マルチキャスト通信のための鍵交換プロトコルの拡張

濱野 淳史† 伊藤 純一郎† 所 真理雄‡

†慶應義塾大学大学院 理工学研究科 計算機科学専攻
‡慶應義塾大学

〒 223–8522 神奈川県横浜市港北区日吉 3–14–1
Tel: 045–560–1150 Fax: 045–560–1151
E-mail: hamano@mt.cs.keio.ac.jp

あらまし IP 層において安全な通信を提供するための機構である IPsec では, 暗号化や認証を行なうために通信者があらかじめ鍵を共有しておく必要がある. IPsec に必要な鍵の共有を動的に行なう機構として, IKE (The Internet Key Exchange) が提案されており, 現在標準化が進められている. IKE の現在の機構では, 1 対 1 通信であるユニキャスト通信のための鍵交換方式のみしか規定されておらず, 1 対多通信であるマルチキャスト通信のための鍵交換は不可能である. そこで本論文では, 鍵交換プロトコル IKE を, IPsec を用いたマルチキャスト通信のための鍵配布が可能となるように拡張を行ない, その有効性を示すために鍵交換デーモンに実装する. さらに, 実装における問題点を示し, その解決方法を述べる.

キーワード IPsec, 鍵交換, IKE, マルチキャスト通信, Join メッセージ, Distribution Mode

Extension of Key Management Protocol for Multicast Communications

Atsushi Hamano † Jun-ichiro Itoh† Mario Tokoro‡

† Department of Computer Science, Faculty of Science and Technology, Keio University
‡ Keio University

3-14-1 Hiyoshi, Kohoku–ku, Yokohama 223–8522, Japan
Tel: 045–560–1150 Fax: 045–560–1151
E-mail: hamano@mt.cs.keio.ac.jp

Abstract IPsec, which provides secure communication on IP layer, needs to share session keys for encription and authentication in advance. IKE (The Internet Key Exchange), which is an automatic key exchange mechanism for IPsec, is proposed and now in progress of standardization. But on current specification of IKE, it is possible to establish session keys only for unicast communication. The paper proposes an extension of IKE for multicast communication, and attempts to implement it to show the effectivity of it. The paper also lists the issues of the implementation, and shows the solution.

key words IPsec, key exchange, IKE, multicast communication, Join Message, Distribution Mode

1 はじめに

近年のインターネットの普及により, 一般家庭や企業においてもインターネットという言葉が聞かれるようになった. それに伴い, 様々なマルチメディアアプリケーションが開発されている. その例として, インターネットを介した映画配送システムや, 自宅に居ながらにして多数の人々と会議ができるビデオ会議システム等が挙げられる. このようなアプリケーションは, 一対多通信であるマルチキャスト通信により実現することが効率的である. ところが, このようなアプリケーションを実現する際にはセキュリティの面で問題が生ずる. 現在のインターネットではユニキャスト通信, マルチキャスト通信を問わず, 暗号化されていないデータがネットワーク上を通過するため, 第三者が容易に盗聴や改竄を行うことが可能である. 例えば映画配送システムの場合には料金の支払いをしていない不当なユーザにデータを受信されてしまう恐れがある. また, ビデオ会議システムの場合にはある企業の会議内容がライバル企業に盗聴や改竄される恐れがある. そのため, 安全な通信を実現するためにはインターネット上を通過するデータを暗号化し, またデータの正当性を保証することが不可欠である.

これまで, 通信における暗号化や認証などのセキュリティの強化は, アプリケーション毎で行なわれることが多かったが, 近年 OS レベルでセキュリティの強化を行なうことが提案されている. 特に, IP 層でのセキュリティの強化を目的とした **IPsec** [3] に関する研究が現在盛んに行なわれており, 標準化が進められている. IPsec では, データの改竄と偽造を防ぐための認証ヘッダ **(AH)** [1] と, 改竄と偽造, 盗聴を防ぐための 暗号化ペイロード **(ESP)** [2] の 2 つが規定されている.

