グループ鍵を利用した相手認証方式の提案

150441022　右京 康也
渡邊研究室

1. はじめに

企業ネットワークにおいて、業務ごとに通信グループを構築することは、セキュリティを高める上で有効な手段である。IPsec を用いれば前述のような通信グループを構築することができる。しかし IPsec を利用するには端末ごとに多くの項目を設定する必要があり、実用性が低い。

そこで、本稿では事前に共有されたグループ鍵と乱数のハッシュ値を通信開始時に交換することにより、相手認証を行う通信方式を提案する。送信側はグループ鍵と乱数から計算したハッシュ値と乱数を送信する。受信側は受け取った乱数と自分が持つグループ鍵からハッシュ値を計算し、送られてきたハッシュ値と比較することで相手を認証することができる。これによりより簡易にセキュアな通信グループを構築することができる。

本研究では、移動透過性と NAT 越え通信の両者を同時に実現する NTMobile（Network Traversal with Mobility)[1] を利用して相手認証を実現するための方式を検討した。グループ鍵の共有方法は手入力またはグループ管理サーバ GMS（Group Management Server）からの配送で行う。

2. 既存技術

セキュアな通信グループを構築する通信方式として IPsec(IP security) を用いたグルーピングが考えられる。IPsec とはネットワーク層において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 では 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、相手認証、暗号化などが可能である。また通信は 1 対 1 であり、使用するにあたり必要な設定項目が多くある。全ての端末間で設定を行い、認証と暗号化を行うことによりセキュアな通信グループを構築することができる。しかし端末ごとに設定を行うことは実用上困難である。また IPsec は NAT を経由する通信では利用できず、システム構成が限定される。

NTMobile は、移動透過性と NAT 越え通信を同時に実現することができる技術である。通信を行う NTM 端末はシグナリングの過程において安全に End Key を共有する。この共通鍵を用いてエンドツーエンドでパケットの暗号化と、パケット認証を行う。しかし、相手認証機能は実装されていなかった。

3. 提案方式

3.1 従来の NTMobile シーケンス

MN(Mobile Node) は Direction Request を DC(Direction Coordinator) に送信して、経路指示を仰ぐ。Direction Request を受け取った DC は、MN と CN(Correspondent Node) に対して Route Direciton で経路を指示する。経路指示を受け取った MN と CN は DC の指示に従って Tunnel Request/Tunnel Respose を交換し、End Key の共有と通信経路の確立を行う。これにより暗号鍵を共有し、セキュアな通信を実現できる。

3.2 提案方式のシーケンス

提案方式では、事前に共有されているグループ鍵 GK を利用することにより Tunnel Request/Tunnel Respose において通信相手の認証を行う。図 1 に提案方式のシーケンスを示す。なおこのシーケンスは提案を説明するための

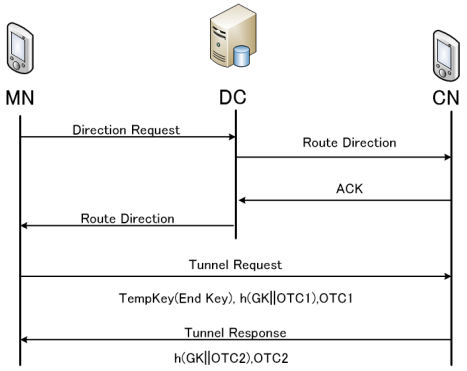


図 1: 提案方式のシーケンス図

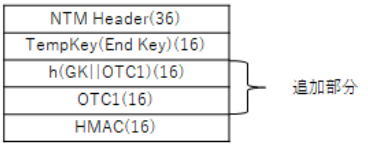


図 2: Tunnel Request のヘッダフォーマット

最も簡易なものであり、NAT は省略している。OTC(one time code) は使い捨てで十分大きな乱数であり、GK はグループ鍵、h（GK||OTC）は GK と OTC のハッシュ値である。MN、CN は同一グループのメンバであり、GK を事前に共有している。MN は、Tunnel Request において、自分が持つ GK と適当に生成した OTC1 のハッシュをとり、OTC1 とともに CN へ送信する。CN は自分が持つ GK と受信した OTC1 のハッシュをとり、受信したハッシュ値と一致するか検証する。一致すれば、Tunnel Respose において、自分が持つ GK と適当に生成した OTC2 のハッシュをとり、OTC2 とともに CN へ送信する。MN が CN から送られてきたハッシュ値の検証に成功すれば相互認証が完了する。図 2 に提案方式における Tunnel Request のヘッダフォーマットを示す。NTM Header は NTMobile 特有のヘッダー、TempKey(End Key）は End Key を DC から配布された TempKey で暗号化したもの、HMAC は認証コードである。ここに今回 h（GK||OTC1）と OTC1 を追加した。

グループ鍵の共有方法は、グループ管理サーバ GMS からの配送、もしくはパスワードの手入力等を選択できるようにする。そのためグループ鍵を格納する所定のファイルを各端末に持たせる。

4. まとめ

グループ鍵を事前に共有しているという前提で、NTMobile においてグループ認証を実現する方式を提案した。

参考文献

[1] 鈴木秀和ほか：NTMobile における通信接続性の確立手法と実装, 情報処理学会論文誌 Vol.54, No.1, pp.367-379 (2013).

グループ鍵を利用した相手認証方式の提案

渡邊研究室
150441022　　右京 康也

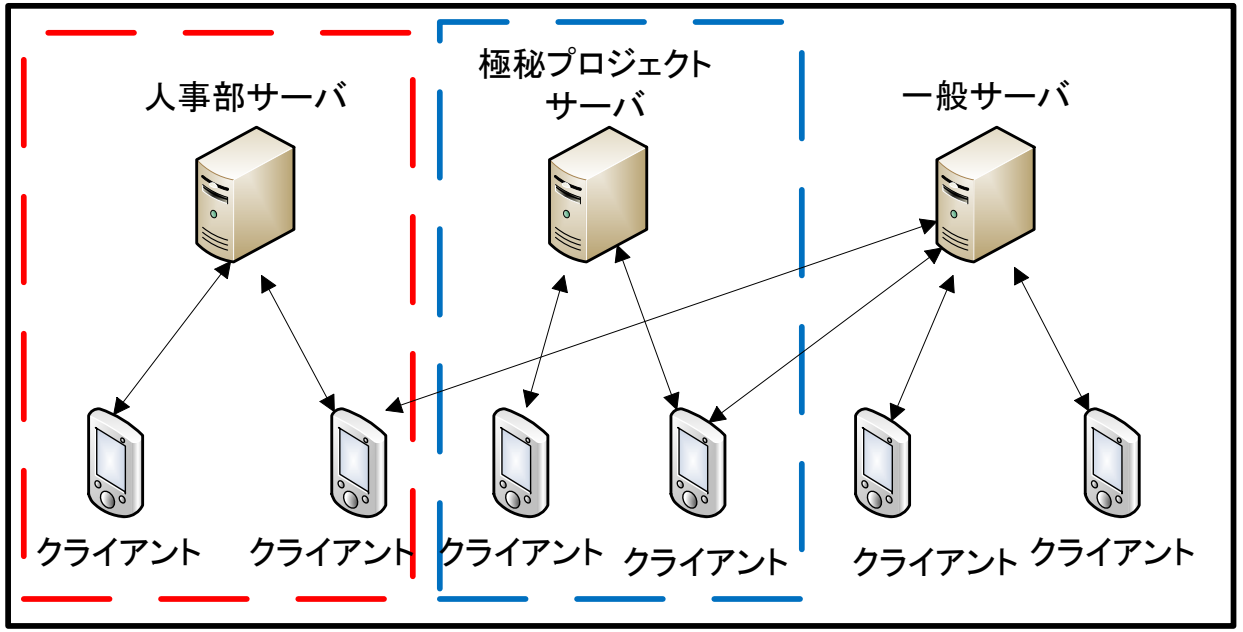
Watanabe Lab.

