目次

概要

このドキュメントでは、IPSec の要点を簡潔に説明しています。 ここでは、事前共有キーによる Internet Key Exchange（IKE; インターネット鍵交換）、認証局による IKE、および IPSec の基本設定を取り上げます。 この文書に、すべてが網羅されているわけではありません。 しかし、これらの作業とその順序を理解するのに役立ちます。

 **警告：**「高度な」暗号化の輸出については、厳しい制限が課せられています。 米国の連邦法に違反すると、シスコではなく違反した本人の責任が問われます。 輸出規制についての疑問な点は、E メールで export@cisco.com までお問い合せください。

注通常の LAN-to-LAN トンネル、あるいは何らかのデバイスで終端されている VPN クライアントでは、マルチキャストとブロードキャストはサポートされていません。 マルチキャストで通過できるのは GRE トンネルだけです。 これがサポートされているのはルータでだけで、VPN 3000 コンセントレータやファイアウォール（ASA/PIX）ではサポートされていません。

前提条件

要件

このドキュメントに関する固有の要件はありません。

使用するコンポーネント

このドキュメントは、特定のソフトウェアやハードウェアのバージョンに限定されるものではありません。

表記法

ドキュメント表記の詳細は、『シスコ テクニカル ティップスの表記法』を参照してください。

背景説明

IPsec は Ciscoセキュリティ プラットフォーム（Cisco IOS® ソフトウェア、PIX、等）のための次世代ネットワーク層 暗号 プラットフォームです。 最初は現在廃止である 1829 によって RFCs 1825 で記述されていて、IPsec は IETF IPセキュリティ ワーキンググループが示すいくつかの文書で現在説明されています。 現在、IPSec では IP バージョン 4 のユニキャスト パケットがサポートされています。 IPv6 とマルチキャストは、将来、サポートされる予定です。

現行の Cisco による暗号化製品全体にわたり、IPSec には次の長所があります。

1. **マルチベンダ**か。IPSecフレームワークが標準化されるので、顧客はあらゆる特定のベンダー製品にロックされていません。 ルータ、ファイアウォール、クライアント デスクトップ（Windows、Mac など）に、IPSec が組み込まれています。
2. **スケーラビリティ**か。IPsec は大企業を考慮して設計されています。 そのため、鍵管理機能が組み込まれています。

注IPsec を使用できるシスコのプラットフォームは複数ありますが、このドキュメントは Cisco IOS ソフトウェアを対象としています。

暗号化に関する用語（用語集）

IPSec を理解し、この文書を読み進むためには、次の用語について理解する必要があります。 この文書の他の部分で略語を見つけたときは、このページで定義を参照してください。

Advanced Encryption Standard（AES）か。AES は、電子データ伝送の保護に使用するために、Federal Information Processing Standard（FIPS: 連邦情報処理標準）が承認した暗号化アルゴリズムとして規定されました（FIPS PUB 197）。). AES は、Rijndael アルゴリズムに基づくもので、128、192、または 256 ビットのブロックの暗号化のための 128、192、または 256 ビット長の鍵の使用法を規定しています。 鍵長とブロック長による 9 通りの組み合せすべてが可能です。

Authentication Header（AH）か。認証とオプションのリプレイ検出サービスを提供するセキュリティ プロトコル。 AH は保護を必要とするデータ（完全な IP データグラムなど）に埋め込まれます。 AH は単体で使用するか、または Encryption Service Payload（ESP）とともに使用できます。 『RFC 2402 』を参照してください。

認証か。IPSec フレームワークの機能の 1 つ。 認証により、データストリームの整合性が確立され、転送中に改ざんされていないことが確認されます。 また、認証によってデータストリームの

発信元に関する確証も得られます。

Certification Authority（CA）か。これは認証を発行し、取り消す責任のサード パーティ エンティティです。 それ自体の証明書と CA の公開鍵を持つ各デバイスは、与えられた CA ドメインの内部で他のすべてのデバイスを認証できます。 この用語は、これらのサービスを提供するサーバ ソフトウェアにも適用されます。

認証か。暗号に署名されたオブジェクトこの識別と関連付けられる識別および公開キーが含まれている。

標準的な暗号か。これは Cisco IOS ソフトウェア リリース 11.2 で使用される Cisco 専用暗号化メカニズムです。 Classic crypto を利用できるのは Cisco IOS ソフトウェア リリース 11.3 です。 ところが、IPSec は Cisco IOS ソフトウェア リリース 11.2 用に改造されてはいません。「classic crypto」という名称は、マーケティングの文脈では「Encryption Express」または「Cisco Encryption Technology（CET; Cisco 暗号化テクノロジー）」と呼ばれる場合もあります。

証明書無効リスト（CRL）か。これはある特定の CA によってリストされている電流取り消された認証すべてをリストするデジタルで署名付きメッセージです。 これは、一般の店舗が不正なクレジット カードを拒否するために使用する盗用カード番号の一覧のようなものです。

クリプト マップか。これは 2 つの主たる機能を行う Cisco IOSソフトウェア 設定 エンティティです。 まず、これによりセキュリティ処理が必要なデータ フローが選択されます。 次に、これらのフローのためのポリシーと、トラフィックが到達すべきクリプト ピアが定義されます。

クリプト マップはインターフェイスに適用されます。 クリプト マップの概念は標準的な暗号で導入されましたが、IPsec のために拡張されました。

データ統合か。これはデータは送信中に修正されなかったことを確認することを保護 データのピースの受信者が可能にする秘密鍵ベースか公開鍵ベースのアルゴリズムの使用によってデータ統合メカニズム、です。

データの機密保持か。これは攻撃者がそれを読むことができないように保護 データが処理される方式です。 一般に、データの機密性は、データの暗号化と、通信に関与している当事者同士でしか使用できないキーによって実現されます。

データ元の認証か。これはレシーバが保護 データは送信側からだけ起きるかもしれませんことを確認できるセキュリティ サービスです。 このサービスでは、データの完全性サービスに加えて、秘密鍵が送信者と受信者の間でだけ共有される鍵配布メカニズムが必要です。

データ暗号規格（DES）か。DES は National Bureau of Standards（NBS; 米国標準局）によって 1977 年に発行された、IBM の Lucifer アルゴリズムに基づく秘密キー暗号化方式です。 DES の対照として公開キーがあります。 Cisco では、classic crypto（40 ビットおよび 56 ビットの鍵長）、IPSec crypto（56 ビットの鍵）、および PIX ファイアウォール（56 ビットの鍵）で、DES を使用しています。

デフィーヘルマンか。これは不確かなメディア上の共有鍵の確立の方式です。 デフィーヘルマンは、（この定義リストにある）Oakley のコンポーネントの 1 つです。

DSS か。公開キー暗号化に基づく米国国立標準技術研究所（NIST）によって設計されている Digital Signature Algorithm。 DSS は、ユーザー データグラムの暗号化は行いません。 DSS は classic crypto および Redcreek IPSec カードのコンポーネントですが、Cisco IOS ソフトウェアに実装されている IPSec のコンポーネントではありません。

Encryption Service Adapter（ESA）か。これは使用されるハードウェアによって基づく暗号化アクセラレータです:

- Cisco 7204 および 7206 ルータ
- Cisco 7500 シリーズの全ルータの第二世代 Versatile Interface Processor2-40（VIP2-40）
- Cisco 7000 シリーズ Route Switch Processor（RSP7000）カードおよび Cisco 7000 シリーズ Chassis Interface（RSP7000CI）カードを搭載した Cisco 7000 シリーズ ルータの VIP2-40

IPSec では ESA アクセラレーションは使用されませんが、ESA カードを搭載した機種ではソフトウェア ベースでアクセラレーションが実行されます。

Encapsulating Security Payload（ESP）か。オプション の 認証およびリプレイ攻撃検出サービスをデータの機密保持および保護に与えるセキュリティプロトコル。 ESP によってユーザ データは完全にカプセル化されます。 ESP は単体で使用するか、または AH と組み合せて使用できます。 RFC 2406 を参照して下さい: IP Encapsulating Security Payload（ESP） 。

ハッシュか。任意 の 長さの入力メッセージを奪取 し、固定長ダイジェストを生成 するこれは 1 方法機能です。 Cisco では、Secure Hash Algorithm（SHA）と Message Digest 5（MD5）の両方のハッシュを、IPSec フレームワークの実装内部で使用しています。 詳細は、HMAC の定義を参照してください。

HMAC か。これは暗号使用が SHA および MD5 のようなハッシュする通報 認証のためのメカニズムです。 HMAC の 徹底的な議論のための RFC 2104 を参照して下さい。

インターネット キー エクスチェンジ（IKE）か。一部 Oakley および別のプロトコル スイートの一部を使用するハイブリッド プロトコルは Internet Security Association and Key Management Protocol（ISAKMP）フレームワークの中の SKEME を呼出しました。 IKE は、共有セキュリティ ポリシーと、鍵を必要とする IPSec などのサービスのための認証鍵を確立するために使用されます。 IPSec トラフィックを通過させる前に、各ルータ/ファイアウォール/ホストで相手ピアのアイデンティティを検証できる必要があります。 これは、両方のホストで事前共有鍵を手動で入力するか、または CA サービスによって、あるいは将来的にはセキュア DNS（DNSSec）によって可能になります。 これは以前 ISAKMP/Oakley として知られているプロトコルで RFC 2409 で定義されます: インターネット キー エクスチェンジ（IKE） 。 ISAKMP と IKE はどちらも Cisco IOS ソフトウェアでは同じことを指す略語として使用されているため、混乱の元になっている場合があります。 この 2 つはやや異なるものです。

Internet Security Association and Key Management Protocol（ISAKMP）か。これはセキュリティ ポリシーの鍵交換プロトコルおよびネゴシエーションの実装の機械工を定義するプロトコル フレームワークです。 ISAKMP は Internet Security Association and Key Management Protocol（ISAKMP）の中で定義されています。