一般に暗号化や認証を行なうためには送信者と受信者の間で秘密裡に鍵を共有することが必要である. IPsec では, 通信者が暗号鍵を共有していることを前提として暗号化や認証を行なうための方法が規定されているが, 暗号化や認証に用いられる鍵の交換方法については規定されていない. 鍵の交換方法の 1 つとして管理者が手動で鍵を設定する方法が考えられる. しかし不特定多数の相手と通信することを考慮した場合, この方法は現実的ではない. そのため現在, 動的に鍵の共有を確立する方法がいくつか提案されている. IPsec に必要な鍵の共有を動的に行なう機構としては **IKE (The Internet Key Exchange)** [5] がある. しかし, 現在の IKE の機構では 1 対 1 通信のための鍵交換方法しか規定されておらず, 1 対多通信のための鍵交換方法は規定されていない.

そこで本論文では, 動的にマルチキャスト通信のための鍵配布が可能となるように鍵交換プロトコルの拡張を行ない, その有効性を示す. さらに, 本実装における問題点を挙げ, その解決方法を示す. 本論文で提案する機構を実現することにより, アプリケーションを変更することなく IPsec を用いたマルチキャスト通信が可能となる.

2 鍵交換プロトコル

IKE は, IPsec で必要な鍵を動的に確立する機構であり, 鍵交換の枠組を提供する **ISAKMP (Internet Security Association and Key Management Protocol)** [7] と, ISAKMP 上で実際の鍵管理機構を定義している **Oakley** [9] から成る.

IKE は, ISAKMP のための Security Association (ISAKMP SA) を確立する **Phase 1** と, IPsec の SA を確立する **Phase 2** から成る. なお SA とは, 通信を保護するために用いられる方針や鍵の集合のことである. Phase 1 には通信者の ID の保護が可能な **Main Mode** と, ID の保護が不可能な **Aggressive Mode** の 2 つがあり, Phase 2 には **Quick Mode** のみがある.

Main Mode では始動者と応答者の間で 6 回 (3 往復) の通信を行ない ISAKMP SA を確立する. 始めの 2 回の通信において両者で使用するアルゴリズム等の取り決めが行なわれ (図 1 (a)), 次の 2 回の通信で鍵交換に必要な Diffie–Hellman 鍵交換アルゴリズム [4] の公開値や乱数などの補助的なデータの交換が行なわれる (図 1 (b)). そして最後 2 回の通信で生成された鍵の認証が行なわれる (図 1 (c)).

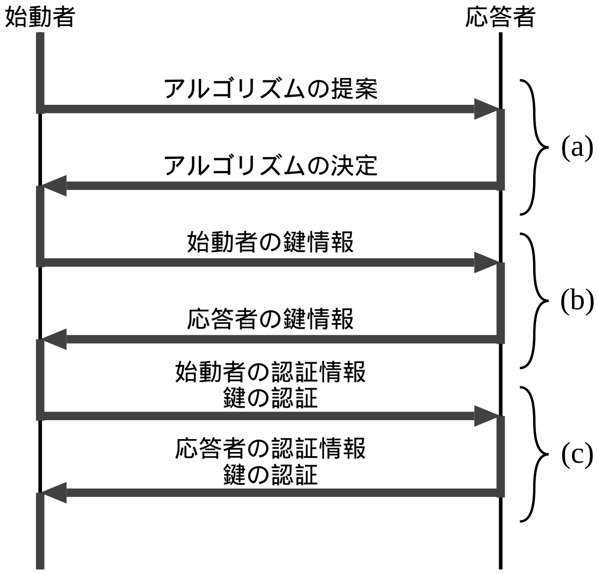


図 1: Main Mode

一方, Aggressive Mode では始動者と応答者の間で 3 回 (1.5 往復) の通信を行ない ISAKMP SA を確立する. 始めの 2 回の通信において両者で使用するアルゴリズム

等の取り決めと, 鍵交換に必要な Diffie–Hellman の公開値や乱数などの補助的なデータの交換が行なわれる (図 2 (a), (b)). さらに 2 回目の通信では, 応答者の認証も行なわれる (図 2 (b)). そして 3 回目の通信において始動者の認証と生成された鍵の認証が行なわれる (図 2 (c)). この 3 回目の通信は, 生成される ISAKMP SA によって暗号化されない. このため, ID の保護が不可能となる.