研究背景

- ネットワーク技術の発展
  - 企業がネットワークサービスをビジネスに利用

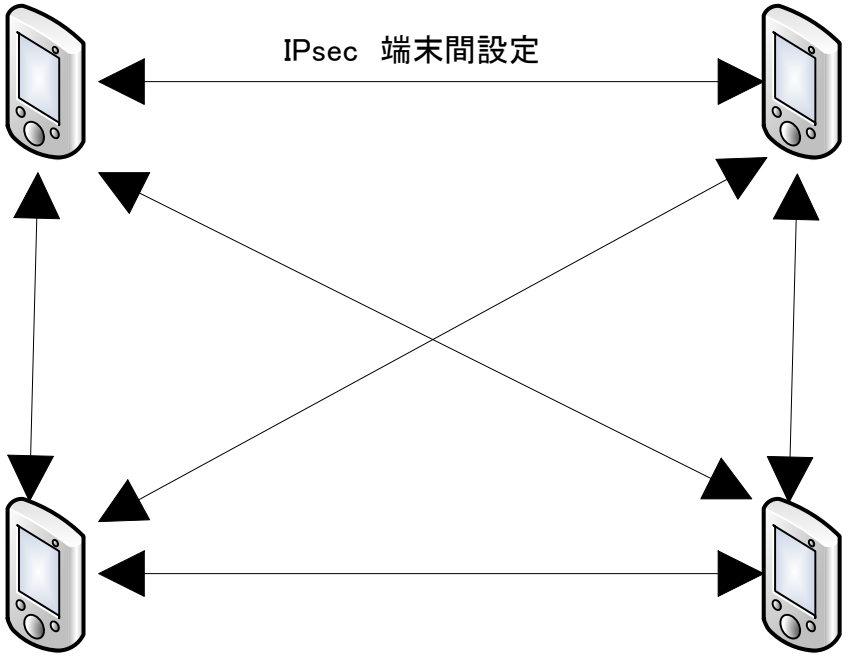
- 企業ネットワークにおける内部脅威
  - 社員による機密情報の持ち出し
    - 業務ごとにセキュアな通信グループを構築

企業ネットワーク



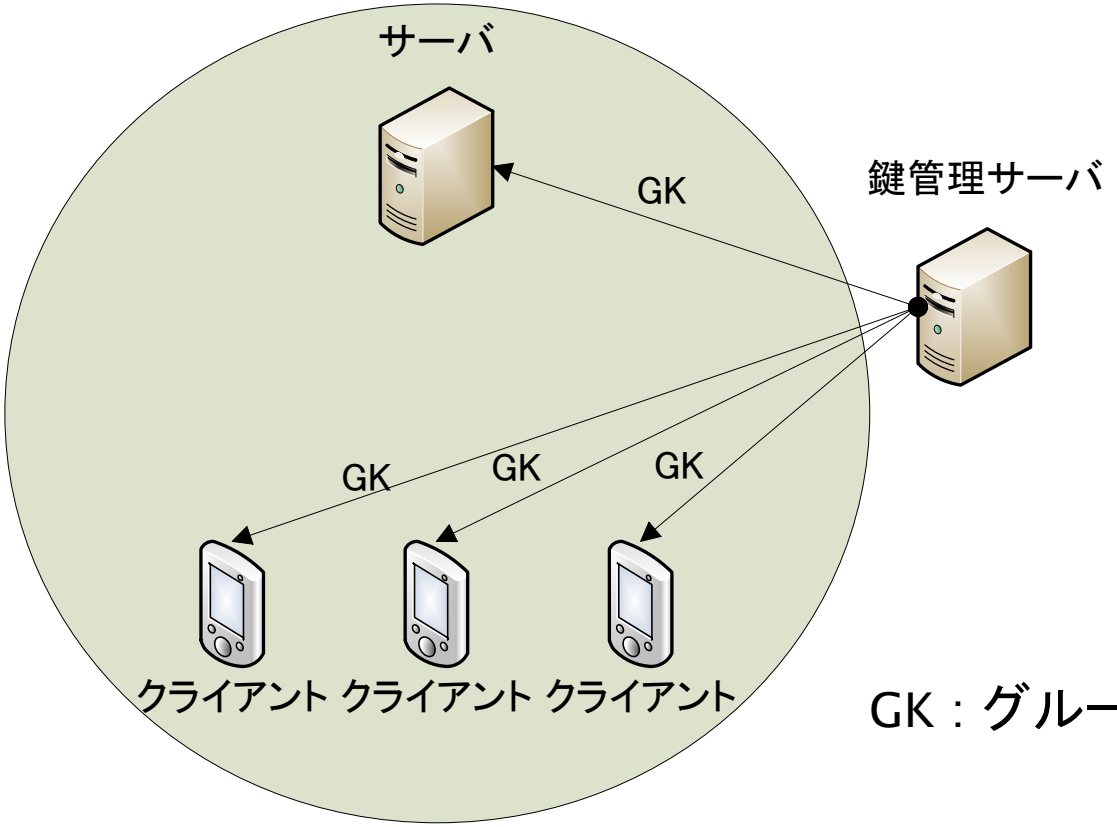
既存技術 –IPsec–

- ネットワーク層においてデータのセキュリティを保護するプロトコル
  - 相手認証
  - 暗号化
- RFC6071として標準化
- セキュアな通信グループを構築
- 課題
  - 管理負荷が膨大
  - NATを経由したネットワークで利用不可



