

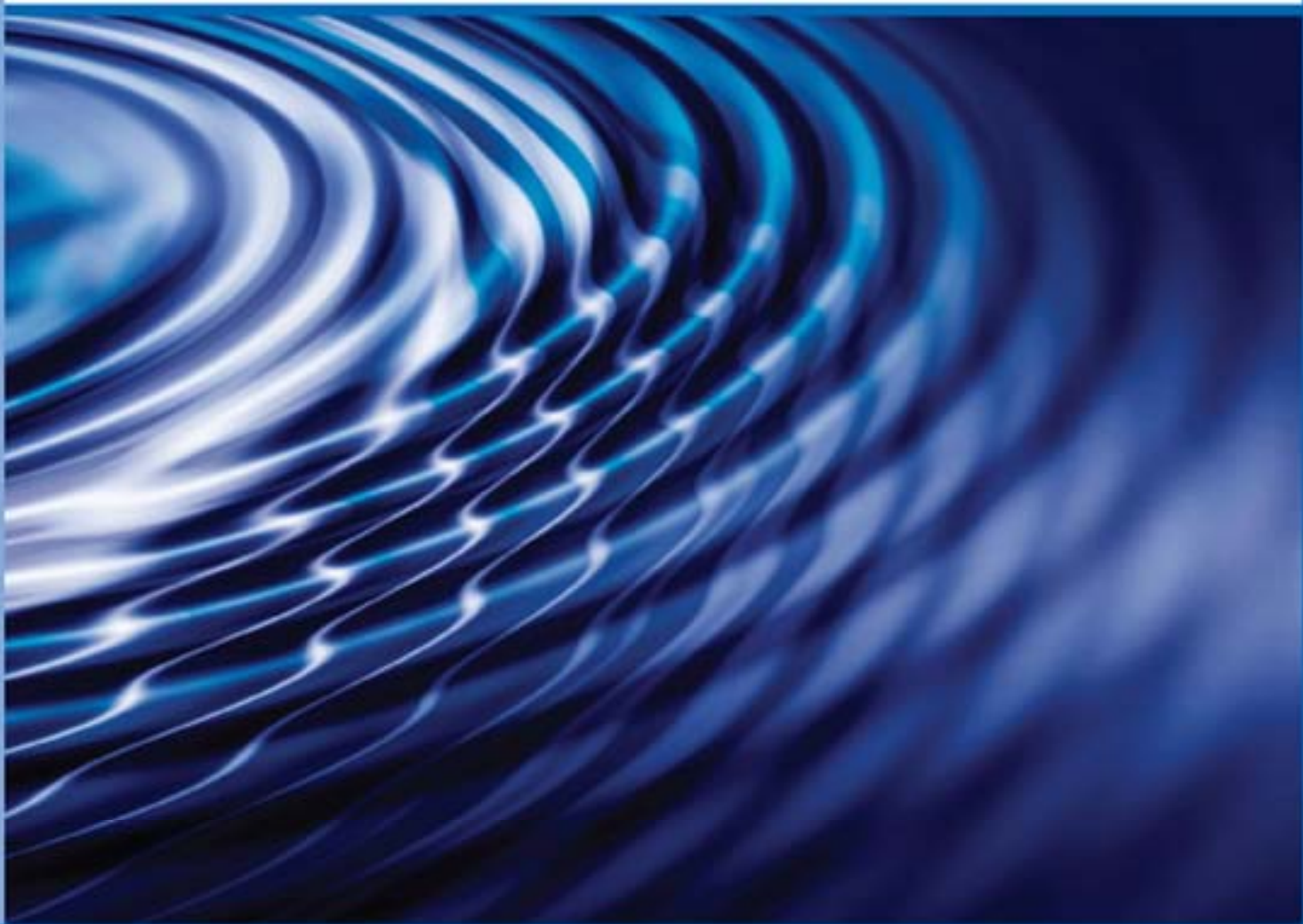
Windows クライアント
ユーザガイド



Attachmate®

Reflection®

for Secure IT



Reflection for Secure IT

Windows クライアント

バージョン 7.1



© 2009 Attachmate Corporation. All rights reserved.

本 Attachmate ソフトウェア製品に付属するマニュアルのいかなる部分も、形式、方法にかかわらず、Attachmate Corporation の書面による許可なく複製、送信、転記したり、ほかの言語へ翻訳することはできません。本書の内容は、本書がエンドユーザ使用許諾契約を含むソフトウェアに付属して配布されていない場合でも、著作権法によって保護されています。

本書の内容は情報提供の目的でのみ提供されており、予告なく変更されることがあります。また、本書の内容を Attachmate Corporation による義務と解釈してはなりません。Attachmate Corporation は、本書に含まれる情報の内容に誤りや不正確な部分があっても、いかなる責任も負いません。

Attachmate、Attachmate のロゴ、および Reflection は、米国およびその他の国における Attachmate Corporation の登録商標です。本書で引用しているその他のすべての商標、商標名、または会社名は、識別の目的でのみ使用されており、その所有権はそれぞれの所有者に帰属します。

Attachmate Corporation
1500 Dexter Avenue North
Seattle, WA 98109
USA
+1.206.217.7100
<http://www.attachmate.com>

目次

概要	7
インストール	9
System Requirements.....	9
Reflection for Secure IT のインストール	10
機能および言語の選択	11
旧バージョンからのアップグレード	11
基本操作	13
新規の端末セッションの開始.....	13
構成ツールバーの表示	13
FTP クライアント を使用したファイルの転送.....	14
Secure Shell とは.....	15
[一般] タブ (Secure Shell の設定).....	16
構成	19
設定ファイル	19
Secure Shell 構成ファイル.....	19
暗号化	21
データの暗号化.....	21
連邦情報処理規格 (FIPS).....	21
[暗号化] タブ (Secure Shell の設定)	22
認証	25
公開鍵を使用したサーバ認証.....	25
証明書を使用したサーバ認証.....	26
クライアント認証方式	27
Secure Shell セッションにおける接続の再利用.....	28
公開鍵認証	31
ユーザ鍵の管理.....	32
公開鍵認証の構成.....	32
ユーザ鍵一覧への鍵の追加.....	32
サーバへのクライアント公開鍵のアップロード	33
ユーザ鍵パスフレーズの変更	34
ユーザ鍵のエクスポート	34
[ユーザ鍵] タブ ([Reflection Secure Shell の設定] ダイアログボックス).....	35
[ユーザ鍵の生成] ダイアログボックス.....	36
ホスト鍵の管理.....	38
ホスト鍵の確認の構成.....	38
優先するホスト鍵の種類の構成.....	39
既知のホストファイル.....	39
[ホスト鍵] タブ ([Reflection Secure Shell の設定] ダイアログボックス).....	40
[ホスト鍵の信頼性] ダイアログボックス.....	41
証明書の認証 (PKI)	43
PKI と証明書.....	43
電子証明書の格納場所	43

証明書を使用したクライアント認証の構成	44
証明書を使用したサーバ認証の構成	45
Windows の証明書格納場所の使用の有効化と無効化	45
証明書取り消しの確認の構成	46
LDAP ディレクトリを使用した中間証明書の配布	47
[PKI] タブ ([Reflection Secure Shell の設定] ダイアログボックス)	48
Reflection 証明書マネージャ	49
Reflection 証明書マネージャの開き方	49
[個人] タブ ([Reflection 証明書マネージャ])	49
[信頼された認証局] タブ ([Reflection 証明書マネージャ])	50
[LDAP] タブ ([Reflection 証明書マネージャ])	51
CRL 確認のための LDAP サーバの設定	52
[OCSP] タブ ([Reflection 証明書マネージャ])	52
[PKCS#11] タブ ([Reflection 証明書マネージャ])	53
[PKCS#11 プロバイダ] ダイアログボックス	54
Secure Shell セッションの GSSAPI (Kerberos) 認証	55
GSSAPI 認証に対する Reflection Kerberos の使用	55
Secure Shell セッションにおける Kerberos チケット転送	55
GSSAPI Secure Shell セッションのサービスプリンシパルの指定	56
[GSSAPI] タブ ([Reflection Secure Shell の設定] ダイアログボックス)	56
ポート転送	59
ローカルポート転送	60
リモートポート転送	62
TCP 通信の転送	62
FTP 通信の転送	64
[トンネリング] タブ ([Reflection Secure Shell の設定] ダイアログボックス)	65
[ローカルポート転送] ダイアログボックス	65
[リモートポート転送] ダイアログボックス	67
マルチホップ Secure Shell セッションの構成	67
[マルチホップ] タブ ([Reflection Secure Shell の設定] ダイアログボックス)	68
[マルチホップサーバの構成] ダイアログボックス	69
ホスト変数およびコマンド	71
[ホストデータ] タブ ([Reflection Secure Shell の設定] ダイアログボックス)	71
プロキシサーバ	73
プロキシサーバ (Secure Shell の設定)	73
トラブルシューティング	75
Secure Shell 接続のトラブルシューティング	75
Secure Shell ログファイルの使用	76
Reflection for Secure IT の問題解決のヘルプ	76
インストールのカスタマイズと配布	79
管理者用インストール	79
インストール	79
インストールと配布の計画	81
管理者用インストールの実行	81
コマンドラインからのインストール	82

インストール記録の開始と終了.....	83
インストールのカスタマイズ.....	83
Attachmate カスタム設定ツールを開く	83
カスタム設定の種類を選択.....	83
コンパニオンインストーラによるカスタム設定のインストール.....	84
FTP クライアント設定のインストール.....	87
既存インストールへのコンパニオンインストーラの追加.....	88
Secure Shell コマンドラインユーティリティ.....	88
ssh コマンドラインユーティリティ	90
ssh2 コマンドラインユーティリティ	94
ssh-keygen コマンドラインユーティリティ	95
sftp コマンドラインユーティリティ	98
sftp2 コマンドラインユーティリティ	103
scp コマンドラインユーティリティ	104
scp2 コマンドラインユーティリティ	107
付録	109
Secure Shell クライアントが使用するファイル.....	110
SSH 構成セクション.....	112
サンプル構成ファイル.....	113
構成ファイルのキーワードのリファレンス - Secure Shell の設定.....	114
構成ファイルのキーワード参照 - 端末エミュレーション設定.....	125
DOD PKI 情報.....	130
用語集.....	135
索引	139

概要

Reflection for Secure IT Windows クライアント は、カスタマイズが容易なフル機能の Windows ベース Secure Shell クライアントです。Reflection を使用すれば、セキュリティが確保されていないネットワーク上でも、信頼するホストと Windows ワークステーション間で安全な暗号化通信を行うことができます。ローカルコンピュータとリモートホストの間の接続はすべて暗号化され、マシン間で送信されるデータが保護されます。Telnet、FTP、rlogin、rsh を使用する場合のように、パスワードが通常のテキスト形式でネットワークに送信されることはありません。

Reflection for Secure IT Windows クライアント は、以下に対応しています。

- プロトコルバージョン 1 とプロトコルバージョン 2 の両方のサーバへの安全な接続。
- TCP ポート転送 (X-11 を含む)、データストリームの圧縮と暗号化、パスワード、キーボード対話型、公開鍵、Kerberos/GSSAPI を使用した認証、ログ記録などの、標準的な Secure Shell 機能に対応します。
- ユーザ鍵の生成ツールにより、RSA 鍵、RSA1 鍵、DSA 鍵を作成できます。
- Secure Shell サーバに公開鍵をアップロードするツール。Reflection は自動的にサーバの種類を検出し、正しい鍵の種類をエクスポートして、サーバの正しい保存場所にインストールすることができます。
- ツールを使用して、信頼されているホスト鍵を表示および管理することができます。
- 鍵エージェントユーティリティにより、1 つのパスフレーズで複数の鍵と証明書を管理し、認証を別のサーバに転送することができます。
- PKI に対応しています。これには、証明書マネージャが含まれており、Reflection 固有の証明書格納場所にある証明書を管理できます。また、Windows の証明書格納場所、スマートカード、その他の PKCS #11 準拠のハードウェアデバイス上にある証明書を使用するように Reflection を設定することもできます。
- 安全な SFTP ファイル転送を実行できます。
- ssh、ssh-keygen、sftp、scp 用の独立した DOS コマンドラインユーティリティを使用できます。

第 1 章

インストール

Reflection for Secure IT をインストールするためにインストールプログラムを用意します。インストールプログラムは、ダウンロードサイトから入手するか、対象の製品 CD に含まれます。SP (サービスパック) を適用する場合は、最新の SP をダウンロードサイトから入手してください。

System Requirements

Supported operating systems:

- Windows XP
- Windows Vista
- Windows Server 2003
- Windows 2000 Professional SP4
- Windows 2000 Server SP4
- Windows Terminal Server (with or without Citrix MetaFrame)

Supported hardware:

- i386 (32bit)
- x86-64 (64 bit AMD x64 and 64 bit EM64T)

Reflection for Secure IT のインストール

注意: Reflection for Secure IT をインストールするには管理者権限でログオンする必要があります。必要なアクセス権限がない場合は、システム管理者に権限の変更を依頼してください。

ワークステーションにインストールするには

- 1 Attachmate 設定ウィザードを実行します。

インストール元	手順
ダウンロードサイト	ダウンロードリンクをクリックして、ダウンロードしたプログラムを実行します。インストーラファイルの場所を選択して [次へ] をクリックします。これにより、ファイルが指定の場所に解凍され、Attachmate 設定ウィザードが起動します。
管理者用インストールイメージ	管理者用インストールポイントから、 <code>setup.exe</code> ファイルをダブルクリックします。

- 2 Attachmate 設定ウィザードで、**[続行]** をクリックして、Attachmate 設定ウィザードのユーザインタフェースの指示に従います。
- 3 (オプション) 特定個人用にインストールを設定するには、**[ユーザ情報]** タブをクリックし、名前、組織、ボリューム購入契約 (VPA) 番号 (VPA がある場合) を入力します。

注意: Attachmate によって発行される VPA 番号の目的は、カスタマーサポートがサービス要求を迅速に処理できるようにすることです。

- 4 (オプション) 既定のインストールフォルダを変更するには、**[ファイルの場所]** タブをクリックし、Reflection for Secure IT をインストールするフォルダを参照します。
- 5 (オプション) インストールされる機能、コンポーネント、言語を選択するには、**[機能の選択]** タブをクリックします。
- 6 **[インストール]** をクリックします。




注記: 管理者用インストールを作成する場合のみ、インストーラの **[詳細設定]** タブを使用します。管理者用インストールでは、製品は実際にはインストールされず、Reflection for Secure IT のイメージをネットワーク上に置き、後で複数のワークステーションにインストールします。このネットワーク上の場所は、配布ツールがワークステーションに配布するパッケージにアクセスして作成するのに使用します。エンドユーザはこの場所にある `setup.exe` を実行して、ワークステーションインストールを実行します。

機能および言語の選択

[機能の選択] タブを使って、インストールする製品機能を選択します。

インストールする機能、コンポーネント、および言語を選択するには

- 1 [機能の選択] タブをクリックします。
- 2 アイテムごとに、以下のオプションから選択します。

選択	目的
 [機能をローカルのハードディスクドライブにインストールする]	アイテムをインストールします。
 [機能を必要時にインストールする]	アイテムをアダプタイズします。例えば、[スタート] メニューからコンポーネントを選択でき、必要な時にそのアイテムをインストールします。
 [機能を使用不可にする]	アイテムがインストールされていない状態にします。この場合でも、後で Windows の [プログラムの追加と削除] コントロールパネルを使ってアイテムをインストールすることができます。

旧バージョンからのアップグレード

旧バージョン (6.x、7.0) からバージョン 7.1 へアップグレードする前に以下の内容を確認ください。

- 前のバージョンをアンインストールする必要はありません。本バージョンをインストールすると、インストーラが自動的に前のバージョンを検出してアップグレードします。既存の Secure Shell 設定はそのまま適用され、既存の設定ファイルを引き続き使用できます。
- 本バージョンをインストールしたら、前のバージョンの Reflection for Secure IT を維持できなくなります。異なるインストール場所を選択した場合でも、インストーラは前のバージョンを自動的にアップグレードし、このアップグレードによって前のバージョンが削除されます。
- 本バージョンをインストール後に削除しても、前のバージョンは復元されません。

第 2 章

基本操作

Secure Shell は、リモートコンピュータへのログインとコマンドの実行を安全に行うためのプロトコルです。これは、Telnet、FTP、rlogin、あるいは rsh の代わりとなる安全な方法です。Secure Shell 接続では、ホスト (サーバ) とユーザ (クライアント) の両方の認証が必要です。また、ホスト間の通信はすべて暗号化された通信チャネルを介して行う必要があります。また、Secure Shell では X11 セッションまたは指定の TCP/IP ポートを、安全なトンネルを介して転送することもできます。

新規の端末セッションの開始

通常、既定の設定のままでもパスワード認証によりリモートホストに接続可能です。

既定値を使用して新規の端末セッションを開始するには

- 1 Windows の [スタート] - [Attachmate Reflection] - [SSH クライアント] をクリックします。
- 2 Reflection for Secure IT ツールバーの [接続/切断] ボタンをクリックします。



- 3 [ホストに接続] ダイアログボックスにホスト名およびユーザ名を入力して、[OK] をクリックします。

注記: リモートホストへの初回アクセス時は、[ホスト鍵の信頼性] ダイアログボックスを表示し、リモートホストが提示したホスト鍵の指紋 (メッセージダイジェスト) を目視確認します。厳密な運用では、事前にリモートホストの管理者からその値を入力し、表示内容と比較確認します。問題なく接続処理を継続する場合は、ローカル PC 内にそのホスト鍵を保存する指示を含め [常時] をクリックします。

- 4 リモートホストへのログインパスワードを入力して、[OK] をクリックします。
- 5 このセッション構成を含む設定ファイルを保存するには、[ファイル] - [保存] をクリックします。

