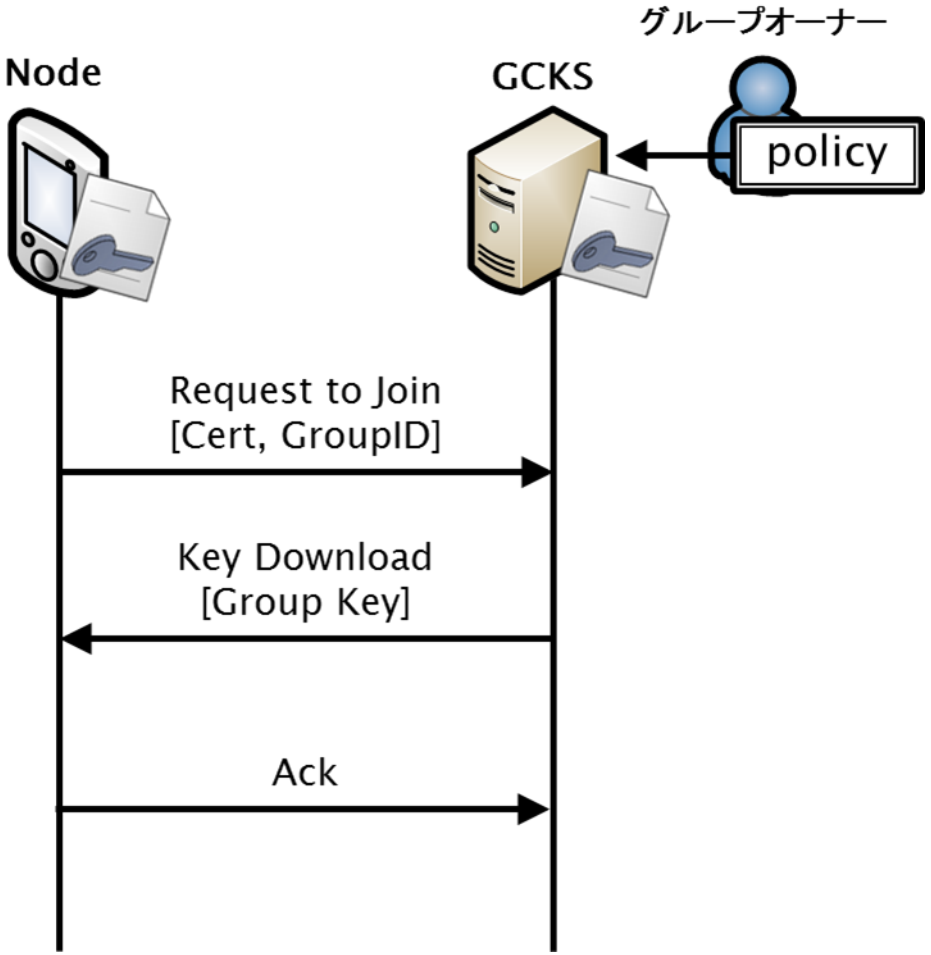
(Group Secure Association Key Management Protocol)

暗号化グループを生成・管理するフレームワーク(RFC4535)

- グループオーナー
  - セキュリティポリシー提供
- 鍵サーバ GCKS
  (Group Controller Key Server)
  - グループ鍵管理
- グループメンバ　Node
  - 公開鍵証明書を所有

課題

- GCKSからグループ鍵を配布
- 悪意のあるメンバの混入



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

提案方式

アウトライン

- 2種類の乱数を使用
  - ユーザのみが使用する乱数RN1
  - 管理サーバが配布する乱数RN2
- 2種類の乱数からグループ鍵GKを生成
  - GKを所有しているメンバのみによる相互通信
  - EFFの評価項目を満たす
- 適切なタイミングでグループ鍵の更新を行う
  - 鍵管理要件を満たす

暗号アルゴリズム

- 公開鍵：RSA (鍵長1024ビット以上)
- 共通鍵：AES (鍵長128ビット以上)

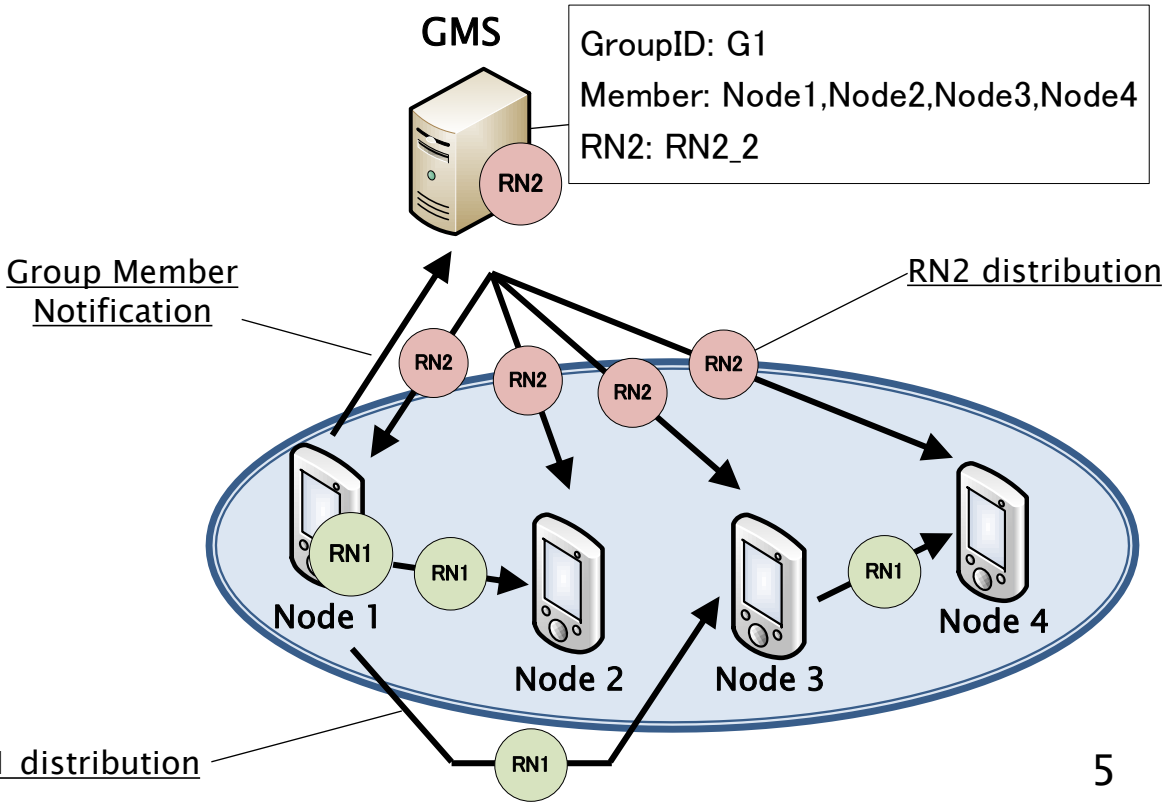
提案方式の構成

グループ管理サーバ GMS(Group Management Server)