グルーピングの手法

- グループメンバの全ての端末で同一の鍵を共有する
- グループ鍵を安全に共有する研究を行っている



GK：グループ鍵

研究の目的

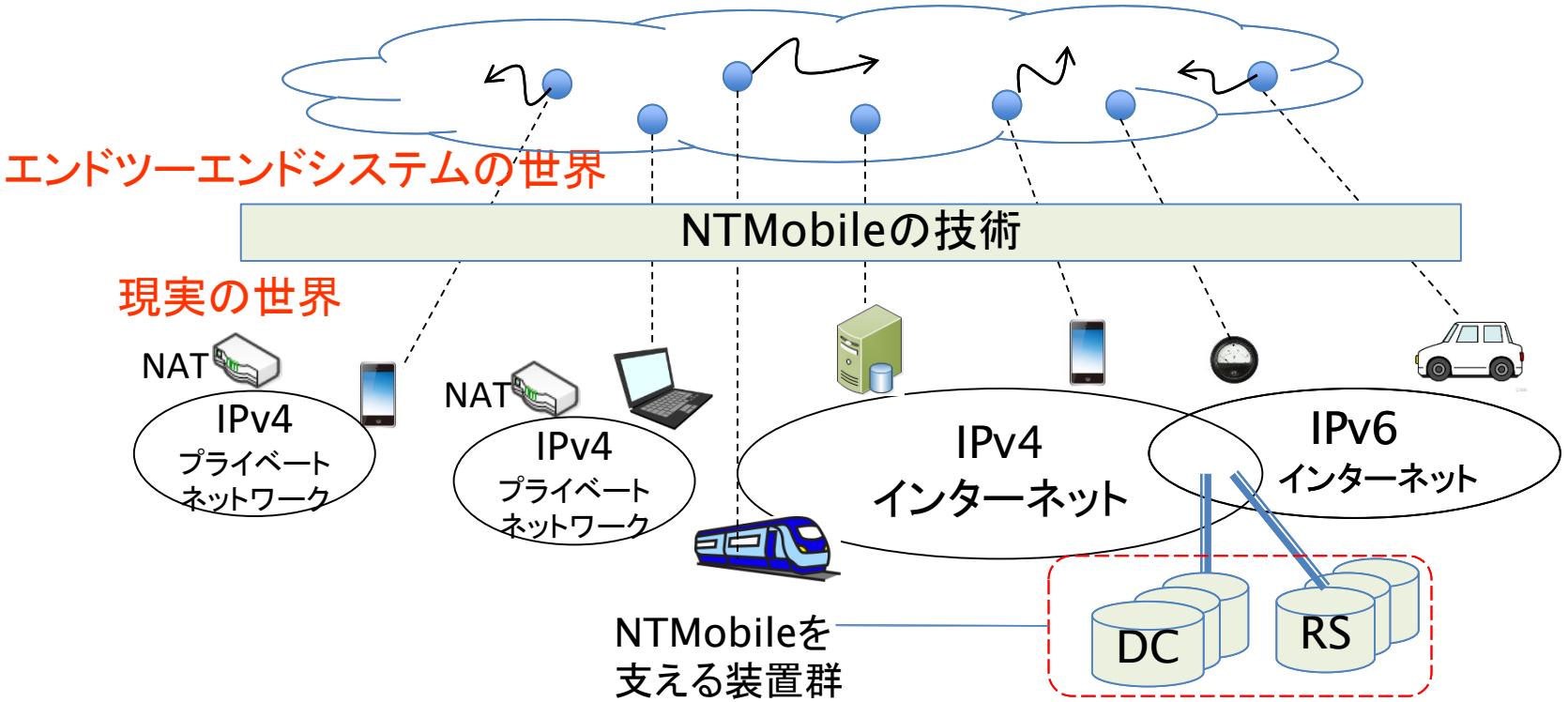
- NTMobile（Network Traversal with Mobility）は相手認証機能がない
- グループ鍵を安全に共有する研究

- グループ鍵を利用してNTMobileに相手認証機能を追加する

NTMobileの構成

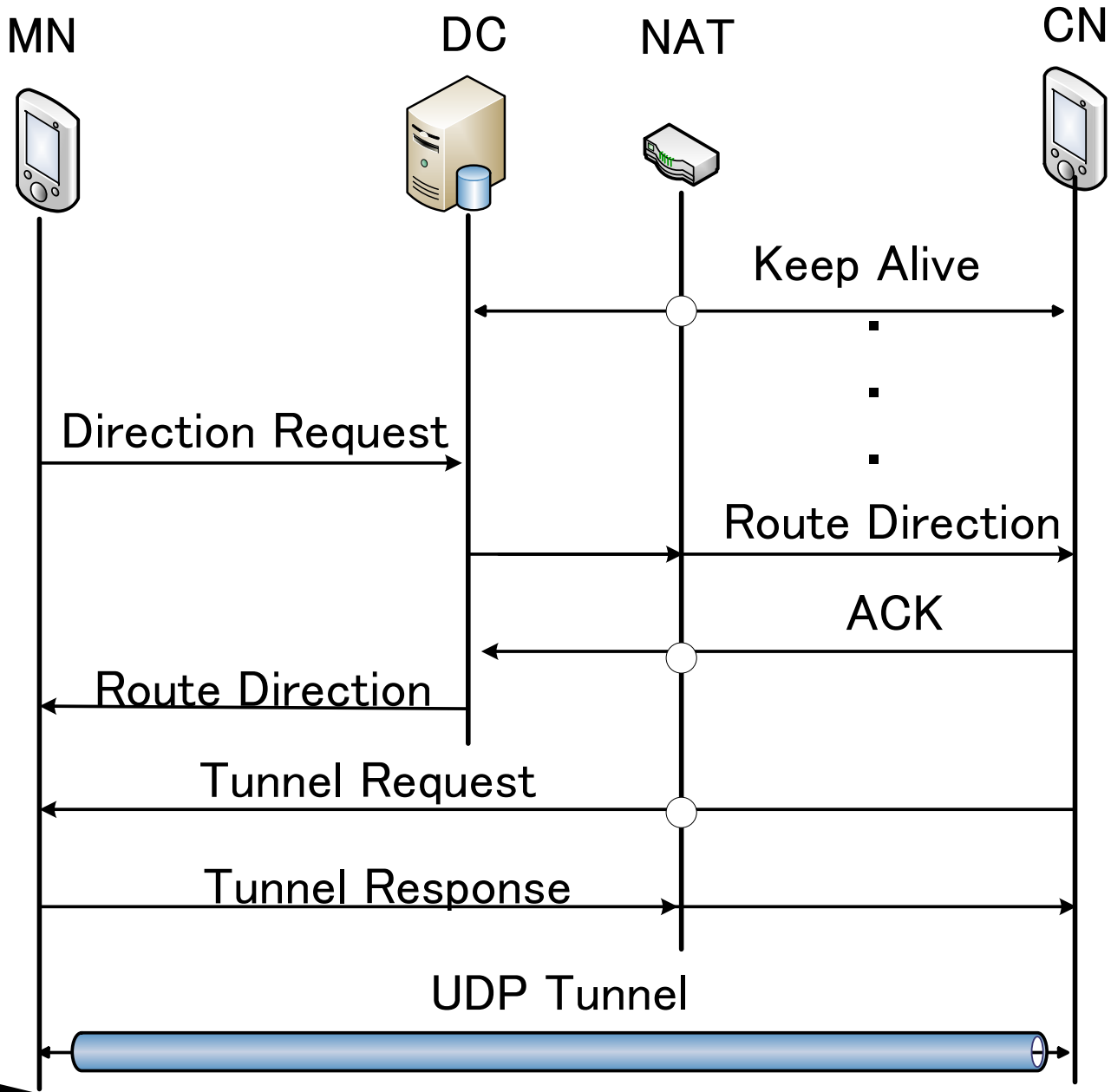
Watanabe Lab.