IPsec NAT 透過か。IPsec NAT 透過 機能は NAT と IPsec の間で多くの既知非互換性を当たることによってネットワークのネットワーク アドレス変換（NAT）または Point Address Translation（PAT）ポイントを移動するために IP Security（IPSec）トラフィックのためのサポートを導入します。 NAT トラバーサルは、VPN デバイスによって自動的に検出される機能です。 Cisco IOS ソフトウェア リリース 12.2(13)T 以降が稼働しているルータでは、設定手順は不要です。 双方の VPN デバイスが NAT-T に対応している場合、NAT トラバーサルは自動的に検出され、ネゴシエートされます。

ISAKMP/Oakley か。IKE を参照してください。

MD5（MD5)?This は 128-bit ハッシュを生成 する 1 つの方法ハッシュアルゴリズムです。 MD5

と Secure Hash Algorithm（SHA）はどちらも MD4 の派生形で、このハッシュ アルゴリズムのセキュリティを強化する設計になっています。 SHA は MD4 や MD5 よりも安全です。 Cisco では IPSec フレームワーク内部での認証にハッシュを使用しています。

Oakley か。これは認証された キー マテリアルを得る方法を定義する鍵交換プロトコルです。 Oakley の基礎となるメカニズムは Diffie-Hellman キー交換アルゴリズムです。 RFC 2412 の規格を見つけることができます: OAKLEY キー判断プロトコル 。

完全転送秘密（PFS）か。PFS はある特定の IPsec SA キーが他のどのシークレットからも得られなかったことを、確認します他のいくつかのキーのように。 つまり、ある攻撃者がある 1 つの鍵を突破したとしても、他のどの鍵も取得できないことが PFS によって保証されます。 PFS がイネーブルになっていないと、攻撃者が IKE SA の秘密鍵を突破し、IPSec で保護されたデータをすべてコピーした後、IKE SA の秘密の情報を使用して、この IKE SA によって設定された IPSec SA を破ることができる可能性があります。 PFS がイネーブルになっていると、IKE が突破されたとしても攻撃者はすぐには IPSec にアクセスできません。 攻撃者は各 IPSec SA を個別に突破しなければならなくなります。 Cisco IOS の IPSec 実装では、PFS グループ 1（D-H 768 ビット）をデフォルトで使用しています。

Replay-detection か。これはレシーバがリプレイアタックを阻止するために古いですか重複パケットを拒否できるセキュリティ サービスです。 リプレイ攻撃では、攻撃者から受信者に古いパケットや重複パケットを送出することにより、受信者にこの不法トラフィックを正規のものと思い込ませることが利用されています。 リプレイ検出は、シーケンス番号と認証を組み合せて使用することにより実現します。リプレイ検出は IPSec の標準機能です。

RSA か。これは発明家の名にちなんで、可変変調長さの Rivest 名付けられる、公開キー 暗号化アルゴリズム シャミールおよび Adleman、です。 RSA には、DES などの一般的な秘密鍵アルゴリズムと比べて計算が非常に遅い欠点があります。 Cisco の IKE 実装では、デフィーヘルマン交換を使用して秘密鍵を取得しています。 この交換は RSA（または事前共有鍵）を使用して認証できます。 RSA の暗号化と署名の技術とは異なり、デフィーヘルマン交換では、DES 鍵が（暗号化形式でも）ネットワークを通過することはありません。 RSA はパブリックドメインではないので、RSA Data Security からライセンスを取得する必要があります。

Security Association（SA）か。これはデータフローに加えられるセキュリティポリシーおよびキーイングマテリアルの例です。 IKE と IPSec はどちらも SA を使用しますが、双方の SA は互いに独立しています。 IPSec SA は単方向であり、セキュリティ プロトコルごとに一意です。 保護されたデータ パイプでは 1 組の SA が必要であり、プロトコルごとに 1 方向当たり 1 つずつ必要です。 たとえば、ピア間で ESP をサポートしているパイプの場合、各方向について ESP SA が 1 つ必要です。 SA は、宛先（IPSec エンドポイント）アドレス、セキュリティ プロトコル（AH または ESP）、および Security Parameter Index（SPI; セキュリティ パラメータ インデックス）によって、一意に識別されます。

IKE は IPSec に代わって SA のネゴシエーションと確立を行います。 ユーザが IPSec SA を手動で確立することもできます。

IKE SA は IKE でだけ使用されます。 IPSec SA とは異なり、IKE SA は双方向です。

セキュアハッシュアルゴリズム（SHA）か。これは NIST によって置かれる 1 つの方法ハッシュです。 SHA は MD4 を厳密なモデルとしており、160 ビットのダイジェストを生成します。 SHA は 160 ビットのダイジェストを生成するため、MD5 などの 128 ビット ハッシュと比べてブルートフォース攻撃に対して耐性がありますが、速度は遅くなります。

分割トンネリングか。これはユーザがリモートオフィスでリソースにアクセスすることができると同時に、パブリックネットワークに、最も一般にアクセスすることをリモート VPN ユーザの

許可のプロセス インターネットです。 このネットワーク アクセスの方式により、ユーザは、パブリック ネットワーク（インターネット）にアクセスするのと同時に、ネットワーク プリンタやサーバなどのリモート デバイスにアクセスできます。 スプリット トンネリングを使用する長所は、インターネットのトラフィックが VPN サーバを通過する必要がないため、ボトルネックが緩和され、帯域幅が節約されることです。 この方式の短所は、パブリックのセキュアではないネットワークからのアクセスが可能なため、必然的に VPN が攻撃に対して脆弱になりがちである点です。

トランスフォームか。トランスフォームは対応した アルゴリズムとのセキュリティプロトコルを（AH か ESP）記述します。 たとえば、DES 暗号化アルゴリズムを使用した ESP や、認証のための HMAC-SHA などです。

トランスポート モードか。これは AH/ESP.トランスポート モードにおけるエンキャプシュレーションモード カプセル化します上位層ペイロードを、オリジナルIPデータグラムの伝送制御 プロトコル（TCP）または User Datagram Protocol （UDP; ユーザ データグラム プロトコル）のような、です。 このモードはピアが通信のエンドポイントであるときにだけ使用できます。 転送モードの逆はトンネル モードです。

トンネルモードか。これは IPsec のための完全な IPデータグラムのカプセル化です。 トンネル モードは、非 IPSec システム（Virtual Private Network（VPN; バーチャル プライベート ネットワーク）のシナリオなど）から発信されるデータグラムや非 IPSec システム宛てのデータグラムを保護するために使用します。

ISAKMP の設定

IKE の存在理由は、IPSec 用に SA を確立するためだけです。 IKE がこれを確立するためには、その前にピアとの間で SA（ISAKMP SA）の関係をネゴシエートする必要があります。 別々の設定文に複数のポリシー文を設定することが可能で、IKE が独自のポリシーでこれをネゴシエートすることにより、2 つのホストが合意することができます。 ISAKMP でネゴシエートされるのは次の項目です。

- **暗号化アルゴリズム**か。これは 56 ビット DES だけに制限されます。
- **ハッシュ Algorithm?MD5** か SHA
- **認証**か。RSA シグニチャ、RSA 暗号化された一時的な情報（乱数）、または事前共有キー
- **SA のライフタイム**か。単位は秒

現在、ISAKMP を設定する方法には、次の 2 つがあります。

1. 事前共有鍵を使用する（設定が簡単であるという利点）。
2. **CA** を使用する（企業全体への拡張が容易であるという利点）。

注IKE ネゴシエーションは UDP 500 で行われます。 IPSec では IP プロトコル 50 と 51 が使用されます。 ピア間にあるどのアクセス リストでも、これらが許可されていることを確認してください。

1. 事前共有キー

これは、IKE を設定するための「迅速ながらダーティな」方法です。 IKE の設定は簡単ですが、CA を使用しないために拡張性が低くなります。

IKE を設定するには、下記の手順が必要です。

- ISAKMP 保護スイートを設定する。
- ISAKMP 鍵を設定する。

ISAKMP 保護スイートの設定

次のコマンドを使用して、ISAKMP ポリシー オブジェクトを作成します。 複数のポリシーを設定することはできますが、この例では 1 つだけです。

```
dt3-45a(config)#crypto isakmp policy 1dt3-45a(config-isakmp)#
```

group コマンドを使用すると、デフィーヘルマンの計算に使用するモジュールのサイズを宣言できます。 グループ 1 の長さは 768 ビットで、グループ 2 の長さは 1024 ビットです。 どちらかを選んで使用する理由は何でしょうか。 グループ 2 はすべてのベンダーでサポートされているわけではありません。 さらに、グループ 2 の CPU 使用率はグループ 1 よりも顕著に高くなっています。 このため、Cisco 2500 シリーズ以下のようなローエンド ルータではグループ 2 の使用は適していません。 ところが、グループ 2 の方がグループ 1 よりも、よりセキュアです。 この例では Cisco 4500 が使用されていて、グループ 2 が使用されているため、ピアでもグループ 2 を使用するように設定されていることを確認します。 デフォルトはグループ 1 です。 デフォルトのプロパティを選択すると、**write terminal** コマンドを実行した際に、グループ 1 の行は表示されません。

```
dt3-45a(config-isakmp)#group 2
```

次の行では、ハッシュ アルゴリズムに MD5 を指定しています。 SHA と MD5 の実装はどちらも必須ですが、すべてのピアがこれらのどちらかでネゴシエートするように設定されているとは限りません。 Cisco IOS のデフォルトは SHA です。これは MD5 より安全です。

```
dt3-45a(config-isakmp)#hash md5
```

次のコマンドでは、SA のライフタイムが 500 秒に設定されています。 ライフタイムを設定しない場合は、デフォルトの 86400 秒、または 1 日になります。 ライフタイムのタイマーが切れると、SA はセキュリティ対策として再度ネゴシエートされます。

```
dt3-45a(config-isakmp)#lifetime 500
```