- ユーザ情報管理
- 乱数RN2 生成・配布・更新
- 公開鍵証明書を所有

ユーザ端末 Node

- RN1 生成・共有
- GK生成
- ユーザ勧誘



提案方式 –乱数RN1共有方式–

(1) 公開鍵証明書方式

公開鍵証明書をユーザ端末に所有させる方式

○確実な認証を行うことができる

×公開鍵証明書の取得費用や管理コストがかかる

(2) エンドツーエンド通信方式

- セキュアなエンドツーエンド経路を用いた方式

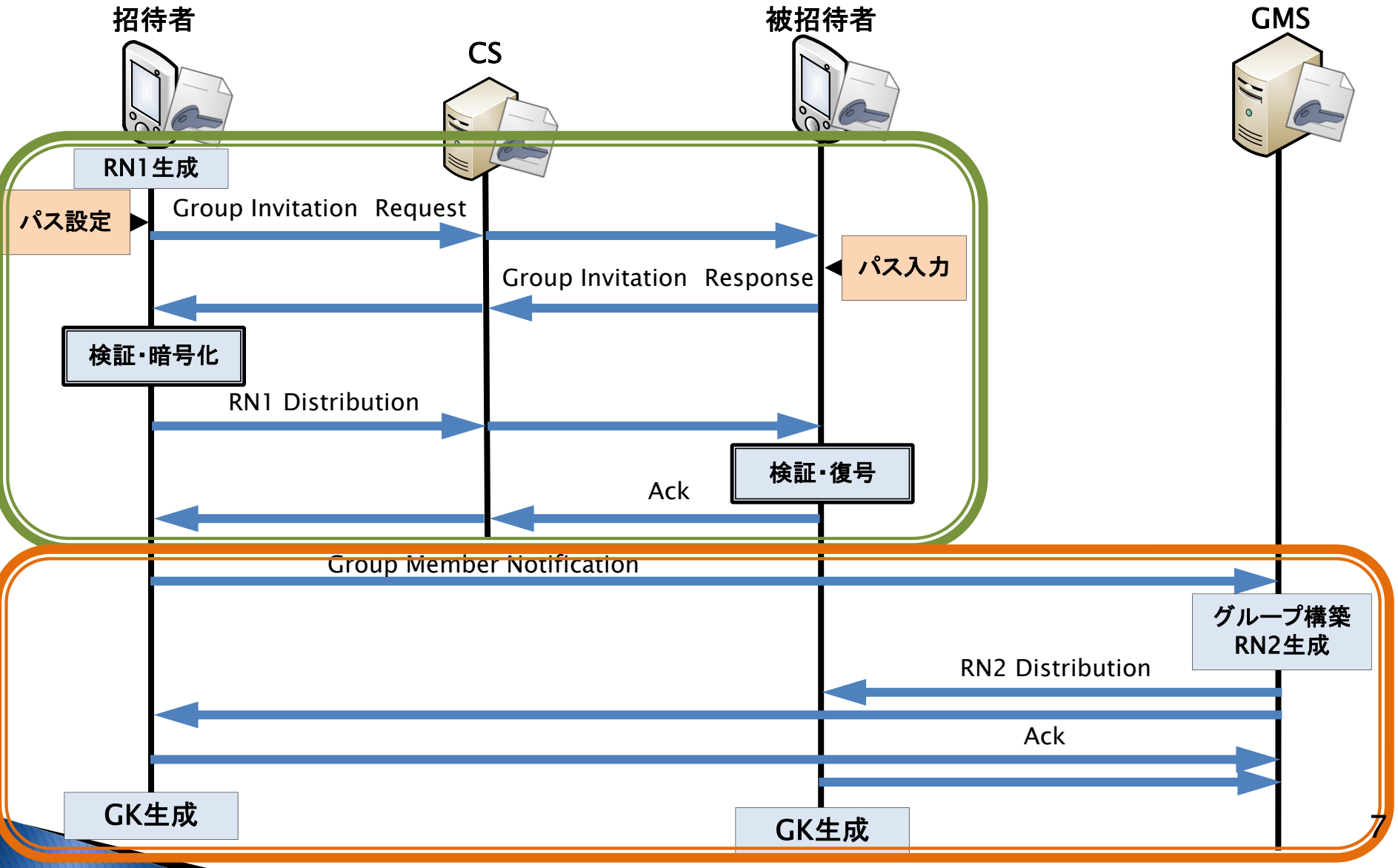
○公開鍵証明書の取得費用や管理コストがかからない

△エンドツーエンド通信が可能なセキュアネットワークが必要

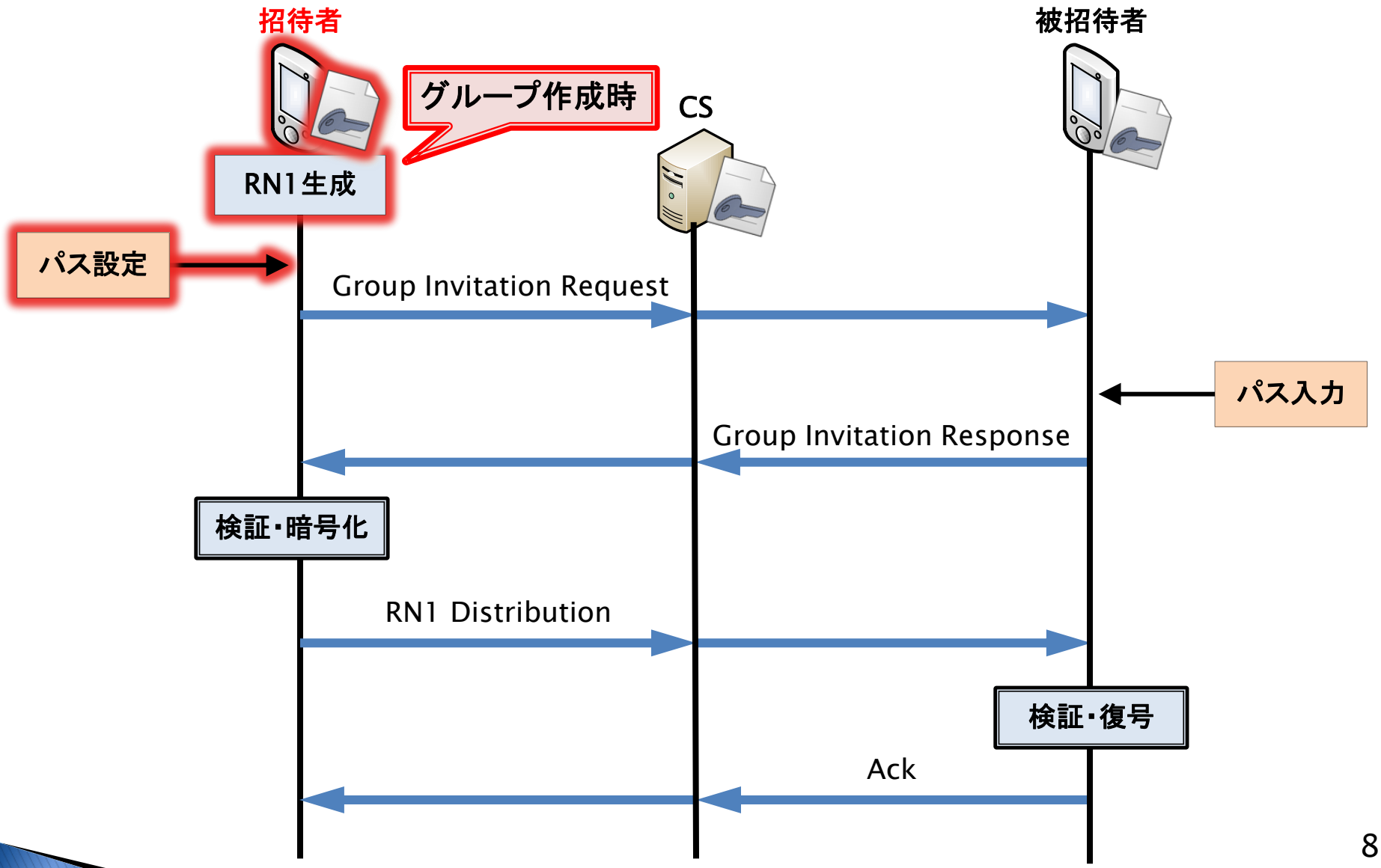
例：NTMobile

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MBL研究会，Vol.2015-MBL-77，No.20，pp.1－8，2015年12月．