エンド端末にNTMobile用アプリケーションをインストールすることによりエンドツーエンドシステムの世界に移行できる
NTMobileをサポートする装置群（DC,RS）をインターネット上に配置する必要がある（ユーザは意識しなくてよい）



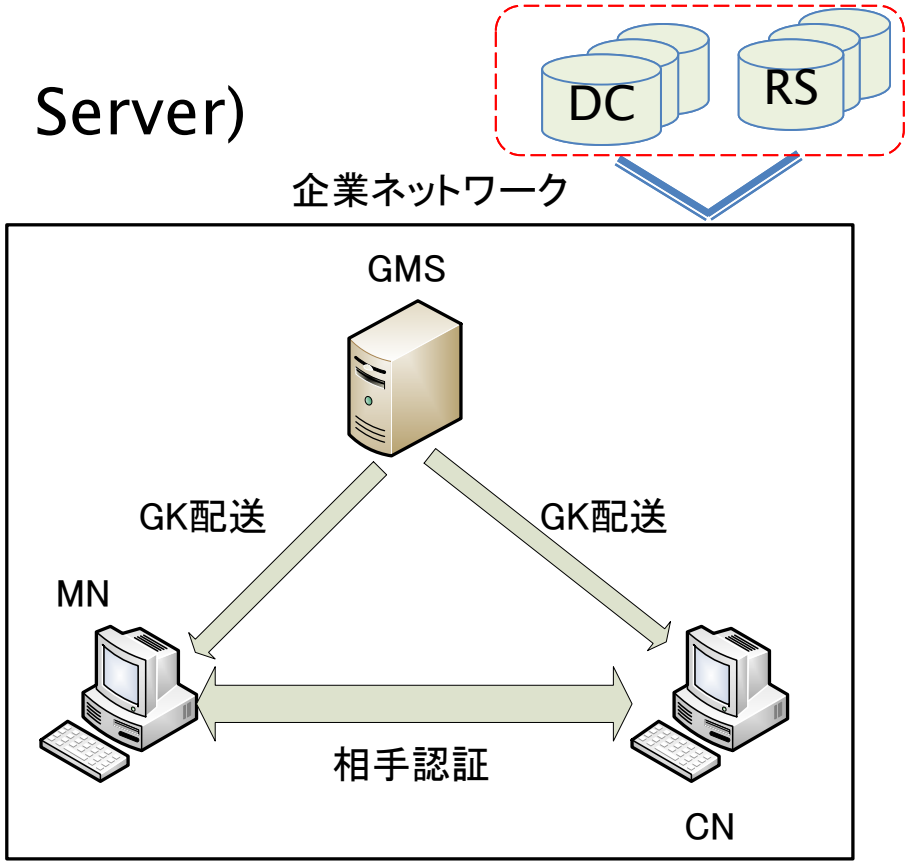
DC (Direction Coordinator)：仮想アドレスの配布と通信経路を指示する装置
RS (Relay Server)：必要に応じてパケットを中継する装置

NTMobileのシーケンス

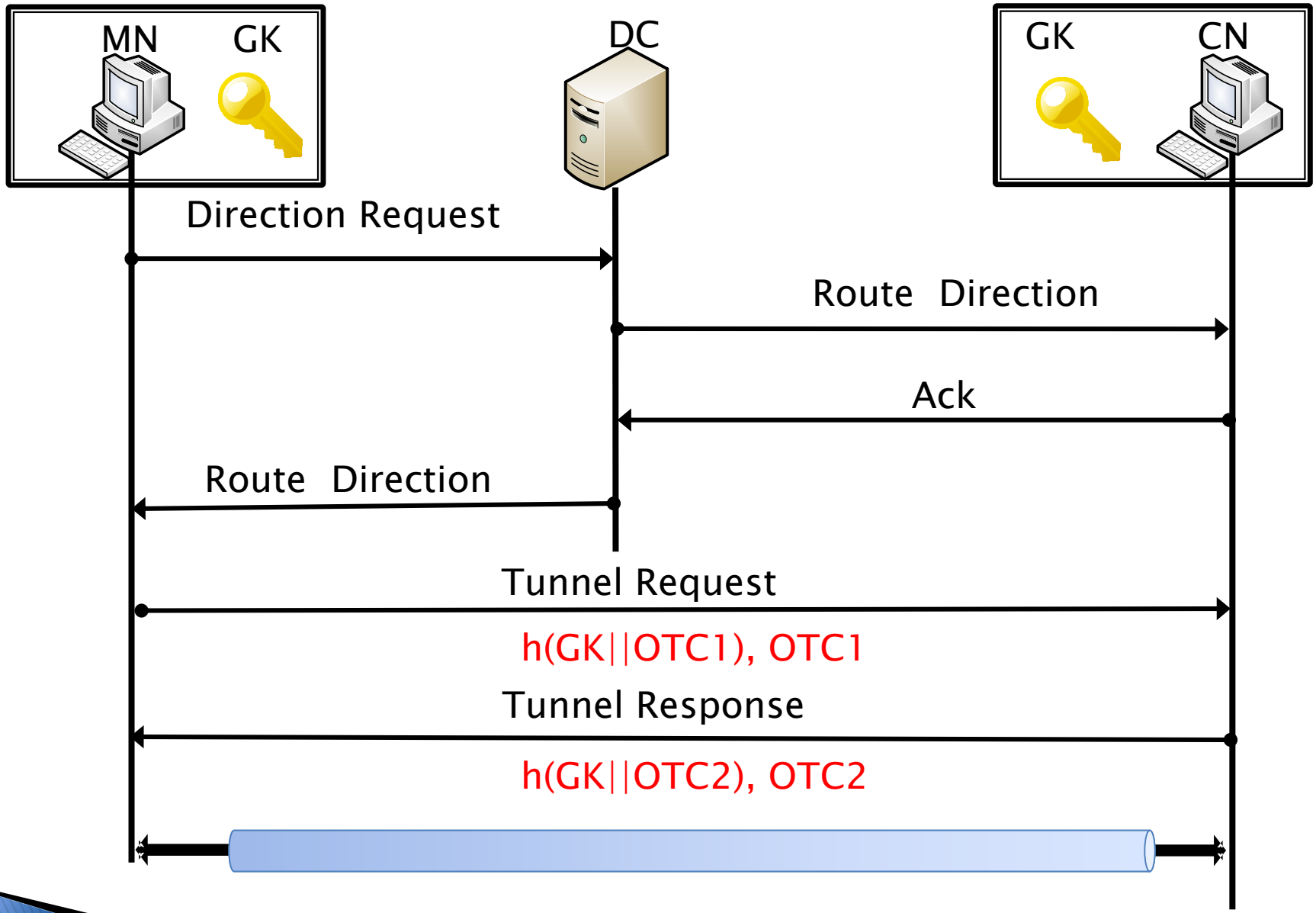


提案方式の構成

- エンド端末(MN,CN)
  - 事前にグループ鍵GKを共有している
    - グループ鍵はGMS(Group Management Server)から配送
- GMS(Group Management Server)
  - グループの作成、管理
  - グループ鍵の配送
  - グループに関する情報を所有