構成ツールバーの表示

構成ツールバーを表示すると、セッションの設定にすばやくアクセスできます。

構成ツールバーを表示するには

- 1 Reflection for Secure IT セッションを開きます。
- 2 [セッション設定の構成] ツールバーボタンをクリックします。



- 3 セッション設定を保存すると ([ファイル] - [保存])、このセッションを開くたびにこのツールバーが表示されるようになります。

FTP クライアント を使用したファイルの転送

Reflection FTP クライアントでは、専用の GUI 画面を開いて、簡単なドラッグ & ドロップ操作やボタン操作によりファイル操作やファイル転送を指示します。単一ファイルを個別に操作したり、複数ファイルやフォルダ全体をまとめて操作することもできます。

サーバに接続してファイルを転送するには

- 1 Windows の [スタート] - [Attachmate Reflection] - [Reflection FTP クライアント] をクリックします。[FTP サイトに接続] ダイアログボックスが自動的に開きます。
- 2 [新規サイト(N)...] をクリックします。
- 3 ウィザードの案内に従い、アクセス先リモートホスト名やユーザ名を入力します。

注記: Reflection for Secure IT とともに FTP クライアントをインストールしている場合、ウィザードは、既定で SFTP 接続を作成するよう構成されています。バージョン 7.0 以降では、[セキュリティのプロパティ] ダイアログボックスを使用して追加の接続の種類を構成することができます。[ログイン情報] ダイアログボックスの [セキュリティ] ボタンを使用します。

- 4 最後のパネルで、今すぐに接続するかどうかの確認を表示します。既定値で [はい] 選択のまま、[完了] ボタンをクリックし、ウィザードを終了して接続処理を開始します。[完了] ボタンをクリックし、ウィザードを終了して接続処理を開始します。

注記: まだ接続していないサーバへの SFTP 接続を構成している場合は、ホストの信頼性を確認するよう求めるダイアログボックスが表示されることがあります。ホスト鍵の有効性を確認するには、そのホストについてシステム管理者に問い合わせてください。このホストを既知のホストの一覧に追加するには [常時] をクリックします。

- 5 画面表示されたフォルダ/ファイル群の中から転送元/転送先の場所を開き、転送ファイルを指定します検索して見つけます。

検索する場所	使用する画面
ローカルフォルダ	左画面
サーバディレクトリ	右画面

- 6 転送対象のファイルまたはフォルダを選択して、転送元の場所から目的の転送先へドラッグします。

リモートホストアクセス情報を一覧に追加保存するには

- [ファイル] - [保存] をクリックして、サイト構成を FTP クライアント 設定ファイルへ保存します。

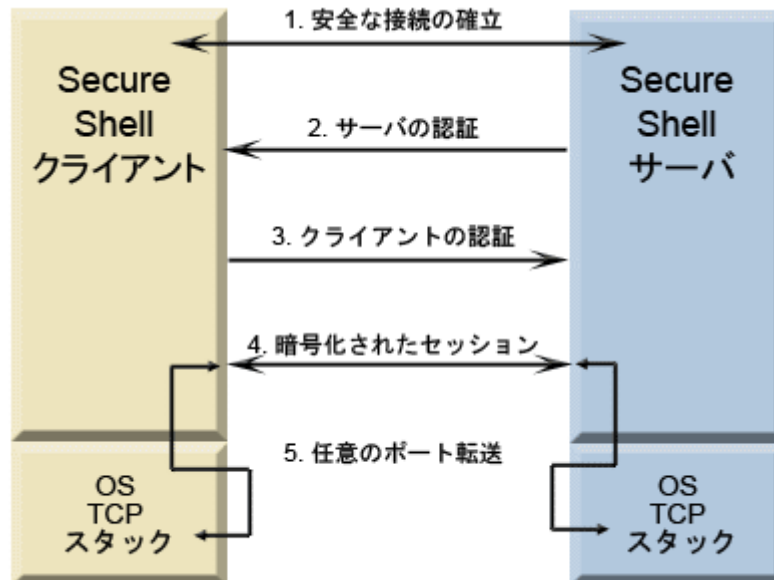
保存したサイトに接続するには

- 1 FTP クライアント を起動します。
- 2 [FTP サイトに接続] ダイアログボックス内の対象ホストの名前を選択します。
- 3 [接続] をクリックします。

注記: FTP クライアントの使い方の詳細については、FTP クライアントアプリケーションのヘルプを参照してください。

Secure Shell とは

ここでは、データを安全に送信するための Secure Shell チャンネルの作成と使用に関する基本手順の概要を記載します。



1. 安全な接続を確立します。

クライアントとサーバは、セッションの暗号化に使用する共有鍵と暗号、およびデータの完全性保証の確認に使用するハッシュを作成するために交渉します。

2. サーバを認証します。

サーバ認証によって、クライアントはサーバの ID を確認できます。サーバからクライアントへの認証は、この認証プロセス中に 1 回だけ可能です。この認証に失敗した場合は、接続できません。

3. クライアントを認証します。

クライアント認証によって、サーバはクライアントユーザの ID を確認できます。既定で、クライアントは認証を複数回試行できます。サーバとクライアントは、1 つまたは複数の認証方式に合意するように交渉します。

4. 暗号化されたセッションを介してデータを送信します。

暗号化されたセッションが確立されると、Secure Shell サーバとクライアント間で交換されるすべてのデータが暗号化されます。この段階で、ユーザはサーバへの安全なリモートアクセスが可能になり、保護されたチャンネルを通じて安全にコマンドを実行し、ファイルを転送することができます。

5. ポート転送を使用して、その他のクライアントとサーバ間の通信を保護します。

トンネリングとしても知られるポート転送は、アクティブなセッションにおける Secure Shell チャンネルを通じて通信をリダイレクトするための方法を提供します。ポート転送が構成されると、指定のポートへ送信されるすべてのデータは、保護されたチャンネルを通じてリダイレクトされます。

SSH クライアントセッションで Secure Shell 設定を構成するには

- 1 [接続] - [接続の設定] コマンドをクリックします。
- 2 [接続オプション] で、[ホスト名] と [SSH 構成セクション] (オプション) に値を入力します([SSH 構成セクション] を空白のままにした場合は、[ホスト名] と同じ内容が SSH 構成セクション 『112 ページ』 に保存されます)。
- 3 [セキュリティ] をクリックします。

注記: ホスト名を入力して初めて [セキュリティ] ボタンが有効化します。

FTP クライアント で Secure Shell 設定を構成するには

- 1 [FTP サイトに接続] ダイアログボックスから接続ホストを 1 つ選択します。
- 2 [セキュリティ] をクリックします。
- 3 [Secure Shell] タブで、[Reflection Secure Shell を使用] を選択します(Reflection for Secure IT とともに FTP クライアントをインストールしている場合、これは既定で選択されています)。
- 4 (オプション) [SSH 構成セクション] を指定します([SSH 構成セクション] を空白のままにした場合は、[ホスト名] と同じ内容が SSH 構成セクション 『112 ページ』 に保存されます)。
- 5 [構成] をクリックします。

[一般] タブ (Secure Shell の設定)

開き方 『15 ページ』

オプションは次のとおりです。

[ポート番号]	サーバ上の接続先ポートを指定します。既定は 22 で、これは Secure Shell 接続の標準ポートです。
[プロトコル]	ホストへの接続の確立時に Reflection で使用する、Secure Shell プロトコルのバージョンを指定します。セキュリティを確保上 [2 のみ] に設定します。
[ユーザ認証]	いずれかの <i>認証方式</i> 『27 ページ』の横のボックス内をクリックして、その方式をオフまたはオンにします。少なくとも 1 つは選択指定する必要があります。矢印を使用して、上下移動させ優先順位を指定します。サーバ側で許可している方式について、上から順に試行します。
[サーバキープアライブ]	[サーバキープアライブ] が選択されている場合、指定間隔でサーバに対して健全性維持確認のための NOOP メッセージを送信します。この設定を使用して、サーバへの接続を維持します。[間隔] を使用して、キープアライブメッセージを送信する間隔を指定します。この設定が無効の場合、サーバがダウンしたり、ネットワーク接続が切断されても、Secure Shell 接続は終了しません。この設定を使用して、TCP セッションのみを転送する接続がサーバで時間切れになるのを防ぐこともできます。この設定を使用しないと、サーバで SSH トラフィックが検出されない場合、これらの接続は時間切れになります。

Windows のレジストリに TCP キープアライブ設定というのがありますが、これはファイアウォールによってすべての TCP/IP 接続が時間切れにならないようにするものです。本設定は、この TCP キープアライブ設定とは関係ありません。TCP/IP のキープアライブの動作を変更するには、Windows のレジストリを編集する必要があります。

[圧縮を使用]

[圧縮を使用] をオンにすると、クライアントは、すべてのデータを圧縮するよう要求します。圧縮は、モデムを介した接続などの低速回線では効果的ですが、高速なネットワークでは逆に応答速度の低下を招きます。圧縮レベルの設定はプロトコルバージョン 1 でのみ使用可能です。プロトコルバージョン 2 接続には適用されません。

[既存の接続がある場合、それを利用]

同一のサーバとクライアント間で複数の接続指定をした場合、既定では 確立済みの既存のセッションを再使用『[28](#) ページ』するので、再度の認証処理手続きを必要としません。本設定の選択を解除すると、同一のサーバとクライアント間でも、接続のつど認証処理手続きを実施し、新たなセッションを確立します。

[記録内容]

Secure Shell のログファイル『[76](#) ページ』にどの程度の情報を記録するのかを指定します。

注意

- このダイアログボックスで構成した設定は、*Secure Shell 設定ファイル*『[19](#) ページ』に保存されます。また、このファイルを任意のテキストエディタで手作業で編集することにより Secure Shell 設定を構成することもできます。
 - この構成ファイルの内の設定は、現在指定されている *SSH 構成セクション*『[112](#) ページ』用に保存されます。
-

第 3 章

構成

Reflection for Secure IT および FTP クライアントでは、設定を保存するために複数のファイルが使用されます。

設定ファイル

設定ファイルは、アプリケーション固有の設定を構成するために使用します。Reflection for Secure IT と FTP クライアントでは、異なる設定ファイルが使用されます。

プログラムコンポーネント	拡張子	構成内容
<SSH クライアント>	*.r3w	ターミナルログイン接続情報、端末エミュレーション、表示の設定、キーマッピング、マウスの構成
<Reflection prod_ftp_short>	*.rfw	SFTP 接続情報、ディレクトリの表示設定、転送設定

設定ファイルを保存するには、[ファイル]- [保存] を使用します。

注記: 設定ファイルの内容やについては、[Reflection for Secure IT] 画面 (SSH クライアント) および [FTP クライアント] 画面のヘルプを参照してください。

Secure Shell 構成ファイル

構成ファイルにて、クライアントとしての SSH 接続動作仕様を構成します。このユーザ固有のファイルは、[Reflection Secure Shell の設定] 『15 ページ』画面上の設定内容と連携し、画面設定を初めて変更した時に新規生成され、以後 画面設定を変更保存する度に更新されます。以下のパスに存在します。

```
<個人用ドキュメントフォルダ>¥Attachmate¥Reflection¥.ssh¥config
```

このファイル内の設定はホストごと (または SSH 構成セクション 『112 ページ』ごと) に適用され、Reflection for Secure IT クライアントと FTP クライアントの両方に影響を与えます。例えば、Reflection for Secure IT を使用して、Acme.com への接続用に既定以外の Secure Shell 設定を構成し、かつ、SSH 構成セクションを指定していない場合、その Secure Shell 設定は、次の行で識別されるセクションにある構成ファイルに保存されます。

```
Host Acme.com
```

さらに、Acme.com へ接続するように FTP クライアントを構成し、かつ、SSH 構成セクションを指定していない場合、FTP クライアントでは、構成ファイルの「Host Acme.com」セクションの設定が使用されます(両方のアプリケーションで同じ SSH 構成セクションを指定している場合も、設定は同じ方法で共有されます)。

注記: [Reflection Secure Shell の設定] ダイアログボックスを閉じる時は、既定の設定を使用する値は構成ファイルに保存されません。既定値が手動でファイルに追加されている場合、その値は、このダイアログボックスを閉じると削除されます。特定のホスト名を使用するスタンザと組み合わせでワイルドカード使用のホストスタンザを使用する場合、この状況はホストの作成上の制約になります。ワイルドカード使用のスタンザに構成されている値を無効にする目的で、特定のホストスタンザ内に既定値を手動で構成している場合、ホスト固有の SSH 構成セクションの設定を表示するために [Secure Shell の設定] ダイアログボックスを開くと、この既定の設定が削除されます。このような状況は、グローバル構成ファイルを使用することで適切に対処できます。グローバル構成ファイルは、[Reflection Secure Shell の設定] ダイアログボックスを開いたり閉じたりしても更新されません。

グローバル構成ファイル (「ssh_config」ファイル)

本 PC からの全ての接続時の共通の設定をグローバル構成ファイルを設定することで構成できます。ファイル名および場所は次のとおりです。

<アプリケーションデータフォルダ> 『[136](#) ページ』¥Attachmate¥Reflection¥ssh_config

このファイル内の設定は、本 PC の全ユーザのクライアント接続に影響を及ぼします。

第 4 章

暗号化

データの暗号化

暗号化によって、転送中のデータの秘密性が保護されます。これは、送信前のデータを秘密鍵および Cipher を使って暗号化することで実施されます。受信されたデータは同じ鍵と Cipher を使用して解読される必要があります。指定したセッションに使用される Cipher は、クライアントの優先順位の最上位にある Cipher であり、サーバもまたこの最上位の Cipher に対応します。

Reflection for Secure IT Windows クライアント は、以下のデータ暗号化規格に対応しています。

- DES (56 ビット)
- Arcfour (40 ビットまたは 128 ビット)
- TripleDES (168 ビット)
- Cast (128 ビット)
- Blowfish (128 ビット)
- AES (Rijndael) (128 ビット、192 ビット、または 256 ビット)

連邦情報処理規格 (FIPS)

Reflection が FIPS モードで動作するよう構成されている場合、Reflection は米国政府の連邦情報処理規格 (FIPS) 140-2 で認定された暗号ライブラリのみを使用します。よって、規格に適合しないオプションは使用できなくなり、接続先リモートホストが提示する仕様とマッチせず接続失敗などの制約を受けることもありますので指定には注記が必要です。各セッションを FIPS モードで動作するように個別に構成することも、すべての Reflection セッションに FIPS モードを設定することもできます。

特定の Secure Shell セッションを FIPS モードで構成する

次の手順を使用して、特定の Secure Shell セッションが FIPS モードで動作するように設定できます。

注記: この手順を実行すると、一部の Secure Shell セッションで FIPS 規格が使用されなくなります。この変更内容は、*Secure Shell 構成ファイル* 『[19](#) ページ』に保存され、特定の *SSH 構成セクション* 『[112](#) ページ』に適用されます(セクションを指定しないと、設定は現在のホストへのすべての接続に適用されます)。同じ SSH 構成セクション (またはホスト名) を使用するように後続の Secure Shell セッションを構成する場合を除いて、この変更は後続のセッションに影響を与えません。

特定のホストまたは SSH 構成セクションに FIPS モードを設定するには

- 1 [Secure Shell の設定] ダイアログボックスを開きます 『[15](#) ページ』。
- 2 [暗号化] タブで [FIPS モードで実行] を選択します。

Secure Shell 構成ファイルを直接編集して、この設定を構成することもできます。FIPS モードを設定するキーワードは、FIPSMode です。

すべての Reflection セッションを FIPS モードで構成する

管理者は、Reflection Group Policies を使用して、FIPS モードで動作するようにすべての Reflection セッションを設定することができます。

すべてのセッションに FIPS モードを設定するには

- 1 グループポリシーエディタを実行するには、次のいずれかの方法を使用します。
 - コマンドラインに次の文字を入力します。
Gpedit.msc
 - [Active Directory ユーザーとコンピュータ] から、[組織単位] のプロパティを開き、[グループ ポリシー] タブをクリックして、ポリシーオブジェクトを編集または新規作成します。
- 2 Reflection テンプレート (ReflectionPolicy) をまだインストールしていない場合はインストールします。
- 3 [ローカルコンピュータポリシー] - [ユーザ構成] - [管理用テンプレート] - [Reflection の設定] で、[FIPS 以外のモードを許可する] を無効にします。

[暗号化] タブ (Secure Shell の設定)

開き方 『[15 ページ](#)』

[Reflection Secure Shell の設定] ダイアログボックスの [暗号化] タブを使用して、Secure Shell 接続が使用する Cipher 『[137 ページ](#)』を指定します。接続に使用する Secure Shell プロトコルに応じて、異なるオプションを使用できます。

オプションは次のとおりです。

[SSH プロトコル 1]

Cipher この設定を使用して、現在のホストへのプロトコル 1 接続で使用する Cipher 『[137 ページ](#)』を選択します。既定値は [Triple DES] で、これが推奨されるオプションです。

[SSH プロトコル 2]

[Cipher 一覧] この一覧を使用して、現在のホストへのプロトコル 2 接続で使用可能にする Cipher 『[137 ページ](#)』を指定します。複数の Cipher を選択した場合、Secure Shell クライアントは指定した順序に従って先頭の Cipher から使用を試みます。順序を変更するには、一覧から Cipher を選択して上向きまたは下向きの矢印をクリックします。指定したセッションに使用される Cipher は、この一覧の先頭の項目であり、サーバもまたこの先頭の Cipher に対応します。

[HMAC 一覧] 使用する HMAC (ハッシュメッセージ認証コード) 方式の候補を指定します。このハッシュは、サーバとの間で交換されるすべてのデータパケットの整合性を確認するために使用されます。複数の HMAC を選択した場合、Secure Shell クライアントは指定した順序に従って先頭の HMAC からサーバとの交渉を試みます。順序を変更するには、一覧から HMAC を選択して上向きまたは下向きの矢印をクリックします。