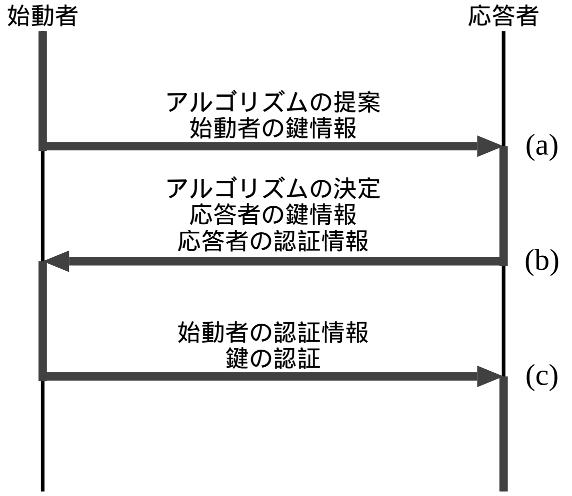


図 2: Aggressive Mode

Main Mode, Aggressive Mode では, 電子署名, 公開鍵暗号, 模擬共有鍵を用いる方法の 3 種類の認証方法が利用可能である.

Phase 2 の Quick Mode では始動者と応答者の間で 3 回 (1.5 往復) の通信を行ない IPsec の SA を確立する. Quick Mode における通信内容は ISAKMP SA によって保護される. 始めの 2 回の通信において両者で使用するアルゴリズム等の取り決めが行なわれ (図 3 (a)), 3 回目の通信で生成された鍵の認証が行なわれる (図 3 (b)).

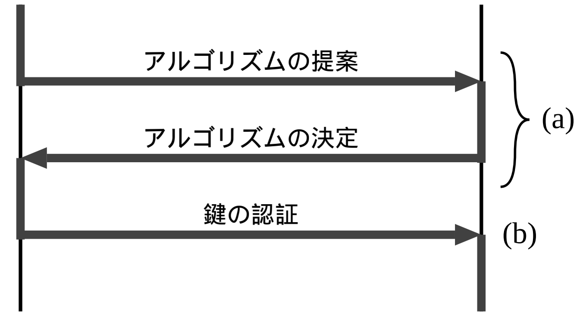


図 3: Quick Mode

現在標準化が進められている IKE の機構では, IPsec を用いたユニキャスト通信の SA の確立のみ可能である.

3 マルチキャスト通信のための拡張

IKE では, マルチキャストグループへの鍵配布について規定されていない. そこで本論文ではマルチキャストグループへの鍵配布を行なう鍵サーバと, 鍵を要求するクライアントの 2 種類の鍵交換デーモンを定義する. そしてこれらが協調して ISAKMP SA を確立し, マルチキャストグループへの鍵配布を行なう方法を提案する.

まずマルチキャストグループに対して共有鍵を配布する際に考慮すべき点を挙げる.

- マルチキャストグループに所属する全てのメンバが同一の鍵を共有しなければならない.
- 正当なメンバのみが鍵を取得可能であり, 不当なホストは取得不可能でなければならない.
- 鍵サーバのアドレスはあらかじめわからないので, 鍵サーバに直接鍵を要求することは不可能である.
- マルチキャストグループの動的な変化に対応できなければならない.

本論文ではこれらを考慮し, 以下のような方式を提案する.

マルチキャストグループの共有鍵の一貫性を保証するために, 鍵サーバが各メンバに対して鍵配布を行なうものとする. また, 鍵サーバはマルチキャストグループの正当なメンバのリストであるアクセス制御リストをあらかじめ保持し, このリストに従い鍵配布を行なうことにより不当なホストへの鍵配布を防止する. さらに, クライアントが鍵サーバのアドレスをあらかじめ知る必要が無いように, 鍵の要求をマルチキャストアドレス宛に送信し, 鍵サーバのみがそれを受信するものとする. そして, マルチキャストグループの動的な変化には, 各メンバがマルチキャストグループに参加する際に鍵サーバにグループの鍵を要求することにより対応する. また, マルチキャストグループの共有鍵に一定時間の寿命を設け, タイムアウト後再び鍵の要求を行なうものとする. これにより, マルチキャストグループから離脱したホストは, それ以降の鍵の要求は行なわない.