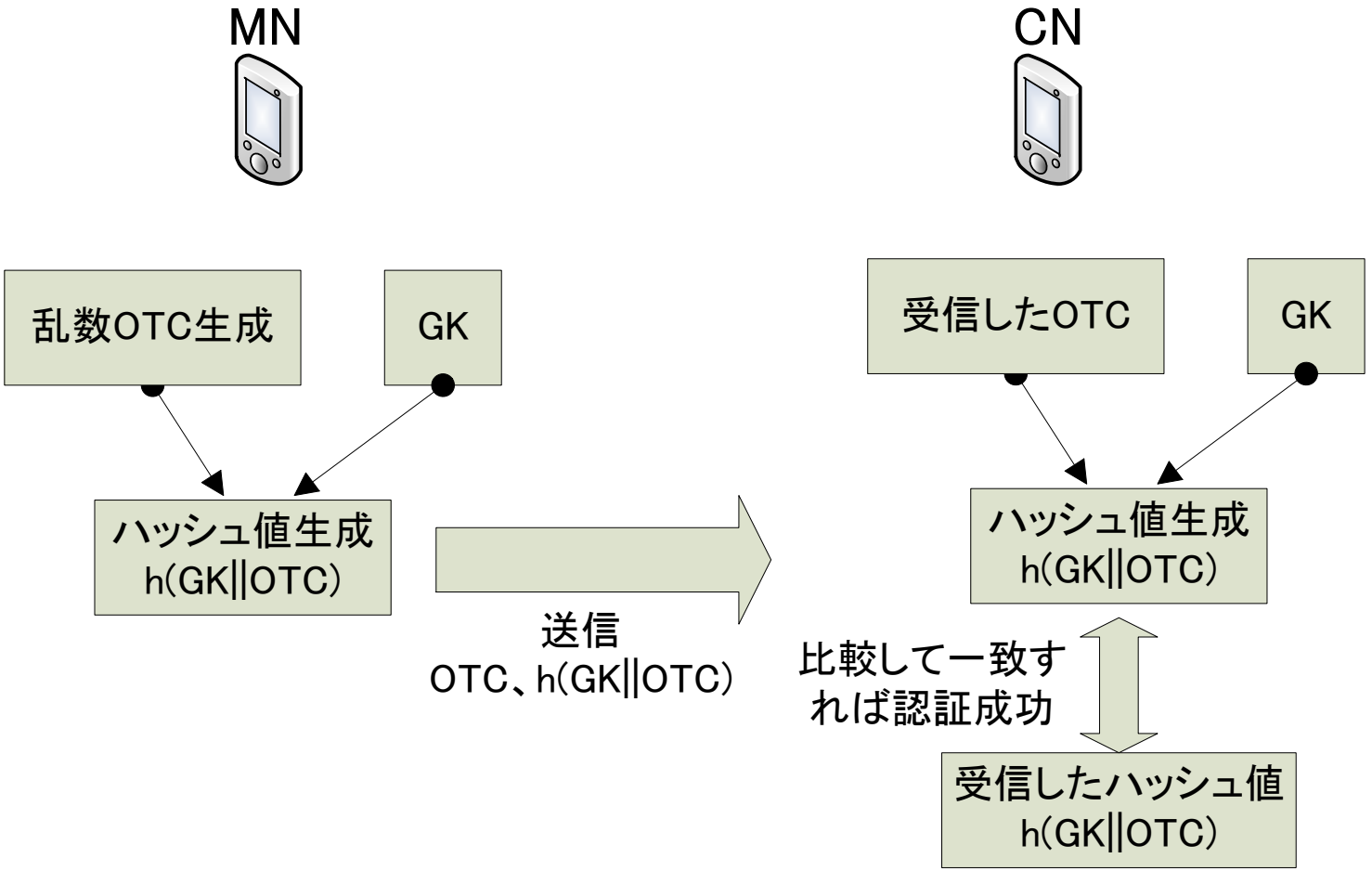


提案方式のシーケンス

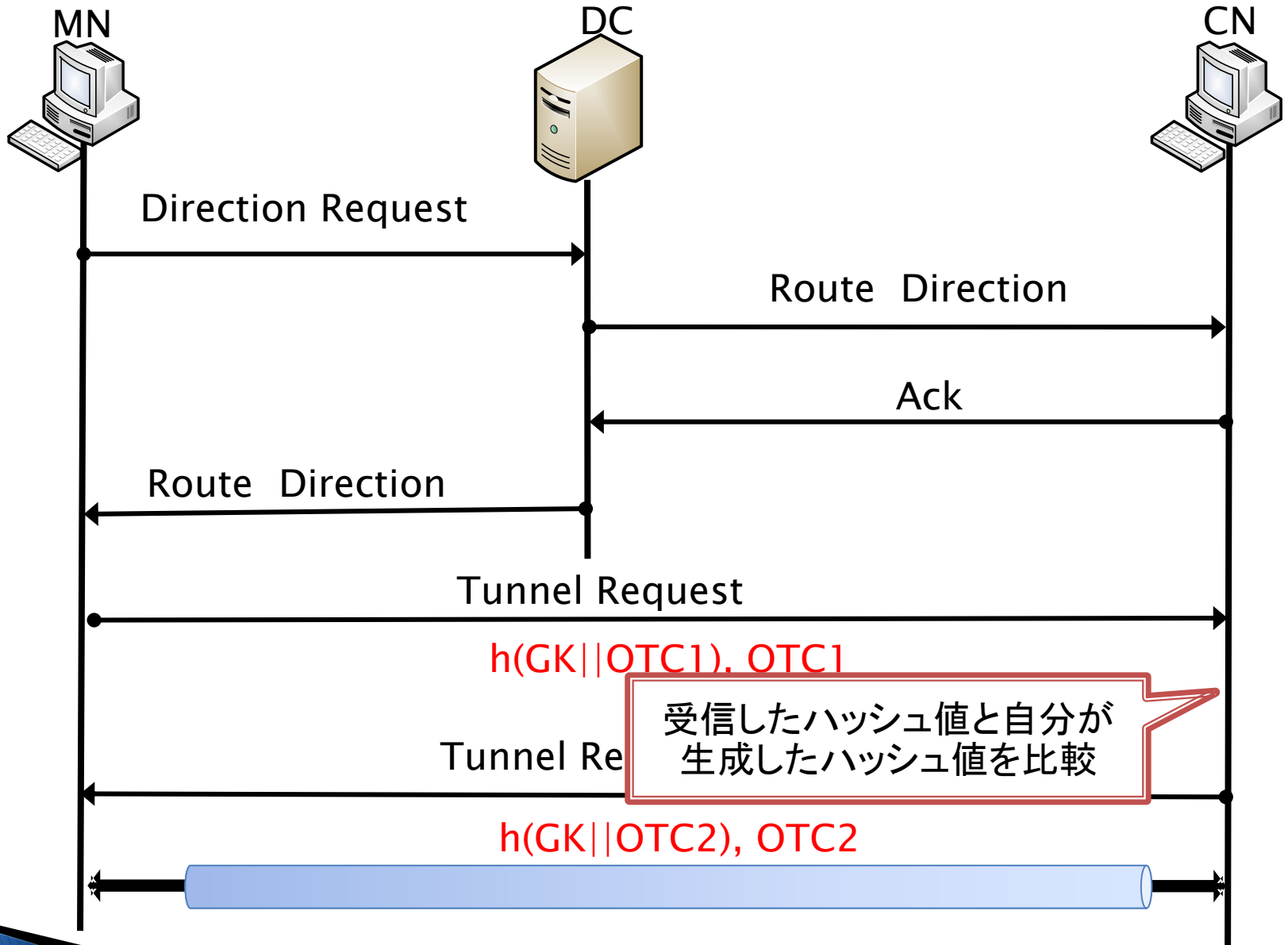

MN
GK
DC
GK
CN
Direction Request
Route Direction
Ack
Route Direction
Tunnel Request
h(GK||OTC1), OTC1
Tunnel Response
h(GK||OTC2), OTC2

