

IPsecにおけるPKIの利用について

CACAnetフリースクール
はやし ゆういち
yu-ichi@gcc.ne.jp

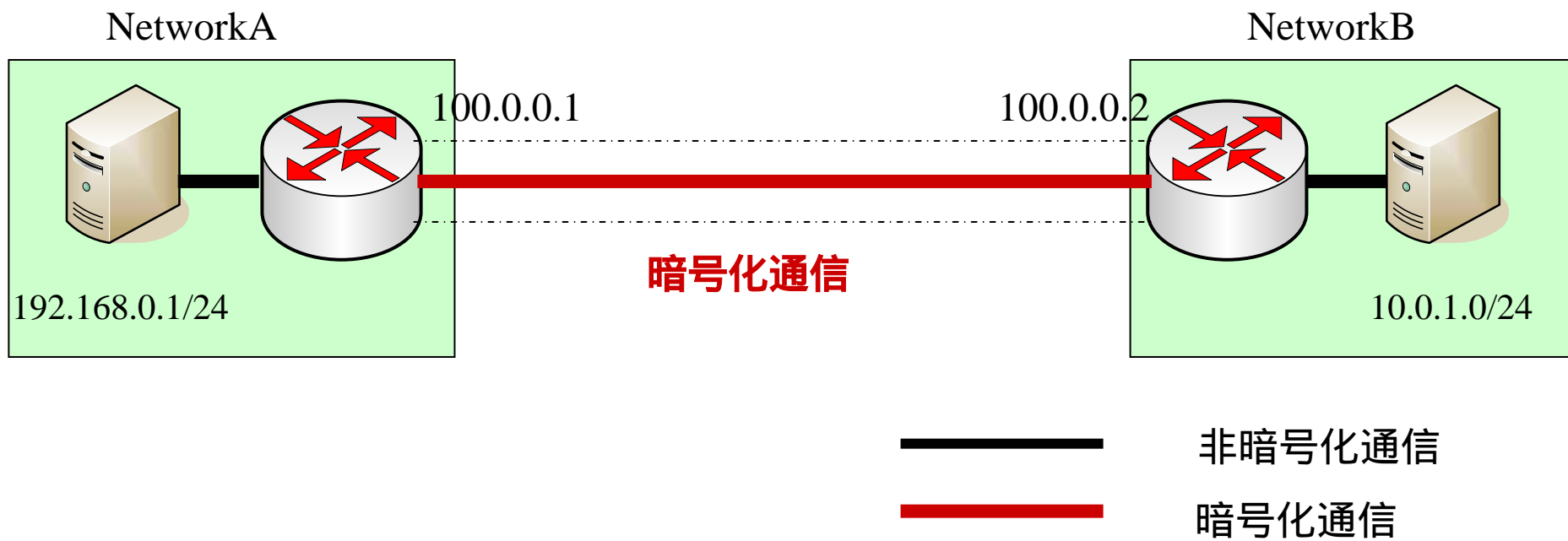
本日の進行内容(目次)

- IPsec Architecture
- IKE (Internet Key Exchange)の役割と仕組み
- IKEを利用したPhase1の流れ

IPsec Architecture

- IPsecの動作イメージ
- IPsecを用いた通信の確立
- セキュリティポリシー
- セキュリティプロトコル(AH,ESP)
- Tunnel mode と Transport mode