3.1 提案モデル

本論文で提案するモデルは, 図 4 に示すように 2 種類の鍵交換デーモンから構成される.

- 鍵サーバ
  グループで用いる鍵等の生成と配布, およびメンバのアクセス制御を行なう鍵交換デーモン. 起動時にマルチキャストグループの共有鍵, セキュリティ・パラメータ・インデックス (SPI), 暗号化に用いる初期

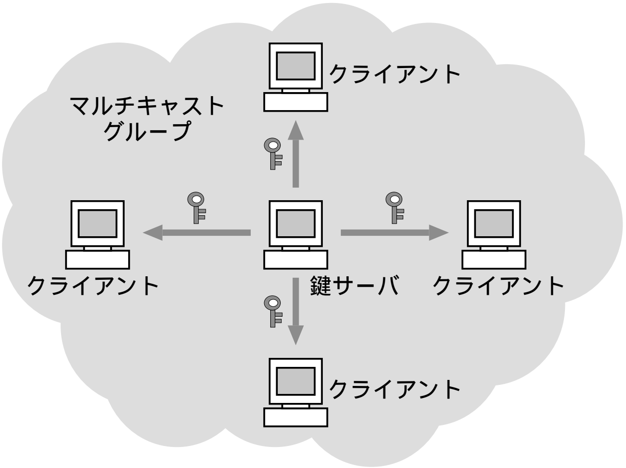


図 4: 提案モデル

ベクタ (Initialization Vector, IV) を生成し, 各メンバへ配布する. 各マルチキャストグループに 1 つ存在する.

- クライアント
  マルチキャストグループのメンバで, 鍵サーバ以外の鍵交換デーモン. マルチキャストグループ参加時に, 鍵サーバに対してグループの共有鍵の要求を行なう.

なお, 鍵サーバ, クライアント共にユニキャスト通信のための SA の確立も可能である.

鍵サーバとクライアントの間で行なわれる処理の概要を以下に示す.

- Phase 1
  クライアントがマルチキャストアドレス宛に **Join** メッセージを送信する. 鍵サーバが Join メッセージを受信し, 両者の間で ISAKMP SA を確立する. Join メッセージは本論文であらたに提案するものである. しかし ISAKMP SA の確立は従来の IKE Phase 1 の処理をそのまま用いる.

- Phase 2
  鍵サーバがマルチキャストグループの鍵情報をクライアントに配布する. クライアントは鍵の完全性を確認するためのメッセージを返送する. Phase 2 での処理は従来の IKE Phase 2 と全く異なるものであり, 本論文であらたに **Distribution Mode** として提案する.

3.3 節以降では, 各 Phase における処理の詳細について述べる.

3.2 表記法

本論文で用いる表記方法を表 1 で説明する.

表 1: 本論文で用いる表記法

表記法	意味
Next Payload	次につながるペイロードの識別子
Reserved	未使用 (0 でなければならない)
Payload Length	ペイロード長
Group Address	マルチキャストグループアドレス
Key Type	鍵の種類
Zero	0 でなければならない
SPI	SPI 値
SPI Length	SPI の長さ
Key	マルチキャストグループの共有鍵
Key Length	鍵の長さ
IV	初期ベクタ
IV Length	初期ベクタの長さ
$prf(key, msg)$	鍵付き疑似乱数発生関数
$SKEYID_a$	認証用の鍵
$\|$	メッセージの結合
SA	SA の情報を含むペイロード
M_KEY	鍵情報ペイロード

3.3 Phase 1

Phase 1 では, あらたにマルチキャストグループに参加するホストが鍵サーバと協調して ISAKMP SA を確立するために以下の通信を行なう.

1. あらたにマルチキャストグループに参加するホストは, そのマルチキャストアドレス宛に Join メッセージを送信する. Join メッセージは, ISAKMP の標準ヘッダ, マルチキャストアドレスと必要とする鍵の種類とを含む Join ペイロードから構成される. 図 5 に Join ペイロードのフォーマットを示す.