[鍵交換アルゴリズム]	<p>使用する鍵交換アルゴリズムとその優先順位を指定します。指定できる値は以下のとおりです。</p> <ul style="list-style-type: none"> ▪ [DH Group1 SHA1] - diffie-hellman-group1-sha1 を指定します。 ▪ [DH Group Ex SHA1] - diffie-hellman-group-exchange-sha1 を指定します。 ▪ [DH Group14 SHA1] - diffie-hellman-group14-sha1 を指定します。 <hr/> <p>そのほかに 2 つの暗号化アルゴリズム (gss-group1-sha1-*) に対応していますが、これらは使用可能な鍵交換アルゴリズムの一覧には表示されません。これらの 2 つのアルゴリズムは、[一般] 『16 ページ』 タブ ([ユーザ認証] の下) で [GSSAPI/Kerberos] を有効にし、[GSSAPI] 『56 ページ』 タブで [Reflection Kerberos] を選択している場合に自動的にクライアントに表示されます。</p> <hr/>
指紋の種類	<p>公開鍵ユーザ認証において、秘密鍵所有確認過程でクライアントが使用するハッシュアルゴリズムを指定します。このハッシュは、公開鍵ユーザ認証時に使用されます。RSA 鍵で使われるハッシュを指定するには [RSA] を使用し、DSA 鍵で使われるハッシュを指定するには [DSA] を使用します。</p>
[FIPS モードで実行]	<p>[FIPS モードで実行] を選択すると、Reflection はこの接続に対して米国政府の連邦情報処理規格 (FIPS) 140-2 を使用します。[FIPS モードで実行] 選択した場合、[暗号化] タブのこれらの規格に適合しないオプションは使用できなくなります。</p>

注意

- このダイアログボックスで構成した設定は、*Secure Shell 設定ファイル* 『19 ページ』に保存されます。また、このファイルを任意のテキストエディタで手作業で編集することにより *Secure Shell 設定* を構成することもできます。
 - この構成ファイルの内の設定は、現在指定されている *SSH 構成セクション* 『112 ページ』用に保存されます。
-

第 5 章

認証

認証は、通信相手の身元を確実に確認する処理のことです。身元の確認は、パスワードなどの既知の情報、秘密鍵やトークンなど所有しているもの、または指紋などの固有の情報を使用して行います。Secure Shell 接続では、ホスト (サーバ) とユーザ (クライアント) の両方の認証が必要です。既定で、ホストは秘密鍵を使用してユーザを認証し、ユーザはパスワードを使用してホストを認証します。

公開鍵を使用したサーバ認証

Reflection for Secure IT では、公開鍵と証明書 (公開鍵認証の特別な形) という 2 種類のサーバ認証に対応しています。

ホスト認証のために公開鍵認証が使用される時は、以下の一連のイベントが行われます。

1. Secure Shell クライアントが接続を開始します。
2. サーバが公開鍵をクライアントに送信します。
3. クライアントが、信頼されているホスト鍵ストアからこの鍵を検索します。

クライアントによる鍵の検索結果

発生するイベント

ホスト鍵を見つけ、クライアントコピーが、サーバによって送信された鍵に一致する

認証は次の段階に進みます。

ホスト鍵を検出できない

クライアントは、ホストが不明であるというメッセージを表示し、ホスト鍵のフィンガープリントを提供します。ユーザが不明な鍵を受け入れることができるようにクライアントが構成されている場合 (既定)、ユーザは、この鍵を受け入れることができ、認証は次の段階に進みます。

厳格なホスト鍵の確認が履行される場合は、クライアントによって接続が終了されます。

ホスト鍵を見つけたが、クライアントコピーが、サーバによって送信された鍵に一致しない

クライアントは、鍵が既存の鍵に一致しないという警告を表示し、サーバによって送信された鍵のフィンガープリントを表示します。ユーザが不明な鍵を受け入れることができるようにクライアントが構成されている場合 (既定値)、ユーザは、この新しい鍵を受け入れることができます。

厳格なホスト鍵の確認が履行される場合は、クライアントによって接続が終了されます。

4. サーバが、受信した公開鍵に対応する秘密鍵を実際に保持していることを確認するために、クライアントはサーバに試行 (任意のメッセージ) を送信して、当該メッセージテキストに基づき [ハッシュ](#) 『[137](#) ページ』を計算します。

5. サーバは、試行メッセージに基づきデジタル署名を作成します。これを行うために、サーバは別個にメッセージのハッシュを計算し、次に、サーバの秘密鍵を使用して、この計算したハッシュを暗号化します。サーバは、当該デジタル署名を元の試行に付加して、この署名付きのメッセージをクライアントに返します。
6. クライアントは、公開鍵を使用して署名を復号化し、クライアント自身が計算したハッシュと当該ハッシュを比較します。値が一致すると、ホスト認証が成功します。

証明書を使用したサーバ認証

公開鍵認証で生じるいくつかの問題は、証明書認証を使用することで解決できます。公開鍵認証では、システム管理者がサーバごとにホストの公開鍵を各クライアントの既知のホストの一覧に追加するか、またはクライアントユーザ側が未知のホストに接続した時にホストを正しく識別する必要がありました。証明書認証では、認証局 (CA) と呼ばれる信頼されたサードパーティを使用してホストからの情報の有効性を検証します。

公開鍵認証と同様に、証明書認証でも公開/秘密鍵のペアを使用してホストの識別情報を確認します。ただし、証明書認証の場合、公開鍵は電子証明書『138 ページ』内に含まれます。この場合、2 つの鍵ペアが使用され、ホストが 1 つの秘密鍵を保持し、CA が 2 番目の秘密鍵を保持します。ホストは CA から証明書を取得します。この証明書には、ホストの識別情報、ホストの公開鍵のコピー、および CA の秘密鍵を使用して作成された電子署名『137 ページ』が含まれます。この証明書は、認証処理中にクライアントに送信されます。ホストから受信した情報の整合性を確認するには、クライアントは、CA ルート証明書に含まれる CA の公開鍵のコピーを保持している必要があります。

ホストの識別情報を確認するために CA ルート証明書をインストールする場合は、ホストの公開鍵をインストールおよび構成した場合と比べて以下のようないくつかの利点があります。

- 1 つの CA 証明書を複数のサーバの認証に使用できます。
- 管理者は、Windows のグループポリシーを使用して CA 証明書を Windows クライアントにインストールできます。
- 商用目的で取得した証明書のルート証明書が、クライアントのコンピュータ上ですでに使用可能な場合があります。Windows コンピュータには、Internet Explorer で使用するために複数のルート証明書が事前にインストールされています。SSL/TLS 接続では、Reflection は既定でこの証明書格納場所にある証明書を確認します。
- 必要な場合、ホストは、クライアントシステム上での変更を必要とせずに、同じ CA から新しい証明書を取得できます。

証明書認証によるサーバ認証は、以下のように行われます。

1. Secure Shell クライアントから接続を開始します。
2. ホストの電子証明書がホストからクライアントに送信されます。
3. クライアントで、CA ルート証明書を使用してサーバ証明書の有効性を確認します。

注記: クライアントの信頼されたルートの格納場所に CA 証明書のコピーがすでに保存されている必要があります(1 つの CA 証明書を複数のサーバの認証に使用できます)。

4. クライアントで、ホストの証明書のサーバ情報が接続先のホストと一致するかどうかを確認します。
5. 証明書内の公開鍵に対応する秘密鍵がホストにあるかどうかを確認するために、クライアントによって、試行 (任意のメッセージ) がサーバへ送信され、そのメッセージテキストに基づいてハッシュ『137 ページ』が計算されます。

6. サーバでは、この試行メッセージに基づいて電子署名が作成されます。この場合、サーバは単独でメッセージのハッシュを計算し、その計算したハッシュを、秘密鍵を使用して暗号化します。次に、サーバはその電子署名を試行に添付し、その署名付きのメッセージをクライアントに返信します。
7. クライアントでは、サーバの公開鍵を使用して署名が解読され、そのハッシュが、クライアントで独自に算出されたハッシュと比較されます。値が一致すれば、ホスト認証は成功です。

注記: Reflection for Secure IT Windows クライアントは、*Reflection の証明書格納場* 『50 ページ』または Windows の証明書格納場のいずれかを使用してホストの証明書を確認できます。

クライアント認証方式

Reflection Secure Shell クライアントでは、4 つの方式のユーザ認証に対応しています。それらは、Kerberos (GSSAPI)、公開鍵、キーボード対話型、パスワードです。[Reflection Secure Shell の設定] 『15 ページ』ダイアログボックスを使用して、認証の設定を構成します。少なくとも 1 つの認証方式を選択する必要があります。複数の方式を選択した場合、Secure Shell クライアントは指定した順番で認証を試行します。既定で、Reflection は最初に公開鍵認証を試行し、次にキーボード対話型、パスワードの順に試行します。

注意: 公開鍵および GSSAPI/Kerberos V5 認証方式では、サーバとクライアントの両方の構成が必要になります。

認証方式	説明
[パスワード]	<p>クライアントユーザに、Secure Shell サーバホスト上の当該ユーザ用のログインパスワードを入力するよう求めます。</p> <p>パスワードは、暗号化されたチャンネルを介してホストに送信されます。</p>
[キーボード対話型]	<p>単純なパスワード認証を含む、キーボードを使用して認証データを入力する手順に対応します。これによって、Secure Shell クライアントは、RSA SecurID トークンまたは RADIUS サーバなどのさまざまな認証機構に対応できるようになります。</p> <p>例えば、クライアント管理者はキーボード対話型認証を構成して、パスワード更新などの複数のプロンプトが必要な状況を処理できます。</p> <p>キーボードデータは、暗号化されたチャンネルを介してホストに送信されます。</p>
[公開鍵]	<p>公開/秘密鍵ペア 『138 ページ』を信頼します。公開鍵認証を構成するには、各クライアントユーザが、鍵ペアを作成して、公開鍵をサーバにアップロードする必要があります。鍵がパスフレーズで保護されている場合は、公開鍵認証を使用して接続を完了する目的で、クライアントユーザは当該パスフレーズを入力するように求められます。</p>

[GSSAPI/Kerberos]

Kerberos は、クライアント認証とサーバ認証の両方の代替メカニズムを提供するセキュリティプロトコルです。Kerberos 認証は、KDC (Key Distribution Center) と呼ばれる信頼されたサードパーティに依存しています。Secure Shell プロトコルは、GSSAPI (Generic Security Services Application Programming Interface) を介して Kerberos 認証に対応しています。

Secure Shell セッションにおける接続の再利用

接続を再利用することによって、すでに確立されている Secure Shell 接続に別の Secure Shell セッションを追加できます。この簡単な例が光ファイバケーブルで、外側のパイプで接続を行い、さまざまな光ファイバストランド (セッションとトンネル) がルーティングされます。追加セッションには、新しい Reflection Secure Shell 端末セッション、新しい Reflection SFTP ファイル転送セッション、転送された X11 接続、SSH トンネルを介したポート転送用に構成された通信、または Reflection Secure Shell コマンドラインユーティリティの 1 つを使用して確立された接続などがあります。

確立されている Secure Shell 接続を再利用する場合、認証処理を繰り返す必要はありません。新しいセッションでは、最初の接続に構成されたすべての Secure Shell 設定を必ず使用します。認証方式、暗号または MAC 設定の違い、あるいはポート転送定義は無視されます。

Reflection ユーザインタフェースを使用して行われるすべての Secure Shell 接続では、接続の再利用が既定で有効になっています。[Reflection Secure Shell の設定] ダイアログボックスの [全般] 『[16](#) ページ』 タブにある **[既存の接続がある場合、それを利用する]** チェックボックスをオフにして、この機能を無効にできます。

[既存の接続がある場合、それを利用する] をオンにして接続を確立すると、以下のすべての条件に合致する場合、以降の Secure Shell セッションでは確立されている接続を再利用します。

- 新しいセッションのホスト名は、確立されている接続のホスト名と完全に一致している必要があります。
- 新しいセッションのユーザ名は、確立されている接続のユーザ名と完全に一致している必要があります。
- 新しいセッションのポート番号は、確立されている接続のポート番号と同じである必要があります (既定ではこの条件が真になります)。
- 元のセッションがホスト名とは異なる *SSH 構成セクション* 『[112](#) ページ』を使用するよう構成されている場合、新しいセッションでは同じセクションを使用するよう構成する必要があります。

注意: SSH 接続にコマンドラインユーティリティを使用している場合、既存の接続を再利用するための追加条件を満たしている必要があります。追加条件の概要を以下に示します。

Reflection コマンドラインセッションにおける接続の再利用

接続の再利用は、Secure Shell 接続が要求され、クライアントと単一のサーバ間に多数の単純な操作が必要になり、認証と鍵交換の間隔が全接続時間のかなりの部分を占めるようなコマンドライン操作に適しています。これは、複数の小さなファイルの転送、または大量の出力を返さない簡単なオペレーティングシステムコマンドの実行が必要な場合などです。このような場合には、ssh (または ssh2) コマンドラインユーティリティを使用して元の SSH 接続を作成してから、以降のコマンドラインユーティリティ操作で接続を再利用するのが便利な場合があります。

既定では、Reflection Secure Shell client コマンドラインユーティリティ (ssh 『90 ページ』、scp 『104 ページ』、sftp、ssh2 『94 ページ』、scp2 『107 ページ』、sftp2 『103 ページ』) の接続の再利用は無効になっています。これらのコマンドユーティリティのいずれかで接続の再利用を有効にするには、以下のいずれかの方法を使用する必要があります。

- 各コマンドラインにスイッチ「`-o ConnectionReuse=yes`」を追加します。最初の接続を確立し、以降のすべてのコマンドラインユーティリティで最初の接続を再利用する場合、このスイッチを使用する必要があります。例えば、以下のコマンドを使用すると、sftp 接続が ssh コマンドで確立した接続を再利用します。

```
ssh "-o connectionReuse=yes" myuser@myhost
sftp "-o connectionReuse=yes" myuser@myhost
```

- DOS コマンドウィンドウ (またはバッチスクリプトファイルの最初) で、環境変数 `SSHConnectionReUse` を次のとおり設定します。

```
set SSHConnectionReUse=yes
```

矛盾する設定が存在する場合、`-o` スイッチが優先されます。

注意:

- OpenSSH サーバは、未認証の同時セッション数を制限するために使用できる `MaxStartups` パラメータに対応しています。この設定は、既存の接続を再利用する、確立可能な Reflection セッション数に影響します。`MaxStartups` パラメータに指定した最大セッション数に到達すると、以降のすべてのセッションで個別の SSH 接続と認証が必要になります。現在許可されている数より多くの未認証の同時セッションを確立する必要がある場合は、ssh サーバ管理者に問い合わせてください。
 - コマンドラインユーティリティでは、Secure Shell *構成ファイル* 『19 ページ』に接続再利用を構成できません。このファイルのキーワード `ConnectionReuse` は、スイッチ `-H` を使用してこの設定を含む *SSH 構成セクション* 『112 ページ』を指定しても、Reflection コマンドラインユーティリティによって必ず無視されます。
-

第 6 章

公開鍵認証

公開鍵認証は、公開/秘密鍵のペアに依存します。公開鍵認証は、サーバ (ホスト) とクライアント (ユーザ) の両方の認証に使用できます。Secure Shell クライアントに公開鍵認証を構成するには、クライアントに鍵のペアを作成 (またはインポート) し、公開鍵をホストにアップロードします。[Reflection Secure Shell の設定] ダイアログボックスの [ユーザ鍵] 『[35](#) ページ』 タブまたは Reflection 鍵エージェントのいずれかを使用して、クライアント認証用の公開鍵を作成および管理できます。鍵の構成方法に応じて、公開鍵認証を使用して接続を完了するために「パスフレーズ」 『[137](#) ページ』 の入力を求められる場合があります。

公開鍵認証の 1 つの形式は、X.509 証明書を使用して行われます。Reflection 証明書マネージャ 『[49](#) ページ』 あるいは Windows 証明書マネージャによって管理される証明書を使用して認証するように Reflection を構成できます。公開鍵認証は、認証に証明書を使用する場合に有効にする必要があります。

公開鍵認証の仕組み

公開鍵暗号では、公開/秘密鍵ペアと数値アルゴリズムを併用して、データの暗号化および復号化を行います。鍵の片方は公開鍵で、これは通信相手に自由に配布できます。もう片方の鍵は秘密鍵で、鍵の所有者が安全に保管しておく必要があります。秘密鍵によって暗号化されたデータは公開鍵によってのみ復号化でき、公開鍵によって暗号化されたデータは秘密鍵によってのみ復号化できます。

鍵が認証に使用される時は、認証される側のユーザが、公開/秘密鍵ペアの秘密鍵を使用してデジタル署名を作成します。受信者は、対応する公開鍵を使用して、このデジタル署名の信頼性を確認する必要があります。これは、受信者が、他方のユーザの公開鍵のコピーを所有し、その鍵の信頼性を信頼しなければならないことを意味します。

ユーザ鍵の管理

このセクション内

公開鍵認証の構成.....	32
ユーザ鍵一覧への鍵の追加.....	32
サーバへのクライアント公開鍵のアップロード.....	33
ユーザ鍵パスフレーズの変更.....	34
ユーザ鍵のエクスポート.....	34
[ユーザ鍵] タブ ([Reflection Secure Shell の設定] ダイアログボックス).....	35
[ユーザ鍵の生成] ダイアログボックス.....	36

公開鍵認証の構成

以下の手順では、公開鍵を使用してクライアント認証を構成します。

クライアントに公開鍵認証を構成するには手順に従います。

- 1 [Reflection Secure Shell の設定] ダイアログボックスを開きます。『[15](#) ページ』
- 2 [全般] タブで、[ユーザ認証] の [公開鍵] がオンになっていることを確認します (公開鍵認証のみを使用する場合は、その他のオプションをオフにします)。
- 3 [ユーザ鍵] タブをクリックします。[使用] 列で、現在指定されているホストへの認証に使用する鍵を 1 つまたは複数選択します。

注意: 鍵をこの一覧に追加するには、「[ユーザ鍵一覧への鍵の追加](#)『[32](#) ページ』」を参照してください。

- 4 [OK] をクリックします。

サーバに公開鍵認証を構成するには手順に従います。

- [公開鍵をホストにアップロード](#)します『[33](#) ページ』。

ユーザ鍵一覧への鍵の追加

[Reflection Secure Shell の設定] ダイアログボックスの [ユーザ鍵] 『[35](#) ページ』タブは、[公開鍵](#) 『[31](#) ページ』認証に使用可能な鍵の一覧を表示します。新しい鍵の生成または既存の鍵のインポートによって、一覧に鍵を追加できます。