IPsecの動作イメージ



IPsecの動作イメージ

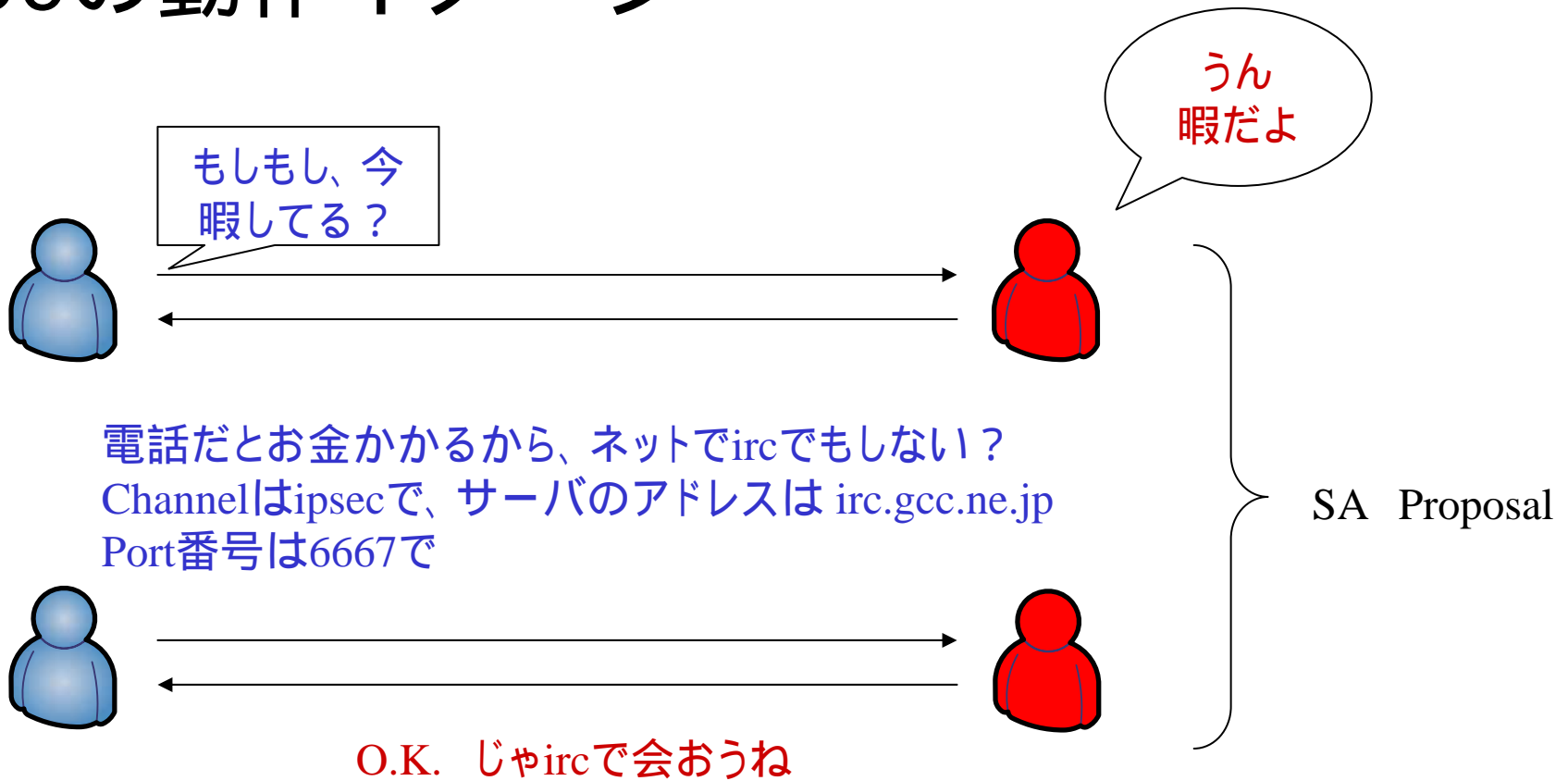
□ アニメのあらすじ

太郎は暇だったので花子に電話をしました。花子の番号にかけて電話にでるのは当然花子のハズです。花子がかかってきた電話を着信通知から太郎と判断しました。ではお話の始まり