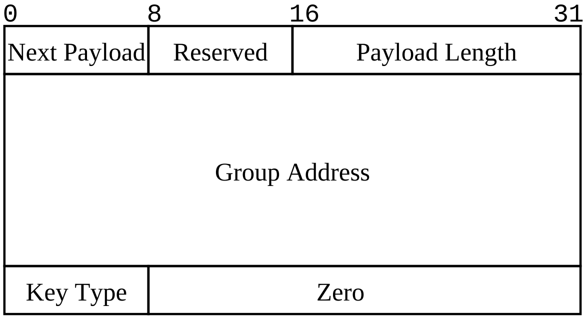


図 5: Join ペイロードのフォーマット

2. Join メッセージを受信した鍵サーバは, アクセス制御リストにより送信者がマルチキャストグループの

正当なメンバであるかを判断する．正当なメンバであることが確認できたら，次のステップに進む．不当なメンバであると判断された場合にはその Join メッセージを無視し，以後の処理は行なわない (図 7).

3. Join メッセージの送信者と鍵サーバの間で ISAKMP SA が確立されていない場合は Main Mode，あるいは Aggressive Mode と同様の処理により ISAKMP SA を確立する．ISAKMP SA の確立後は Phase 2 に進む．既に ISAKMP SA が確立されている場合には，即 Phase 2 に移る．

図 6 に Phase 1 における処理を示す．

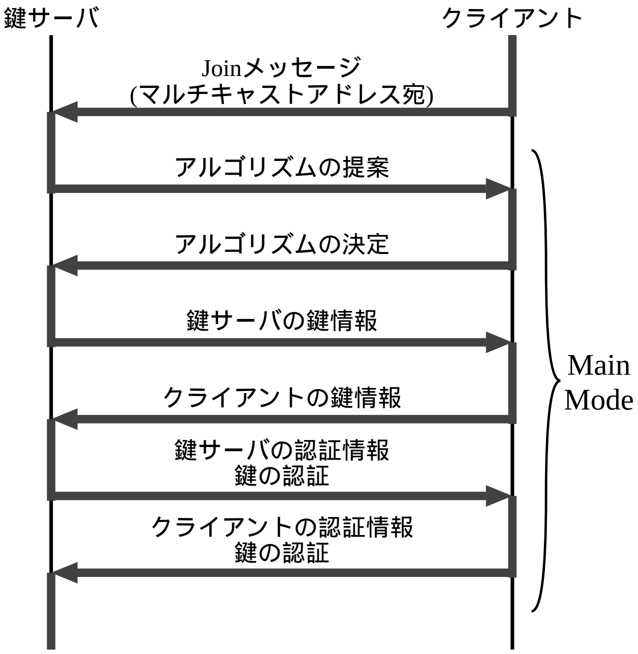


図 6: Phase 1 における処理

本論文において用いるアクセス制御リストには，マルチキャストグループの正当なメンバであると認められたホストの IP アドレスを列挙する．したがって鍵サーバでは IP アドレスによりホストの正当性を判断することになるので，ホストレベルでのアクセス制御が可能となる．アクセス制御リストの動的な作成方法については，本論文の機構では定義しておらず，各マルチキャストグループの管理者により行なわれることを仮定している．図 7 に，本論文におけるアクセス制御機構を示す．