Reflection を使用して新しい鍵のペアを生成するには

- 1 [Reflection Secure Shell の設定] ダイアログボックスを開きます。『[15](#) ページ』
- 2 [ユーザ鍵] タブをクリックします。
- 3 [鍵の生成] をクリックします。
- 4 鍵の名前、種類、長さを指定します (既定以外の鍵の名前または場所を指定する場合は、[参照] ボタンをクリックします)。
- 5 パスフレーズ『[137](#) ページ』を指定するか、[パスフレーズなし] をオンにします。

警告: [パスフレーズなし] を選択した場合、コンピュータに保存された秘密鍵は暗号化されず、この鍵にアクセスできる人は鍵を使用して認証できるようになります。

- 6 [生成] をクリックします。

注意: 既定で、鍵はユーザの `.ssh` フォルダ『135 ページ』に生成されます。

鍵エージェントを使用して新しい鍵のペアを生成するには

- 1 Reflection 鍵エージェントを起動してロック解除します (Windows の [スタート] メニューから、[すべてのプログラム] > [Attachmate Reflection] > [ユーティリティ] > [鍵エージェント] に進みます)。
- 2 [鍵の生成] をクリックします。
- 3 鍵の名前、種類、長さを指定し、[OK] をクリックします。

注意: 鍵エージェントを使用して作成する鍵は、暗号化された形でエージェントにより保存されます。

Reflection の鍵格納場所に鍵をインポートするには

- 1 [Reflection Secure Shell の設定] ダイアログボックスを開きます。『15 ページ』
- 2 [ユーザ鍵] タブをクリックします。
- 3 [インポート] をクリックします。
- 4 インポートしたい秘密鍵を検索して指定します。鍵のペアごとに、*.pub ファイルと拡張子のないファイルの 2 つが保存されています。秘密鍵は、拡張子のないファイルです。

注意: インポートした鍵は、ユーザの `.ssh` フォルダ『135 ページ』にある Reflection の鍵格納場所にコピーされます。

サーバへのクライアント公開鍵のアップロード

[ユーザ鍵] タブの [アップロード] ボタンを使用して、公開鍵を Secure Shell サーバにアップロードします。公開鍵は、安全な SFTP プロトコルを使用して転送されます。公開鍵をアップロードするには、パスワード認証 (または別の認証方式) を使用できる必要があります。公開鍵のアップロードに成功すると、その他の認証方式を無効にできます。

鍵をアップロードするには

- 1 [Reflection Secure Shell の設定] ダイアログボックスを開きます。『15 ページ』
- 2 [ユーザ鍵] タブから鍵を選択し、[アップロード] をクリックします (鍵を選択していない場合または証明書を選択していない場合、[アップロード] ボタンを使用できません)。
- 3 入力を求められたら、ホスト名、認証するユーザ名、ユーザパスワードを入力します。
- 4 ホストへの安全な接続が確立されると、ダイアログボックスが開き、この鍵をアップロードするホスト上の場所に関する情報が表示されます。通常は、この設定を変更する必要はありません。詳細については、以下の「注意」を参照してください。

[公開鍵のアップロード] ダイアログボックスに、転送に関する情報が表示されます。

- 5 [OK] をクリックしてこのダイアログボックスを閉じます。

注意

- Reflection for Secure IT、F-Secure、および SSH Communications (SSH Tectia) サーバが実行されているホストにアップロードした鍵は、RFC 4716 準拠形式でエクスポートされます。既定で、これらの鍵はユーザの `.ssh2` ディレクトリにインストールされ、適切な Key エントリが `authorization` ファイルに追加されます。このファイルがまだない場合は、新規作成され、適切なファイル権限が付与されます。
 - OpenSSH サーバが実行されているホストにアップロードした鍵は、OPENSSH 形式でエクスポートされます。既定で、これらの鍵はユーザの `.ssh` ディレクトリにある `authorized_keys` ファイルに追加されます。このファイルがまだない場合は、新規作成され、適切なファイル権限が付与されます。
-

ユーザ鍵パスフレーズの変更

ユーザ鍵の保護に使用するパスフレーズ『[137](#) ページ』を変更できます。

パスフレーズを変更するには

- 1 [Reflection Secure Shell の設定] ダイアログボックスを開きます。『[15](#) ページ』
- 2 [ユーザ鍵] タブをクリックし、一覧から鍵を選択します。
- 3 [パスフレーズの変更] をクリックします (鍵を選択していない場合、あるいは、Reflection 証明書マネージャまたは Windows 証明書マネージャのいずれかによって管理される証明書を選択している場合、このボタンは使用できません)。

ユーザ鍵のエクスポート

以下の手順を使用して、ユーザ鍵を新しい場所および形式にエクスポートします。

注意: 公開鍵を Secure Shell サーバにアップロードしたい場合、この手順を使用する必要はありません。[アップロード] ボタンを使用するだけでアップロードできます。指定したサーバに適合した鍵の形式が自動的に決定されます。詳細については、「サーバへの鍵のアップロード『[33](#) ページ』」を参照してください。

鍵をエクスポートするには

- 1 [Reflection Secure Shell の設定] ダイアログボックスを開きます。『[15](#) ページ』
- 2 [ユーザ鍵] タブから鍵を選択し、[エクスポート] をクリックします (鍵を選択していない場合、あるいは、Reflection 証明書マネージャまたは Windows 証明書マネージャのいずれかによって管理される証明書を選択している場合、このボタンは使用できません)。
- 3 選択した鍵のパスフレーズ『[137](#) ページ』を変更します。
- 4 (オプション)

目的	操作
秘密鍵をエクスポートに含める	[秘密鍵をエクスポートする] をオンにします。
鍵を OpenSSH 形式でエクスポートする	[OpenSSH 形式で保存する] をオンにします。

- 5 [公開鍵ファイル名] ダイアログボックスで、エクスポートする鍵の名前と場所を指定します。
- 6 [保存] をクリックします。

[ユーザ鍵] タブ ([Reflection Secure Shell の設定] ダイアログボックス)

表示方法 『15 ページ』

[ユーザ鍵] タブには、公開鍵認証 『31 ページ』を使った Secure Shell 接続の確立時にホストに対してクライアントセッションを認証する鍵を作成および管理するためのツールがあります。

Reflection は、使用可能なユーザ鍵の一覧を保存しています。Reflection で現在のホストに対する認証に使用したい鍵を指定するには、[使用] 列のチェックボックスをオンまたはオフにします。使用するために選択した鍵の一覧は、現在指定されている SSH 構成セクション 『112 ページ』用に保存されます。

一覧に含まれている鍵は次のとおりです。

- [ユーザ鍵の生成] 『36 ページ』 ダイアログボックスを使って作成した鍵
- [インポート] ボタンを使って追加した鍵
- Reflection の *Secure Shell* フォルダ 『135 ページ』に手作業でコピーした鍵
- Reflection 鍵エージェント内の鍵
- F-Secure の設定を Reflection に移行中にコピーされたユーザ鍵と認証エージェント鍵
- 個人の格納場所にある Windows 証明書マネージャの証明書
- 個人の格納場所にある Reflection 証明書マネージャ 『49 ページ』の証明書

次の鍵管理オプションも使用できます。

[鍵の生成]	[ユーザ鍵の生成] 『36 ページ』 ダイアログボックスが開きます。このダイアログボックスは、公開/秘密鍵のペアをユーザ鍵認証用に構成するために使用できます。
[パスフレーズの変更]	選択した鍵の保護に使用するパスフレーズ 『137 ページ』を変更します。
[鍵エージェント起動]	Reflection 鍵エージェントを起動します。
[インポート]	使用可能な鍵の一覧に秘密鍵を追加します。この機能を使用すると、他のアプリケーションを使用して作成された鍵に Reflection 内から簡単にアクセスできます。鍵をインポートすると、その鍵が Reflection の <i>Secure Shell</i> フォルダ 『135 ページ』にコピーされます。
[エクスポート]	公開鍵をエクスポートする 『34 ページ』か、公開/秘密鍵のペアをエクスポートします。
[エージェントに追加]	選択した鍵を Reflection 鍵エージェントに追加します。初回起動時に鍵エージェントを起動しなかった場合、または鍵エージェントがロックされている場合、鍵エージェントパスフレーズの入力を求められます。さらに、鍵をエージェントに追加する前に秘密鍵のパスフレーズを入力するよう求められます。

[アップロード]	現在指定されているホストに公開鍵をアップロードし ず『33 ページ』。
[削除]	選択した鍵を削除します。
[表示]	選択した鍵または証明書の内容を表示します。
[エージェントの転送を 許可する]	Reflection 鍵エージェント接続の転送を有効にします。 エージェント転送を有効にする場合は注意が必要です。 エージェントの UNIX ドメインソケットのリモートホ ストでファイル権限を回避できるユーザは、転送された 接続を介してローカルエージェントにアクセスできま す。攻撃者は鍵の情報を取得できませんが、エージェン トに読み込まれた識別情報を使用して、その鍵で操作を 実行して認証を有効にすることができます。

[ユーザ鍵の生成] ダイアログボックス

表示方法

- 1 [Reflection Secure Shell の設定] ダイアログボックスを開きます。『15 ページ』
- 2 [ユーザ鍵] タブをクリックします。
- 3 [鍵の生成] をクリックします。

このダイアログボックスは、ユーザ鍵認証用に公開/秘密鍵のペアを構成するために使用
します。

オプションは次のとおりです。

鍵ファイルの名前お よびパスのテキスト ボックス	生成された鍵のペアの名前とフォルダが表示されます。公開 鍵は、*.pub ファイル拡張子を付けて作成されます。秘密鍵 は、ファイル拡張子を付けずに同じ名前を使用して作成さ れます。
[参照]	その鍵に対して異なるファイル名または場所を指定する場 合は、[参照] ボタンをクリックします。
[鍵の種類]	鍵の生成に使用する鍵のアルゴリズムを指定します。
[鍵の長さ]	鍵のサイズを指定します。鍵のサイズを大きくすると、ある 程度までセキュリティは向上します。鍵のサイズを大きくす ると最初の接続が遅くなりますが、正常に接続した後は、鍵 のサイズはデータストリームの暗号化や解読の速度に影響 しません。使用する鍵の長さは、多くの要素に依存します。 その要素には、鍵の種類、鍵の有効期間、保護するデータ の値、潜在的な攻撃者にとって利用可能なリソース、この非対 称鍵とともに使用する対称鍵のサイズなどがあります。ニー ズに合った最適な選択をするには、セキュリティ管理者にお 問い合わせください。
[パスフレーズ]	この鍵を使った接続時に要求されるパスフレーズ『137 ページ』を指定します。注意: パスフレーズを使用しない場 合は、[パスフレーズなし] チェックボックスをオンにする必 要があります。

- [確認] 確認のためにパスワードを再入力します。
- [パスワードなし] パスワードの入力を求められることなく接続したい場合は、このチェックボックスをオンにします。注意: [パスワードなし] チェックボックスをオンにした場合、コンピュータに保存されている秘密鍵は、暗号化されません。
- [生成] 指定された場所に鍵のペアを生成します。公開鍵には *.pub ファイル拡張子を使用します。秘密鍵にはファイル拡張子がありません。

ホスト鍵の管理

このセクション内

ホスト鍵の確認の構成.....	38
優先するホスト鍵の種類の構成.....	39
既知のホストファイル.....	39
[ホスト鍵] タブ ([Reflection Secure Shell の設定] ダイアログボックス).....	40
[ホスト鍵の信頼性] ダイアログボックス	41

ホスト鍵の確認の構成

この手順を使用して、不明なホストへの接続時に Reflection がどのように動作するかを指定します。

ホスト鍵の確認を構成するには

- 1 [Reflection Secure Shell の設定] ダイアログボックスを開きます。『[15](#) ページ』
- 2 [ホスト鍵] タブをクリックします。
- 3 [厳格なホスト鍵の確認の履行] をクリックします。
- 4 次のオプションのどちらかを選択します。

選択する項目	結果
[ユーザに尋ねる] (既定値)	不明なホストに接続すると、[ホスト鍵の信頼性] 『 41 ページ』 確認のダイアログボックスが表示されます。
[はい]	厳格なホスト鍵の確認を履行します。ホストが信頼するホストでない場合、Reflection は接続しません。接続前に、ホスト鍵を信頼するホスト鍵の一覧に追加する必要があります。
[いいえ]	厳格なホスト鍵の確認を履行しません。確認のダイアログボックスが表示されずに、Reflection は接続します。信頼する鍵の一覧にホスト鍵を追加しません。

注意

- ホストが X.509 証明書を使用して認証するように構成されている場合、[厳格なホスト鍵の確認の履行] には影響しません。ホストがホスト認証のために証明書を提示し、信頼されたルート格納場所に必要な CA 証明書がない場合、接続に失敗します。
- この設定に加えた変更は、現在指定されている SSH 構成セクション 『[112](#) ページ』 に保存されます。
- Secure Shell 設定は、Secure Shell 構成ファイル 『[19](#) ページ』 に保存されます。また、このファイルを任意のテキストエディタで手作業で編集することにより Secure Shell 設定を構成することもできます。この設定の構成に使用されるキーワード 『[114](#) ページ』 は StrictHostKeyChecking です。

優先するホスト鍵の種類構成

[証明書よりも SSH 鍵を優先する] を使用して、ホスト鍵アルゴリズムの優先順位を指定します。この設定は、証明書と標準ホスト鍵認証の両方をサーバに構成する場合に便利です。SSH プロトコルでは、ホストの認証試行を 1 回しか許可しません。ホストが証明書を提示し、証明書を使用したホスト認証がクライアントに構成されていない場合、接続に失敗します (これは、複数の認証試行に対応しているユーザ認証とは異なります)。

優先するホスト鍵の種類 (標準 SSH 鍵または証明書) を構成するには

- 1 [Reflection Secure Shell の設定] ダイアログボックスを開きます。『15 ページ』
- 2 [ホスト鍵] タブをクリックします。
- 3 ホストが認証に標準ホスト鍵を使用するようにするには、[証明書よりも SSH 鍵を優先する] を選択します。
または
認証に証明書を使用するには、[証明書よりも SSH 鍵を優先する] をオフにします。

注意

- この設定に加えた変更は、現在指定されている *SSH 構成セクション* 『112 ページ』に保存されます。
- Secure Shell 設定は、*Secure Shell 構成ファイル* 『19 ページ』に保存されます。また、このファイルを任意のテキストエディタで手作業で編集することにより Secure Shell 設定を構成することもできます。この設定の構成に使用されるキーワードは HostKeyAlgorithms です。

既知のホストファイル

Reflection Secure Shell クライアントでは、既知のホストの一覧を既知のホストファイルに保存します。Reflection は、ユーザ固有のホストファイルとグローバルな既知のホストファイルの両方に対応しています。

ユーザ既知のホストファイル

ユーザ固有の既知のホストファイルは `known_hosts` と呼ばれ、ユーザの `.ssh` フォルダ 『135 ページ』にあります。これは、既定の既知のホストファイルです。Reflection は、以下の場合にこのファイルを自動的に更新します。

- [Reflection Secure Shell の設定] ダイアログボックスの [ホスト鍵] 『40 ページ』 タブにある [信頼されているホスト鍵] 一覧を更新した場合

または

- これまで知らなかったホストに接続し、ホスト鍵の信頼性 『41 ページ』メッセージに対して [常時] と応答した場合

グローバルな既知のホストファイル

システム管理者は、`ssh_known_hosts` という名前のシステム規模で既知のホストファイルを *Reflection アプリケーションデータフォルダ* 『136 ページ』に追加できます。

この場所で、既知のホストファイルにより PC のすべてのユーザのホスト一覧が提供されます。この一覧の鍵を表示できますが、[Reflection Secure Shell の設定] ダイアログボックスの [ホスト鍵] 『40 ページ』 タブにある [グローバルホスト鍵] 一覧で編集できません。

[ホスト鍵] タブ ([Reflection Secure Shell の設定] ダイアログボックス)

表示方法 『15 ページ』

[ホスト鍵] タブは、クライアントセッションに対してホストを認証する鍵の管理に使用します。このタブを使うと、信頼するホストの一覧を表示したり、ホスト鍵を追加または削除したり、不明なホストの処理方法を指定したりできます。

ホスト認証により、Secure Shell クライアントは Secure Shell サーバを確実に識別することができます。この認証は、公開鍵認証を使用して行われます。ホストの公開鍵がクライアントに事前にインストールされていない場合は、ユーザの初回接続時に、不明のホストであることを示すメッセージが表示されます。このメッセージには、ホストを識別するための指紋が含まれています。このホストが正しいホストであることを確認するには、正しい指紋であることを確認できるホストシステム管理者に問い合わせる必要があります。ホストが実際にユーザのホストであることが確認されるまでは、ユーザは「中間者」攻撃 (別のサーバがユーザのホストを装う) のリスクにさらされることとなります。プロンプトに答えて [常時] を選択すると、ホストが [信頼されているホスト鍵] 一覧に追加されます。ホスト管理者に問い合わせずに済むように、ホスト鍵を最初の接続前に [信頼されているホスト鍵] の一覧に追加することができます。

オプションは次のとおりです。

[厳格なホスト鍵の確認の履行]

不明なホストへの接続時に **ホスト鍵の確認** 『38 ページ』がどのように処理されるかを指定します。

[証明書よりも SSH 鍵を優先する]

Reflection での **ホスト鍵アルゴリズムの優先順位** 『39 ページ』を指定します。この設定がオフになっている場合 (既定)、Reflection はホスト鍵の前にホスト証明書を要求します。この設定がオンになっている場合、Reflection はホスト証明書の前にホスト鍵を要求します。

[信頼されているホスト鍵]

現在の Windows ユーザの信頼されているホストの一覧を表示します。この一覧の内容は、[インポート] および [削除] コマンドを使って変更できます。

既定で、この一覧にないホストへの接続を試行すると、その新しいホスト鍵を信頼するかどうかを確認するメッセージが表示されます。プロンプトに答えて [常時] を選択すると、ホストが [信頼されているホスト鍵] 一覧に追加されます。