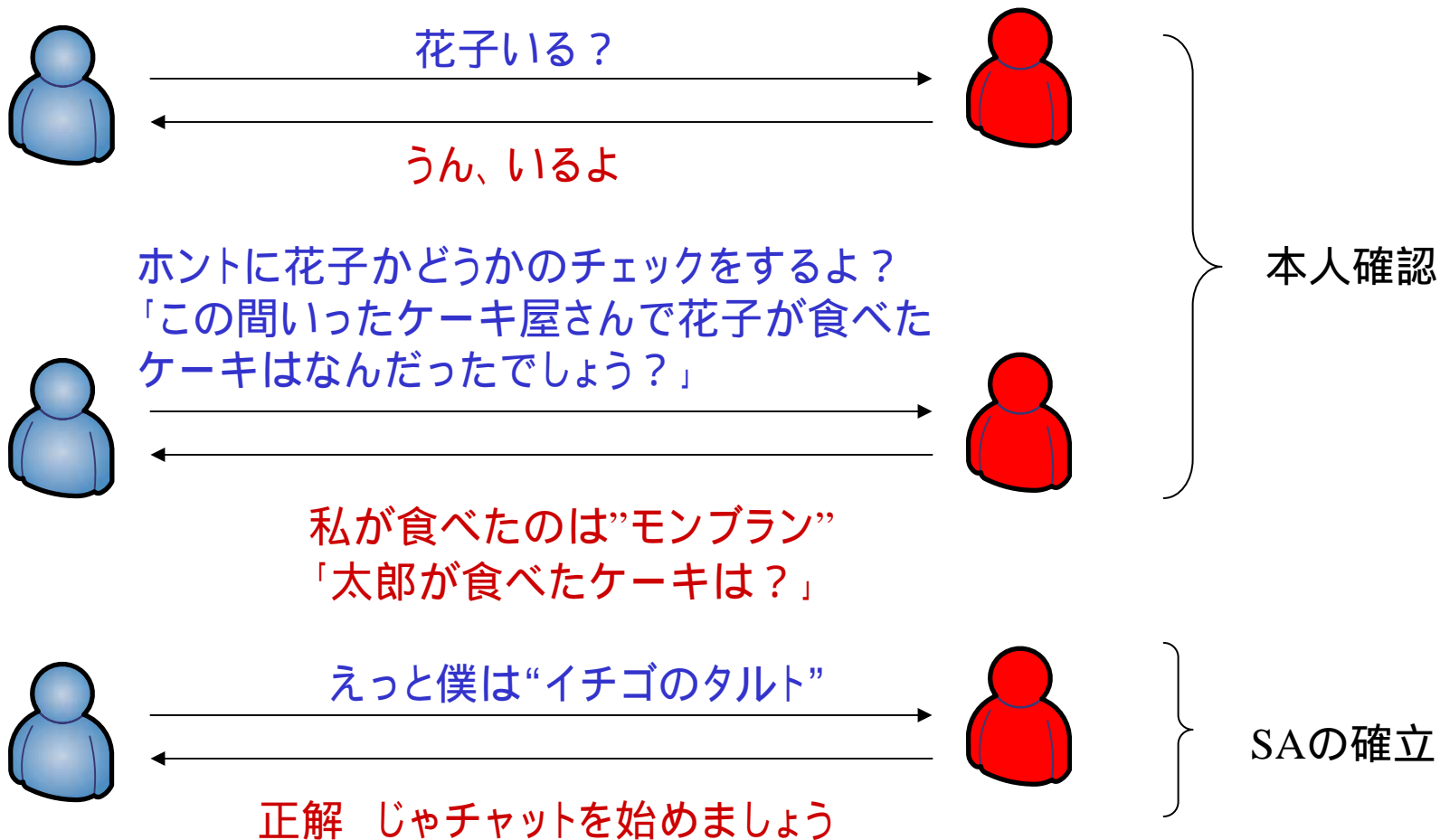
(注意)

アニメはあくまで概要でIKEや暗号通信の部分について概念を省いている部分があります

IPsecの動作イメージ



IPsecの動作イメージ



IPsecを用いた通信の確立

- IPsecを利用するための各ルールの作成 (SAD:security Association Database)と各コネクシヨンの確立(SA: Security Association)

暗号化方式・認証方式の決定

IPsecの適応範囲 (単一ホストのみorネットワーク)

SA (Security Association)

- IPsec通信ごとにおけるコネクション
- SADに従いSAが生成される

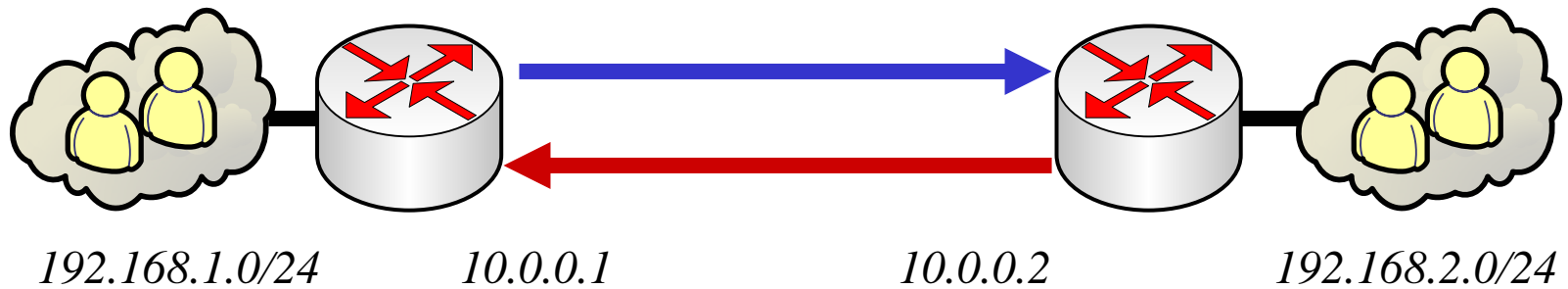
例: SADが

- ネットワーク単位で暗号化(tunnel mode)
- Packetを暗号化
- TCPのpacketのみ暗号化

上記のルールに従い暗号化通信を行う相手にSAを確立する

SAは必ず片方向であり、相手から自分も同じルールを適用し、コネクションを確立する

SAのイメージ



SA:192.168.1.0/24から192.168.2.0/24へ向かうパケットは暗号化



SA:192.168.2.0/24から192.168.1.0/24へ向かうパケットは暗号化

セキュリティプロトコル(AH,ESP)