3.4 Phase 2

Phase 2 では，マルチキャストグループへ共有鍵の配布を行なうために以下の通信を行なう．この過程を Distribution Mode とする．

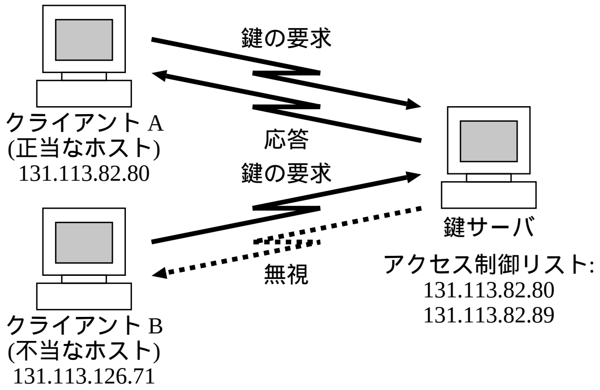
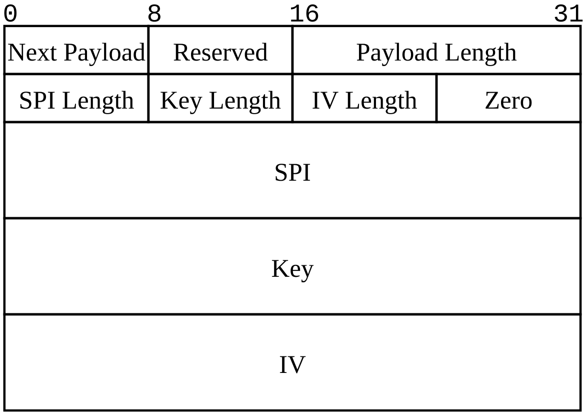


図 7: アクセス制御機構

1. 鍵サーバは鍵情報メッセージを，Phase 1 において確立された ISAKMP SA を用いて暗号化し，クライアントに送信する．鍵情報メッセージは，ISAKMP の標準ヘッダ，メッセージのハッシュ値 $HASH(D_1)$ を含むハッシュペイロード，SA の情報を含むペイロード，SPI 値，共有鍵，初期ベクタを含む 鍵情報ペイロードとから構成される．図 8 に鍵情報ペイロードのフォーマットを示す．



0	8	16	31
Next Payload	Reserved	Payload Length	
SPI Length	Key Length	IV Length	Zero
SPI			
Key			
IV			

図 8: 鍵情報ペイロードのフォーマット

2. クライアントは ISAKMP SA を用いて復号化を行ないハッシュ値 $HASH(D_1)$ を照合し，メッセージの完全性を確認する．その中から鍵情報を取り出しカーネルに渡す．これによりクライアントは，IPsec を用いたマルチキャスト通信のための SA をカーネル内に所持する．

3. クライアントは IPsec を用いたマルチキャスト通信のための SA が正しく確立されたことを確認するために，確認メッセージを ISAKMP SA を用いて暗号化して鍵サーバに送信する．確認メッセージは，ISAKMP

の標準ヘッダとメッセージのハッシュ値 $HASH(D_2)$ を含むハッシュペイロードから構成される.

4. 鍵サーバは ISAKMP SA を用いて復号化を行ないハッシュ値 $HASH(D_2)$ を照合し, SA が正しく確立されたことを確認する. SA が正しく確立されていない場合はクライアントにエラーメッセージを送信し, 確立した SA を無効とする.

図 9 に Phase 2 における処理を示す.

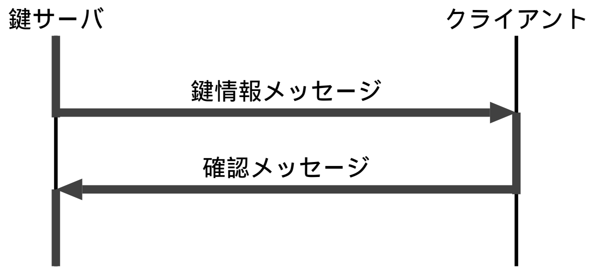


図 9: Phase 2 における処理

また, ハッシュ値の計算には次式を用いる.

$$HASH(D_1) = prf(SKEYID_a,\ SA \mid M\_KEY) \quad (1)$$

$$HASH(D_2) = prf(SKEYID_a,\ SPI \mid KEY \mid IV) \quad (2)$$

4 実装および評価

4.1 実装

本論文では, 3 章で提案した機構の実装を行なった. OS には FreeBSD 2.2.1–RELEASE に伊藤 純一郎氏が作成した IPsec カーネルへの変換パッチ[1] を当てたものを用いた. また, 鍵交換デーモンは, 祢宜 知孝氏が作成した鍵交換デーモン[2] をもとに, 本論文で提案した拡張を加えた.