[インポート]

ホストの公開鍵を [信頼されているホスト鍵] 一覧に追加します。

[削除]

選択した鍵を [信頼されているホスト鍵] 一覧から削除します。

警告: この場合、確認を求めるメッセージは表示されず、操作を元に戻すことはできません。

[グローバルホスト鍵]

コンピュータのすべてのユーザが使用できる、信頼されているホスト鍵の一覧を表示します。この一覧の項目は表示のみが可能で、編集できません。

システム管理者は、グローバルな既知のホストファイル 『39 ページ』を使って [グローバルホスト鍵] 一覧を変更することができます。

[ホスト鍵の信頼性] ダイアログボックス

この確認ダイアログボックスは、接続しているホストが信頼するホストではない場合に表示されます。この新しいホスト鍵を信頼して、接続を継続しますか？

ホスト認証により、Secure Shell クライアントは Secure Shell サーバを確実に識別することができます。この認証は、公開鍵認証を使用して行われます。ホストの公開鍵がクライアントに事前にインストールされていない場合は、ユーザの初回接続時に、不明のホストであることを示すメッセージが表示されます。このメッセージには、ホストを識別するための指紋が含まれています。このホストが正しいホストであることを確認するには、正しい指紋であることを確認できるホストシステム管理者に問い合わせる必要があります。ホストが実際にユーザのホストであることが確認されるまでは、ユーザは「中間者」攻撃（別のサーバがユーザのホストを装う）のリスクにさらされることとなります。オプションは次のとおりです。

- [常時] 接続を行って、信頼するホストの一覧にこのホストを追加します。信頼するホストの一覧からこのホストを削除するか、ホスト鍵が変更されないかぎり、以降、同じホストに接続する際に上記のメッセージが表示されることはありません。
- [今回のみ] 接続を行いますが、信頼するホストの一覧にこのホストを追加しません。次に同じホストに接続する時は上記のメッセージが表示されます。
- [いいえ] 接続を行わず、信頼するホストの一覧にこのホストを追加しません。

第 7 章

証明書の認証 (PKI)

PKI と証明書

PKI (Public Key Infrastructure) は、電子証明書を使用して安全に通信できるようにするためのシステムです。Reflection for Secure IT では、ホストとユーザの両方の認証に PKI を使用できます。

公開鍵認証のように、証明書認証では公開/秘密鍵のペアを使用してホストの身元を確認します。ただし、証明書認証を使用すると、公開鍵が [電子証明書『138 ページ』](#) 内に含まれ、この場合 2 つの鍵のペアが使用されます。例えばサーバ認証の場合、ホストは秘密鍵を 1 つ保有し、CA がもう 1 つの鍵を保有します。ホストは、CA から証明書を取得します。この証明書には、ホストに関する識別情報、ホスト公開鍵のコピー、CA の秘密鍵を使用して作成された [電子署名『137 ページ』](#) が含まれています。この証明書は、認証処理時にクライアントに送信されます。ホストから送信される情報の整合性を検証するには、クライアントが CA ルート証明書に含まれる CA の公開鍵のコピーを保有している必要があります。クライアントが、ホスト公開鍵のコピーを保有する必要はありません。

証明書認証により、公開鍵認証により発生するいくつかの問題が解決されます。例えばホスト公開鍵認証の場合、システム管理者はすべてのサーバのホスト鍵を各クライアントの既知のホスト格納場所に配布したり、クライアントユーザが既知のホストに接続する場合に、クライアントユーザに依頼してホストの身元を正しく確認する必要があります。証明書をホスト認証に使用すると、単一の CA ルート証明書を使用して複数のホストを認証できます。多くの場合、必要な証明書は Windows の証明書格納場所ですで使用可能です。

同様に、公開鍵をクライアント認証に使用すると、各クライアント公開鍵をサーバにアップロードし、その鍵を認識するようにサーバを構成する必要があります。証明書認証を使用すると、単一の CA ルート証明書を使用して複数のクライアントユーザを認証できます。

電子証明書の格納場所

電子証明書は、コンピュータの証明書格納場所に保存されています。証明書格納場所には、相手の身元を確認するのに使用する証明書が含まれています。また、自分の身元を相手に示すのに使用する個人用証明書が含まれていることもあります。個人用証明書は、コンピュータにある秘密鍵に関連付けられています。

Reflection は、次のどちらかまたは両方の格納場所にある電子証明書を使用するように設定できます。

▪ Windows の証明書格納場所

この格納場所は、Reflection、Web ブラウザ、メールクライアントなど、複数のアプリケーションが使用できます。この格納場所の証明書のいくつかは、Windows オペレーティングシステムのインストール時にインストールされます。また、インターネットサイトに接続して信頼関係を確立したり、ソフトウェアをインストールしたりした時や、暗号化された電子メールや電子署名のある電子メールを受け取った時にも追加されます。Windows の格納場所に証明書を手作業でインポートすることもできます。この格納場所の証明書は、Windows の証明書マネージャを使用して管理します。

▪ Reflection の証明書格納場所

この格納場所は、Reflection アプリケーションだけが使用します。この格納場所に証明書を追加するには、証明書を手作業でインポートする必要があります。証明書はファイルからインポートでき、スマートカードなどのハードウェアトークン上の証明書を使用することもできます。この格納場所の証明書は、Reflection の証明書マネージャ『49 ページ』を使用して管理します。

Reflection のアプリケーションは、Reflection の格納場所にある証明書のみ、または Windows の格納場所と Reflection の格納場所を両方を使用して認証するように構成することができます。Windows の証明書格納場所の証明書を使用したホスト認証を有効にすると、既存の証明書を使用して認証できるので、証明書をインポートしなくて済むことがあります。Windows の証明書格納場所の証明書を使用した認証を無効にすると、認証で使用する証明書を詳細に制御できるようになります。詳細については、「Windows の証明書格納場所を使用した認証の有効化と無効化『45 ページ』」を参照してください。

証明書を使用したクライアント認証の構成

電子証明書『138 ページ』は、Secure Shell クライアントセッションでのホスト認証とクライアント認証『138 ページ』の両方またはそのいずれかで使用できます。証明書は必須ではなく、既定では使用されません。このトピックでは、証明書を使用して認証されるように Reflection for Secure IT クライアントを構成する方法について説明します。Secure Shell サーバの構成方法については、サーバのマニュアルを参照してください。

クライアントに証明書認証を構成するには

- 1 個人用証明書および関連する秘密鍵を含むファイル (*.pfx または *.p12 ファイル) を入手します (証明書は、認証局から入手できます)。
- 2 このファイルを使用して、証明書を Reflection 証明書マネージャ『49 ページ』または Windows の証明書格納場所の [個人] タブにインポートします。
- 3 Reflection for Secure IT で、[Reflection Secure Shell の設定]『15 ページ』ダイアログボックスを開きます。
- 4 [全般] タブで、[ユーザ認証] の [公開鍵] が選択されていること (既定) を確認します。
- 5 [ユーザ鍵] タブで、使用可能な鍵の一覧から使用したい証明書を検索し、使用できるようにするために [使用] 列のチェックボックスをオンにします。

証明書を使用したサーバ認証の構成

証明書を使用してホストを認証するために Reflection クライアント側で必要な構成は、サーバが提示する証明書の認証に必要な CA 証明書をインストールすることだけです。証明書をインストールする要件は、Reflection がどのように設定されていて、証明書がどのように作成されたかによって異なります。

- 証明書が VeriSign や Thawte などのよく知られた認証局 (CA) から取得されたもので、Windows の証明書格納場所を使用してホスト証明書に対応するように Reflection を構成した場合は、使用しているコンピュータに証明書をインストールする必要がないことがあります。また、発行者を信頼された CA として識別する証明書が、使用しているシステムの信頼されたルート認証局の一覧にすでに含まれている場合があります。
- Reflection の格納場所を使用して認証を要求するように Reflection を構成した場合、各クライアントコンピュータに必要な CA 証明書を Reflection の格納場所にインポートする必要があります。
- 会社独自の認証局を作成した場合は、各クライアントコンピュータはその認証局のルート証明書をインポートする必要があります。構成に応じて、Windows の証明書格納場所または Reflection の証明書格納場所にインポートします。

Windows の証明書格納場所の使用の有効化と無効化

Reflection Secure Shell セッションおよび SSL/TLS セッションでは、ホスト認証およびユーザ認証の両方で [電子証明書『138 ページ』](#)を使用することができます。Reflection のアプリケーションは、Reflection の格納場所にある証明書のみ、または Windows の格納場所と Reflection の格納場所の両方を使用して認証するように設定することができます。

ホスト認証

Windows の証明書格納場所を使用した認証を有効にすると、ホスト認証で使用する証明書をインポートしなくて済むことがあります。ホスト証明書が VeriSign や Thawte などのよく知られた [認証局『138 ページ』](#) (CA) から取得されたものである場合、発行者を信頼された CA として識別する証明書が、使用しているシステムの信頼されたルート認証局の一覧にすでに含まれています。システムの格納場所を使用できるように設定すると、Reflection クライアントは Reflection の格納場所とシステムの格納場所の両方で証明書を探します。

Windows の証明書格納場所の使用を無効にすると、認証で使用する証明書を詳細に制御できるようになります。証明書は、さまざまな方法で Windows の格納場所に追加できます。また、Reflection セッションの認証で使用する証明書を限定できます。Windows の格納場所を使用できないようにした場合、Reflection の格納場所にインポートした証明書だけがホスト認証で使用されます。

Windows の格納場所にある証明書を使用したホスト認証を有効 (または無効) にするには、以下の手順に従います。

- 1 [\[Reflection 証明書マネージャ\]『49 ページ』](#)を開きます。
- 2 [\[信頼された認証局\]『50 ページ』](#) タブをクリックします。
- 3 [\[システムの格納場所にある証明書を使用して SSH に接続する\]](#) または [\[システムの格納場所にある証明書を使用して SSL/TLS に接続する\]](#) を選択 (または選択解除) します。

ユーザ認証

Reflection は、Windows の格納場所と Reflection の格納場所にある個人用証明書を同じように使用します。利用可能な個人用証明書には、Windows の個人用保存場所、Reflection の個人用保存場所『49 ページ』、およびスマートカードなどの構成済みのハードウェアトークン『53 ページ』上の証明書が含まれます。

- Reflection Secure Shell セッションを設定した場合は、ユーザ認証で使用する証明書を [Reflection Secure Shell の設定] ダイアログボックスの [ユーザ鍵] タブで指定する必要があります。
- Reflection SSL/TLS セッションを設定した場合は、それらの格納場所にあるすべての証明書が自動的にユーザ認証で使用可能になります。

証明書取り消しの確認の構成

Reflection SSL/TLS 接続と Secure Shell 接続では、電子証明書『138 ページ』を使用したホスト認証を構成できます。失効していない証明書を実際に使用するには、CRL『135 ページ』または OCSP レスポンダを使用して証明書の取り消しを確認するように Reflection を構成します。

Reflection で CRL の確認が有効になっている場合は、証明書の CRL Distribution Point (CDP) フィールドに指定されているすべての場所で CRL が必ず確認されます。また、LDAP ディレクトリにある CRL を確認したり、OCSP レスポンダを使用するように Reflection を構成することもできます。

Reflection では、証明書取り消しの確認の既定値は現在のシステム設定に基づいて決まります。システムが CRL の確認を行うように構成されている場合は、既定ですべての Reflection セッションにおいて CRL を使用して証明書取り消しを確認されます。

注意: Reflection が DOD PKI モード『130 ページ』で実行されている場合、証明書取り消しは常に有効であり、無効にすることはできません。

システムでの CRL の確認を有効にするには

- 1 Internet Explorer を起動します。
- 2 [ツール] > [インターネットオプション] > [詳細設定] を選択します。
- 3 [セキュリティ] の下の [サーバー証明書の取り消しを確認する] をオンにします。

Reflection では、CRL または OCSP レスポンダを使用した証明書取り消し確認を行えます。

Secure Shell セッションでの CRL の確認を有効にするには

- 1 [Reflection Secure Shell の設定] ダイアログボックスの [PKI] タブを開きます。
- 2 [OCSP を使用する] または [CRL を使用する] をオンにします。

SSL/TLS セッションでの CRL の確認を有効にするには

- 1 [セキュリティのプロパティ] ダイアログボックスを開きます。
- 2 [SSL/TLS] タブで [PKI の構成] をクリックします([SSL/TLS セキュリティを使用する] がオンになっている必要があります)。
- 3 [OCSP を使用する] または [CRL を使用する] をオンにします。

注意: 証明書に必要な CRL レスポンドおよび OCSP レスポンドは、証明書の AIA 拡張および CDP 拡張に指定されます。この情報が証明書で提供されない場合、[Reflection 証明書マネージャ]の [OCSP] 『52 ページ』 タブおよび [LDAP] 『51 ページ』 タブを使用して構成できます。

LDAP ディレクトリを使用した中間証明書の配布

Reflection SSL/TLS 接続と Secure Shell 接続では、*電子証明書* 『138 ページ』を使用したホスト認証を構成できます。*Reflection 証明書マネージャ* 『45 ページ』の構成方法に応じて、Reflection の格納場所にある証明書のみ、または Windows および Reflection の両方の格納場所にある証明書が Reflection で使用されます。Windows の格納場所には、中間証明書と信頼されたルート証明書が保存されます。Reflection の格納場所には、信頼されたルート証明書のみが保存されます。また、LDAP サーバから中間証明書を検索するように Reflection を構成することもできます。

LDAP ディレクトリに保存されている中間証明書を検索するように Reflection を構成するには、[Reflection 証明書マネージャ] の [LDAP] 『51 ページ』 タブで LDAP サーバ (1 台または複数) を指定します。

LDAP サーバの構成

Reflection で LDAP ディレクトリ内の証明書を検索できるのは、LDAP 識別名 (DN) が証明書の件名フィールドの内容と完全に一致する場合のみです。例えば、証明書の件名フィールドに以下のオブジェクトが表示されるとします。

- CN = Some CA
- O = Acme
- C = US

この場合、LDAP ディレクトリのエントリの DN は「CN = Some CA, O=Acme, C = US」である必要があります。

この DN で識別される LDAP エントリの属性は、以下のいずれかを含む必要があります (Reflection ではこれらの属性を上から下に検索します)。

属性	OID (Object Identifier - オブジェクト識別子)
userCertificate;binary	2.5.4.36
cACertificate;binary	2.5.4.37
userCertificate	2.5.4.36
cACertificate	2.5.4.37
mosaicKMandSigCertificate	2.16.840.1.101.2.1.5.5
sdnsKMandSigCertificate	2.16.840.1.101.2.1.5.3
fortezzaKMandSigCertificate	2.16.840.1.101.2.1.5.5
crossCertificatePair;binary	2.5.4.40
crossCertificatePair	2.5.4.40

[PKI] タブ ([Reflection Secure Shell の設定] ダイアログボックス)

表示方法 『15 ページ』

このタブを使って、Reflection Secure Shell セッション用に PKI を構成します。

オプションは次のとおりです。

- | | |
|---------------------------------|---|
| [証明書のホスト名と対象ホスト名が一致するかどうかを確認する] | ホストの証明書の検証時にホスト名の一致を確認するかどうかを指定します。この設定が有効になっている場合 (既定値)、Reflection で構成したホスト名が、証明書の CommonName フィールドまたは SubjectAltName フィールドに入力されているホスト名に一致していなければなりません。 |
| [OCSP を使用する] | ホストの証明書の検証時に、OCSP (Online Certificate Status Protocol) レスポンダを使用して Reflection で証明書の失効を確認するかどうかを指定します。証明書自体の AIA 拡張に OCSP レスポンダが指定される場合もあります。Reflection 証明書マネージャの [OCSP] 『52 ページ』 タブを使用して OCSP レスポンダを指定することもできます。 |
| [CRL を使用する] | ホストの証明書の検証時に、CRL 『135 ページ』 (Certificate Revocation Lists) を使用して Reflection で証明書の失効を確認するかどうかを指定します。証明書自体の CDP 拡張に CRL が指定される場合もあります。Reflection 証明書マネージャの [LDAP] 『51 ページ』 タブを使用して CRL を指定することもできます。 |
| | 注意: この設定の既定値は、システムの現在の CRL チェックの設定によって決まります。システム設定を表示して編集するには、Internet Explorer を起動して、[ツール] - [インターネットオプション] - [詳細設定] に進みます。[セキュリティ] の下の [サーバー証明書の取り消しを確認する] を探します。 |
| [Reflection 証明書マネージャ] | Reflection 証明書マネージャを開きます。ここで Reflection の格納場所内の証明書を管理し、PKI 構成を指定することができます。 |
| [システム証明書の表示] | Windows 証明書マネージャを開きます。ここでシステムの格納場所にある証明書を管理することができます。 |

注意

- このダイアログボックスで構成した設定は、*Secure Shell 設定ファイル* 『19 ページ』に保存されます。また、このファイルを任意のテキストエディタで手作業で編集することにより Secure Shell 設定を構成することもできます。
 - この構成ファイルの内の設定は、現在指定されている *SSH 構成セクション* 『112 ページ』用に保存されます。
-

第 8 章

Reflection 証明書マネージャ

表示方法 『[49](#) ページ』

Reflection アプリケーションは、Windows の証明書格納場所または Reflection の証明書格納場所 (あるいはその両方) にある **電子証明書** 『[138](#) ページ』を使用して認証できます。Reflection の証明書格納場所は、Secure Shell セッションまたは SSL/TLS セッションでの認証で使用することができます。

Reflection 証明書マネージャを使用して、Reflection の格納場所にある電子証明書を管理したり、Reflection の PKI 対応のほかのオプションを構成します。