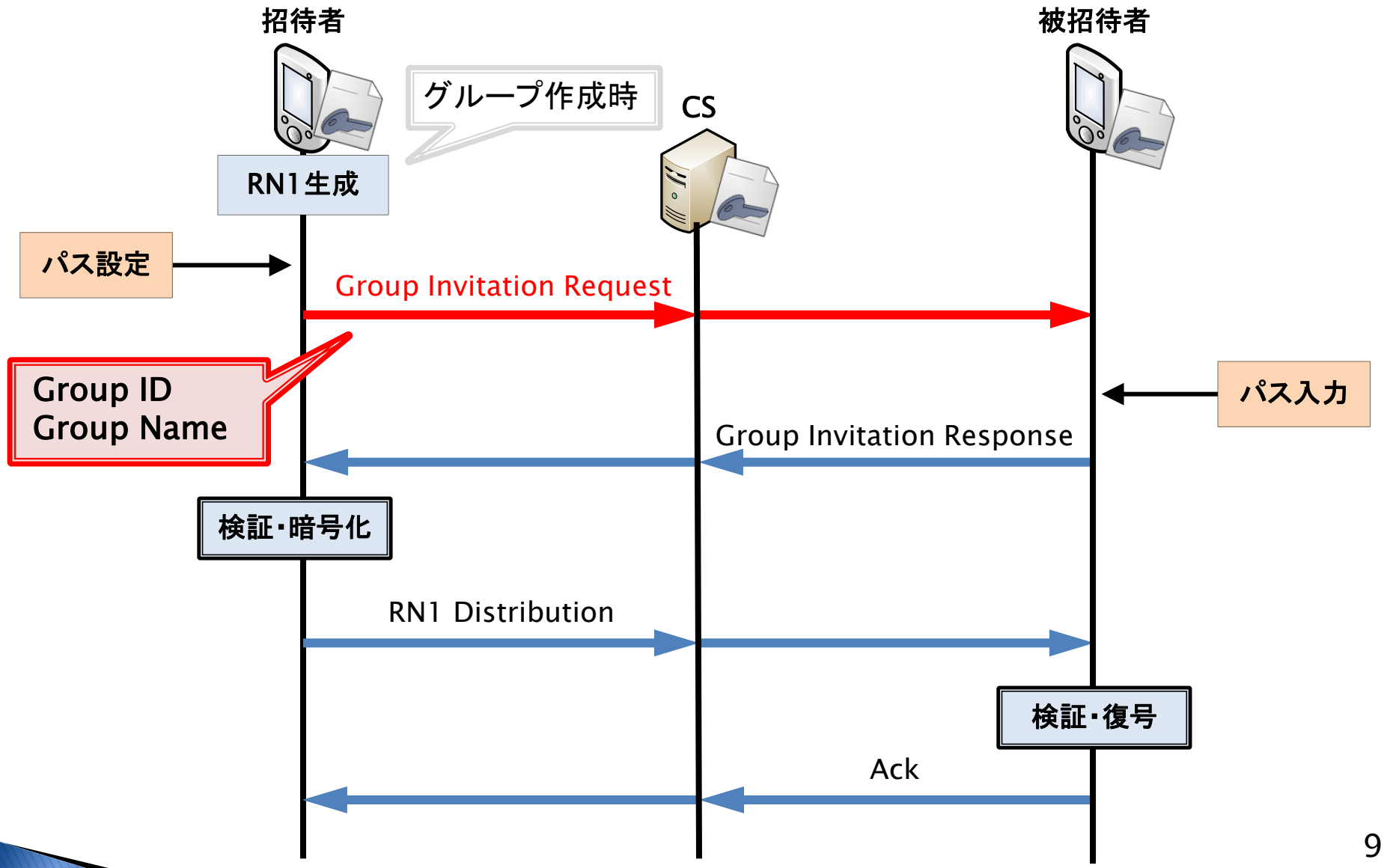
提案方式 –グループ鍵共有シーケンス–



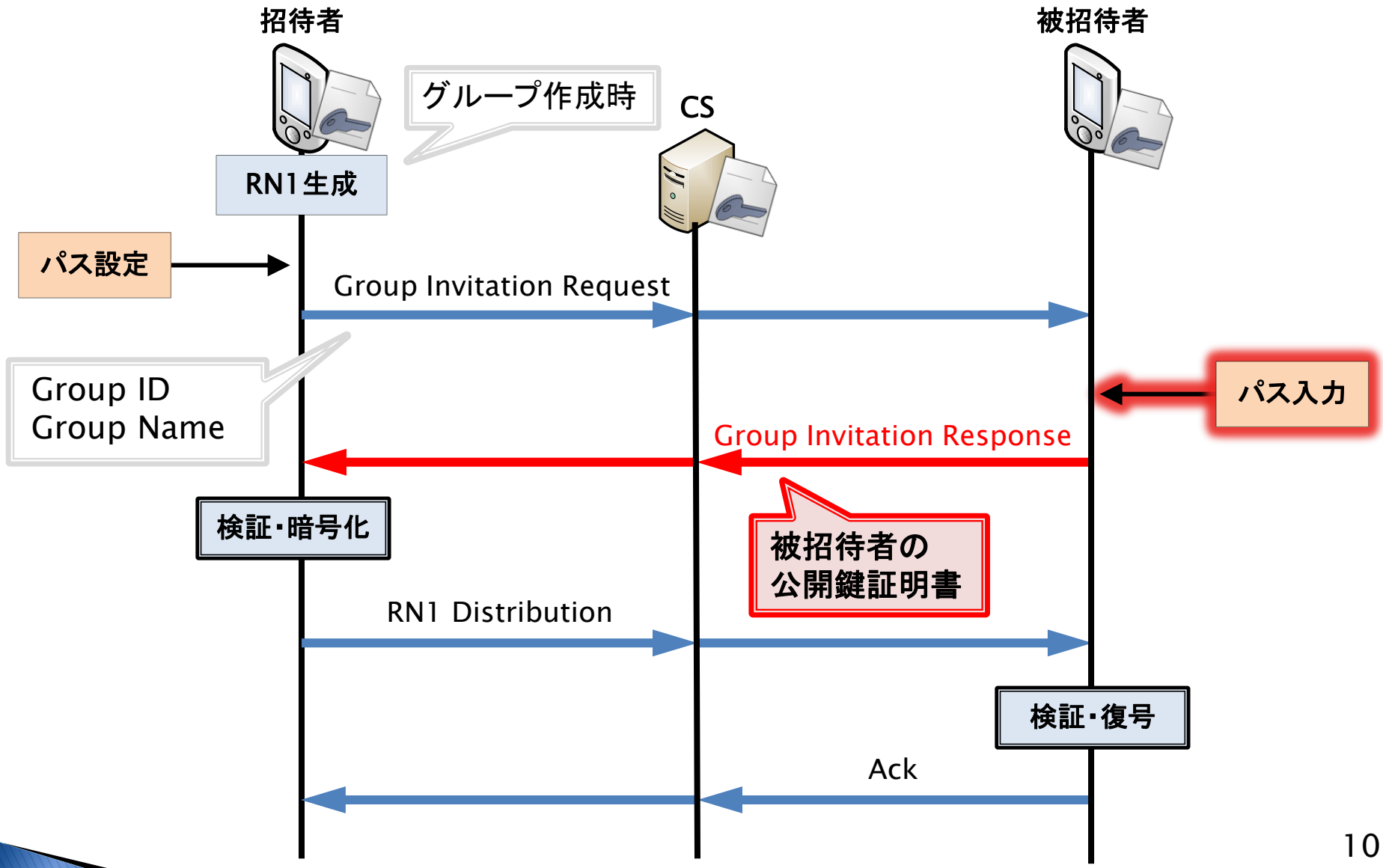
提案方式 -RN1の共有-


招待者
被招待者
グループ作成時
CS
RN1生成
パス設定
Group Invitation Request
パス入力
Group Invitation Response
検証・暗号化
RN1 Distribution
検証・復号
Ack

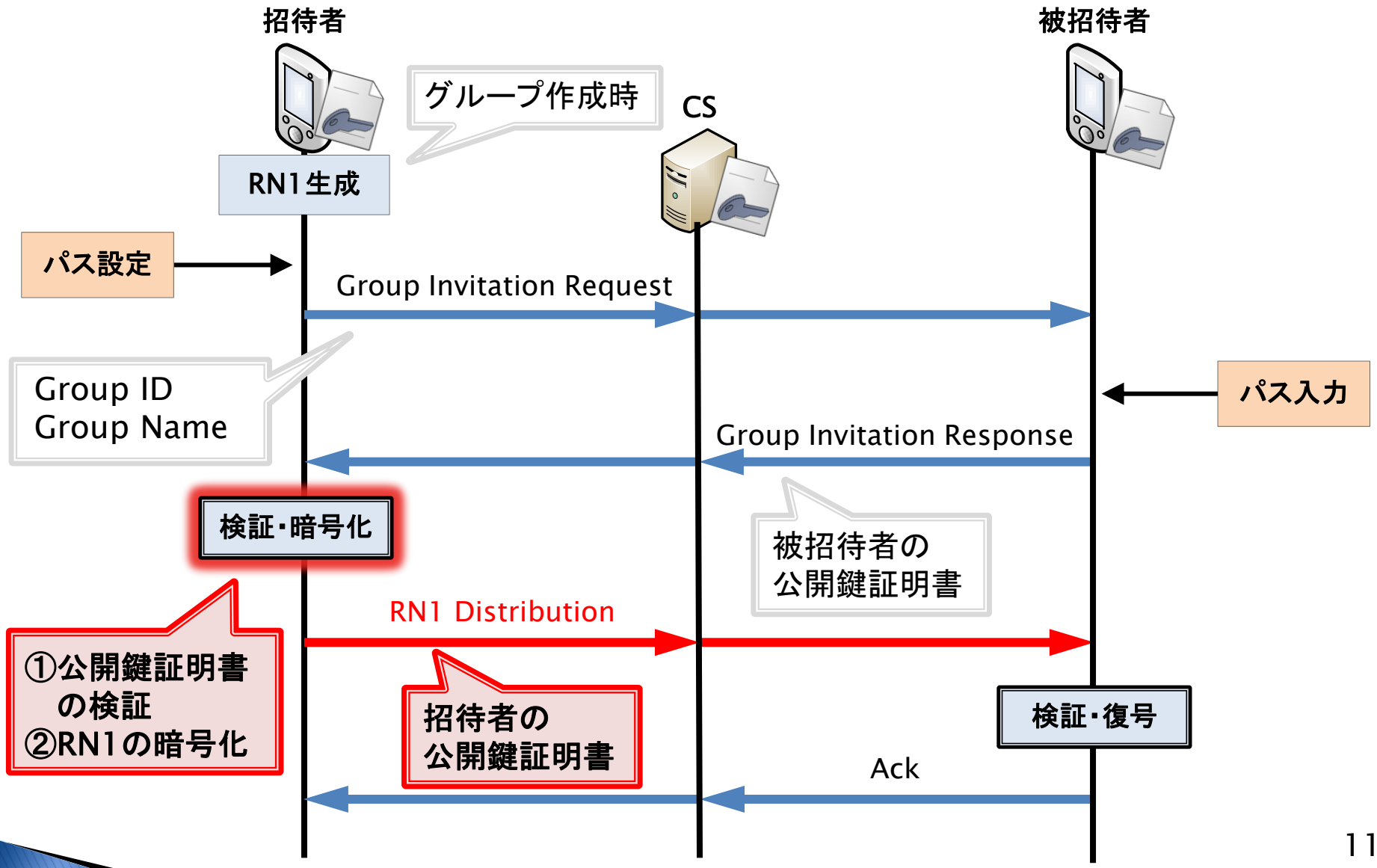
提案方式 -RN1の共有-


招待者
被招待者
グループ作成時
CS
RN1生成
パス設定
Group Invitation Request
Group ID
Group Name
パス入力
Group Invitation Response
検証・暗号化
RN1 Distribution
検証・復号
Ack