4.2 評価および検討

4.1 節で行なった実装を, 表 2 の環境で評価した結果を表 3 に示す. なお ISAKMP SA の確立には Main Mode を用いた. また認証方法として, 公開鍵暗号を用いる方法を採用した.

表 3 より, Join メッセージに伴う処理時間の増加は約 3 % であり, オーバーヘッドは非常に小さいことがわかる. また Phase 2 では, Distribution Mode の処理時間は Quick Mode の処理時間の約 21 % であることがわかる. これは, Quick Mode では通信者が協調して鍵の生成等を行なう必要があるが, Distribution Mode ではこの必要がないためである.

表 2: 評価環境

	鍵サーバ	クライアント
機種	PC–AT 互換機	PC–AT 互換機
CPU	P6-200	P5-75
メモリ	32 M	24 M

表 3: 評価結果 ($msec$)

Phase	unicast	multicast
1	913.1 (Main Mode)	940.4 (Join Mesg + Main Mode)
2	24.1 (Quick Mode)	0.5 (Distribution Mode)
Total	937.2	945.4

以上より, Join メッセージの処理のために多少オーバーヘッドがあるものの, Phase 2 の処理時間がユニキャスト通信の場合と比較してかなり小さいので, 全体の処理時間はほとんど同じであることがわかる. したがって, 本論文の提案する機構は妥当であるといえる.

5 議論

本論文において提案した機構には以下の問題点が存在する.

- 鍵サーバによるマルチキャストグループの共有鍵管理は, 鍵の一貫性の保証という点では理想的である. しかし一極集中の構造であるため鍵サーバの故障時に一点障害となり, それ以降の鍵配布が不可能となる.
- 本論文の機構では, 鍵配布を行なうのは正当な鍵サーバであると暗に仮定している. したがって不当な鍵サーバを実装し, マルチキャストグループへ偽の共有鍵を配布することが可能である.

これらの問題は, 以下のような方法で解決することが可能である.

まず第一の問題点は一極集中の構造である. そこで, 各マルチキャストグループに複数の鍵サーバを設けることにする. このとき複数の鍵サーバが同時に異なる鍵を生成した場合, マルチキャストグループの各メンバが同一の鍵を所持できない. そこで鍵サーバにマスターとスレーブを設け, マスターがマルチキャストグループの鍵を

---

[1] ftp://ftp.itojun.org/pub/ipsec/FreeBSD221–ipsec.970714.tar.gz.pgp

[2] http://www.mt.cs.keio.ac.jp/person/negi/implement-j.html

生成し, スレーブに配布する. マスターとスレーブの相違点は, 起動時にマルチキャストグループの共有鍵を生成するか否かのみであり, 起動後の機能は同じである. クライアントがどの鍵サーバからマルチキャストグループの共有鍵を取得するかは, どの鍵サーバの応答メッセージを一番最初に受信するかに依存する.

マルチキャストグループの共有鍵が必要となったホストは, マルチキャストアドレス宛に Join メッセージを送信する. この Join メッセージに対して, そのマルチキャストグループの全ての鍵サーバが応答メッセージを送信する. クライアントは, 一番最初に応答メッセージを送信した鍵サーバと協調し Main Mode, あるいは Aggressive Mode により ISAKMP SA を確立し, Distribution Mode によりマルチキャストグループの共有鍵を取得する.

一方, 第二の問題点を解決するためにクライアントは鍵サーバの認証を行なう. なお, 鍵サーバの認証には **X.509** [6], [8] を用いる予定である. Join メッセージに対する応答メッセージに X.509 で規定されている証明書を付加することにより鍵サーバを認証することが可能となる.

以上を考慮に入れると, Phase 1 は図 10 のように変更される.