相手認証の流れ

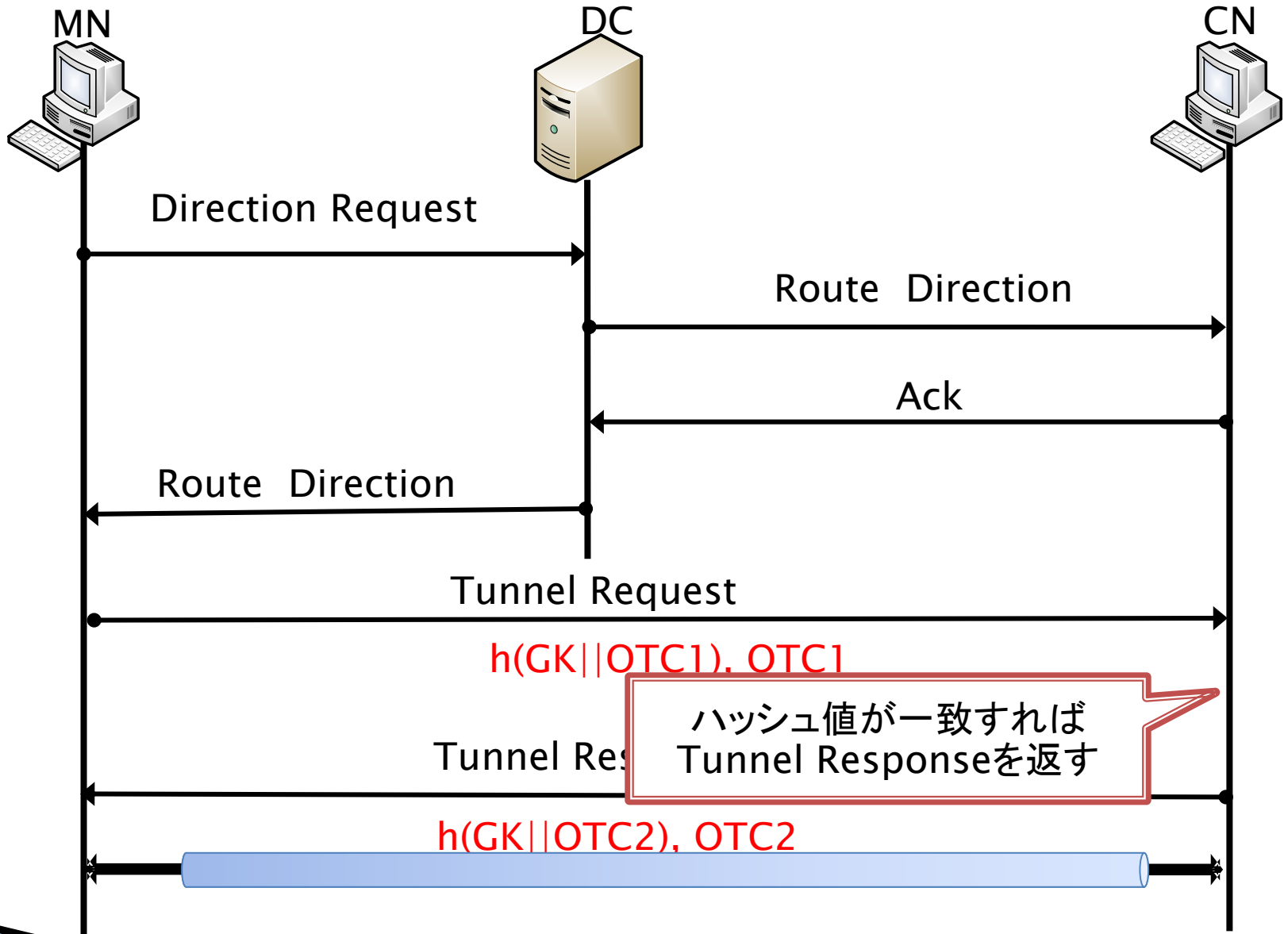
- グループ鍵と乱数のハッシュ値を交換する



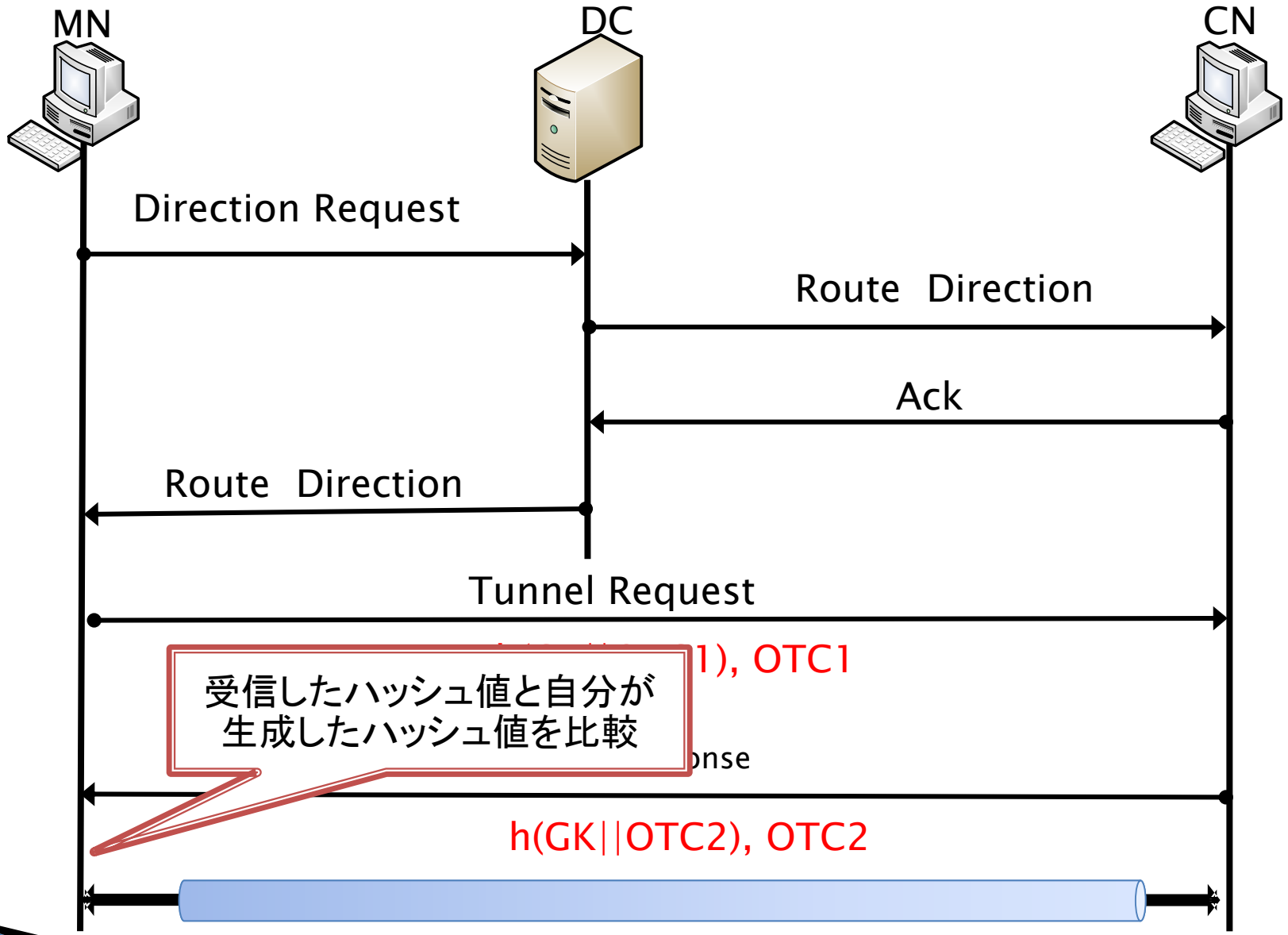
提案方式のシーケンス


MN
DC
CN
Direction Request
Route Direction
Ack
Route Direction
Tunnel Request
h(GK||OTC1), OTC1
受信したハッシュ値と自分が
生成したハッシュ値を比較
Tunnel Re
h(GK||OTC2), OTC2

提案方式のシーケンス

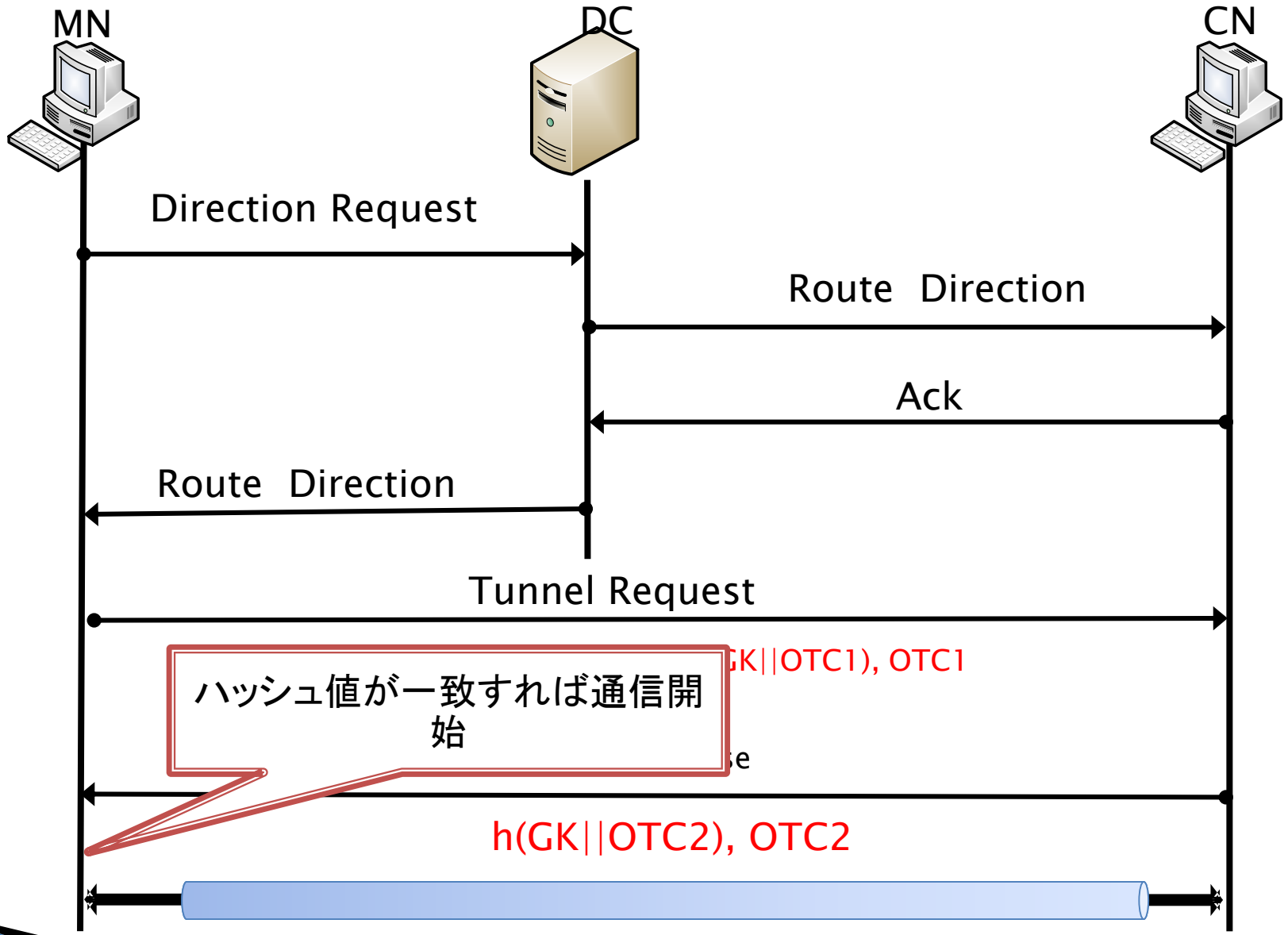

MN
DC
CN
Direction Request
Route Direction
Ack
Route Direction
Tunnel Request
h(GK||OTC1), OTC1
ハッシュ値が一致すれば
Tunnel Responseを返す
Tunnel Res
h(GK||OTC2), OTC2

提案方式のシーケンス


MN
DC
CN
Direction Request
Route Direction
Ack
Route Direction
Tunnel Request
1), OTC1
受信したハッシュ値と自分が
生成したハッシュ値を比較
nse
h(GK||OTC2), OTC2

提案方式のシーケンス

Watanabe Lab.