□ セキュリティプロトコル

そのSAがESPで処理されるか、AHで処理されるかの区別(AH, ESPは後述)

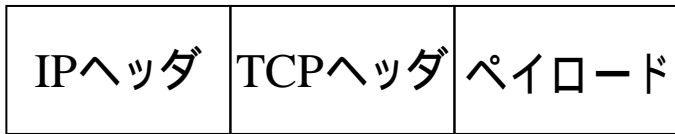
ESPとAHを併用する場合はバンドルを行う

- バンドルとはあるSAをもう一つのSAで Wrapperするようなもの

ESP (Encapsulating Security Payload)

- Tunnel modeではIPヘッダを含め全てのペイロードが暗号化
 - 新しいIPヘッダが付加され新しいヘッダは暗号化対象外
- Transport modeではIPヘッダ以外のペイロードが暗号化される
 - 元のIPヘッダは暗号対象外

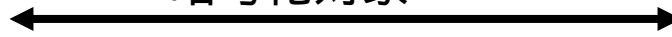
ESP



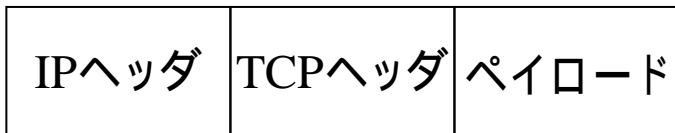
Transport mode



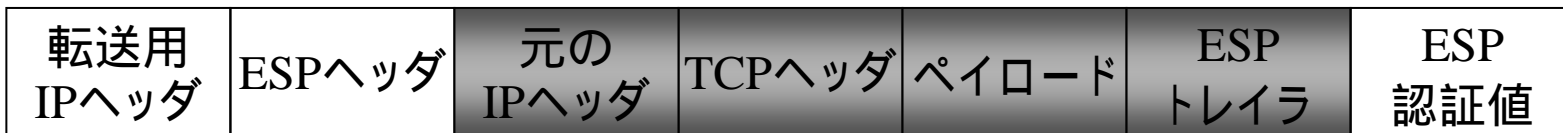
暗号化対象



認証対象



Tunnel mode



暗号化対象



認証対象



AH (Authentication Header)

- パケット全体を対象とするICVを付加することにより、メッセージの改ざんやIPの書き換えが行われていないかどうかを確認できる
- またICVを送信先IPアドレスから秘密対象鍵を用いて生成するため本人の確認も行うことができる（本人性確認）

ESPとAHの違い

□ ESP

暗号化を担当

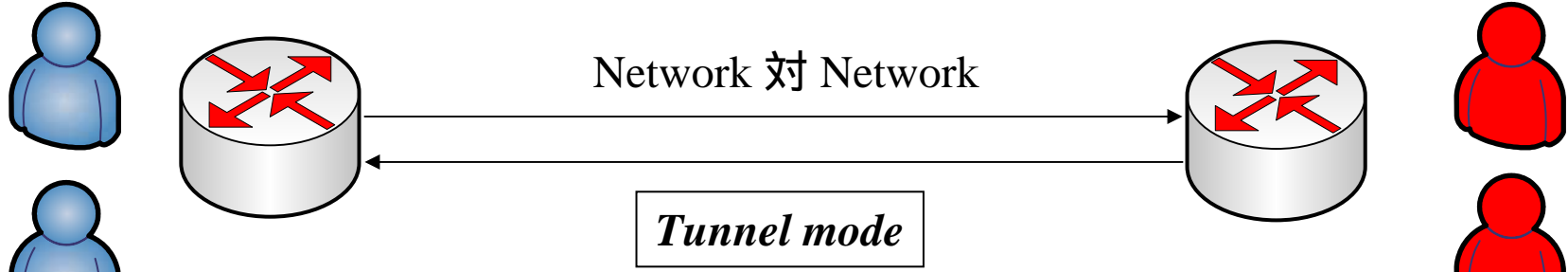
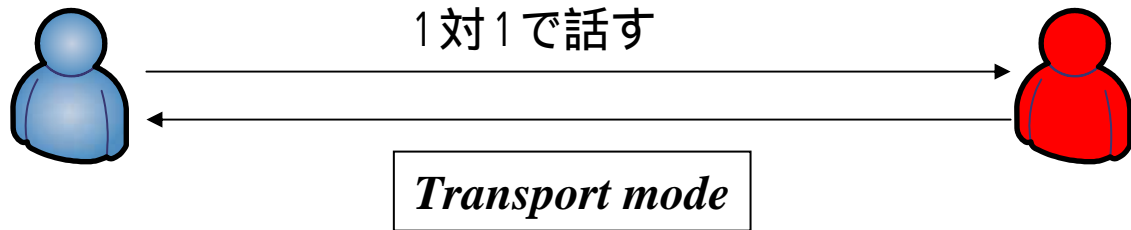
完全性保証はAHより劣るためIKEと組み合わせて利用