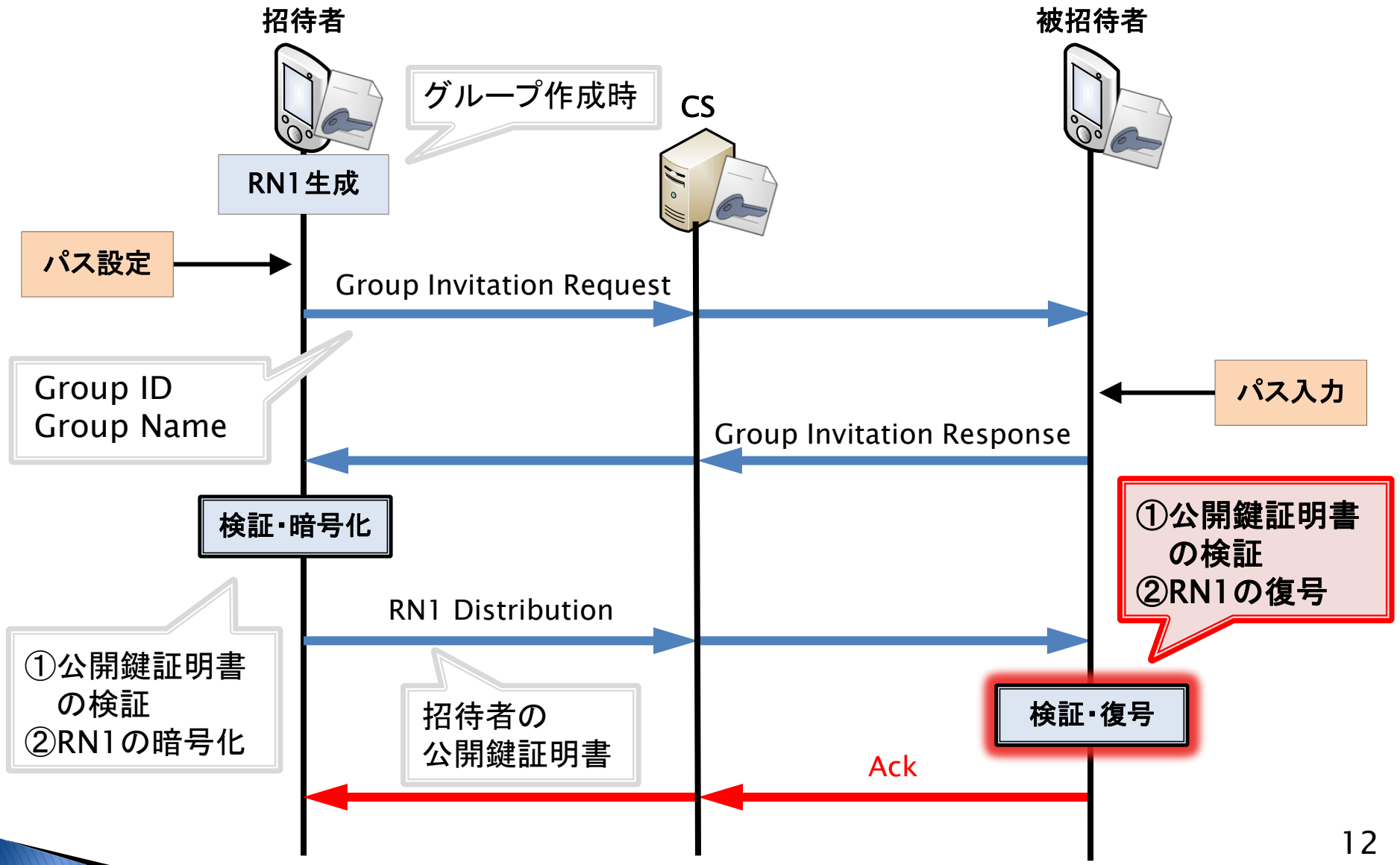
提案方式 -RN1の共有-


招待者
被招待者
グループ作成時
CS
RN1生成
パス設定
Group Invitation Request
Group ID
Group Name
パス入力
Group Invitation Response
検証・暗号化
被招待者の
公開鍵証明書
RN1 Distribution
検証・復号
Ack

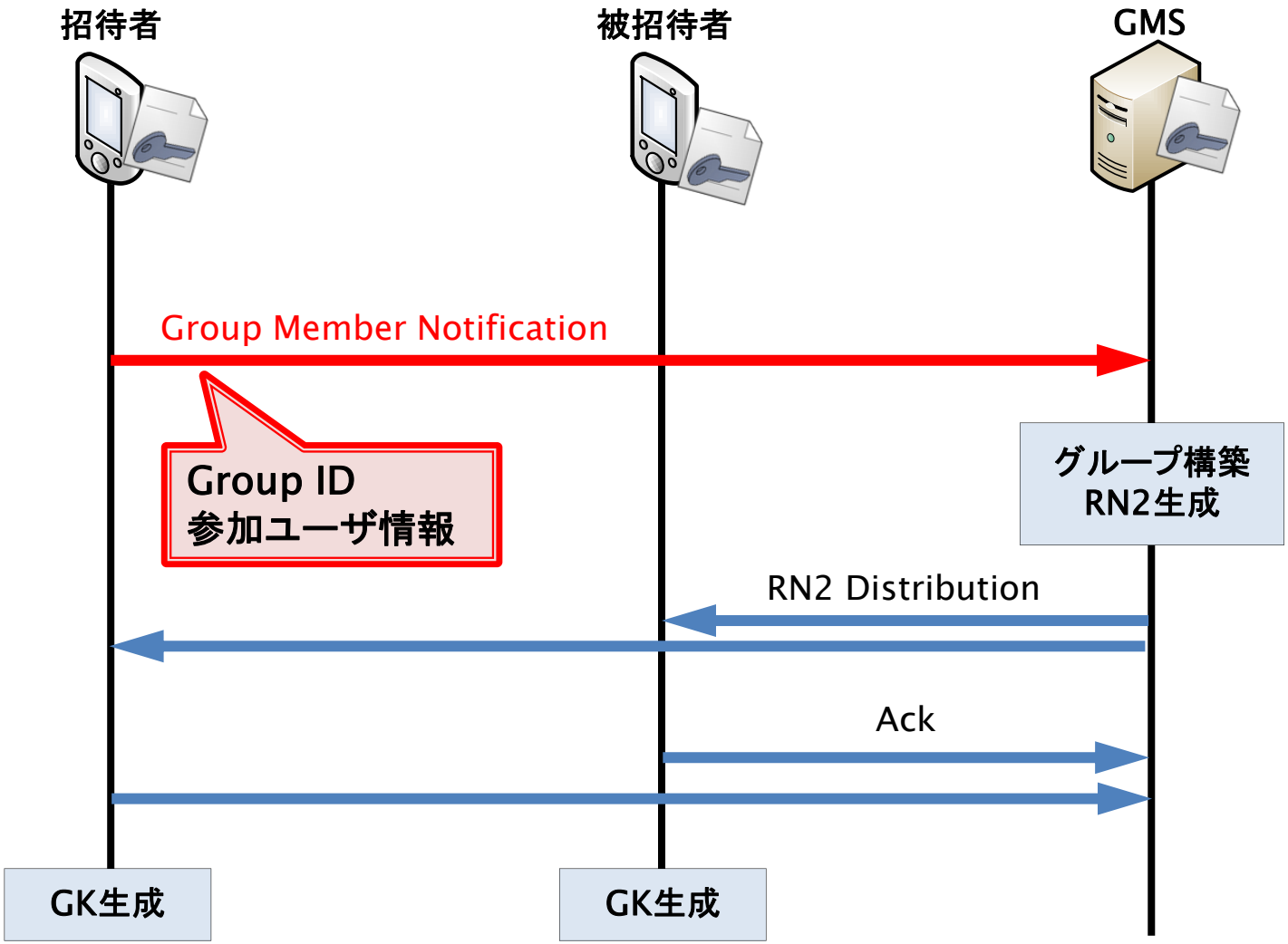
提案方式 -RN1の共有-


招待者
被招待者
グループ作成時
CS
RN1生成
パス設定
Group Invitation Request
Group ID
Group Name
パス入力
Group Invitation Response
検証・暗号化
被招待者の
公開鍵証明書
RN1 Distribution
①公開鍵証明書
の検証
②RN1の暗号化
招待者の
公開鍵証明書
検証・復号
Ack