このセクションの内容

Reflection 証明書マネージャの開き方.....	49
[個人] タブ ([Reflection 証明書マネージャ]).....	49
[信頼された認証局] タブ ([Reflection 証明書マネージャ]).....	50
[LDAP] タブ ([Reflection 証明書マネージャ]).....	51
CRL 確認のための LDAP サーバの設定.....	52
[OCSP] タブ ([Reflection 証明書マネージャ]).....	52
[PKCS#11] タブ ([Reflection 証明書マネージャ]).....	53
[PKCS#11 プロバイダ] ダイアログボックス.....	53

Reflection 証明書マネージャの開き方

Reflection 証明書マネージャを起動するには、以下の手順を使用します。

[Reflection 証明書マネージャ] を開くには

- 1 [Reflection Secure Shell の設定] ダイアログボックスを開きます。 『[15](#) ページ』
- 2 [PKI] タブで [Reflection 証明書マネージャ] をクリックします。

[個人] タブ ([Reflection 証明書マネージャ])

表示方法 『[49](#) ページ』

このタブは、Reflection の格納場所にある個人用 **証明書** 『[138](#) ページ』の管理に使用します。個人用証明書はユーザ (クライアント) 認証に使用されます。

オプションは次のとおりです。

- [**インポート**] Reflection の格納場所に証明書を追加します。インポートしたファイル (通常、*.pfx または *.p12) には秘密鍵が含まれている必要があります。ファイルがどのように作成されたかによっては、ファイルをインポートする前にパスワードの入力が要求されることがあります。

Reflection の格納場所にある秘密鍵を保護できるように、パス

フレーズ『[137](#) ページ』の入力が要求されます。パスフレーズを指定すると、この証明書を使用してホストに認証する時に、このパスフレーズを入力するように要求されます。

- [削除] 選択した証明書を Reflection の格納場所から削除します。
- [表示] 選択した証明書を表示します。
- [パスフレーズの変更] 選択した証明書のパスフレーズ『[137](#) ページ』を変更します。

[信頼された認証局] タブ ([Reflection 証明書マネージャ])

表示方法『[49](#) ページ』

このタブは、信頼された認証局から提供された、Reflection の格納場所にある証明書の管理に使用します。Reflection は、信頼された認証局の格納場所にあるあらゆる証明書をホスト (サーバ) の認証に使用します。

- [インポート] Reflection の格納場所に証明書 (通常、*.cer または *.crt) を追加します。
- [削除] 選択した証明書を Reflection の格納場所から削除します。
- [表示] 選択した証明書を表示します。
- [システムの格納場所にある証明書を使用して SSH に接続する] この項目をオンにすると、Reflection は Secure Shell 接続を確立する際に、Windows の格納場所にある証明書を使用してホストを認証します。

Reflection アプリケーションが、Reflection の格納場所にある証明書だけを使用してホストを認証するには、この設定をオフにします。
- [システムの格納場所にある証明書を使用して SSL/TLS に接続する] この項目をオンにすると、Reflection は SSL/TLS 接続を確立する際に、Windows の格納場所にある証明書や Reflection の格納場所にインポートした証明書を使用してホストを認証します。

Reflection アプリケーションが、Reflection の格納場所にある証明書だけを使用してホストを認証するには、この設定をオフにします。

[LDAP] タブ ([Reflection 証明書マネージャ])

表示方法 『[49](#) ページ』

LDAP (Lightweight Directory Access Protocol) は、情報の中央位置への保存およびユーザへの情報の配布に使用できる標準的なプロトコルです。管理者は、LDAP サーバを構成して、証明書を使用して認証するユーザが必要とする情報を配布することができます。これには、次の情報が含まれます。

- CRL (Certificate Revocation List) - 使用されている証明書が認証局によって失効されていないことを保証するために使用します。
- 中間証明書 - サーバ証明書から信頼されたルート認証局への有効な証明書経路を確立するのに必要です。

[Reflection 証明書マネージャ] の [LDAP] タブで、この情報を配布する LDAP サーバを一覧表示します。オプションは次のとおりです。

[追加] LDAP サーバをリストに追加します。次の URL 形式を使用してサーバを指定します。

```
ldap://hostname[:portnumber]
```

例えば、次のように入力します。

```
ldap://ldapservers.myhost.com:389
```

[変更] サーバの URL を編集します。

[削除] 選択したサーバを一覧から削除します。

LDAP ディレクトリの構成

LDAP ディレクトリに保存されている情報を Reflection で処理する方法の詳細については、以下のリンクを参照してください。

- CRL 『[52](#) ページ』
- 中間証明書 『[47](#) ページ』

注意

- LDAP サーバで CRL を確認するように構成する必要はありません。Reflection で CRL の確認が有効になっている場合は、証明書の CRL Distribution Point (CDP) フィールドに指定されているすべての場所で CRL が必ず確認されます。LDAP サーバの構成は、CRL の一覧を取得する別の方法として使用できます。
 - Reflection は、SSL を使用して LDAP データを転送するのに LDAPS 方式 (例: ldaps://hostname:port) を使用するサーバの URL に対応していません。
-

CRL 確認のための LDAP サーバの設定

Reflection で LDAP ディレクトリ内の CRL を検索できるのは、LDAP 識別名 (DN) が CRL の発行者フィールドの内容と完全に一致する場合のみです。例えば、CRL の発行者フィールドに以下のオブジェクトが表示されるとします。

- CN = Some CA
- O = Acme
- C = US

この場合、LDAP ディレクトリのエントリの DN は「CN = Some CA, O=Acme, C = US」である必要があります。

この DN で識別される LDAP エントリの属性は、以下のいずれかを含む必要があります (Reflection ではこれらの属性を上から下に検索します)。

属性	OID (Object Identifier - オブジェクト識別子)
certificateRevocationList;binary	2.5.4.39
authorityRevocationList;binary	2.5.4.38
certificateRevocationList	2.5.4.39
authorityRevocationList	2.5.4.38
deltaRevocationList;binary	2.5.4.53
deltaRevocationList	2.5.4.53
mosaicCertificateRevocationList	2.16.840.1.101.2.1.5.45
sdnsCertificateRevocationList	2.16.840.1.101.2.1.5.44
fortezzaCertificateRevocationList	2.16.840.1.101.2.1.5.45

[OCSP] タブ ([Reflection 証明書マネージャ])

表示方法 『[49](#) ページ』

証明書取り消しの確認のために 1 つまたは複数の OCSP レスポンダを構成します。一覧にサーバを追加するには、[追加] をクリックします。既定では、追加するすべてのサーバの証明書の有効性が、一覧の最初のサーバから問い合わせられます。一覧からサーバを削除せずにサーバの証明書確認を無効にするには、そのサーバのチェックボックスをオフにします。

[追加] OCSP サーバをリストに追加します。次の URL 形式を使用してサーバを指定します。

URL:portnumber

例えば、次のように入力します。

www.ocspresponder.com:80

[変更] サーバの URL を編集します。

[削除] 選択したサーバを一覧から削除します。

[PKCS#11] タブ ([Reflection 証明書マネージャ])

表示方法 『49 ページ』

[PKCS#11] タブで、スマートカードや USB トークンを使用した認証を構成します。ハードウェアデバイスは、PKCS#11 『135 ページ』仕様に従っている必要があります。

このタブには、現在使用可能なすべてのデバイスと、これらのデバイスに存在しているすべての証明書または公開鍵が表示されます。チェックボックスをオンにしてデバイスの使用を有効にすると、Reflection はデバイス上の証明書や鍵をユーザ認証に自動的に使用します。

Reflection がハードウェアトークンを使用して認証できるようにするには、トークンプロバイダによって提供されたソフトウェアをインストールする必要があります。トークンを使用して認証を構成する場合は、ハードウェアデバイスへのアクセスを提供できるように、そのプロバイダが使用するライブラリファイル (*.dll) の名前と場所も知っている必要があります。

オプションは次のとおりです。

[プロバイダ] 一覧	現在使用可能なデバイスを表示します。表示されているデバイスでの認証を無効にするには、チェックボックスの選択をオフにします。
[デバイスの内容]	選択したデバイス上で利用可能な鍵と証明書が表示されます。
[証明書の表示]	選択した証明書を表示します。
[トークンが削除された場合は自動的に切断する]	選択すると、トークンが存在している時のみ接続が確立されます。
[切断待ち時間 (秒)]	トークンが削除されてから切断するまでの時間を秒で指定します。

[PKCS#11 プロバイダ] ダイアログボックス

表示方法 『[49](#) ページ』

オプションは次のとおりです。

- | | |
|--------------|---|
| [プロバイダの DLL] | ハードウェアデバイスへのアクセスに使用するライブラリのファイル名と場所を指定します。通常、このファイルは Windows の System フォルダにインストールされます。不明な場合は、デバイスの製造元に問い合わせてください。 |
| [スロット ID] | 認証に使用するカードが入っているカードスロットを識別します。 |
| [追加パラメータ] | ハードウェアデバイス上の情報にアクセスするのに必要な追加パラメータを指定します。 |

第 9 章

Secure Shell セッションの GSSAPI (Kerberos) 認証

GSSAPI 認証に対する Reflection Kerberos の使用

この手順では、Secure Shell サーバに対する認証に Windows のドメイン資格情報を使用するように Reflection Kerberos を構成します。Secure Shell 接続の Kerberos 認証は、Secure Shell 構成ファイルに現在指定されている SSH 構成セクションで有効になります。

システム管理者が Kerberos 構成ファイルを PC にインストールしている場合、Reflection アプリケーションの初回起動時に、Reflection Kerberos が自動的に構成されます。Reflection Kerberos 設定はユーザ単位でレジストリに保存され、Kerberos クライアントを使用するすべての Reflection アプリケーションで使用できます。

GSSAPI 認証に Reflection Kerberos を使用するには

- 1 [Reflection Secure Shell の設定] ダイアログボックスを開きます。『[15](#) ページ』
- 2 [全般] タブの [ユーザ認証] で、[GSSAPI/Kerberos] チェックボックスをオンにします。
- 3 [GSSAPI] タブで、[Reflection Kerberos] をオンにします。
- 4 [構成] をクリックします。
- 5 [Reflection Kerberos 初期構成] ダイアログボックスで、プリンシパル名、レルム、KDC ホストを入力します。システムに Kerberos がすでに構成されている場合、代わりに Reflection Kerberos マネージャが起動します。

注意: 認証後、Reflection Kerberos は Kerberos 発券許可チケット (TGT) をホストに転送します。チケット転送を無効にするには、「*Secure Shell セッションにおける Kerberos チケット転送* 『[55](#) ページ』」を参照してください。

Secure Shell セッションにおける Kerberos チケット転送

既定で、認証後 Reflection は Kerberos 発券許可チケット (TGT) をホストに転送します。

以下のいずれかの方法で、チケット転送を無効にできます。

- [Reflection Secure Shell の設定] ダイアログボックスの [GSSAPI] 『[56](#) ページ』 タブの [資格情報を委任する] 設定をオフにします。この設定は、Secure Shell プロトコルバージョン 2 接続に影響します。
- *Secure Shell 構成ファイル* 『[19](#) ページ』を編集します。使用するプロトコルに応じて、以下の行の一方または両方を使用します。最初の行ではプロトコルバージョン 1 のチケット転送を無効にし、2 番目の行ではプロトコルバージョン 2 のチケット転送を無効にします。

```
KerberosTgtPassing no
```

```
GssapiDelegateCredentials no
```


- プリンシパルプロファイルで使用されるレルムのチケット転送を無効にするには、Reflection Kerberos マネージャを使用します (システムで使用可能な場合)。これらの変更は Reflection Kerberos を使用するように構成された Secure Shell セッションに影響しますが、SSPI を使用するように構成されたセッションには影響しません。上記の方法のいずれかを使用してチケット転送を構成している場合、Reflection Kerberos マネージャで行われた変更は無視されます。

GSSAPI Secure Shell セッションのサービスプリンシパルの指定

サービスプリンシパル名とは、Reflection が Kerberos Key Distribution Center (KDC) へサービスチケット要求を送信する時に使用する名前です。形式は次のとおりです。

```
hostname.domain.com@REALM
```

Reflection によって使用される名前は、[Reflection Secure Shell の設定] ダイアログボックスの [GSSAPI] 『56 ページ』 タブで構成する設定によって決まります。[既定のサービスプリンシパル名を使用する] がオン (既定) の場合、ホスト名の値は接続先の Secure Shell サーバの名前で、レルムの値は選択した GSSAPI プロバイダによって決まります。

- *Reflection Kerberos* 『55 ページ』 を使用している場合、レルム名は、既定プリンシパルプロファイルで指定したものになります。
- SSPI を使用している場合、レルム名は Windows のドメイン名です。

既定以外の値を指定するには、[サービスプリンシパル] 設定を使用します。GSSAPI プロバイダとして SSPI を選択している場合、この設定を使用して、Windows ドメインとは異なるレルムのサービスプリンシパルを指定できます。次のように、完全なホスト名の後に @、そしてレルム名を続けてください。

```
myhost.myrealm.com@MYREALM.COM
```

[GSSAPI] タブ ([Reflection Secure Shell の設定] ダイアログボックス)

表示方法 『15 ページ』

[Reflection Secure Shell の設定] ダイアログボックスの [GSSAPI] タブを用いて、GSSAPI 認証に関する設定を指定します。このタブの項目は、[全般] 『16 ページ』 タブの [ユーザ認証] 一覧で [GSSAPI/Kerberos] が選択されている場合にのみ使用できます。

オプションは次のとおりです。

[SSPI]	Microsoft の Security Services Provider Interface (SSPI) を使用すると、Secure Shell サーバに対する認証に Windows ドメインのログイン資格情報を使用できます。この設定を行うと、Reflection Kerberos クライアントの構成が不要になるため、セットアップが簡単になります。
[Reflection Kerberos]	Kerberos/GSSAPI 認証のために Reflection Kerberos クライアントを使用します。Reflection Kerberos クライアントを使用して接続する前に、使用するコンピュータで Reflection Kerberos を構成する必要があります。
[構成]	Reflection Kerberos クライアントを構成します。このボタンは、[Reflection Kerberos] が GSSAPI プロバイダとして選択されている場合のみ使用できます。
[資格情報を委任する]	GSSAPI がホストへ Kerberos 発券許可チケット (TGT) を転送するかどうかを指定します。

[既定のサービスプリンシパル名を使用する]

Kerberos Key Distribution Center (KDC) へサービスチケット要求を送信する時に `Reflection` が使用する名前を指定します。ホスト名の値は、接続先の Secure Shell サーバの名前になります。レルムの値は、選択した GSSAPI プロバイダによって異なります。

[サービスプリンシパル]

既定ではないサービスプリンシパル値を指定します。

第 10 章

ポート転送

トンネリングとしても知られるポート転送は、アクティブなセッションにおける Secure Shell チャンネルを通じて通信をリダイレクトするための方法を提供します。ポート転送が構成されると、指定のポートへ送信されるすべてのデータは、保護されたチャンネルを通じてリダイレクトされます。ローカルポート転送またはリモートポート転送のいずれかを構成できます。「ローカル」および「リモート」という用語は、Secure Shell クライアントを基準にしてリダイレクトされたポート位置を示します。Reflection では、TCP 通信と FTP 通信の両方のローカルポート転送に対応しています。リモートポート転送は、TCP 通信のみに対応します。

用語集

ポート転送には、クライアントアプリケーションとサーバアプリケーションの 2 つのセット (Secure Shell クライアントとサーバ、およびデータが転送されるクライアント/サーバのペア) が必要です。このガイドでは、ポート転送に関連して以下のように定義された用語を使用します。

用語	定義
Secure Shell サーバ	Reflection for Secure IT サーバデーモン
Secure Shell サーバホスト	Secure Shell サーバを実行しているコンピュータ
Secure Shell クライアント	Reflection for Secure IT クライアントアプリケーション
Secure Shell クライアントホスト	Secure Shell クライアントを実行しているコンピュータ
アプリケーションクライアント	そのデータを転送したいクライアント/サーバのペアのクライアントプログラム。例えば、メールクライアントまたは Web ブラウザです。
アプリケーションクライアントホスト	アプリケーションクライアントを実行しているコンピュータ。通常は Secure Shell サーバホストまたは Secure Shell クライアントホストのいずれかになりますが、3 番目のホストにすることもできます。
アプリケーションサーバ	メールサーバまたは Web サーバなど、アプリケーションクライアントと通信するサーバプログラム
アプリケーションサーバホスト	サーバアプリケーションを実行しているコンピュータ。Secure Shell サーバホストまたは Secure Shell クライアントホストのいずれか、または 3 番目のホストにすることもできます。

ローカルポート転送

ローカルポート転送を使用して、Secure Shell クライアントと同じコンピュータ上で実行されているアプリケーションクライアントからデータを安全に転送できます。ローカルポート転送を構成する時は、データの転送に使用する任意のローカルポート、およびデータを受信する着信先ホストとポートを指定します。ローカルポート転送は以下のように機能します。

1. Secure Shell 接続が確立されると、Secure Shell クライアントは、指定されたローカルポートを使用して、ローカルコンピュータ (Secure Shell クライアントを実行しているコンピュータ) 上のリスニングソケットを開きます。ほとんどの場合、このソケットは、Secure Shell クライアントホストで実行されているアプリケーションのみが使用できます。

ゲートウェイポート設定は、ローカルに転送されるポートをリモートアプリケーションが使用できるかどうかを制御します。既定ではこの設定は無効で、クライアントは、ローカルポート転送のためにソケットを開いた時に、ループバックアドレス (「localhost」または 127.0.0.1) を使用します。これにより、他のコンピュータで実行中のアプリケーションは、転送されるポートに接続できなくなります。ゲートウェイポートを有効にすると、リモートアプリケーションクライアントは、Secure Shell クライアントの Ethernet アドレス (IP アドレス、URL、DNS 名など) を使用してソケットを開くことができます。例えば、acme.com で実行中の Secure Shell クライアントがポート 8088 を転送するよう構成されているとします。ゲートウェイポートが無効な場合、転送されるソケットは localhost:8088 です。ゲートウェイポートが有効な場合、転送されるソケットは acme.com:8088 です。

注意: ゲートウェイポートを有効化すると、使用しているクライアントホスト、ネットワーク、接続のセキュリティの低下を招きます。この理由は、リモートアプリケーションが、認証なしで、システム上の転送されたポートを使用することが可能になるからです。