□ AH

完全性保証用認証値

暗号化が行えない

Tunnel mode / Transport mode



ESP利用時のTunnel modeではsrc addr, dst addr がルータで付加されるIP addrでカプセル化されるため、どのpeer同士で通信を行っているかを秘密にすることが可能

IKEの役割と仕組み

- IKE
- ISAKMP SAとIPsec SA
- Diffie-Hellman動作原理
- Proposal交換の概要
- Phase1とPhase2
- 各Phaseのモード
- Main/Aggressive mode(Phase 1)
- Quick mode(Phase 2)
- Phase2 + PFSの役割

IKE (Internet Key Exchange)

- IPsecの通信が必要になると自動的にSAを生成するプロトコル

SAは手動で設定しても良いが、パラメータが多く、設定が煩雑なためIKEを利用することがほとんどである。

- 基本機能

Proposal交換

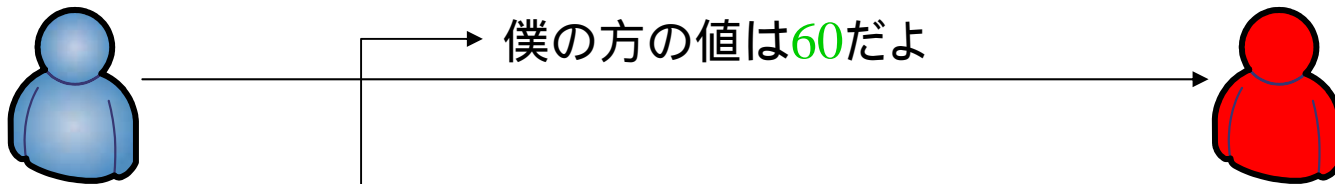
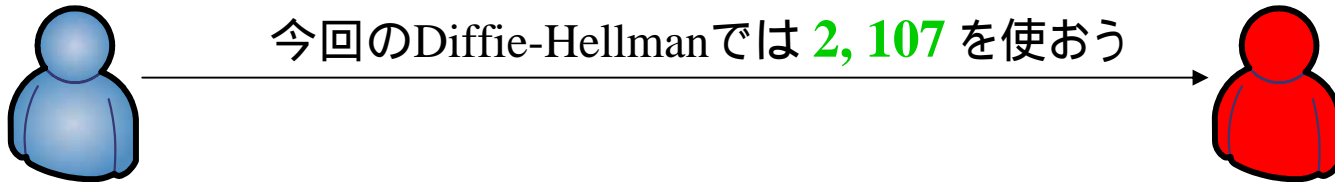
Diffie-Hellman交換

IKE相手の認証

秘密対称鍵の生成(Diffie-Hellman)

- 公開値と秘密値を利用して、通信者同士しかわからない秘密対称鍵を生成
- 手法（簡単です）
次のスライドで

Diffie-Hellman交換 (1)