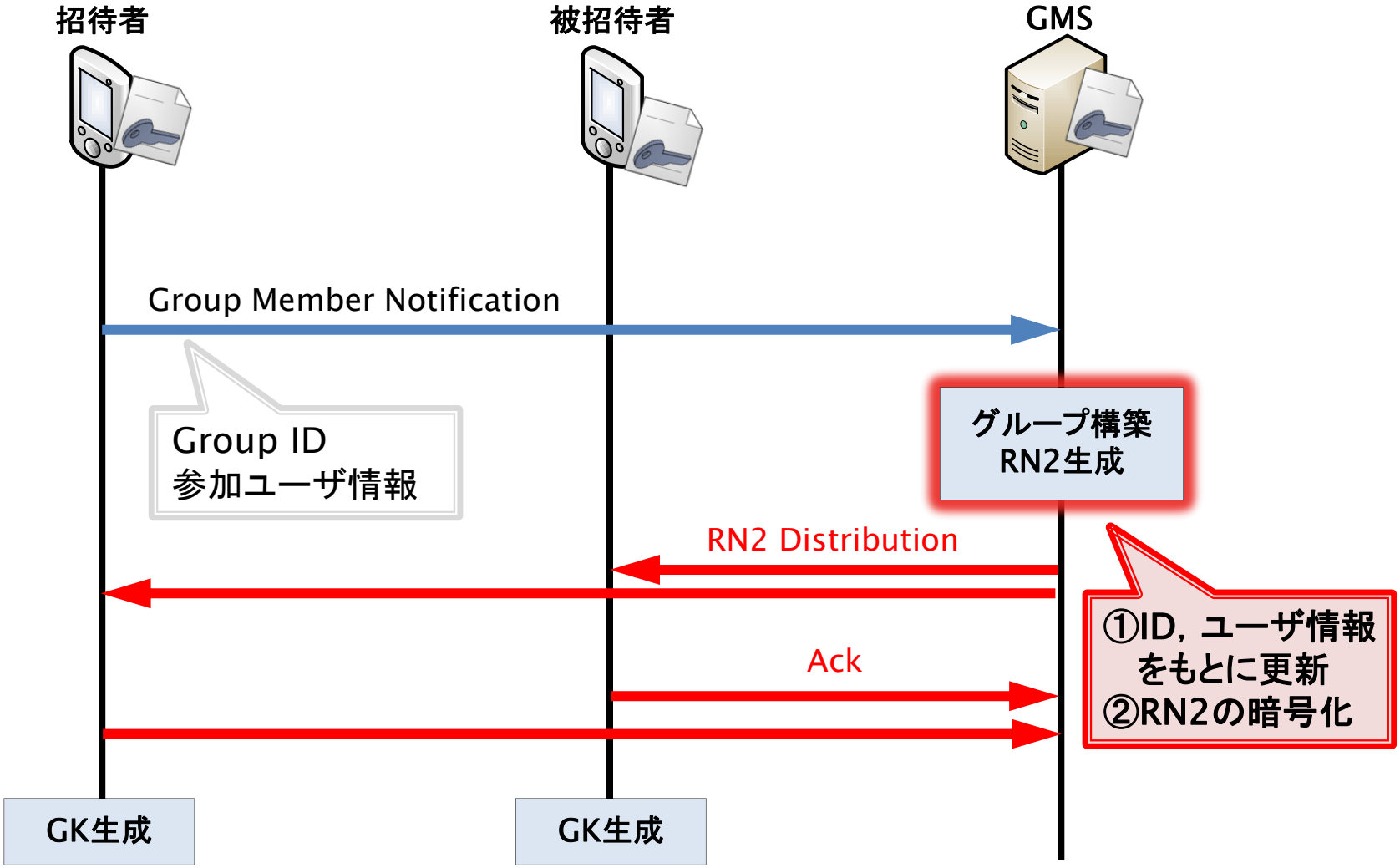
提案方式 -RN1の共有-


招待者
被招待者
グループ作成時
CS
RN1生成
パス設定
Group Invitation Request
Group ID
Group Name
パス入力
Group Invitation Response
検証・暗号化
①公開鍵証明書
の検証
②RN1の復号
RN1 Distribution
①公開鍵証明書
の検証
②RN1の暗号化
招待者の
公開鍵証明書
検証・復号
Ack

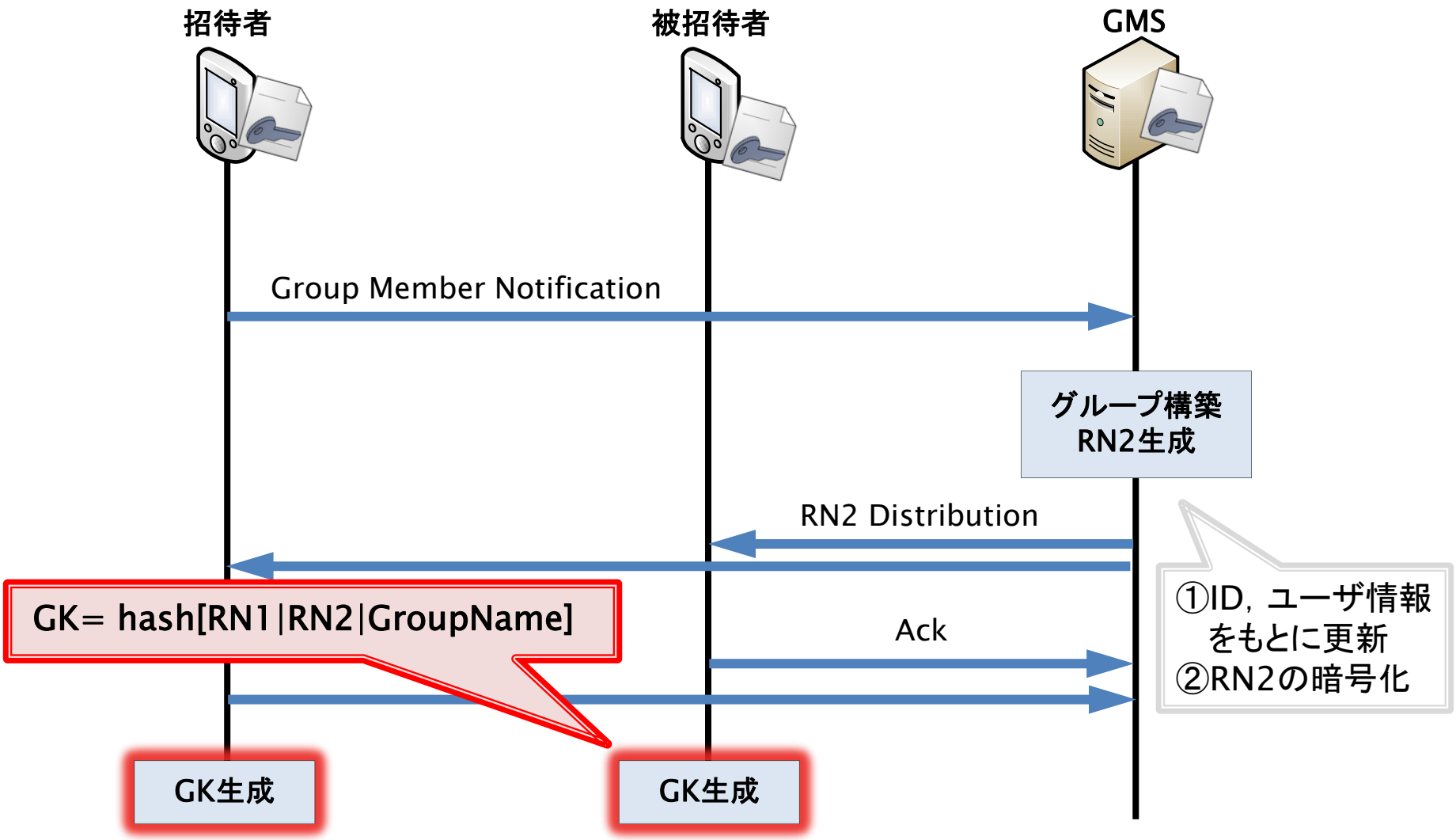
提案方式 –RN2の配布–


招待者
被招待者
GMS
Group Member Notification
Group ID
参加ユーザ情報
グループ構築
RN2生成
RN2 Distribution
Ack
GK生成
GK生成

提案方式 –RN2の配布–


招待者
被招待者
GMS
Group Member Notification
Group ID
参加ユーザ情報
グループ構築
RN2生成
RN2 Distribution
①ID，ユーザ情報
をもとに更新
②RN2の暗号化
Ack
GK生成
GK生成

提案方式 -GKの生成-


招待者
被招待者
GMS
Group Member Notification
グループ構築
RN2生成
RN2 Distribution
GK= hash[RN1|RN2|GroupName]
Ack
①ID，ユーザ情報
をもとに更新
②RN2の暗号化
GK生成
GK生成