次のコマンドでは、IKE に対して使用する鍵を指定しています。 そのため、**pre-share** コマンドが使用されます。 **pre-share** コマンド以外の 2 つのオプションは、**rsa-encr** コマンドと **rsa-sig** コマンドです。 **rsa-encr** コマンドは RSA 暗号化 nonce を設定し、**rsa-sig** コマンドは RSA シグニチャを設定します。 **rsa-encr** コマンドと **rsa-sig** コマンドは、「CA の使用」セクションで取り上げています。 ここでは、**rsa-sig** がデフォルトであることを憶えておいてください。

```
dt3-45a(config-isakmp)#authentication pre-share
```

ISAKMP 鍵の設定

次のコマンドでは、IKE に対して使用する鍵を指定しています。 ここでのピアは 192.168.10.38 ですが、ピアではコンフィギュレーションに同一の鍵の「Slurpee-Machine（生成機）」が設定されている必要があります。

```
dt3-45a(config-isakmp)#exitdt3-45a(config)#crypto isakmp key Slurpee-Machine address 192.168.10.38
```

これで IKE が設定されました。 次に示す行はピアの IKE 設定です。 両方のルータの完全な設定は、このドキュメントの「設定例」セクションに示されています。

```
dt3-45a(config-isakmp)#exitdt3-45a(config)#crypto isakmp key Slurpee-Machine address 192.168.10.38
```

2. CA の使用

CA の使用は、IKE の設定に使用される複雑な方法です。 IPSec ではこれがきわめてスケーラブ

ルなので、classic crypto に代えて、IPSec を使用する必要があります。 Cisco IOS ソフトウェア リリース 11.3(3) がリリースされる際の、製品を出荷している CA ベンダーの数は限られています。 最初、ほとんどの設定は**事前共有**鍵を使用して行われます。 CA 製品を扱っているのは、VeriSign、Entrust、Microsoft および Netscape (およびおそらくは他のホストも) です。 この例では、VeriSign CA を使用しています。

CA を使用するには、下記の手順が必要です。

- ルータ用の RSA 鍵ペアを作成する。
- CA の証明書を要求する。
- クライアント ルータ用の証明書を登録する。
- ISAKMP 保護スイートを設定する。

ルータ用の RSA 鍵ペアの作成

crypto key gen rsa usage-keys コマンドには、わかりにくい部分があります。 このコマンドにより、RSA 用の鍵ペアが 2 つ作成されます。

- 暗号化用の 1 つの鍵ペア
- デジタル署名用の 1 つの鍵ペア

1 つの鍵ペアは公開鍵と、それに対応する秘密鍵です。 コマンドの最後に「usage-keys」を指定しない場合、ルータでは RSA 鍵ペアが 1 つだけ生成され、それが暗号化とデジタル署名の両方に使用されます。 警告ですが、このコマンドは DSS 鍵を作成するために使用される場合があります。 ところが、DSS は classic crypto の一部であり、IPSec の一部ではありません。

```
dt3-45a(config)#crypto key gen rsa usage-keysThe name for the keys will be: dt3-45a.cisco.com%You already
have RSA keys defined for dt3-45a.cisco.com.%Do you really want to replace them? [yes/no] yes
```

すでにこのルータには使用できる RSA 鍵があるため、既存の鍵を削除するかどうかが問い合せられます。 この回答は「はい」なので、コマンドを確認します。 次のプロンプトが返されます。

```
dt3-45a(config)#crypto key gen rsa usage-keysThe name for the keys will be: dt3-45a.cisco.com%You already
have RSA keys defined for dt3-45a.cisco.com.%Do you really want to replace them? [yes/no] yes
```

これでデフォルトの 512 ビット モジュラスの RSA 鍵ペアが作成されました。 コンフィギュレーション モードを抜けて、**show crypto key mypubkey rsa** コマンドを入力します。 これで、RSA 公開鍵が表示されます。 鍵ペアの秘密鍵部分が表示されることはありません。 既存の鍵がない場合でも、上記と同じ表示になります。

注キーペアが生成されたら、必ず設定を保存してください。

CA の証明書の要求

ここで、CA と通信を行うようにルータを設定する必要があります。 これには次のステップが必要です。 結論としては、CA 管理者との共同作業が必要です。

下記の設定行で、ドメイン名がルータに付加されます。 これにより、ホスト名 **ciscoca-ultra** が作成され、ルータに対して、その IP アドレスとネーム サーバが通知されます。 ルータで使用する CA または DNS に対して定義されたホスト名のいずれかが必要です。 Cisco では、ルータで DNS を使用することを推奨いたします。

```
dt3-45a(config)#ip host ciscoca-ultra 171.69.54.46dt3-45a(config)#ip domain-name cisco.comdt3-
45a(config)#ip name-server 171.692.132dt3-45a(config)#ip name-server 198.92.30.32
```

CA パラメータを設定し始めて下さい。 **verisign カリフォルニアは**ちょうど任意 の 名前です。

```
dt3-45a(config)#crypto ca identity verisign-cadt3-45a(ca-identity)#
```

次に示すように、Cisco の登録プロトコルでは CA との通信に HTTP が使用されます。 **dt3-45a(ca-identity)#enrollment url http://ciscoca-ultra** コマンドでは、CA との相互対話を行うために、指定された URL に移動することをルータに指示しています。 **dt3-45a(ca-identity)#crypto ca authenticate verisign-ca** コマンドでは、CA 自身の証明書を取得するようにルータに指示しています。 CA で登録できる前に信頼性を確認するために CA 管理者との CA の認証を確認するように実質 CA.に話確かめる必要があります。

```
dt3-45a(ca-identity)#enrollment url http://ciscoca-ultra dt3-45a(ca-identity)#exitdt3-45a(ca-identity)#crypto ca authenticate verisign-ca
```

クライアント ルータ用の証明書の登録

CA での登録を開始するには、**crypto ca enroll verisign-ca** コマンドを発行します。 これにはいくつかの手順ががあります。 まず、CA の ID を検証します。次に CA がルータの ID を検証します。 有効期限が切れる前に証明書を破棄する必要がある場合は (ルータのインターフェイスの番号を付け替えた場合や、証明書が侵害されたと思われる場合など) 、CA 管理者にパスワードを提出する必要があります。 次に示すように、パスワードを入力します。 パスワードを入力すると、ルータは次の手順に進みます。

```
dt3-45a(config)#crypto ca enroll verisign-ca%Start certificate enrollment ..%Create a challenge password. You will need to verbally provide this password to the CA Administrator in order to revoke your certificate.  For security reasons your password will not be saved in the configuration.  Please make a note of it.Password:Re-enter password:
```

今 CA.からのフィンガープリントがフィンガープリントが CA 管理者と正しいことを確認するのを見ます。 また、**show crypto ca cert** コマンドを実行すると、自分の証明書のほかに CA の証明書も表示されます。 この時点では、CA の証明書は「pending」としてリストに入れられています。

```
dt3-45a(config)#crypto ca enroll verisign-ca%Start certificate enrollment ..%Create a challenge password. You will need to verbally provide this password to the CA Administrator in order to revoke your certificate.  For security reasons your password will not be saved in the configuration.  Please make a note of it.Password:Re-enter password:
```

CA 管理者は証明書を発行する前にホストの ID を確認する必要があるので、次は CA 管理者に連絡を取ります。 CA によって証明書が発行されると、証明書のステータスが「pending」から「available」に変わります。 これで、CA への登録が完了します。 ただし、作業はまだ完了していません。 次は ISAKMP ポリシー オブジェクトを設定する必要があります。

ISAKMP 保護スイートの設定

次の出力では、**rsa-sig** のデフォルトが使用されています。 複数の保護スイートを使用することができますが、この例で使用しているのは 1 つだけです。 複数の保護スイートのイベントでは、ポリシーが番号順にピアに提示され、ピアは使用するポリシーをネゴシエートします。 すべてのピアで特定の機能がサポートされているわけではないことがわかっている場合は、このようにする必要があります。 これにより、ルータが無意味な事柄をネゴシエートをすることがなくなります。 たとえば、ポリシーに **rsa-sig** が設定されていて証明書がない場合、ルータはこれをネゴシエートすることはありません。

```
dt3-45a(config)#crypto isakmp policy 1dt3-45a(config-isakmp)#hash md5dt3-45a(config-isakmp)#lifetime 4000dt3-45a(config-isakmp)#exit
```

IPSec の設定

事前共有キーを使用する場合も、CA を設定する場合も、インターネット鍵交換 IKE を設定した場合は、IPSec も設定する必要があります。 どの IKE 方式を使用する場合でも、IPSec の設定手

順は同じです。

IPSec を設定するには、下記の手順が必要です。

- 拡張 ACL の作成
- IPSec トランスフォームの作成
- クリプト マップの作成
- インターフェイスへのクリプト マップの適用

拡張 ACL の作成

次のコマンドは、ルータが互いに通信することを許可するための非常に簡単な ACL です（たとえば、1 つのルータから次のルータへの Telnet など）。

```
dt3-45a(config)#access-list 101 permit ip host 192.168.10.38                host 192.168.10.66
```

より現実的な ACL は、次のコマンドのようになります。 このコマンドは、一般的な拡張 ACL です。192.168.3.0 は、対象のルータの背後のサブネットで、10.3.2.0 はピア ルータの背後のどこかにあるサブネットです。 **permit** は暗号化することを意味し、**deny** は暗号化を行わないことを意味することを憶えておいてください。

```
dt3-45a(config)#access-list 101 permit ip 192.168.3.0 0.0.0.255                10.3.2.0 0.0.0.255
```