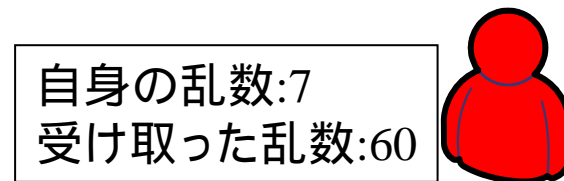
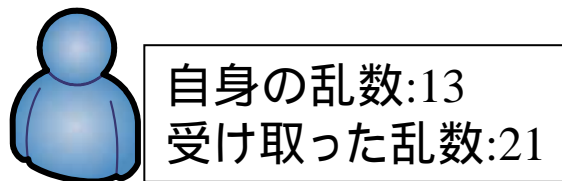
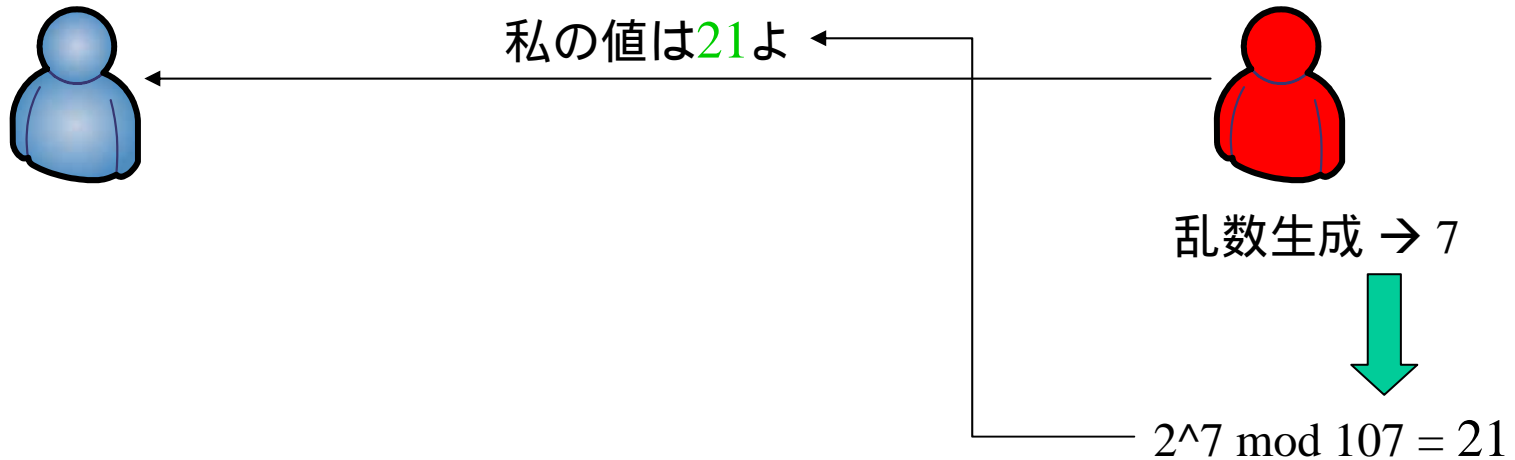


乱数生成 → 13



$$2^{13} \bmod 107 = 60$$

Diffie-Hellman交換 (2)



$$21^{13} \bmod 107 = 70$$

$$60^7 \bmod 107 = 70$$

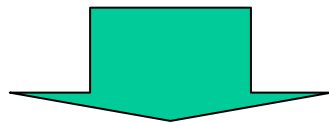
結果は一致

Diffie-Hellman交換 (3)

$$\square 21^{13} \bmod 107 = 2^{(13 \cdot 7)} \bmod 107 = 60^7 \bmod 107$$

- 60, 21は盗聴可能であり、2, 107は公開値だとし、未知の乱数13, 7をそれぞれ x, y とすると下記の式を解けば乱数を見つけることが可能

$$21^x \bmod 107 = 2^{(x \cdot y)} \bmod 107 = 60^y \bmod 107$$



離散対数問題とほぼ同じ計算量が必要であり事実上解析不可能

Proposal交換の概要

- IPsec通信に関する提案をする
 - AH/ESP
 - Tunnel mode / Transport mode
 - 暗号化アルゴリズム
- 受信者はSPDを検索し、受け入れ可能かどうか判断
- 双方が通信に関する合意が取れればISAKMP SAを確立

Phase 1 と Phase 2

□ Phase 1

IPsecの通信を行う前準備

- ISAKMP SAのコネクションの確立
- 秘密対象鍵の生成
- 本人確認 (本人同士しか知らない秘密の値を秘密鍵で暗号化)

IKEにおける認証(本人確認)の方式

Pre-Shared Key認証	事前に通信相手と“秘密の文字列”を共有しておく、通信相手に対してその数だけ必要となる
公開鍵認証	通信相手の公開鍵を事前に入手しておく必要がある、複数の通信相手がいる際は、その数だけ相手の公開鍵を入手しておく必要がある
デジタル署名認証	事前に通信相手の鍵情報を取得しておく必要がない、CAによって署名された公開鍵を利用するため、通信に対する否認ができない

Phase1 とPhase2

□ Phase2

IPsecのproposalの提案と受諾

IPsec SAコネクションの確立

IPsec通信の開始

通信路の寿命が来たら再度上記3つの手順
をやりなおす

Main / Aggressive mode (Phase 1)

□ Aggressive mode

短い手続きでISAKMP SAを確立Phase 1

□ Main mode

SAパラメータの交換と決定

鍵交換とNonceペーロードの交換

IDペイロードとHashペイロードの交換

ISAKMP SAの確立

Quick mode(Phase 2)

□ Phase2

Quick mode

- IPsec SAの確立(計2本のSAを生成する)
- 3回の送信を行い確立する
- IDにはほとんどの場合IP addressを使用

Phase2 + PFSの役割

- Phase2で利用する秘密鍵はすでに生成してある秘密対称鍵を利用するため、Phase1の秘密対称鍵が漏洩した場合には全ての通信が危険にさらされるためPhase2でもういちどDiffie-Hellmanを行う、その結果、より安全にPhase2におけるIPsec SAによる通信を行うことができる