性能評価における端末仕様

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

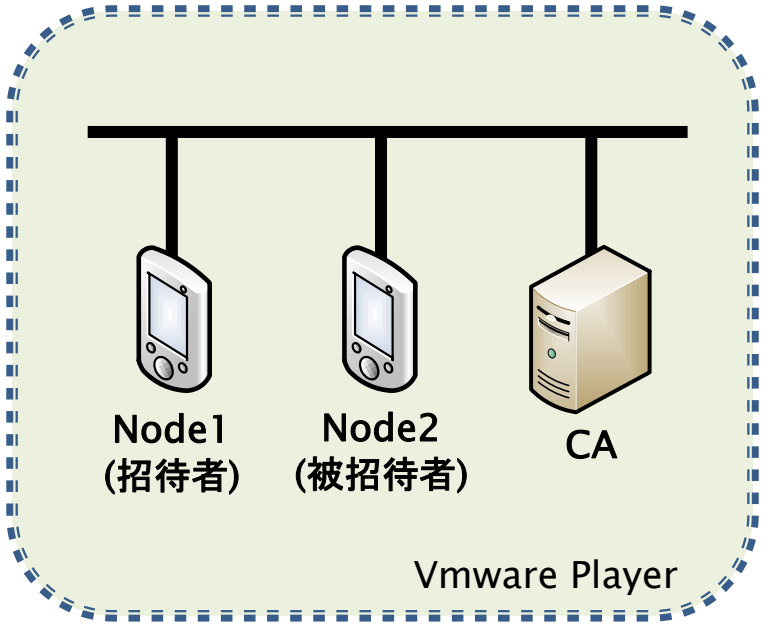
仮想マシン	Node1, Node2, CA
OS	Ubuntu 14.04
Kernel Version	3.13.0-24-generic
Memory	各1GB

Node1：招待者

Node2：被招待者

認証局CA：ルートCA

各端末へ証明書発行

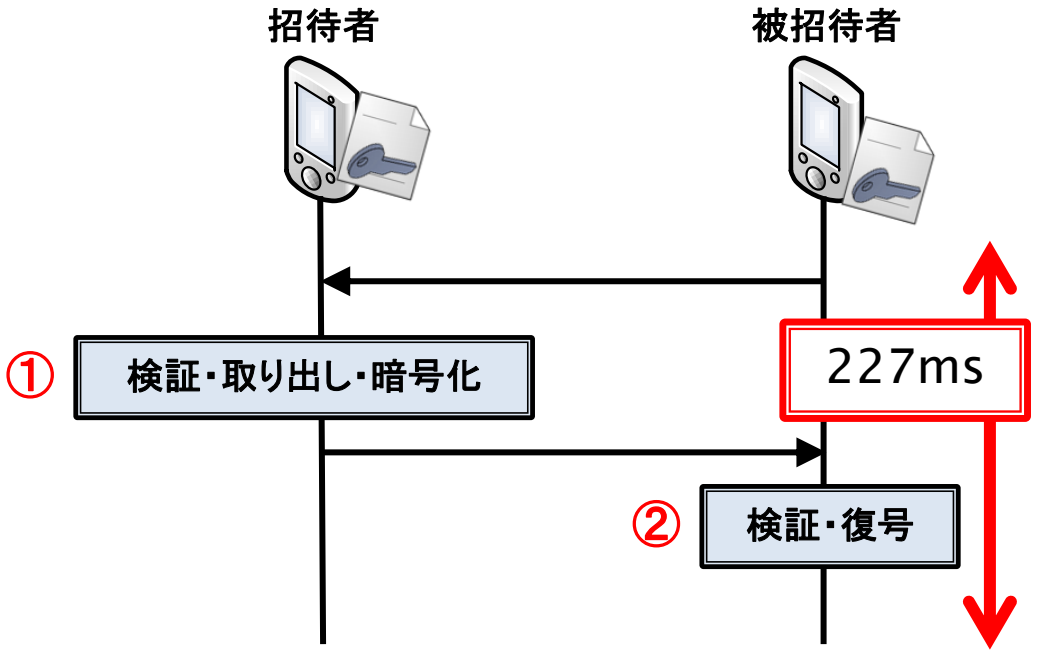


各処理における性能評価

各試行回数10回の平均値

証明書検証と暗号化/復号
- 多くの時間を要する

処理は招待時のみ
- 実用上問題がない



招待者		[ms]	[ms]
①	検証	46	139
	取出し	45	
	暗号化	48	
送信処理		0.4	

被招待者		[ms]	[ms]
②	検証	46	86
	復号	40	
送信処理		0.4	

関連技術との比較

(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.

	(1)	(2)
GSAKMP	×	○
TextSecure	○	×
提案方式	○	○

結論

2種類の乱数からグループ鍵を共有するグループ通信方式を提案

- 端末間におけるセキュアな乱数共有
  - サーバ管理者にも情報が漏洩しないグループ通信が可能
- グループ鍵更新処理
  - 鍵管理要件を満たす

提案方式の一部を実装，計測

- 実用上問題がない時間で実行可能

研究業績

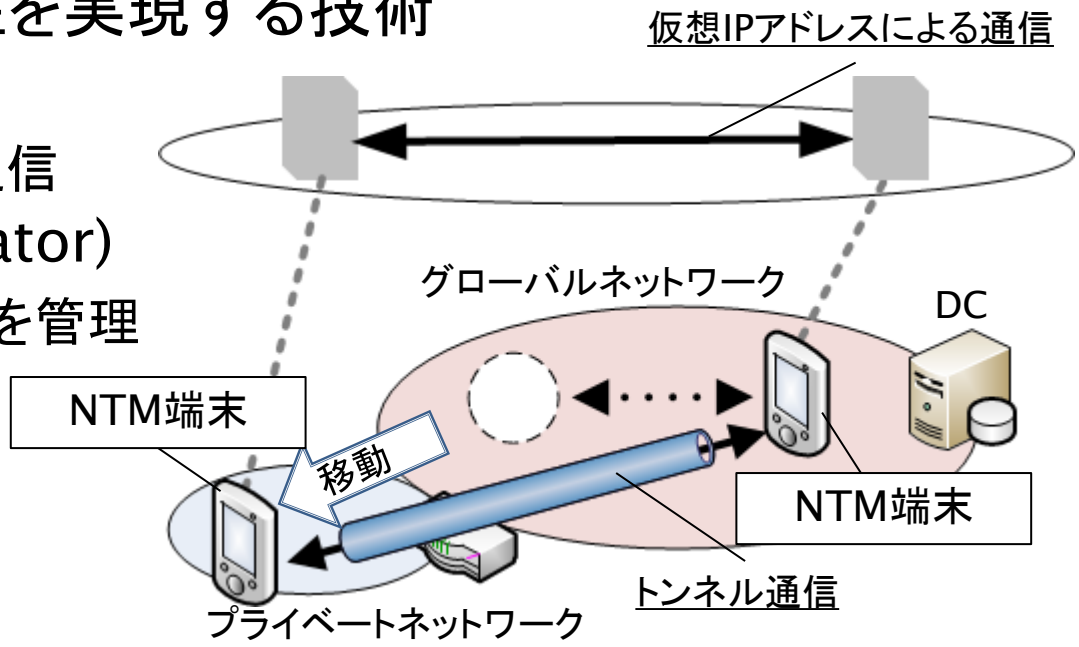
国際会議：ICMU2016 オーラルセッション

国内発表：DICOMO2016など5件

補足資料

NTMobile (Network Traversal with Mobility)

- 移動透過性と通信接続性を実現する技術
  - NTM端末
    - 仮想IPアドレスを用いた通信
  - DC(Direction Coordinator)
    - NTM端末のアドレス情報を管理
    - 経路指示を行う