実装

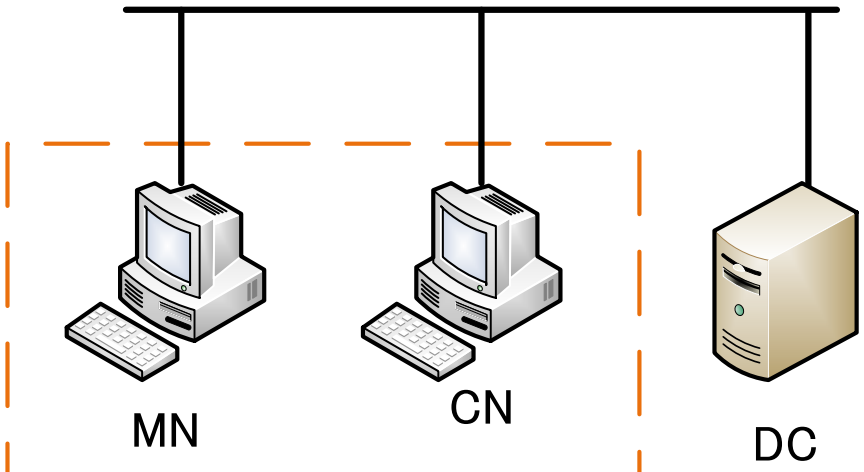
- Tunnel Request/Response生成処理
  - 乱数生成、ハッシュ値生成処理を追加
- Tunnel Request/Response受信処理
  - ハッシュ値検証処理を追加
- GKを手入力で設定する機能を追加

実装

- 装置の仕様と構成
  - 1台のホストPC上に仮想マシンを2台構築
  - DCはローカルネットワーク上に構築されているものを利用

	ホストPC
OS	Windows10 64bit
CPU	Intel Core i3-3220 3.30GHz
Memory	4.00GB

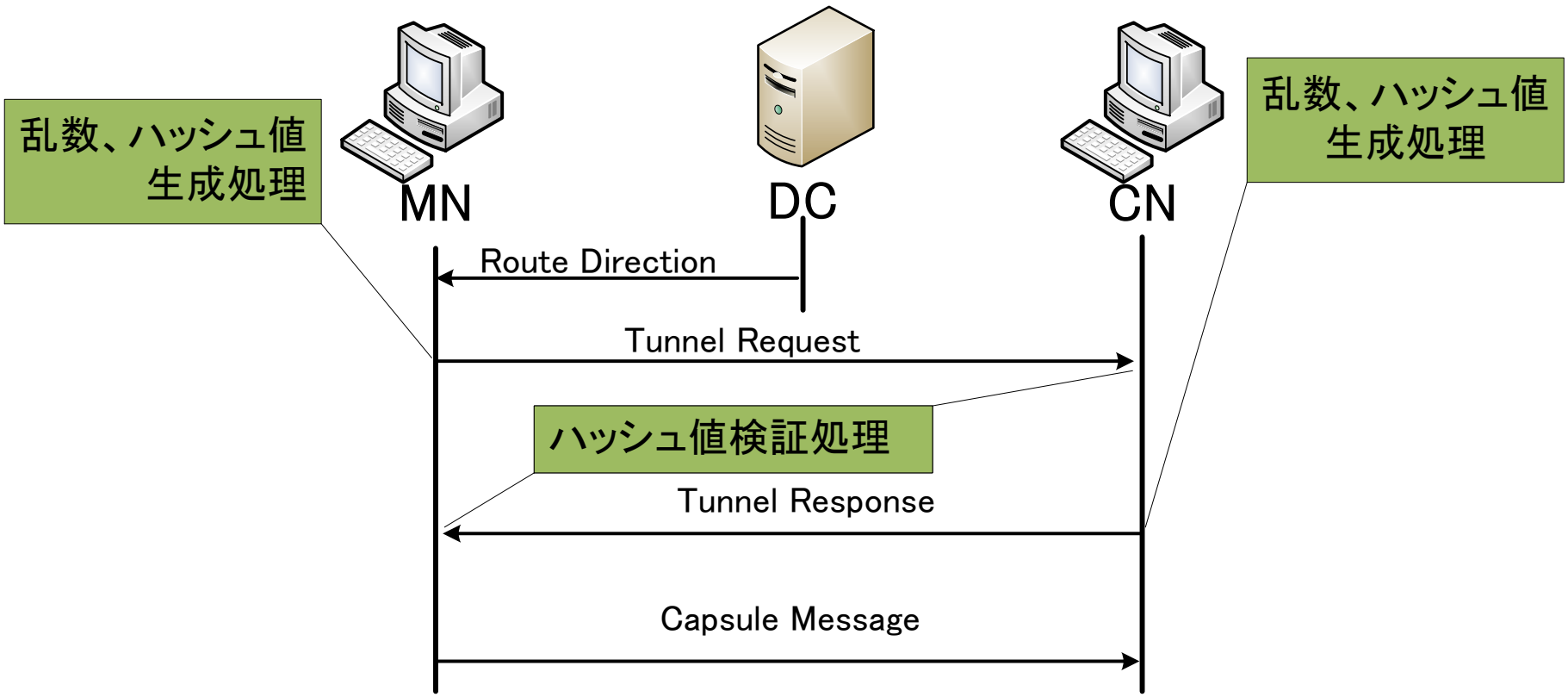
	MN, CN
OS	Ubuntu 14.04
CPU	Intel Core i3-3220 3.30GHz
Memory	各 1.00GB



- GKの共有を手入力で行った

性能測定

- 図の処理時間をclock_gettimeを用いて計測
  - 10回の平均を計測



性能測定

- 図の処理時間を計測
  - ほとんど性能低下はない

従来方式	処理時間(ms)
Tunnel Request Respons生成処理	129.2
Tunnel Request Respons受信処理	144.3
合計	273.5

追加処理	処理時間(μs)
乱数、ハッシュ値生成処理時間	9.063
ハッシュ値検証処理時間	485.2
合計	494.3

まとめ

- グループ鍵を用いた相手認証方式
  - NTMobileおいてグループ鍵を用いた相手認証機能を追加
- 実装と評価
  - 仮想環境にて動作することを確認
  - ほとんど性能低下がない処理時間で実装
- 今後の方針
  - GMSを用いた実装