IPSec トランスフォームの作成

トランスフォーム セットを 3 つ作成します。 最初のものは ESP だけを使用し、2 番目のものは ESP と組み合わせた AH を使用し、最後のものは AH だけを使用します。 IPSec SA ネゴシエーションの間、これら 3 つはすべてピアに提示され、ピアによって 1 つが選択されます。 また、これら 3 つのトランスフォームセットには、デフォルトの**トンネル モード**を使用します。 トランスポート モードは、暗号化のエンドポイントが通信のエンドポイントでもある場合にだけ使用できます。 トランスポート モードは、トランスフォームセット設定で **mode transport** コマンドを実行することによって指定できます。 トンネル モードは、主に VPN シナリオで使用されます。 また、**esp-rfc1829** および **ah-rfc1828** は、この技術に関する元の RFC に基づくもので、後方互換性のために残されている廃止されたトランスフォームです。 これらのトランスフォームをサポートしていないベンダーもありますが、これらのトランスフォームだけをサポートしているベンダーもあります。

次のコマンド内のトランスフォーム セットが最も実用的なものであるとは限りません。 たとえば、「PapaBear」と「BabyBear」には、両方とも標準外のトランスフォームセットが含まれています。 **esp-rfc1829** と **ah-rfc1828** は、両方とも同一のトランスフォームセットで使用する必要があります。

```
dt3-45a(config)#crypto ipsec transform-set PapaBear esp-rfc1829 dt3-45a(cfg-crypto-trans)#exit dt3-45a(config)#crypto ipsec transform-set MamaBear ah-md5-hmac esp-des dt3-45a(cfg-crypto-trans)#exit dt3-45a(config)#crypto ipsec transform-set BabyBear ah-rfc1828 dt3-45a(cfg-crypto-trans)#exit dt3-45a(config)#
```

暗号化マップの作成

ipsec-isakmp タグを使用することで、このクリプト マップが IPSec クリプト マップであることがルータに示されています。 このクリプト マップでは 1 つのピアだけが宣言されていますが、所定のクリプト マップ内に複数のピアを含めることができます。 **session key lifetime** は、次のコマンドで示すように、キロバイト（トラフィックが x キロバイトに達したら鍵を変更）、または秒で指定します。 この目的は、潜在的な攻撃者による攻撃を、より困難にすることにあります。 **set transform-set** コマンドによって、トランスフォームとクリプト マップが結合されます。 また

、トランスフォームを宣言する順序も重要です。 この設定では、「MamaBear」を最初に宣言し、残りは「BabyBear」まで優先順位を降順で進みます。 **match address 101** コマンドは、関連のあるトラフィックを判別するためにアクセス リスト 101 を使用するという意味です。 複数のクリプト マップに同一の名前（次の例では「armadillo」）と異なるシーケンス番号（次の例では「10」）を使用できます。 複数のクリプト マップと異なるシーケンス番号の組み合せによって、classic crypto と IPSec を組み合せることができます。 また、ここでは PFS 設定を修正することができます。 次の例のデフォルトは、PFS group1 です。 PFS を group2 に変更することができます。また、すべてオフにすることもできますが、これは避けるべきです。

```
dt3-45a(config)#crypto map armadillo 10 ipsec-isakmpdt3-45a(config-crypto-map)#set peer 192.168.10.38dt3-45a(config-crypto-map)#set session-key lifetime seconds 4000dt3-45a(config-crypto-map)#set transform-set MamaBear PapaBear BabyBeardt3-45a(config-crypto-map)#match address 101
```

インターフェイスへの暗号化マップの適用

次のコマンドにより、インターフェイスにクリプト マップが適用されます。 インターフェイスに適用できるクリプト マップ セットは 1 つだけです。 複数のクリプト マップ エントリに同じマップ名が付いていて、シーケンス番号が異なっている場合、それらは同じセットの一部となり、すべてインターフェイスに適用されます。 セキュリティ アプライアンスでは、最もシーケンス番号の低い **crypto map** エントリが最初に評価されます。

```
dt3-45a(config)#interface e0dt3-45a(config-if)#crypto map armadillo
```

メモリおよびCPU 考察

IPSec によって処理されるパケットでは、classic crypto によって処理されるパケットよりも速度が低下します。 これには複数の理由がありますが、パフォーマンスに重大な問題を引き起こす可能性があります。

1. IPSec にはパケット拡張が取り入れられているが、これによって IPSec データグラムのフラグメンテーションと、それにともなうリアセンブリが必要になる可能性が高い。
2. 暗号化されたパケットは認証手続きを経る可能性が高いため、各パケットについて暗号化操作が 2 回行われることになる。
3. 認証アルゴリズムの処理速度が遅い（デフィーヘルマン計算のような高速化対策は取られています）。

さらに、IKE で使用される Diffie-Hellmanキー交換は大きな 番号の非常に累乗（768 のおよび 1024 バイト間で）で、Cisco 2500 の 4 秒程かかることができます。 RSA のパフォーマンスは RSA キーペアのために選択される素数のサイズに依存しています。

SA データベースはルータごとに約 300 バイトを使用し、これに加えて SA ごとに 120 バイトを使用します。 着信に 1 つ送信に 1 つで IPSec SA が 2 つある状況では、540 バイトが必要です。これは、ほとんどの場合にあてはまります。 IKE SA のエントリは、それぞれ約 64 バイトです。データフローに IPSec SA が 1 つしかない状況とは、通信が単方向の場合だけです。

IPSec と IKE がアクティブになっていると、パフォーマンスに影響します。 デッフィーヘルマン鍵交換、公開鍵認証、および、暗号化/復号化により多大なリソースが消費されます。 一方で、この影響を少なくするために多大な努力が払われています。

暗号化を実行するインターフェイスを通過する非暗号化パケットでも、小さいながらパフォーマンス上の劣化があります。 すべてのパケットをクリプト マップでチェックする必要があることが、この理由です。 暗号化を実行するインターフェイスを避けてルータを通過するパケットによるパフォーマンス上の影響はありません。 最も大きな影響を受けるのは、暗号化されたデータ フロー自体です。

クリプト サブシステムによるルータの他の部分への影響を最小限にとどめるには、IKE 内のデフィーヘルマン鍵交換にグループ 1 を使用し、MD5 をハッシュ アルゴリズムとして、より長いライフタイムを指定します。 このパフォーマンス チューニングを行うと、その代わりに暗号化の強度が弱くなります。 究極的には、使用する機能と使用しない機能と判別するためのユーザのセキュリティ ポリシーに依存します。

show コマンドの出力

注このセクションのキャプチャは、このドキュメントの前のセクションで使用されたテストとは異なるシリーズのテストで取得されたものです。 そのため、これらのキャプチャでは異なる IP アドレスが使用されており、設定もいくらか異なっています。 **show** コマンドの別のシリーズは、このドキュメントの「デバッグ情報」セクションに記載されています。

IKE 関連の出力

VeriSign CA 登録をチェックするために、次のコマンドを学習します。 これらのコマンドでは、RSA 暗号化およびシグネチャに使用している公開鍵が示されます。

```
dt1-45a#show crypto key mypubkey rsa% Key pair was generated at: 11:31:59 PDT Apr 9 1998Key name: dt1-
45a.cisco.com Usage: Signature Key Key Data:  305C300D 06092A86 4886F70D 01010105 00034B00 30480241
00C1185439A9C75C  4E34C987 B4D7F36C A058D697 13172767 192166E1 661483DD 0FDB907BF9C10B7A  CB5A034F
A41DF385 23BEB6A7 C14344BE E6915A12 1C86374F 83020301 0001% Key pair was generated at: 11:32:02 PDT Apr 9
1998Key name: dt1-45a.cisco.com Usage: Encryption Key Key Data:  305C300D 06092A86 4886F70D 01010105
00034B00 30480241 00DCF5AC360DD5A6C69704CF 47B2362D 65123BD4 424B6FF6 AD10C33E 89983D08 16F1EA583700BCF9
1EF17E71 5931A9FC 18D60D9A E0852DDD 3F25369C F09DFB75 05020301 0001
```

次のコマンドでは、ルータが認識する証明書が示されています。 **pending** ステータスの証明書が、承認のために CA に送信されています。

```
dt1-45a#show crypto ca certificatesCertificate  Subject Name    Name: dt1-45a.cisco.com    Serial Number:
01193485  Status: Available  Certificate Serial Number: 650534996414E2BE701F4EF3170EDFAD  Key Usage:
Signature CA Certificate  Status: Available  Certificate Serial Number: 3051DF7169BEE31B821DFE4B3A338E5F
Key Usage: Not SetCertificate  Subject Name    Name: dt1-45a.cisco.com    Serial Number: 01193485
Status: Available  Certificate Serial Number: 1e621faf3b9902bc5b49d0f99dc66d14  Key Usage: Encryption
```

次の出力には、ルータの公開鍵と、ルータがそれを学習した場所が示されています。

```
dt1-45a#show crypto key pubkey-chain rsaCodes: M - Manually configured, C - Extracted from certificate
Code Usage   IP-Address       NameC    Signing                 Cisco SystemsDevtestCISCOCA-ULTRAC
General 172.21.30.71     dt1-7ka.cisco.com
```

これは、ISAKMP (IKE) SA テーブルです。 ここでは、現在 SA が 172.21.30.71 と 172.21.30.70 の間に存在していることがわかります。 ピアには、このルータの出力と同じ状態の SA エントリが必要です。

```
dt1-7ka#show crypto isakmp sa    dst           src          state        conn-id   slot172.21.30.70
172.21.30.71   QM_IDLE          47      5
```