2. アプリケーションクライアントは、(直接、アプリケーションサーバホストおよびポートへではなく) 転送されたポートへ接続するように構成されます。当該クライアントが接続を確立すると、すべてのデータがリスニングポートへ送信され、次に Secure Shell クライアントへリダイレクトされます。
3. Secure Shell クライアントはデータを暗号化して、Secure Shell チャンネルを介して Secure Shell サーバへ安全に送信します。
4. Secure Shell サーバはデータを受信し、復号化して、そのデータをアプリケーションサーバによって使用される着信先ホストおよびポートへリダイレクトします。

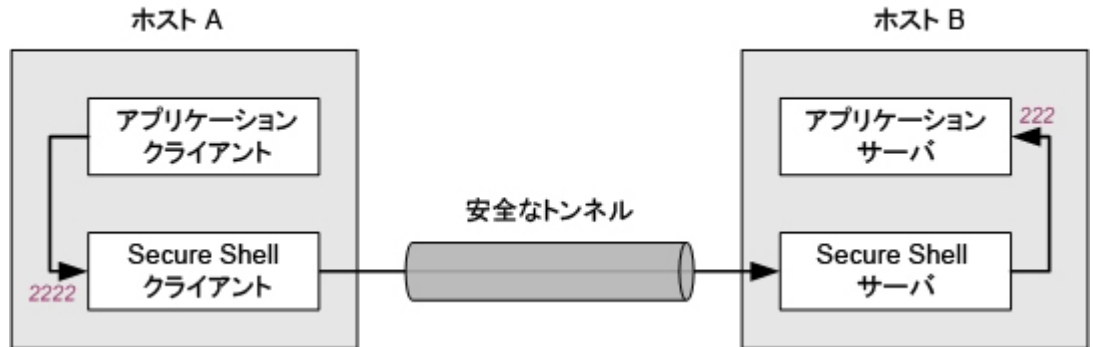
注意: 最終宛先ホストおよびポートが Secure Shell サーバホスト上にない場合、Secure Shell ホストとアプリケーションサーバホスト間でデータは平文で送信されます。

5. アプリケーションサーバから返されたデータは Secure Shell サーバへ転送されます。Secure Shell サーバはそのデータを暗号化して、SSH トンネルを介して Secure Shell クライアントへ安全に送信します。Secure Shell クライアントはデータを復号化し、そのデータを元のアプリケーションクライアントへリダイレクトします。

ローカルポート転送の一般的なコマンドライン構文は以下のとおりです。

```
ssh -L listening_port:app_host:hostport user@sshserver
```

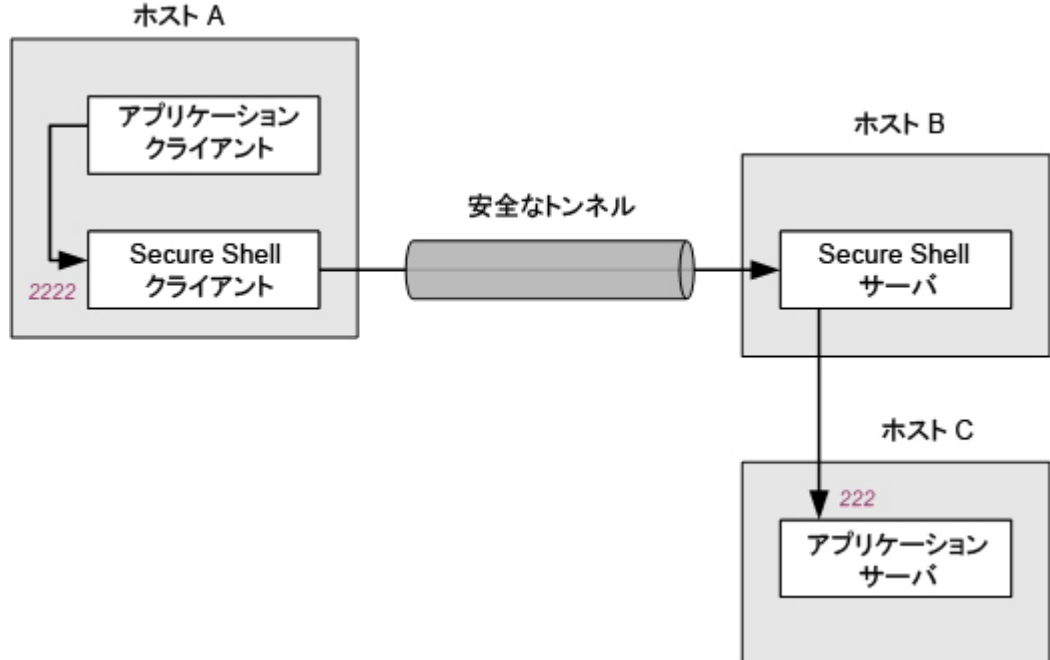
以下に示す図は、ローカルポート転送の 2 とおりの使用法を示しています。



上記の構成では、アプリケーションクライアントと Secure Shell クライアントの両方がホスト A で実行されます。Secure Shell サーバとアプリケーションサーバの両方はホスト B で実行されます。ホスト A のポート 2222 へ送信されたすべてのデータはホスト B のポート 222 へ転送されます。この配置では、転送中のすべてのデータが安全に暗号化されます。これは、以下のコマンド (localhost はホスト B のループバックアドレスを識別します) によって構成します。

```
ssh -L 2222:localhost:222 user@HostB
```

以下の図は、3 番目のホストへのローカルポート転送を示しています。この構成では、アプリケーションサーバが、Secure Shell サーバとは異なるホストで実行されます。ホスト A のポート 2222 へ送信されたすべてのデータはホスト C のポート 222 へ転送されます。



これは、以下のコマンドによって構成します。

```
ssh -L 2222:HostC:222 user@HostB
```

注意: ホスト B とホスト C 間で送信されるデータは暗号化されません。

リモートポート転送

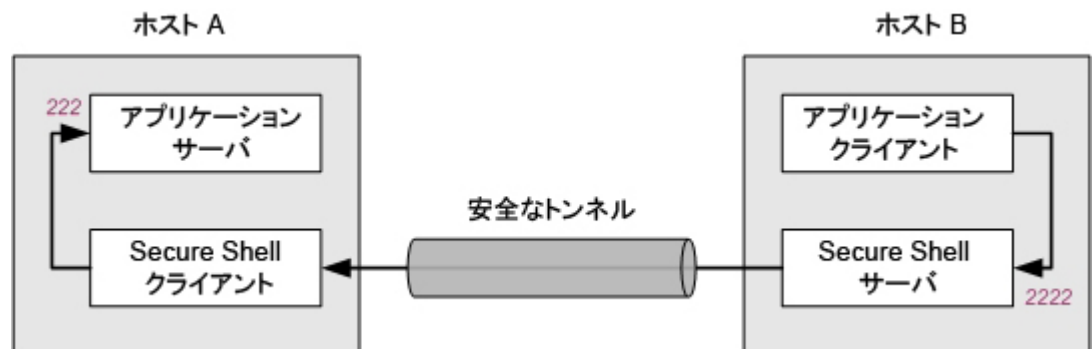
リモートポート転送を使用して、Secure Shell サーバホストで実行されているアプリケーションクライアントからデータを安全に転送できます。リモートポート転送を構成する時は、データの転送に使用する任意のリモートポート、およびデータを受信する着信先ホストとポートを指定します。リモートポート転送は、以下のように行われます。

1. Secure Shell 接続が確立されると、Secure Shell サーバは、受信を待っている指定のポートを使用して Secure Shell サーバホスト上に受信待ちのソケットを開きます。
2. Secure Shell サーバホスト上で実行されているクライアントアプリケーションは、(直接、アプリケーションサーバのホストおよびポートではなく) 受信を待っているポートに接続するよう構成されます。そのクライアントが接続を確立すると、すべてのデータが受信待ちのポートに送信され、次に、Secure Shell サーバへリダイレクトされます。
3. データは Secure Shell サーバで暗号化され、SSH トンネルを介して Secure Shell クライアントに安全に送信されます。
4. Secure Shell クライアントでは、データが受信され、解読されて、サーバアプリケーションで使用される宛先のホストおよびポート (Secure Shell クライアントホスト上) へリダイレクトされます。
5. サーバアプリケーションからの戻りデータが Secure Shell クライアントに送信されます。データはここで暗号化され、Secure Shell サーバに SSH トンネルを介して安全に送信されます。データは Secure Shell サーバで解読され、元のクライアントアプリケーションにリダイレクトされます。

リモートポート転送の一般的なコマンドライン構文は以下のとおりです。

```
ssh -R listening_port:app_host:hostport user@sshserver
```

以下に示す図は、考えられる 1 つのリモートポート転送構成を示しています。



アプリケーションサーバと Secure Shell クライアントの両方はホスト A で実行されま
す。Secure Shell サーバとアプリケーションクライアントの両方はホスト B で実行されま
す。ホスト B のポート 2222 へ送信されたすべてのデータは ホスト A のポート 222
へ転送されます。この配置では、転送中のすべてのデータが安全に暗号化されます。こ
れは、以下のコマンドによって構成します。

```
ssh -R 2222:localhost:222 user@HostB
```

TCP 通信の転送

この手順を使用して、アプリケーションクライアントとサーバ間でプレーンテキストで送信される TCP 通信内容を暗号化します (括弧で示した例では、Reflection for Secure IT を実行しているコンピュータの Web ブラウザとリモート Web サーバ間でデータを安全に送信するように Reflection for Secure IT クライアントを構成します)。

TCP 通信を転送するには

- 1 Reflection for Secure IT クライアントを開き、Secure Shell サーバホスト (例、MySSHserver.com) に接続するように構成します。
- 2 **[Reflection Secure Shell の設定]** ダイアログボックスを開きます。『15 ページ』**[トンネリング]** タブに移動します。
- 3 **[ポートのローカル転送]** の **[追加]** をクリックします。
- 4 **[ローカルポートの転送]** で、使用可能なローカルポートを指定します。通常、1024 より大きい値 (例、8080) を入力できます。通常、1024 以下の値のポートはサービス用に予約されているため、使用できません。
- 5 **[転送先のホスト]** で、アプリケーションサーバホストの **[名前]** (例、WebServer.Acme.com) を指定します。

注意: このサーバホストが Secure Shell サーバホストと異なる場合、Secure Shell サーバと指定したサーバ間の通信は暗号化されません。指定したサーバが Secure Shell サーバホストと同じリモートコンピュータで実行されている場合、値 localhost (または IP 接続 127.0.0.1) を指定できます。この場合、すべての通信が暗号化されません。

- 6 **[ポート]** で、アプリケーションサーバで使用されるポート (例、Web サーバの場合 80 またはメールサーバの場合 110) を指定します。

注意: 次の 2 つの手順は不要ですが、これらの手順を完了すると、Secure Shell トンネルの確立後、アプリケーションクライアントが自動的に起動されるように Reflection for Secure IT が構成されます。

- 7 (オプション) **[起動するアプリケーション]** で、トンネルを介して転送したいデータのクライアントアプリケーション名 (例えば iexplore.exe) を指定します。システムパスにないアプリケーションの場合、完全なパス情報を含める必要があります。実行ファイルの検索に **[参照]** ボタンを使用して完全なパス情報を含めることができます。
- 8 (オプション) **[引数]** で、このアプリケーションに使用したいコマンドライン引数を指定します (例えば、http:¥¥localhost:8080 を使用してリダイレクトポート 8080 に接続するようにブラウザを設定できます)。アプリケーションクライアントを実行し、指定したポートに接続するよう構成する必要がある場合もあります。
- 9 **[OK]** をクリックして開いているダイアログボックスを閉じます。

注意: **[ローカルポート転送]** ダイアログボックスの **[OK]** ボタンは、すべての必須情報が入力されるまで使用できません。

- 10 Secure Shell ホストに接続します。

Secure Shell 接続の確立後、手順 7 で指定したアプリケーションが起動されます。転送されるローカルポート (この例では 8080) への接続が正しく構成されている場合、このポートからサーバアプリケーションにデータがリダイレクトされます。クライアントは、そのサーバに直接接続するよう構成されているかのように正確に実行されます。

FTP 通信の転送

Secure Shell のポート転送を使用して FTP プロトコル通信 (FTP コマンドチャネルおよびすべてのデータチャネルを含む) を暗号化するには、次の手順に従います。ポート転送を使用することで、FTP サーバに安全に接続でき、SFTP 接続では使用できないオプションやコマンドも含め、すべての FTP オプションおよびコマンドにアクセスできます。

注意: データチャネルの転送を有効にするためには、FTP クライアントは、パッシブ (PASV) モード (既定値) で通信するように構成されている必要があります。

FTP 通信を転送するには

- 1 FTP クライアントを開きます。
- 2 **[FTP サイトに接続]** ダイアログボックスで、**[新規サイト]** をクリックします。
- 3 **[FTP サイトの追加]** ダイアログボックスで、FTP サーバホストの名前または IP アドレスを入力します。
- 4 **[ログイン情報]** ダイアログボックスで、**[ユーザ]** を選択し、**[セキュリティ]** をクリックします。
- 5 **[Secure Shell]** タブをクリックし、次を構成します。
 - **[Reflection Secure Shell を使用する]** チェックボックスをオンにします。
 - **[ポート転送を使用した FTP コマンドのトンネリング]** チェックボックスをオンにします。
- 6 この手順は、Secure Shell サーバが FTP サーバとは別のホスト上にある場合のみ、実行する必要があります。
 - **[FTP ホストが Secure Shell ホストと異なる]** チェックボックスをオンにします。
 - **[SSH サーバアドレス]** で、Secure Shell サーバのホスト名または IP アドレスを入力します。
 - **[SSH ユーザ名]** で、Secure Shell サーバでのログイン名を入力します。

注意: **[FTP ホストが Secure Shell ホストと異なる]** をオンにした場合、FTP コマンドとデータは、クライアントコンピュータから Secure Shell サーバに安全なトンネルを介して安全に送信されます。コマンドとデータは、Secure Shell サーバと FTP サーバ間で暗号化されずに送信されます。

- 7 **[OK]** をクリックして、**[セキュリティのプロパティ]** ダイアログボックスを閉じ、**[次へ]** をクリックします。
- 8 **[FTP ユーザログイン]** ダイアログボックスで、FTP サーバでのユーザ名を入力して、**[次へ]** をクリックします。
- 9 **[完了]** をクリックします。

注意: Secure Shell サーバと FTP サーバの両方に認証を行う必要があります。

[トンネリング] タブ ([Reflection Secure Shell の設定] ダイアログボックス)

表示方法 『15 ページ』

ポート転送 『137 ページ』を使用すると、TCP/IP トラフィックを SSH トンネルを介して転送することができます。これによって、保護されていない TCP/IP チャネル上のデータを、Reflection Secure Shell クライアントを使って保護するように設定できます。

オプションは次のとおりです。

[X11 トンネル接続する]	X11 リモートポートから送信されるすべてのデータが安全なトンネルを介して自動的に正しいローカルポートへ転送されるように指定します。
[ゲートウェイポートを許可する]	ゲートウェイポートを有効にします。リモートホストは、ローカル転送ポートへの接続を許可されます。既定で、Reflection Secure Shell は、ローカルポート転送をループバックアドレスに結合します (これは、「ローカルホスト」を使用することに相当します)。これによって、ほかのリモートホストが、転送ポートに接続できないようにしています。[ゲートウェイポートを許可する] は、Reflection Secure Shell がローカルポート転送をローカルのイーサネットアドレス (IP アドレス、URL、DNS 名など) に結合して、リモートホストが転送ポートへ接続できるようにすることを指定するために使用できます。 この設定を有効にする場合は注意が必要です。この設定によって、リモートホストで、認証なしにシステムで転送されたポートを使用できるようになるため、ネットワークと接続の安全性が低下します。
[ポートのローカル転送]	構成済みのローカルポート転送を表示します。[追加] をクリックすると、[ローカルポート転送] 『65 ページ』ダイアログボックスが開きます。
[ポートのリモート転送]	構成済みのリモートポート転送を表示します。[追加] をクリックすると、[リモートポート転送] 『66 ページ』ダイアログボックスが開きます。

注意

- このダイアログボックスで構成した設定は、*Secure Shell 設定ファイル* 『19 ページ』に保存されます。また、このファイルを任意のテキストエディタで手作業で編集することにより *Secure Shell 設定* を構成することもできます。
 - この構成ファイルの内の設定は、現在指定されている *SSH 構成セクション* 『112 ページ』用に保存されます。
-

[ローカルポート転送] ダイアログボックス

表示方法

- 1 [Reflection Secure Shell の設定] ダイアログボックスを開きます。 『15 ページ』
- 2 [トンネリング] タブをクリックします。

3 [ポートのローカル転送] の [追加] をクリックします。

ローカルポート転送を構成するには、このダイアログボックスを使用します。指定したローカルポートへの送信データは、安全なトンネルを介して、指定したリモートホストのポートへ転送されます。

以下のオプションをすべて指定する必要があります。

[ローカルポートの転送]	PC の使用可能なポートを指定します。このポートに送信されたデータは、SSH トンネルを介して転送されます。
[転送先のホスト] の [名前]	データの送信先ホストコンピュータを指定します (<code>localhost</code> を指定して、すでに Secure Shell 接続を確立した同じリモートホストの異なるポートにデータを転送できます)。
[ポート]	データの送信先のリモートホストのポートを指定します ([リモートデスクトップにトンネル接続する] を選択した場合は、Reflection for Secure IT で正しいリモートポートが自動的に構成され、このボックスは選択できなくなります)。
[転送の種類]	[TCP] と [FTP] の 2 つのオプションがあります。FTP クライアントとサーバ間で通信を転送する場合を除いて、TCP を使用してください。