IKEを利用したPhase1の流れ

- Pre-Shared Key (Main Mode)
- Pre-Shared Key (Aggressive Mode)
- Pre-Shared Key利用時の利点・欠点
- デジタル署名方式 (Main Mode)
- デジタル署名方式 (Aggressive Mode)
(おまけ程度)
- デジタル署名方式利用時の利点・欠点

Pre-Shared Key (Main Mode)[1]

Initiator

HDR, SA

HDR, KE, Ni

HDR*, IDii, HASH_I

Responder

HDR, SA

HDR, KE, Nr

HDR*, IDir, HASH_R

Pre-Shared Key (Main Mode)[2]

□ HDR, SA ,

Proposalの提案と受諾

- このペイロードはProposal-payload, Transform-payloadを含む

□ HDR, KE, Ni ,

Diffie-Hellman交換による秘密対称鍵の生成

- このペイロードはDiffie-Hellman公開値とNounceによる乱数の交換を含む
- SKEYID = prf (Pre-Shared Key, Ni_b|Nr_b)

Pre-Shared Key (Main Mode)[3]

- DH後生成されたSKEYIDを利用して用途に応じた鍵素材を生成する

$$\text{SKEYID}_d = \text{prf}(\text{SKEYID}, g^{xy} \mid \text{CKY-I} \mid \text{CKY-R} \mid 0)$$

- SKEYID_d は、非 ISAKMP SA のための鍵を生成するために使用される鍵素材

$$\text{SKEYID}_a = \text{prf}(\text{SKEYID}, \text{SKEYID}_d \mid g^{xy} \mid \text{CKY-I} \mid \text{CKY-R} \mid 1)$$

- SKEYID_a は、ISAKMP SA が自分のメッセージを認証するために使用する鍵素材

$$\text{SKEYID}_e = \text{prf}(\text{SKEYID}, \text{SKEYID}_a \mid g^{xy} \mid \text{CKY-I} \mid \text{CKY-R} \mid 2)$$

- SKEYID_e は、ISAKMP SA が自分のメッセージの秘匿性を守るために使用する鍵素材

Pre-Shared Key (Main Mode)[4]

- HDR*, ID_i, HASH_*

最初のproposalで合意された暗号化とアルゴリズムを用いて、 ID_i の交換で生成された秘密対称鍵SKEYID_eによって暗号化されている

$\text{HASH}_I = \text{prf}(\text{SKEYID}, g^{x_i} | g^{x_r} | \text{CKY-I} | \text{CKY-R} | \text{SA}_{i,b} | \text{ID}_{i,b})$

$\text{HASH}_R = \text{prf}(\text{SKEYID}, g^{x_r} | g^{x_i} | \text{CKY-R} | \text{CKY-I} | \text{SA}_{i,b} | \text{ID}_{i,b})$

- 暗号化されるのはID-payloadとHash-payload

- ここでの利用可能なIDはIP_addressでなくてはならない

ID_i の交換はSKEYID_eによって暗号化されるが、その暗号化を行うためには相手のPre-Shared Keyが事前に分かっている必要があり、この交換が終わったあとにIDが特定できたのでは遅いため、IP_addressを利用して予め相手を特定しておく必要がある

ID情報

□ IDの種類はRFC2407に規定に従っている

(例) ID_IPV4_ADDR → IPv4アドレス

Pre-Shared Key (Aggressive Mode)

Initiator

HDR, SA, KE, Ni,
IDi

HDR*, HASH_I

Responder

HDR, SA, KE, Nr, IDir,
HASH_R

Pre-Shared Key (Aggressive Mode)[2]

□ HDR, SA, KE, Ni, IDii

Proposalの提案, Diffie-Hellman交換, Nonce, ID

- 最初の packets で上記のパラメータを送信するため
 - IDの暗号化はされない
 - DHグループもネゴシエートできない

Pre-Shared Key (Aggressive Mode)[3]

□ HDR, SA, KE, Nr, IDir, HASH_R

Proposalの受諾, Diffie-Hellman交換,
Nonce, ID, レスポンダHashの送信

- この段階でDiffie-Hellmanの秘密対称鍵の生成が完了(SKEYID, SKEYID_d, SKEYID_a, SKEYID_e)
- レスポンダからの秘密対称鍵を利用し生成したHashの送信

Pre-Shared Key (Aggressive Mode)[3]

□ HDR*, HASH_I

イニシエータHashの送信

- 秘密対称鍵を利用してイニシエータのHashを生成し、送信する。この場合、パケットの暗号化はしなくてもかまわない

Pre-Shared Key利用時の利点・欠点

□ Pre-Shared Key

設定が容易である

事前に接続者同士で“秘密のワード”をネゴシエーションしておく必要がある

接続するpeerが増えれば増えるほどワードの管理が煩雑に
(通常はplain textで保存)