下記の行には、設定済みのポリシー オブジェクトを示されています。 この場合、デフォルトに加えてポリシー **1**、**2**、および **4** が使用されています。 このポリシーは、**1** を最優先として、順番にピアに対して提示されます。

```
dt1-45a#show crypto isakmp policyProtection suite of priority 1encryption algorithm:   DES - Data
Encryption Standard (56 bitkeys).hash algorithm:         Message Digest 5authentication method:  Rivest-
Shamir-Adleman SignatureDiffie-Hellman group:   #1 (768 bit)lifetime:               180 seconds, no
volume limitProtection suite of priority 2encryption algorithm:   DES - Data Encryption Standard (56 bit
keys).hash algorithm:         Secure Hash Standardauthentication method:  Pre-Shared KeyDiffie-Hellman
group:   #2 (1024 bit)lifetime:               180 seconds, no volume limitProtection suite of priority
4encryption algorithm:   DES - Data Encryption Standard (56 bit keys).hash algorithm:         Message
```

```
Digest 5authentication method:  Pre-Shared KeyDiffie-Hellman group:   #2 (1024 bit)lifetime:
180 seconds, no volume limitDefault protection suiteencryption algorithm:   DES - Data Encryption
Standard (56 bit keys).hash algorithm:         Secure Hash Standardauthentication method:  Rivest-Shamir-
Adleman SignatureDiffie-Hellman group:   #1 (768 bit)lifetime:               86400 seconds, no volume
limit
```

IPSEC 関連のSHOW コマンド

次のコマンドでは、クリプト マップ **ToOtherRouter**、ACL、および、このクリプト マップに適用されるトランスフォーム プロポーザル、ピア、および鍵のライフタイムが示されています。

```
S3-2513-2#show crypto mapCrypto Map "ToOtherRouter" 10 ipsec-isakmp        Peer = 192.168.1.1
Extended IP access list 101            access-list 101 permit ip                source: addr =
192.168.45.0/0.0.0.255                dest:   addr = 192.168.3.0/0.0.0.255        Connection Id = UNSET
(0 established,    0 failed)        Current peer: 192.168.1.1        Session key lifetime: 4608000
kilobytes/3600 seconds        PFS (Y/N): N        Transform proposals={ Elvis, Bubba, BarneyDino, }
```

次の設定では、上記の出力と同じルータを使用していますが、コマンドが異なっています。 ネゴシエートされる設定になっているすべてのトランスフォーム プロポーザルとデフォルトが示されています。

```
S3-2513-2#show crypto ipsec transform-setTransform proposal Elvis: { ah-sha-hmac  }   supported settings
= { Tunnel,  },   default settings = { Tunnel,  },   will negotiate = { Tunnel,  },    { esp-des  }
supported settings = { Tunnel,  },   default settings = { Tunnel,  },   will negotiate = { Tunnel,  },
Transform proposal Bubba: { ah-rfc1828  }   supported settings = { Tunnel,  },   default settings = {
Tunnel,  },   will negotiate = { Tunnel,  },    { esp-des esp-md5-hmac  }supported settings = { Tunnel,
},   default settings = { Tunnel,  },   will negotiate = { Tunnel,  }, Transform proposal BarneyDino: {
ah-md5-hmac  }   supported settings = { Tunnel,  },   default settings = { Tunnel,  },   will negotiate =
{ Tunnel,  },
```

このコマンドでは、このルータの現在の IPSec セキュリティ結合が示されています。 このルータでは、着信と発信の両方に単一の AH SA を使用しています。

```
S3-2513-2#show crypto ip sessionSession key lifetime: 4608000 kilobytes/3600 secondsS3-2513-2#show crypto
ipsec sa interface: Ethernet0    Crypto map tag: ToOtherRouter, local addr. 192.168.1.2    local  ident
(addr/mask/prot/port): (192.168.45.0/255.255.255.0/0/0)   remote ident (addr/mask/prot/port):
(192.168.3.0/255.255.255.0/0/0)   current_peer: 192.168.1.1     PERMIT, flags={origin_is_acl,}    #pkts
encaps: 0, #pkts encrypt: 0, #pkts digest 0    #pkts decaps: 0, #pkts decrypt: 0, #pkts verify 0    #send
errors 5, #recv errors 0      local crypto endpt.: 192.168.1.2, remote crypto endpt.: 192.168.1.1
path mtu 1500, media mtu 1500     current outbound spi: 25081A81     inbound esp sas:       inbound ah
sas:      spi: 0x1EE91DDC(518594012)        transform: ah-md5-hmac ,        in use settings ={Tunnel, }
slot: 0, conn id: 16, crypto map: ToOtherRouter        sa timing: remaining key lifetime (k/sec):
(4608000/3423)        replay detection support: Y       outbound esp sas:       outbound ah sas:
spi: 0x25081A81(621288065)        transform: ah-md5-hmac ,in use settings ={Tunnel, }        slot: 0,
conn id: 17, crypto map: ToOtherRouter        sa timing: remaining key lifetime (k/sec): (4608000/3424)
replay detection support: Y
```

設定例

次の設定では、**事前共有**鍵が使用されています。 「デバッグ情報」セクションに一覧表示されているデバッグ出力の作成には、このルータ設定が使用されています。 この設定では、「Source Router」の背後にある「X」という名前のネットワークが、「Peer Router」の背後にある「Y」という名前のネットワークと通信することが許可されています。 使用している Cisco IOS のバージョンについては、『Cisco IOS ソフトウェア』のドキュメントを調べるか、特定のコマンドについての詳細は、Command Lookup Tool（登録ユーザ専用）を使用してください。 このツールを使用すると、特定のコマンドについての詳細な説明や設定のガイドラインを検索できます。一部ツールについては、ゲスト登録のお客様にはアクセスできない場合がありますことを、ご了承ください。

ネットワーク図


```
S3-2513-2#show crypto ip sessionSession key lifetime: 4608000
```

```
interface: Ethernet0    Crypto map tag: ToOtherRouter, local
addr. 192.168.1.2    local  ident (addr/mask/prot/port):
(192.168.45.0/255.255.255.0/0/0)   remote ident
(addr/mask/prot/port): (192.168.3.0/255.255.255.0/0/0)
current_peer: 192.168.1.1     PERMIT, flags={origin_is_acl,}
#pkts encaps: 0, #pkts encrypt: 0, #pkts digest 0    #pkts
decaps: 0, #pkts decrypt: 0, #pkts verify 0    #send errors
5, #recv errors 0      local crypto endpt.: 192.168.1.2,
remote crypto endpt.: 192.168.1.1     path mtu 1500, media
mtu 1500     current outbound spi: 25081A81     inbound esp
sas:       inbound ah sas:      spi: 0x1EE91DDC(518594012)
transform: ah-md5-hmac ,         in use settings ={Tunnel, }
slot: 0, conn id: 16, crypto map: ToOtherRouter        sa
timing: remaining key lifetime (k/sec): (4608000/3423)
replay detection support: Y       outbound esp sas:
outbound ah sas:      spi: 0x25081A81(621288065)
transform: ah-md5-hmac ,in use settings ={Tunnel, }
slot: 0, conn id: 17, crypto map: ToOtherRouter        sa
timing: remaining key lifetime (k/sec): (4608000/3424)
replay detection support: Y
```

ピア ルータ

```
S3-2513-2#show crypto ip sessionSession key lifetime: 4608000
kilobytes/3600 secondsS3-2513-2#show crypto ipsec sa
interface: Ethernet0    Crypto map tag: ToOtherRouter, local
addr. 192.168.1.2    local  ident (addr/mask/prot/port):
(192.168.45.0/255.255.255.0/0/0)   remote ident
(addr/mask/prot/port): (192.168.3.0/255.255.255.0/0/0)
current_peer: 192.168.1.1     PERMIT, flags={origin_is_acl,}
#pkts encaps: 0, #pkts encrypt: 0, #pkts digest 0    #pkts
decaps: 0, #pkts decrypt: 0, #pkts verify 0    #send errors
5, #recv errors 0      local crypto endpt.: 192.168.1.2,
remote crypto endpt.: 192.168.1.1     path mtu 1500, media
mtu 1500     current outbound spi: 25081A81     inbound esp
sas:       inbound ah sas:      spi: 0x1EE91DDC(518594012)
transform: ah-md5-hmac ,         in use settings ={Tunnel, }
slot: 0, conn id: 16, crypto map: ToOtherRouter        sa
timing: remaining key lifetime (k/sec): (4608000/3423)
replay detection support: Y       outbound esp sas:
outbound ah sas:      spi: 0x25081A81(621288065)
transform: ah-md5-hmac ,in use settings ={Tunnel, }
slot: 0, conn id: 17, crypto map: ToOtherRouter        sa
timing: remaining key lifetime (k/sec): (4608000/3424)
replay detection support: Y
```

デバッグ情報

このセクションには、2 つのルータ間で行われる通常の IKE/IPSec セッションのデバッグ出力が記載されています。 設定は、この文書の「設定例」と同じです。 これらのルータは、事前共有キーを使用しています。 どちらのルータでも **debug crypto isakmp** コマンド、**debug crypto ipsec** コマンド、および **debug crypto engine** コマンドがイネーブルになっています。 これは、発信元ルータのイーサネット インターフェイスから、ピア ルータのイーサネット インターフェイスに対して (60.60.60.60 から 50.50.50.50) 拡張 PING を実行してテストされました。