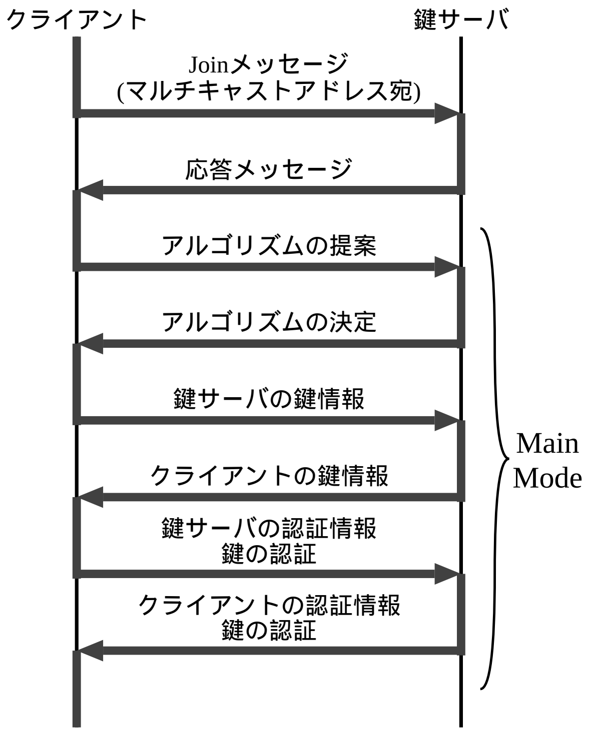


図 10: Phase 1

なお, Join メッセージには変更点はない. 応答メッセージは ISAKMP の標準ヘッダ, X.509 の証明書を含む証明書ペイロードから成る. 証明書ペイロードは ISAKMP [7] において規定されているものを用いる.

6　まとめ

本論文では, 現在標準化が進められている鍵交換プロトコル IKE を, マルチキャストグループに対して共有鍵の配布が可能となるように拡張を行なった.

本論文で提案した拡張は以下の 4 つの特徴をもつ.

- マルチキャストグループ毎に鍵サーバを設け, メンバに鍵配布を行なう. したがって, すべてのメンバが同一の鍵を矛盾なく所持することが可能である.
- アクセス制御機構により正当なメンバのみがマルチキャストグループの鍵を取得可能であり, 不当なホストによる鍵の取得を防止できる.
- クライアントは鍵の要求時に Join メッセージをマルチキャストアドレス宛に送信するので, 鍵サーバのアドレスをあらかじめ知る必要がない.
- クライアントのマルチキャストグループへの参加と同時に鍵の要求を行なうので, 動的なメンバの変更に対応できる.

これらは, IKE に対して比較的少ない変更で実現できた. また本論文で提案した機構を実装し評価を行った結果, 拡張による処理時間のオーバーヘッドは微小であることが確認できた. さらに提案した機構の問題点を挙げ, その解決策を示した.

今後は, この解決策をもとに再び実装を行なう予定である.

参考文献

[1] R. Atkinson. *IP Authentication Header*. RFC 1826, *IETF*, August 1995.

[2] R. Atkinson. *IP Encapsulating Security Payload (ESP)*. RFC 1827, *IETF*, August 1995.

[3] R. Atkinson. *Security Architechture for IP*. RFC 1825, *IETF*, August 1995.

[4] W. Diffie and M. Hellman. New Directions in Cryptograpy. In *Transactions on Information Theory*, IEEE, November 1976.

[5] D. Harkins and D. Carrel. *The Internet Key Exchange (IKE) <draft–ietf–ipsec–isakmp–oakley–06.txt>*. Internet Draft, *IPSEC Working Group*, February 1998. work in progress.

[6] C. I'Anson and C. Mitchell. Security Defects in CCITT Recommendation X.509 – The Directory

Authentication Framework. *Computer Communication Review*, April 1990.

[7] D. Maughan, M. Schertler, M. Schneider, and J. Turner. *Internet Security Association and Key Management Protocol (ISAKMP) <draft–ietf–ipsec–isakmp–09.txt>*. Internet Draft, *IPSEC Working Group*, March 1998. work in progress.

[8] C. Mitchell, M. Walker, and D. Rush. CCITT/ISO Standards for Secure Message Handling. *IEEE Journal on Selected Areas in Communications*, May 1989.

[9] H. Orman. *The OAKLEY Key Determination Protocol <draft–ietf–ipsec–oakley–02.txt>*. Internet Draft, *IPSEC Working Group*, 1996. work in progress.