□ Main Mode

IDにIPアドレスを利用する必要があり、移動先からのIPsec
利用を行うことができない

□ Aggressive Mode

IDを保護することができない

DHグループが予め決定されてしまう

デジタル署名方式 (Main Mode)[1]

Initiator

HDR, SA

HDR, KE, Ni

HDR*, ID_{ii}, [CERT,]
SIG_I

Responder

HDR, SA

HDR, KE, Nr

HDR*, ID_{ir}, [CERT,]
SIG_R

デジタル署名方式 (Main Mode)[2]

□ 通信の概要(1)

イニシエータ(ini)はレスポнда(res)にIKE通信のPhase1を開始する

SA Proposalの合意, Diffie-Hellman交換などによる秘密鍵対称鍵の生成などを行う

IniはPre-Shared Key認証の場合に送信していたHash-payloadを同様に計算し,そのHashに公開鍵とペアになっている秘密鍵を用いてデジタル署名をつけ, Signatureペイロードで送る(必要であれば、CAによって署名された公開鍵証明書と同時に送る)

デジタル署名方式 (Main Mode)[3]

□ 通信の概要(2)

resはSignatureペイロードからデジタル署名を取り出し、デジタル署名が確かにiniによって生成されたものかどうかを確認する(公開鍵証明書の確認にはCAの公開鍵で検証)

resも同様に自らの秘密鍵でHashに署名をし、iniに送信する

iniは受け取ったSignatureペイロードを検証する

デジタル署名方式 (Main Mode)[4]

□ r, s, z はPre-Shared Key認証と同様

□ SKEYIDの生成

$$\text{SKEYID} = \text{prf} (N_{i_b} \parallel N_{r_b}, g^{xy})$$

デジタル署名方式 (Main Mode)[4]

□ HDR*, IDii, [CERT,] SIG_* ,

Pre-Shared Keyの時と同様にHashを生成し、お互いに秘密鍵でデジタル署名をしたものを送信する(SIG_I or SIG_R)

相手の要求に応じて公開鍵証明書([CERT])を添付する(要求がない場合はしなくともよい)

デジタル署名の検証に使用する公開鍵が本物があるかどうかは、公開鍵証明書のsubjectNameもしくはsubjectAltNameの中に相手のIDと同じものがあり、さらにその公開鍵証明書が本物であることを、公開鍵証明書を発行したCAの公開鍵を使用して検証する

デジタル署名方式 (Main Mode)[4]

□ 使用可能なID

IKEではX.500で規定されているDNをIDとして利用するには不便なため、X.509 Version 3で拡張されたsubjectAltNameフィールドもIDとして利用可能になっている（IKEの実装によって差異がある）

デジタル署名方式 (Aggressive Mode)[備考程度]

Initiator

HDR, SA, KE, Ni,
IDi

HDR, [CERT,]
SIG_I

Responder

HDR, SA, KE, Nr, IDir, [CERT,]
SIG_R

デジタル署名方式利用時の利点・欠点

□ デジタル署名方式

通信相手ごとに事前に鍵情報などを共有する必要がないためpeerが増えた際にも管理が楽

事前に鍵を用意するための敷居がかなり高い

各OSにおけるIPsecの実装

□ Linux

FreeS/WAN(開発終了)

FreeS/WAN(X.509)

□ FreeBSD

KAME + racoon

□ Windows

標準添付

Configuration Fileにみる X.509の設定 (FreeBSD, racoon 未実験)

```
remote 192.168.0.200
{
    #exchange_mode main,aggressive;
    exchange_mode aggressive,main;
    doi ipsec_doi;
    situation identity_only;

    my_identifier user_fqdn "peer1@gcc.ne.jp";
    peers_identifier user_fqdn "peer2@gcc.ne.jp";
    #certificate_type x509 "mycert" "mypriv";

    nonce_size 16;
    lifetime time 1 min; # sec,min,hour

    proposal {
        encryption_algorithm 3des;
        hash_algorithm sha1;
        authentication_method pre_shared_key;
        dh_group 2;
    }
}
```

racoon.conf.inより引用

デジタル署名方式による相互接続実験と試験環境の提供の重要性

□ 実際に触ってみる

実際にconfigurationをし、相互接続の実験を行い設定等をまとめ、公開することが重要 → 利用が困難なため手を出しにくい実情がある

□ 試験的な証明書を簡単に発行できる環境を

証明書を手に入れる、作成する段階で挫折するユーザも少なくない

まとめ

- 口頭で :-)

END

ご静聴ありがとうございました