注次のデバッグ出力例で青色で示されているのは、デバッグ出力の一部ではなく、動作の内容を理解するための説明です。

- 発信元ルータ
- IKE/IPSec ネゴシエーションの後の発信元ルータによる **show** コマンド出力

発信元ルータ

```
goss-e4-2513#show clockgoss-e4-2513#pingProtocol [ip]: Target
IP address: 50.50.50.50Repeat count [5]: 10Datagram size
[100]: Timeout in seconds [2]: Extended commands [n]: ySource
address or interface: 60.60.60.60Type of service [0]: Set DF
bit in IP header? [no]: Validate reply data? [no]: Data
pattern [0xABCD]: Loose, Strict, Record, Timestamp,
Verbose[none]: Sweep range of sizes [n]: Type escape sequence
to abort.Sending 10, 100-byte ICMP Echos to 50.50.50.50,
timeout is 2 seconds:Apr  2 12:03:55.347: IPSEC(sa_request):
,  (key eng. msg.) src= 20.20.20.20, dest= 20.20.20.21,
src_proxy= 60.60.60.0/255.255.255.0/0/0 (type=4),
dest_proxy= 50.50.50.0/255.255.255.0/0/0 (type=4),
protocol= ESP, transform= esp-rfc1829 ,     lifedur= 190s and
4608000kb,     spi= 0x0(0), conn_id= 0, keysize= 0, flags=
0x4004Apr  2 12:03:55.355: IPSEC(sa_request): ,  (key eng.
msg.) src= 20.20.20.20, dest= 20.20.20.21,     src_proxy=
60.60.60.0/255.255.255.0/0/0 (type=4),     dest_proxy=
50.50.50.0/255.255.255.0/0/0 (type=4),    protocol= AH,
transform= ah-md5-hmac ,     lifedur= 190s and 4608000kb,
spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4004Apr  2
12:03:55.363: IPSEC(sa_request): ,  (key eng. msg.) src=
20.20.20.20, dest= 20.20.20.21,     src_proxy=
60.60.60.0/255.255.255.0/0/0 (type=4),     dest_proxy=
50.50.50.0/255.255.255.0/0/0 (type=4),    protocol= ESP,
transform= esp-des ,     lifedur= 190s and 4608000kb,
spi= 0x0(0), conn_id= 0, keysi.ze= 0, flags= 0x4004Apr  2
12:03:55.375: IPSEC(sa_request): ,  (key eng. msg.) src=
20.20.20.20, dest= 20.20.20.21,     src_proxy=
60.60.60.0/255.255.255.0/0/0 (type=4),     dest_proxy=
50.50.50.0/255.255.255.0/0/0 (type=4),    protocol= AH,
transform= ah-rfc1828 ,     lifedur= 190s and 4608000kb,
spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4004!--- Note
that the router offers to the peer all of the !--- available
transforms.Apr 2 12:03:55.391: ISAKMP (14): beginning Main
Mode exchangeApr 2 12:03:57.199: ISAKMP (14): processing SA
payload. message ID = 0Apr 2 12:03:57.203: ISAKMP (14):
Checking ISAKMP transform 1 against priority 1 policyApr 2
12:03:57.203: ISAKMP: encryption DES-CBCApr 2 12:03:57.207:
ISAKMP: hash MD5Apr 2 12:03:57.207: ISAKMP: default group
1Apr 2 12:03:57.207: ISAKMP: auth pre-shareApr 2
12:03:57.211: ISAKMP (14): atts are acceptable. Next payload
is 0Apr 2 12:03:57.215: Crypto engine 0: generate alg
paramApr 2 12:03:5.8.867: CRYPTO_ENGINE: Dh phase 1 status:
0Apr 2 12:03:58.871: ISAKMP (14): SA is doing pre-shared key
authentication..Apr 2 12:04:01.291: ISAKMP (14): processing
KE payload. message ID = 0Apr 2 12:04:01.295: Crypto engine
0: generate alg paramApr 2 12:04:03.343: ISAKMP (14):
processing NONCE payload. message ID = 0Apr 2 12:04:03.347:
Crypto engine 0: create ISAKMP SKEYID for conn id 14Apr 2
12:04:03.363: ISAKMP (14): SKEYID state generatedApr 2
12:04:03.367: ISAKMP (14): processing vendor id payloadApr 2
12:04:03.371: ISAKMP (14): speaking to another IOS box!Apr 2
12:04:03.371: generate hmac context for conn id 14Apr 2
12:04:03.615: ISAKMP (14): processing ID payload. message ID
= 0Apr 2 12:04:03.615: ISAKMP (14): processing HASH payload.
message ID = 0Apr 2 12:04:03.619: generate hmac context for
conn id 14Apr 2 12:04:03.627: ISAKMP (14): SA has been
authenticatedApr 2 12:04:03.627: ISAKMP (14): beginning Quick
```

```
Mode exchange, M-ID of 1628162439!--- These lines represent
verification that the policy !--- attributes are fine, and
the final authentication of the IKE SA. !--- Once the IKE SA
is authenticated, a valid IKE SA exists. !--- New IKE kicks
off IPsec negotiation:Apr 2 12:04:03.635: IPSEC(key_engine):
got a queue event...Apr 2 12:04:03.635: IPSEC(spi_response):
getting spi 3035648241d for SA .!!!from 20.20.20.21 to
20.20.20.20 for prot 3Apr 2 12:04:03.639:
IPSEC(spi_response): getting spi 4239562801d for SA from
20.20.20.21 to 20.20.20.20 for prot 2Apr 2 12:04:03.643:
IPSEC(spi_response): getting spi 4153056211d for SA from
20.20.20.21 to 20.20.20.20 for prot 3Apr 2 12:04:03.647:
IPSEC(spi_response): getting spi 2183089761d for SA from
20.20.20.21 to 20.20.20.20 for prot 2Apr 2 12:04:03.891:
generate hmac context for conn id 14Apr 2 12:04:04.!!Success
rate is 50 percent (5/10), round-trip min/avg/max =
264/265/268 msgoss-e4-2513#723: generate hmac context for
conn id 14Apr 2 12:04:04.731: ISAKMP (14): processing SA
payload. message ID = 1628162439Apr 2 12:04:04.731: ISAKMP
(14): Checking IPSec proposal 1Apr 2 12:04:04.735: ISAKMP:
transform 1, ESP_DES_IV64Apr 2 12:04:04.735: ISAKMP:
attributes in transform:Apr 2 12:04:04.735: ISAKMP: encaps is
1Apr 2 12:04:04.739: ISAKMP: SA life type in secondsApr 2
12:04:04.739: ISAKMP: SA life duration (basic) of 190Apr 2
12:04:04.739: ISAKMP: SA life type in kilobytesApr 2
12:04:04.743: ISAKMP: SA life duration (VPI) of 0x0 0x46 0x50
0x0 Apr 2 12:04:04.747: ISAKMP (14): atts are acceptable.!---
The ISAKMP debug is listed because IKE is the !--- entity
that negotiates IPsec SAs on behalf of IPsec.Apr 2
12:04:04.747: IPSEC(validate_proposal_request): proposal part
#1, (key eng. msg.) dest= 20.20.20.21, src= 20.20.20.20,
dest_proxy= 50.50.50.0/255.255.255.0/0/0 (type=4), src_proxy=
60.60.60.0/255.255.255.0/0/0 (type=4), protocol= ESP,
transform= esp-rfc1829 , lifedur= 0s and 0kb, spi= 0x0(0),
conn_id= 0, keysize= 0, flags= 0x4Apr 2 12:04:04.759: ISAKMP
(14): processing NONCE payload. message ID = 1628162439Apr 2
12:04:04.759: ISAKMP (14): processing ID payload. message ID
= 1628162439Apr 2 12:04:04.763: ISAKMP (14): processing ID
payload. message ID = 1628162439Apr 2 12:04:04.767: generate
hmac context for conn id 14Apr 2 12:04:04.799: ISAKMP (14):
Creating IPSec SAsApr 2 12:04:04.803: inbound SA from
20.20.20.21 to 20.20.20.20 (proxy 50.50.50.0 to
60.60.60.0)Apr 2 12:04:04.803: has spi 303564824 and conn_id
15 and flags 4Apr 2 12:04:04.807: lifetime of 190 secondsApr
2 12:04:04.807: lifetime of 4608000 kilobytesApr 2
12:04:04.811: outbound SA from 20.20.20.20 to 20.20.20.21
(proxy 60.60.60.0 to 50.50.50.0)Apr 2 12:04:04.811: has spi
183903875 and conn_id 16 and flags 4Apr 2 12:04:04.815:
lifetime of 190 secondsApr 2 12:04:04.815: lifetime of
4608000 kilobytesApr 2 12:04:04.823: IPSEC(key_engine): got a
queue event...Apr 2 12:04:04.823: IPSEC(initialize_sas): ,
(key eng. msg.) dest= 20.20.20.20, src= 20.20.20.21,
dest_proxy= 60.60.60.0/255.255.255.0/0/0 (type=4), src_proxy=
50.50.50.0/255.255.255.0/0/0 (type=4), protocol= ESP,
transform= esp-rfc1829 , lifedur= 190s and 4608000kb, spi=
0x12180818(303564824), conn_id= 15, keysize= 0, flags= 0x4Apr
2 12:04:04.831: IPSEC(initialize_sas): , (key eng. msg.) src=
20.20.20.20, dest= 20.20.20.21, src_proxy=
60.60.60.0/255.255.255.0/0/0 (type=4), dest_proxy=
50.50.50.0/255.255.255.0/0/0 (type=4), protocol= ESP,
transform= esp-rfc1829 , lifedur= 190s and 4608000kb, spi=
0xAF62683(183903875), conn_id= 16, keysize= 0, flags= 0x4Apr
2 12:04:04.839: IPSEC(create_sa): sa created, (sa) sa_dest=
```

```
20.20.20.20, sa_prot= 50, sa_spi= 0x12180818(303564824),
sa_trans= esp-rfc1829 , sa_conn_id= 15Apr 2 12:04:04.843:
IPSEC(create_sa): sa created, (sa) sa_dest= 20.20.20.21,
sa_prot= 50, sa_spi= 0xAF62683(183903875), sa_trans= esp-
rfc1829 , sa_conn_id= 16!--- These lines show that IPsec SAs
are created and !--- encrypted traffic can now pass.
```