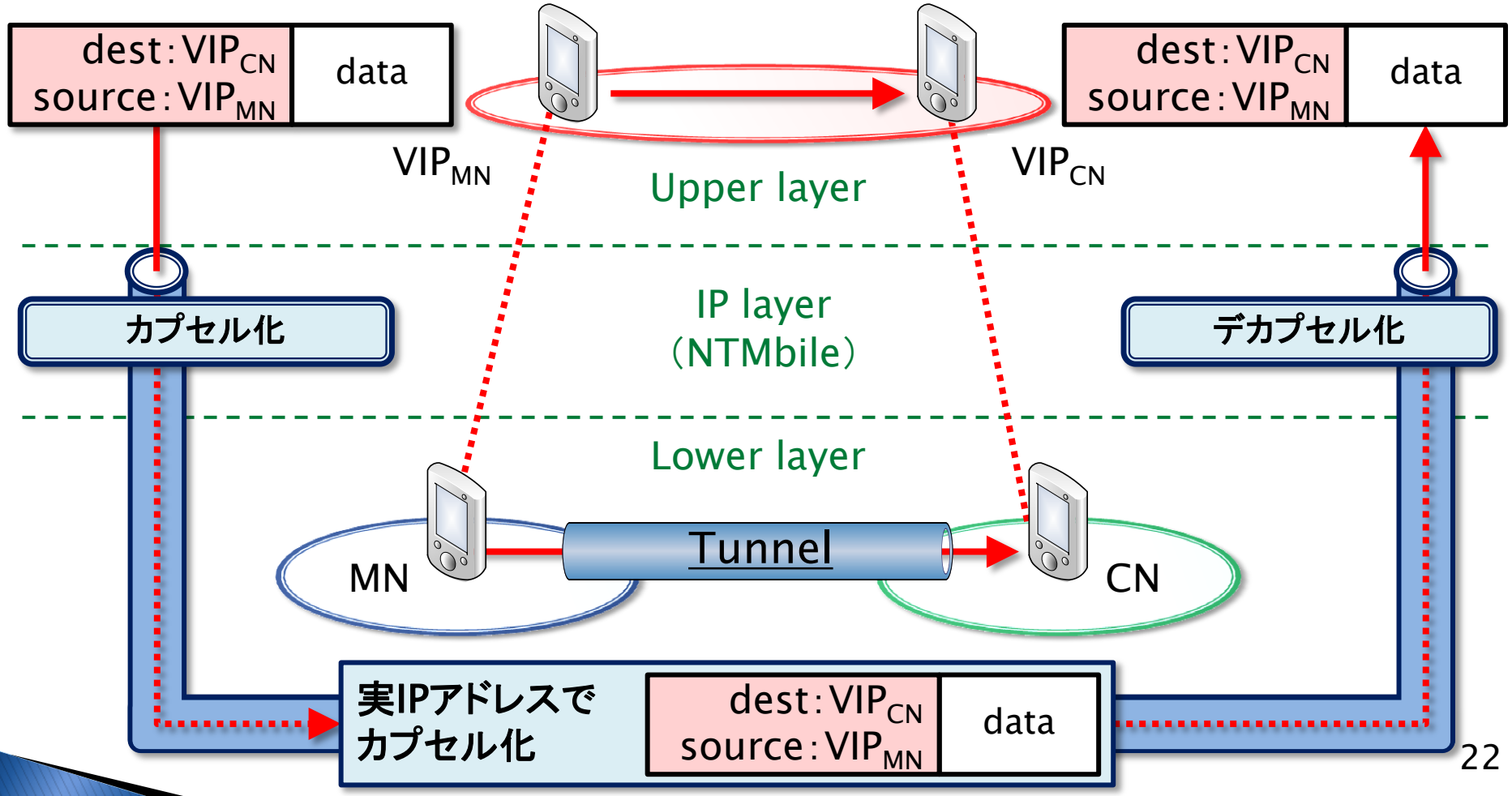


- トンネルによるエンドツーエンド通信
  - サーバや他端末が仲介することがないためRN1の共有が可能

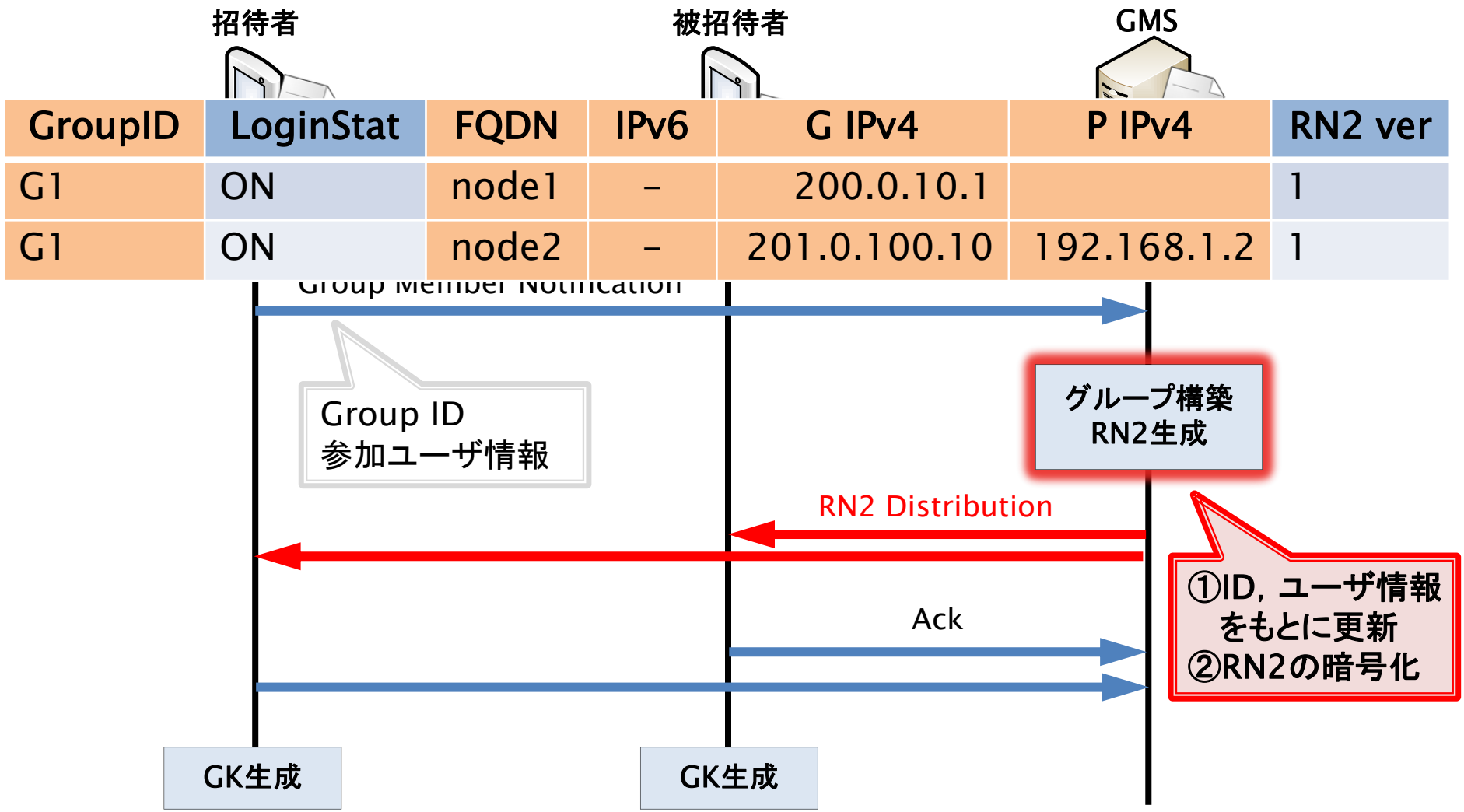
[1] 納堂博史，杉原史人，鈴木秀和，内藤克浩，渡邊晃:
NTMobileの実用化に向けた統合的枠組の検討,
情報処理学会研究報告MBL研究会, Vol.2015-MBL-77, No.20, pp.1－8, 2015年12月.