IKE/IPSec ネゴシエーションの後の発信元ルータによる show コマンド出力

```
goss-e4-2513#goss-e4-2513#show crypto isakmp sa    dst
src          state        conn-id   slot20.20.20.21
20.20.20.20    QM_IDLE           14      0goss-e4-2513#show
crypto ipsec sainterface: Serial0    Crypto map tag:
armadillo, local addr. 20.20.20.20   local  ident
(addr/mask/prot/port): (60.60.60.0/255.255.255.0/0/0)
remote ident (addr/mask/prot/port):
(50.50.50.0/255.255.255.0/0/0)   current_peer: 20.20.20.21
PERMIT, flags={origin_is_acl,}    #pkts encaps: 5, #pkts
encrypt: 5, #pkts digest 0    #pkts decaps: 5, #pkts decrypt:
5, #pkts verify 0    #send errors 5, #recv errors 0     local
crypto endpt.: 20.20.20.20, remote crypto endpt.: 20.20.20.21
path mtu 1500, media mtu 1500     current outbound spi:
AF62683     inbound esp sas:      spi: 0x12180818(303564824)
transform: esp-rfc1829 ,        in use settings ={Var len IV,
Tunnel, }        slot: 0, conn id: 15, crypto map: armadillo
sa timing: remaining key lifetime (k/sec): (4607999/135)
IV size: 8 bytes        replay detection support: N
inbound ah sas:     outbound esp sas:      spi:
0xAF62683(183903875)        transform: esp-rfc1829 ,
in use settings ={Var len IV, Tunnel, }        slot: 0, conn
id: 16, crypto map: armadillo        sa timing: remaining key
lifetime (k/sec): (4607999/117)        IV size: 8 bytes
replay detection support: N     outbound ah sas:goss-e4-
2513#show crypto isakmp policyProtection suite of priority 1
encryption algorithm:   DES - Data Encryption Standard (56
bit keys).        hash algorithm:         Message Digest 5
authentication method:  Pre-Shared Key        Diffie-Hellman
group:   #1 (768 bit)        lifetime:               86400
seconds, no volume limitDefault protection suite
encryption algorithm:   DES - Data Encryption Standard (56
bit keys).        hash algorithm:         Secure Hash
Standard        authentication method:  Rivest-Shamir-Adleman
Signature        Diffie-Hellman group:   #1 (768 bit)
lifetime:               86400 seconds, no volume limitgoss-
e4-2513#show crypto mapCrypto Map "armadillo" 1 ipsec-isakmp
Peer = 20.20.20.21        Extended IP access list 101
access-list 101 permit ip 60.60.60.0 0.0.0.255 50.50.50.0
0.0.0.255        Current peer: 20.20.20.21        Security
association lifetime: 4608000 kilobytes/190 seconds
PFS (Y/N): N        Transform sets={ BearPapa, BearMama,
BearBaby, }
```

同一の PING シーケンスを使用したピア ルータを反対側から見た場合

```
goss-c2-2513#show debugCryptographic Subsystem:  Crypto
ISAKMP debugging is on  Crypto Engine debugging is on  Crypto
IPSEC debugging is ongoss-c2-2513#Apr  2 12:03:55.107: ISAKMP
(14): processing SA payload. message ID = 0Apr  2
12:03:55.111: ISAKMP (14): Checking ISAKMP transform 1
against      priority 1 policyApr  2 12:03:55.111: ISAKMP:
encryption DES-CBCApr  2 12:03:55.111: ISAKMP:      hash
MD5Apr  2 12:03:55.115: ISAKMP:      default group 1Apr  2
12:03:55.115: ISAKMP:      auth pre-shareApr  2 12:03:55.115:
```

```
ISAKMP (14): atts are acceptable. Next payload is 0!--- IKE
performs its operation, and then kicks off IPsec.Apr 2
12:03:55.119: Crypto engine 0: generate alg paramApr 2
12:03:56.707: CRYPTO_ENGINE: Dh phase 1 status: 0Apr 2
12:03:56.711: ISAKMP (14): SA is doing pre-shared key
authenticationApr 2 12:03:58.667: ISAKMP (14): processing KE
payload. message ID = 0Apr 2 12:03:58.671: Crypto engine 0:
generate alg paramApr 2 12:04:00.687: ISAKMP (14): processing
NONCE payload. message ID = 0Apr 2 12:04:00.695: Crypto
engine 0: create ISAKMP SKEYID for conn id 14Apr 2
12:04:00.707: ISAKMP (14): SKEYID state generatedApr 2
12:04:00.711: ISAKMP (14): processing vendor id payloadApr 2
12:04:00.715: ISAKMP (14): speaking to another IOS box!Apr 2
12:04:03.095: ISAKMP (14): processing ID payload. message ID
= 0Apr 2 12:04:03.095: ISAKMP (14): processing HASH payload.
message ID = 0Apr 2 12:04:03.099: generate hmac context for
conn id 14Apr 2 12:04:03.107: ISAKMP (14): SA has been
authenticatedApr 2 12:04:03.111: generate hmac context for
conn id 14Apr 2 12:04:03.835: generate hmac context for conn
id 14Apr 2 12:04:03.839: ISAKMP (14): processing SA payload.
message ID = 1628162439Apr 2 12:04:03.843: ISAKMP (14):
Checking IPSec proposal 1Apr 2 12:04:03.843: ISAKMP:
transform 1, ESP_DES_IV64Apr 2 12:04:03.847: ISAKMP:
attributes in transform:Apr 2 12:04:03.847: ISAKMP: encaps is
1Apr 2 12:04:03.847: ISAKMP: SA life type in secondsApr 2
12:04:03.851: ISAKMP: SA life duration (basic) of 190Apr 2
12:04:03.851: ISAKMP: SA life type in kilobytesApr 2
12:04:03.855: ISAKMP: SA life duration (VPI) of 0x0 0x46 0x50
0x0 Apr 2 12:04:03.855: ISAKMP (14): atts are acceptable.Apr
2 12:04:03.859: IPSEC(validate_proposal_request): proposal
part #1, (key eng. msg.) dest= 20.20.20.21, src= 20.20.20.20,
dest_proxy= 50.50.50.0/255.255.255.0/0/0 (type=4), src_proxy=
60.60.60.0/255.255.255.0/0/0 (type=4), protocol= ESP,
transform= esp-rfc1829 , lifedur= 0s and 0kb, spi= 0x0(0),
conn_id= 0, keysize= 0, flags= 0x4Apr 2 12:04:03.867: ISAKMP
(14): processing NONCE payload. message ID = 1628162439Apr 2
12:04:03.871: ISAKMP (14): processing ID payload. message ID
= 1628162439Apr 2 12:04:03.871: ISAKMP (14): processing ID
payload. message ID = 1628162439Apr 2 12:04:03.879:
IPSEC(key_engine): got a queue event...Apr 2 12:04:03.879:
IPSEC(spi_response): getting spi 183903875ld for SA from
20.20.20.20 to 20.20.20.21 for prot 3Apr 2 12:04:04.131:
generate hmac context for conn id 14Apr 2 12:04:04.547:
generate hmac context for conn id 14Apr 2 12:04:04.579:
ISAKMP (14): Creating IPSec SAsApr 2 12:04:04.579: inbound SA
from 20.20.20.20 to 20.20.20.21 (proxy 60.60.60.0 to
50.50.50.0)Apr 2 12:04:04.583: has spi 183903875 and conn_id
15 and flags 4Apr 2 12:04:04.583: lifetime of 190 secondsApr
2 12:04:04.587: lifetime of 4608000 kilobytesApr 2
12:04:04.587: outbound SA from 20.20.20.21 to 20.20.20.20
(proxy 50.50.50.0 to 60.60.60.0)Apr 2 12:04:04.591: has spi
303564824 and conn_id 16 and flags 4Apr 2 12:04:04.591:
lifetime of 190 secondsApr 2 12:04:04.595: lifetime of
4608000 kilobytesApr 2 12:04:04.599: IPSEC(key_engine): got a
queue event...Apr 2 12:04:04.599: IPSEC(initialize_sas): ,
(key eng. msg.) dest= 20.20.20.21, src= 20.20.20.20,
dest_proxy= 50.50.50.0/255.255.255.0/0/0 (type=4), src_proxy=
60.60.60.0/255.255.255.0/0/0 (type=4), protocol= ESP,
transform= esp-rfc1829 , lifedur= 190s and 4608000kb, spi=
0xAF62683(183903875), conn_id= 15, keysize= 0, flags= 0x4Apr
2 12:04:04.607: IPSEC(initialize_sas): , (key eng. msg.) src=
20.20.20.21, dest= 20.20.20.20, src_proxy=
50.50.50.0/255.255.255.0/0/0 (type=4), dest_proxy=
```

```
60.60.60.0/255.255.255.0/0/0 (type=4), protocol= ESP,
transform= esp-rfc1829 , lifedur= 190s and 4608000kb, spi=
0x12180818(303564824), conn_id= 16, keysize= 0, flags= 0x4Apr
2 12:04:04.615: IPSEC(create_sa): sa created, (sa) sa_dest=
20.20.20.21, sa_prot= 50, sa_spi= 0xAF62683(183903875),
sa_trans= esp-rfc1829 , sa_conn_id= 15Apr 2 12:04:04.619:
IPSEC(create_sa): sa created, (sa) sa_dest= 20.20.20.20,
sa_prot= 50, sa_spi= 0x12180818(303564824), sa_trans= esp-
rfc1829 , sa_conn_id= 16!--- The IPsec SAs are created, and
ICMP traffic can flow.
```

ピア ルータの show コマンド

```
!--- This illustrates a series of show command output after
!--- IKE/IPsec negotiation takes place.goss-c2-2513#show
crypto isakmp sa    dst            src           state
conn-id   slot20.20.20.21    20.20.20.20    QM_IDLE
14      0goss-c2-2513#show crypto ipsec sainterface: Serial0
Crypto map tag: armadillo, local addr. 20.20.20.21   local
ident (addr/mask/prot/port): (50.50.50.0/255.255.255.0/0/0)
remote ident (addr/mask/prot/port):
(60.60.60.0/255.255.255.0/0/0)   current_peer: 20.20.20.20
PERMIT, flags={origin_is_acl,}    #pkts encaps: 5, #pkts
encrypt: 5, #pkts digest 0    #pkts decaps: 5, #pkts decrypt:
5, #pkts verify 0    #send errors 0, #recv errors 0     local
crypto endpt.: 20.20.20.21, remote crypto endpt.: 20.20.20.20
path mtu 1500, media mtu 1500     current outbound spi:
12180818     inbound esp sas:      spi: 0xAF62683(183903875)
transform: esp-rfc1829 ,        in use settings ={Var len IV,
Tunnel, }        slot: 0, conn id: 15, crypto map: armadillo
sa timing: remaining key lifetime (k/sec): (4607999/118)
IV size: 8 bytes        replay detection support: N
inbound ah sas:     outbound esp sas:      spi:
0x12180818(303564824)        transform: esp-rfc1829 ,
in use settings ={Var len IV, Tunnel, }        slot: 0, conn
id: 16, crypto map: armadillo        sa timing: remaining key
lifetime (k/sec): (4607999/109)        IV size: 8 bytes
replay detection support: N     outbound ah sas:goss-c2-
2513#show crypto isakmp policyProtection suite of priority 1
encryption algorithm:   DES - Data Encryption Standard (56
bit keys).        hash algorithm:         Message Digest 5
authentication method:  Pre-Shared Key        Diffie-Hellman
group:   #1 (768 bit)        lifetime:               86400
seconds, no volume limitDefault protection suite
encryption algorithm:   DES - Data Encryption Standard (56
bit keys).        hash algorithm:         Secure Hash
Standard        authentication method:  Rivest-Shamir-Adleman
Signature        Diffie-Hellman group:   #1 (768 bit)
lifetime:               86400 seconds, no volume limitgoss-
c2-2513#show crypto mapCrypto Map "armadillo" 1 ipsec-isakmp
Peer = 20.20.20.20        Extended IP access list 101
access-list 101 permit ip 50.50.50.0 0.0.0.255 60.60.60.0
0.0.0.255        Current peer: 20.20.20.20        Security
association lifetime: 4608000 kilobytes/190 seconds
PFS (Y/N): N        Transform sets={ MamaBear, PapaBear,
BabyBear, }
```

IPSec の実装に関するヒント

ここでは、IPSec の実装に関するヒントを紹介しています。

- クリプトを設定する前に、通信のエンドポイント間が接続されていることを確認してくださ

い。
- ルータで DNS が動作していること、または CA ホスト名が入力されていること（CA を使用している場合）を確認してください。
- IPSec では IP プロトコルの 50 と 51 を使用し、IKE トラフィックはプロトコル 17、ポート 500（UDP 500）を通過します。 これらが適切に許可されていることを確認してください。
- ACL では、「any」という語は使用しないでください。 これを使用すると問題が発生します。 詳細は、『PIX コマンド リファレンス』に記載されている**アクセスリスト**に関する使用上のガイドラインを参照してください。
- 推奨するトランスフォームの組み合わせは、次のとおりです。*!--- This illustrates a series of* **show** command output after !--- IKE/IPsec negotiation takes place.goss-c2-2513#**show crypto isakmp sa** dst src state conn-id slot20.20.20.21 20.20.20.20 QM_IDLE 14 0goss-c2-2513#**show crypto ipsec sa**interface: Serial0 Crypto map tag: armadillo, local addr. 20.20.20.21 local ident (addr/mask/prot/port): (50.50.50.0/255.255.255.0/0/0) remote ident (addr/mask/prot/port): (60.60.60.0/255.255.255.0/0/0) current_peer: 20.20.20.20 PERMIT, flags={origin_is_acl,} #pkts encaps: 5, #pkts encrypt: 5, #pkts digest 0 #pkts decaps: 5, #pkts decrypt: 5, #pkts verify 0 #send errors 0, #recv errors 0 local crypto endpt.: 20.20.20.21, remote crypto endpt.: 20.20.20.20 path mtu 1500, media mtu 1500 current outbound spi: 12180818 inbound esp sas: spi: 0xAF62683(183903875) transform: esp-rfc1829 , in use settings ={Var len IV, Tunnel, } slot: 0, conn id: 15, crypto map: armadillo sa timing: remaining key lifetime (k/sec): (4607999/118) IV size: 8 bytes replay detection support: N inbound ah sas: outbound esp sas: spi: 0x12180818(303564824) transform: esp-rfc1829 , in use settings ={Var len IV, Tunnel, } slot: 0, conn id: 16, crypto map: armadillo sa timing: remaining key lifetime (k/sec): (4607999/109) IV size: 8 bytes replay detection support: N outbound ah sas:goss-c2-2513#**show crypto isakmp policy**Protection suite of priority 1 encryption algorithm: DES - Data Encryption Standard (56 bit keys). hash algorithm: Message Digest 5 authentication method: Pre-Shared Key Diffie-Hellman group: #1 (768 bit) lifetime: 86400 seconds, no volume limitDefault protection suite encryption algorithm: DES - Data Encryption Standard (56 bit keys). hash algorithm: Secure Hash Standard authentication method: Rivest-Shamir-Adleman Signature Diffie-Hellman group: #1 (768 bit) lifetime: 86400 seconds, no volume limitgoss-c2-2513#**show crypto map**Crypto Map "armadillo" 1 ipsec-isakmp Peer = 20.20.20.20 Extended IP access list 101 access-list 101 permit ip 50.50.50.0 0.0.0.255 60.60.60.0 0.0.0.255 Current peer: 20.20.20.20 Security association lifetime: 4608000 kilobytes/190 seconds PFS (Y/N): N Transform sets={ MamaBear, PapaBear, BabyBear, }
- AH は、認証ヘッダーであることに注意してください。 実際のユーザ データストリームは暗号化されていません。 データストリームの暗号化には、ESP が必要です。 AH だけを使用している場合は、ネットワーク上をクリアテキストが通過しても驚くにはあたりません。 AH を使用する場合は、ESP も使用します。 ESP では認証も実行できることに留意してください。 したがって、**esp-des** と **esp-sha-hmac** のようなトランスフォームの組み合せを使用できます。
- **ah-rfc1828** と **esp-rfc1829** は、古い IPSec 実装との後方互換性を保つために残されている、廃止されたトランスフォームです。 ピアが新しいトランスフォームをサポートしていない場合は、これらを使用してみてください。
- SHA は MD5 よりも低速ですがセキュアさでは勝り、MD5 は SHA よりも高速ですがセキュアさでは劣ります。 コミュニティによっては、MD5 の快適度は非常に低くなります。
- 疑問がある場合は、トンネル モードを使用してください。 デフォルトはトンネル モードになっており、トランスポート モードで使用することや、その VPN ケイパビリティにも使用できます。
- Cisco IOS ソフトウェア リリース 11.3 にアップグレードする classic crypto ユーザは、設定の crypto コマンドの保管方法が変更され、IPSec を使用できるようになりました。 したがって、classic crypto ユーザが Cisco IOS ソフトウェア リリース 11.2 に戻す場合は、クリプト設定を再度指定する必要があります。
- 設定が終了する際に暗号化リンクに PING テストを実行する場合は、ネゴシエーションのプ

ロセスに多少の時間がかかることがあります (Cisco 4500 では 6 秒、Cisco 2500 では 20 秒) 。これは、SA がまだネゴシエートされていないからです。 すべてが正しく設定されていても、最初は PING が失敗することがあります。 **debug crypto ipsec** コマンドおよび **debug crypto isakmp** コマンドを使用すると、現在の状況を確認できます。 暗号化データストリームの設定が終了すると、PING は正常に動作します。
- ネゴシエーションで問題が発生したり、設定変更を行った際は、**clear crypto is** コマンドや **clear crypto sa** コマンドを使用してデータベースをフラッシュしてから再試行してください。 これにより、途中で止まっている前のネゴシエーションを無効にして、新しくネゴシエーションを開始できます。 この方法では、**clear crypto is** コマンドと **clear cry sa** コマンドが非常に便利です。

ヘルプと関連情報のリンク

IPSec の情報

- **IPsec に関するサポート ページ**
- ECRA 暗号化ポリシーおよび手順か。export@cisco.com に E メールを送信 して下さい

IPSec のその他の設定例

- Cisco のネットワークレイヤ暗号化の設定およびトラブルシューティング: IPSec と ISAKMP
- IPSec ネットワーク セキュリティの概要
- PIX ファイアウォールでの IPSec 設定に関するドキュメントPIX 5.1PIX 5.2PIX 5.3PIX 6.0PIX 6.1PIX 6.2PIX 6.3

IPSec に関してさらに支援が必要な場合は、Cisco のテクニカルサポートに、電話で (800) 553-24HR、(408) 526-7209 か、または E メールで tac@cisco.com にお問い合せください。注：弊社とのサポート契約がないお客様は、製品をご購入いただきました販売店経由でお問い合わせください。

参考資料

Harkins, D. *ISAKMP/Oakley Protocol Feature Software Unit Functional Specification.* ENG-0000 Rev. A.シスコシステムズ。

Madson、C. *IPSec ソフトウェアユニットの機能仕様 ENG-17610 Rev. F.*シスコシステムズ。

Kaufman, C. Perlman R. and Spencer, M. *Network Security: Private Communication in a Public World.* Prentice Hall, 1995.

Schneier, B. *Applied Cryptography: Protocols, Algorithms, and Source Code in C.* 第 2 ED。 John Wiley & Sons， Inc.

『Various IETF IP Security working-drafts 』

関連情報

- **IPsec に関するサポート ページ**

- バーチャル プライベート ネットワークの動作のしくみ
- 一般的な L2L およびリモート アクセス IPSec VPN のトラブルシューティング方法について
- テクニカルサポートとドキュメント - Cisco Systems