NTMobileによるカプセル化通信

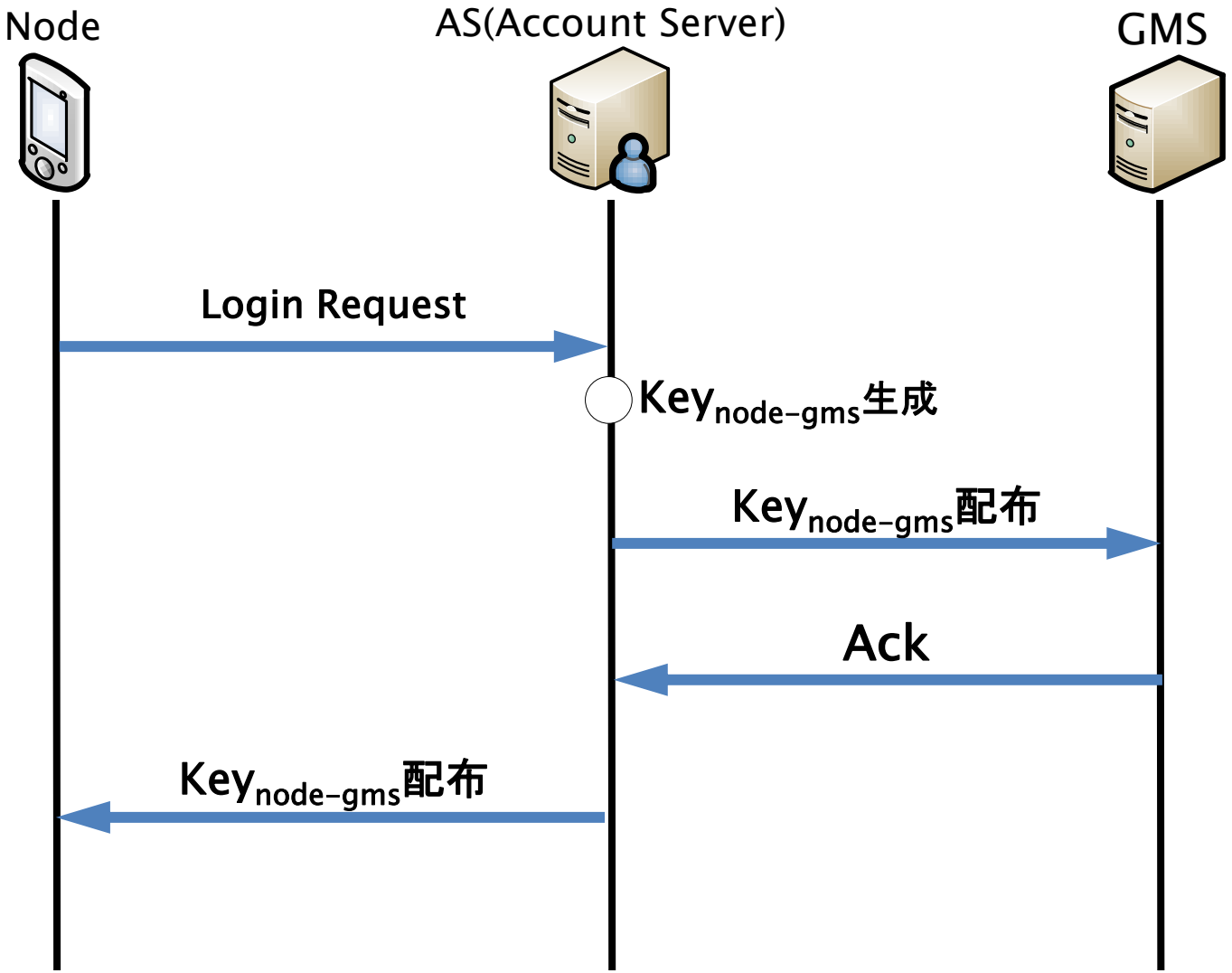
- 仮想IPアドレスを使用



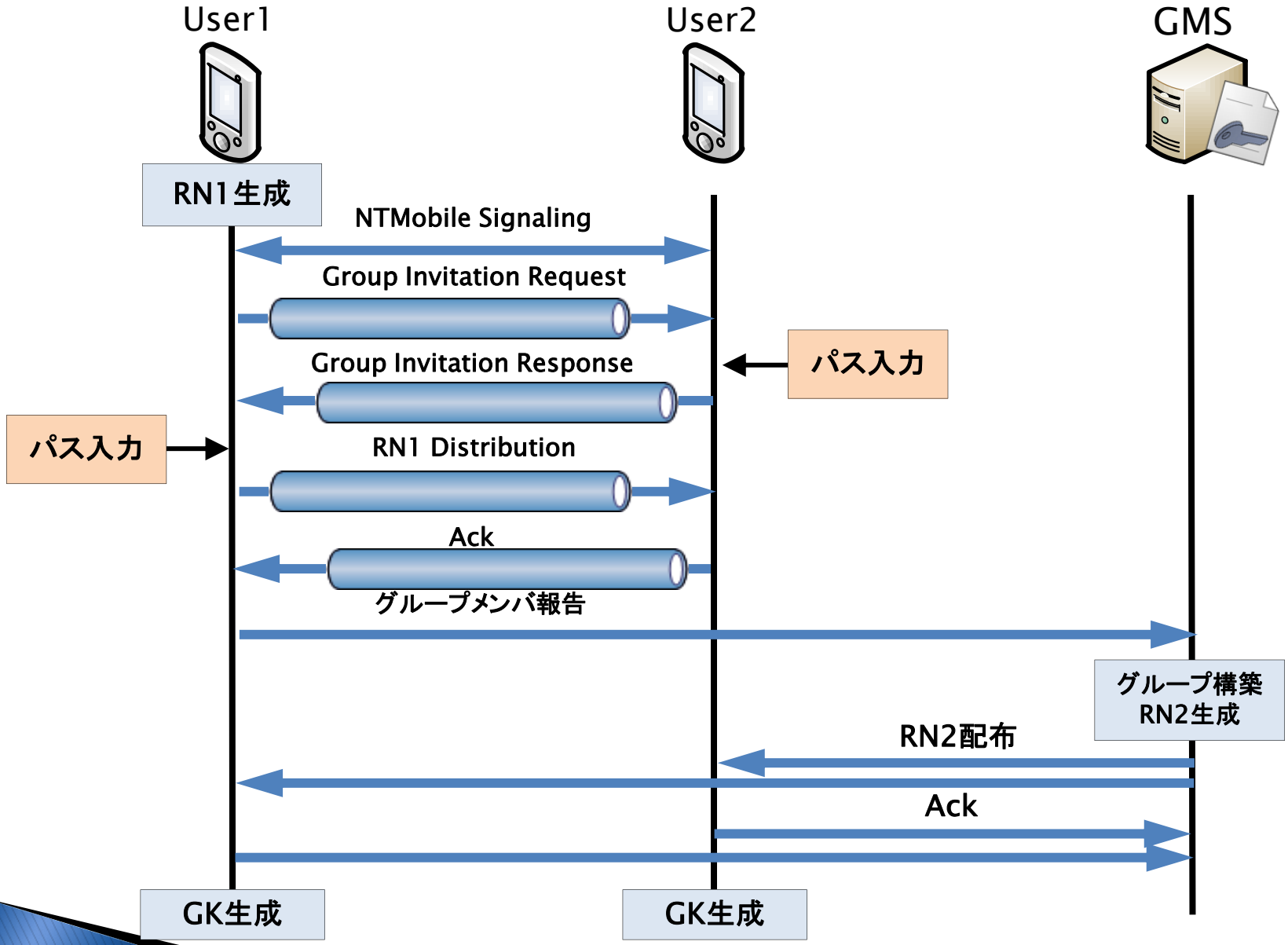
提案方式 –RN2の配布・GKの生成–



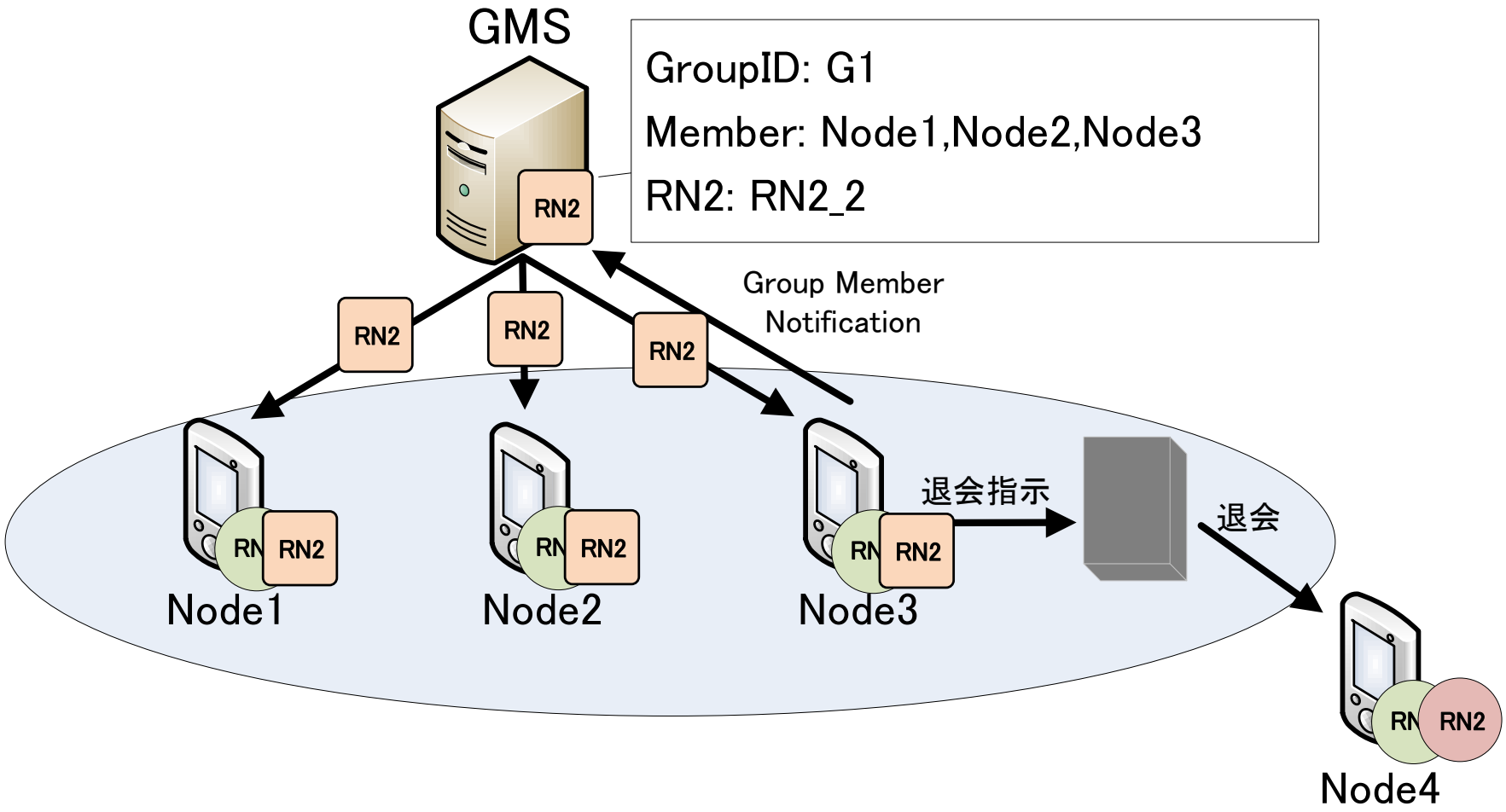
ユーザ端末-GMS間共通鍵共有



提案方式–エンドツーエンド方式–



グループメンバ変更処理



EFFによる Secure Messaging Scorecard

1. トランジットで暗号化がされているか?
2. プロバイダーが読めないように暗号化されているか?
3. 通信相手の確認が可能か?
4. 鍵が盗まれても過去の通信内容が安全か?
5. コードが公開されていて, 個別の評価が可能か?
6. セキュリティの方針は適切に文書化されているか?
7. コードは監査を受けているか?