構成可能なオプション

[リモートデスクトップにトンネル接続する]	Windows リモートデスクトップセッションにトンネル接続する場合は、このチェックボックスをオンにします。このオプションをオンにするとその他のオプションが選択できなくなり、Reflection for Secure IT で自動的にセッション転送設定が正しく構成されます。
[Reflection FTP を使用]	このボタンは、[転送の種類] が [FTP] に設定されている場合のみ表示されます。クリックすると、[起動するアプリケーション] に、Reflection Secure FTP クライアントを起動して FTP 通信をトンネル接続するための正しい値が自動的に入力されます。
[起動するアプリケーション] の [名前]	Secure Shell 接続確立後に、Reflection for Secure IT が自動的に起動するアプリケーション (メールクライアント、FTP クライアント、または Web ブラウザ) の名前を入力します。安全なトンネルを使用するには、[ローカルポートの転送] に設定したポートに接続するようにアプリケーションを構成する必要があります。一部のアプリケーションでは、コマンドライン引数を使用してこのように構成できます。引数は [引数] テキストボックスに指定します。
[引数]	指定したアプリケーションが起動した時に使用するオプションのコマンドライン引数を指定します。

注意: ポート転送の設定は、現在指定されている *SSH 構成セクション* 『[112](#) ページ』に保存されます。

[リモートポート転送] ダイアログボックス

表示方法

- 1 [Reflection Secure Shell の設定] ダイアログボックスを開きます。『15 ページ』
- 2 [トンネリング] タブをクリックします。
- 3 [ポートのリモート転送] の [追加] をクリックします。

リモートポート転送を構成するには、このダイアログボックスを使用します。指定したリモートポートからの送信データは、安全なトンネルを介して、指定したローカルコンピュータのポートへ転送されます。

以下のオプションをすべて指定する必要があります。

[リモートサーバポート
の転送] ホストコンピュータのポートを指定します。このポートから送信されたデータは、SSH トンネルを介して PC に転送されます。

[名前] データの送信先のローカルコンピュータを指定します。

[ポート] データの送信先のローカルホストのポートを指定します。

注意: ポート転送の設定は、現在指定されている *SSH 構成セクション* 『112 ページ』に保存されます。

マルチホップ Secure Shell セッションの構成

一連の Secure Shell サーバによって安全な接続を確立する必要がある場合、マルチホップ接続を使用します。これは、直接リモートサーバにアクセスすることはできないが、中間サーバを介してアクセスすることができるネットワーク構成で役に立ちます。ここでは、そのような一連のサーバについて示します。Windows ワークステーションにはサーバ C への安全なアクセスが必要ですが、サーバ B にもサーバ C にも直接接続することはできません。サーバ A はサーバ B に接続でき、サーバ B はサーバ C に接続できます。

Windows ワークステーション ➡ サーバ A ➡ サーバ B ➡ サーバ C

マルチホップ一覧を構成すると、Reflection for Secure IT では一連の安全なトンネルを確立することにより、安全なエンドツーエンド接続を作成します。各トンネルは既存のトンネル内に確立され、チェーンに沿ってさらにトンネルが確立されます。

チェーン内の最後のサーバは、最初の Secure Shell 接続の設定時に指定したホストになります。その他のサーバを順番に (クライアント側を基準に上から下に) マルチホップサーバ一覧に追加します。以下の手順では、この方法について説明します。

マルチホップセッションを構成するには

- 1 Reflection Secure Shell セッションを サーバ C に構成します。
- 2 [Reflection Secure Shell の設定] ダイアログボックスを開きます。『15 ページ』

- 3 [マルチホップ] タブをクリックします。
- 4 [追加] をクリックし、以下のように サーバ A への転送情報を構成します。
 - a. [ローカルポートの転送] に値を指定します。これは未使用のポート (この例ではポート 2022 を使用) にすることができます。
 - b. [転送先のホスト] で、ホストの [名前] (この例ではサーバ A) を指定します。

注意: このサーバに必要なユーザ名が手順 1 で指定したユーザ名 (この例ではサーバ C) と異なる場合は、ユーザ名 JoeA@ServerA を使用します。

- c. 既定以外の Secure Shell 設定をこのトンネルで使いたい場合は、[構成] をクリックします。
- d. [OK] をクリックします。
- 5 再度 [追加] をクリックし、サーバ B に転送情報を構成します。
 - a. [ローカルポートの転送] に値を指定します。これは未使用のポート (この例ではポート 3022 を使用) にすることができます。
 - b. [転送先のホスト] で、ホストの [名前] (この例ではサーバ B) を指定します。

注意: このサーバに必要なユーザ名が手順 1 で指定したユーザ名 (この例ではサーバ C) と異なる場合は、ユーザ名 JoeB@ServerB を使用します。

- c. 既定以外の Secure Shell 設定をこのトンネルで使いたい場合は、[構成] をクリックします。
- d. [OK] をクリックします。
- 6 [Reflection Secure Shell の設定] ダイアログボックスを閉じ、Reflection セッションに接続します。

注意: この接続を別のアプリケーション (ブラウザまたはメールクライアントなど) のデータのトンネリングに使用している場合、[トンネリング] 『64 ページ』 タブをクリックしてポート転送を構成します。例えば、メールサーバがサーバ C で実行されている場合、このマルチホップの構成後、以下のように新しいローカルポート転送を作成できます。[ローカルポートの転送] に、未使用のポート (例 1110) を指定し、リモートホストの [名前] に localhost (このコンテキストで「localhost」は、上記の例の一連のサーバ C の最後のサーバを示します) を入力し、メールサーバポートと同じ [ポート] 値 (通常 110) を設定します。Reflection マルチホップトンネルが確立されると、localhost:1110 に接続するようにローカルのメールクライアントを構成することによって、メールサーバに安全にアクセスできるようになります。

[マルチホップ] タブ ([Reflection Secure Shell の設定] ダイアログボックス)

表示方法 『15 ページ』

このタブを使って、マルチホップ 『67 ページ』 Secure Shell セッションを構成します。オプションは次のとおりです。

[マルチホップサーバ] マルチホップシーケンスに含まれるサーバを表示します。Reflection は、指定したローカルポートから、リモートサーバ上の指定したポートへ新しい SSH トンネルを確立します。一覧の各接続は、その上の接続で確立されたトンネルを介して送信されます。一覧内の順序を変更するには、矢印ボタンを使用します。

[追加]	[マルチホップサーバの構成] 『69 ページ』 ダイアログボックスを使って新しいサーバを一覧へ追加します。
[変更]	選択したサーバを変更します。
[削除]	選択したサーバを削除します。

注意

- このダイアログボックスで構成した設定は、*Secure Shell 設定ファイル* 『19 ページ』に保存されます。また、このファイルを任意のテキストエディタで手作業で編集することにより *Secure Shell 設定* を構成することもできます。
 - この構成ファイルの内の設定は、現在指定されている *SSH 構成セクション* 『112 ページ』用に保存されます。
-
-

[マルチホップサーバの構成] ダイアログボックス

表示方法

- 1 [Reflection Secure Shell の設定] ダイアログボックスを開きます。 『15 ページ』
- 2 [マルチホップ] タブをクリックします。
- 3 [追加] をクリックします。

このダイアログボックスで、マルチホップ 『67 ページ』の一覧にサーバを追加します。以下のオプションをすべて指定する必要があります。

[ローカルポートの転送]	ローカルの Windows ワークステーションのポートを指定します。このポートに送信されたデータは、SSH トンネルを介してサーバに転送されます。
[名前]	送信データを通させるホストコンピュータを指定します。このサーバに必要なユーザ名が、元の接続で指定したユーザ名と異なる場合は、ユーザ名 <ユーザ名>@<ホスト名> を使用します。
[構成]	[Reflection Secure Shell の設定] ダイアログボックスを開きます。このダイアログボックスでは、このトンネルに [全般] 『16 ページ』、[暗号化] 『22 ページ』、[GSSAPI] 『56 ページ』 の設定を指定できます。
[ポート]	データの送信先リモートホストのポートを指定します。既定では、多くの SSH サーバで使用されているポートである「22」が設定されています。

注意

- このダイアログボックスで構成した設定は、*Secure Shell* 設定ファイル『[19](#) ページ』に保存されます。また、このファイルを任意のテキストエディタで手作業で編集することにより *Secure Shell* 設定を構成することもできます。
 - この構成ファイルの内の設定は、現在指定されている *SSH* 構成セクション『[112](#) ページ』用に保存されます。
-
-

第 11 章

ホスト変数およびコマンド

[ホストデータ] タブ ([Reflection Secure Shell の設定] ダイアログボックス)

表示方法 『15 ページ』

[ホストデータ] タブを使用して環境変数を設定し、サーバでコマンドを実行します。オプションは次のとおりです。

[環境変数]

- [追加] [新規環境変数] ダイアログボックスが開き、新しい変数と値を指定できます。
- [変更] 選択した変数を編集します。
- [削除] 選択した変数を削除します。

[リモートコマンド]

- [コマンド] リモートサーバで実行する 1 つまたは複数のコマンドを指定します。セミコロン (;) を使用して、複数のコマンドを区切ります。接続の確立後、サーバは指定したコマンドを実行 (または実行を試行) し、セッションが終了します。サーバは、クライアントから受信したコマンドの実行を許可するよう構成されている必要があります。

コマンドは、正しい形式でサーバに指定する必要があります。例えば、UNIX サーバに一覧表示されているディレクトリをキャプチャするには、以下のように指定する必要があります。

```
ls > list.txt
```

Windows サーバで同等のコマンドは、サーバの構成方法に応じて以下のいずれかになります。

```
dir > list.txt  
cmd /c dir > list.txt
```


第 12 章

プロキシサーバ

プロキシサーバ (Secure Shell の設定)

開き方 『[15](#) ページ』

Reflection Secure Shell セッションでプロキシの使用を有効にするには [プロキシ] タブを使用します。

オプションは次のとおりです。

[なし]	プロキシは構成されません(これは既定の動作です)。
[SOCKS]	SOCKS プロキシを経由する Secure Shell 接続を構成するには、[SOCKS] を選択します。
[HTTP]	HTTP プロキシを経由する Secure Shell 接続を構成するには、[HTTP] を選択します。
[構成]	プロキシサーバ設定を構成します。

注記:

- Secure Shell 接続では、Secure Shell 構成ファイルの Proxy 設定を使用して、現在指定されている *SSH 構成セクション* 『[112](#) ページ』でプロキシの使用を有効にできません。プロキシサーバアドレスは、Windows レジストリにユーザ別に格納され、すべての Reflection セッションに適用されます。
 - Reflection FTP クライアントでは、[セキュリティのプロパティ] ダイアログボックスにプロキシの構成と Secure Shell の構成を行うタブがあります。[セキュリティのプロパティ] ダイアログボックスの [Secure Shell] タブで [Reflection Secure Shell を使用] を有効にしている場合は、[プロキシ] タブの [プロキシサーバを使用] を有効にできません。SOCKS プロキシを構成するには、[Reflection Secure Shell の設定] ダイアログボックスの [一般] タブで、[SOCKS の構成] の設定を使用します。
-

第 13 章

トラブルシュート

Secure Shell 接続のトラブルシュート

Secure Shell 接続に問題がある場合、問題の原因は Reflection がホストを検索できない、またはホスト認証またはユーザ認証のいずれかの問題の可能性があります。

ログファイルの使用

接続の問題が **ホスト認証** 『25 ページ』にある場合、Reflection クライアントの **ログファイル** 『76 ページ』に有効な情報を見つけることができます。

問題が **ユーザ認証** 『27 ページ』にある場合、Secure Shell サーバの管理者への問い合わせが必要な場合があります。ユーザ認証の問題は共通しており、失敗したユーザ認証に関する詳細な情報は、クライアントログではなくサーバデバッグログでのみ使用できません。Secure Shell プロトコルでは、失敗した認証試行に関する固有の情報をクライアントに提供しないようになっています。これは攻撃者がエラーメッセージを使用できないようにし、認証失敗の理由を判断して攻撃の成功する可能性を減らすためです。

トラブルシュートの候補

パスワードの認証

- パスワードが誤っています。Caps Lock が無効であることを確認します。
- パスワードの期限が切れています。パスワード認証ではなくキーボード対話型認証を使用して、パスワード更新を有効にしなければならない可能性があります。
- パスワードメッセージが表示されない場合、パスワード認証が無効の可能性があります。

公開鍵認証

- ユーザの公開鍵が、ホストの正しい位置にアップロードされていません。
- ユーザの公開鍵が正しい位置にアップロードされていますが、所有権またはファイル許可が間違っています。
- 鍵がパスフレーズで保護され、誤ったパスフレーズを入力しています。
- [Reflection Secure Shell の設定] ダイアログボックスの **[ユーザ鍵]** 『35 ページ』 タブで認証に選択した鍵が誤っています。
- 特に古いバージョンの OpenSSH を実行するサーバへの接続を試行している場合、選択した公開鍵が多すぎます。

証明書認証

- ホストの認証に使用される証明書が使用できません。Reflection の信頼されたルート格納場所と Microsoft の信頼されたルート格納場所と中間格納場所を確認します (Microsoft の格納場所の使用が無効になっている場合、証明書を Reflection 格納場所にする必要があります)。
- ユーザの認証に使用される証明書が使用できません。Reflection の個人の格納場所と Microsoft の個人の格納場所を確認します。

- ホストまたはユーザの認証に使用される証明書の有効期限が切れています。
- 『[証明書](#)のホスト名と対象ホスト名が一致するかどうかを確認する』『[48](#) ページ』がオンになっていますが、この接続に指定したホスト名が証明書のホスト名と一致していません。
- 『[証明書取り消しの確認が有効](#)』『[48](#) ページ』で、Certificate Revocation List が使用できません。
- 『[証明書の取り消しの確認が有効](#)』『[48](#) ページ』で、ホスト証明書が失効していません。

Secure Shell ログファイルの使用

このログファイルには、Secure Shell 接続の問題の解決に使用できる情報が含まれます。

注記: [記録内容] 設定を使用して、Secure Shell のログファイルにどの程度の情報を記録するかを指定できます。この設定は、『[Reflection Secure Shell の設定](#)』ダイアログボックスの『[一般](#)』タブ『[16](#) ページ』にあります。

Reflection for Secure IT クライアントでログファイルを使用するには

- 1 トレースを有効にします ([接続] - [トレース] - [トレースの開始])。
- 2 接続を確立します。
- 3 トレースを無効にします ([接続] - [トレース] - [トレースの終了])。
- 4 トレースを処理します ([接続] - [トレース] - [トレースの処理])。
- 5 『[ネットワークプロトコルの詳細](#)』を選択して、[OK] をクリックします。
- 6 Logs フォルダ内のトレースファイル (*.rev) を選択して、[開く] をクリックします。
- 7 ログの出力用にファイル名と形式を選択して、[OK] をクリックします。

FTP クライアントのログファイルを使用するには

- 次のいずれかを実行します。

目的	選択
ファイルへのログ情報の送信	[ツール] > [記録の開始] で、[ファイルの種類] を「診断ファイル (*.txt)」に変更します。
FTP コマンドウィンドウのログ情報の表示	[表示] > [コマンドウィンドウ]。

Reflection for Secure IT の問題解決のヘルプ

問題:Reflection for Secure IT を Vista で実行した時にヘルプが表示されない。

Reflection for Secure IT アプリケーションのヘルプ (*.hlp) では、Windows ヘルププログラム (WinHlp32.exe) を使用します。Vista オペレーティングシステムはこのプログラムに対応していません。ヘルプファイルを表示するには、Microsoft ダウンロードセンターから WinHlp32.exe をダウンロードしてインストールできます。

To download the Windows ヘルププログラムをダウンロードするには

- 「Windows Vista 用 Windows ヘルププログラム (WinHlp32.exe)」
(<http://go.microsoft.com/fwlink/?LinkID=82148>) という名称の Microsoft ダウンロードサイトにアクセスし、そのサイトの手順に従います。

Windows ヘルプサポートをインストールした後は、アプリケーションのヘルプを表示できるようになります。ただし、Reflection for Secure IT アプリケーションのヘルプで使用するマクロの一部は Vista では既定で無効になっています。これらのマクロを有効にしなくてもヘルプを表示することはできますが、一部の機能が失われることになり、マクロを実行できないことを示すメッセージが時折表示されることがあります。この場合、Windows レジストリを編集することによって、完全なマクロへの対応を可能にすることができます。

レジストリを変更してマクロを有効にするには

- 1 次の新しいサブキーをレジストリに追加します。

Vista 32 ビット版:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WinHelp
```

Vista 64 ビット版:

```
HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432node\Microsoft\WinHelp
```

- 2 AllowProgrammaticMacros という DWORD 値をサブキーに追加します。
- 3 AllowProgrammaticMacros の値を 1 に設定します。これで、マクロは有効になります。

注記: このレジストリ値が存在しないか、0 に設定された場合、マクロは無効になります。

追加情報については、Microsoft サポート技術情報 917607
(<http://support.microsoft.com/kb/917607>) を参照してください。

第 14 章

インストールのカスタマイズと配布

管理者用インストール

インストール

以下の 2 種類のインストールを実行できます。

実行するインストール	目的
ワークステーションインストール	少数のコンピュータに Reflection for Secure IT をインストールする。 または、 カスタマイズされたアプリケーション設定を含むファイルを作成する。
管理者用インストール	配布元として使用できる管理者用インストールポイントを作成する。

上記のインストールはインストールウィザードのグラフィカルインタフェースまたはコマンドラインから実行できます。また、直接 MSI を使って Reflection for Secure IT をインストールすることもできます。

第 15 章

インストールと配布の計画

Reflection for Secure IT のインストールと配布にはさまざまな方法があります。どの方法を選ぶかは、通常、承認されている業務プロセス、配布の規模、配布に使用するツール、およびカスタムインストールをやるかどうかなどのさまざまな要因で決まります。

例えば小規模の配布では、Attachmate 設定ウィザードを使用して数台のワークステーションに Reflection for Secure IT をインストールするだけかもしれませんが、企業規模の配布を行うには、膨大なカスタマイズとテストが必要です。

要件に応じて、以下のいずれかの方法を使用してください。

- **各ワークステーション上でのワークステーションインストールの実行**
Reflection for Secure IT のファイルをすべて PC のハードディスクにインストールします。この方法は、Reflection for Secure IT をインストールするコンピュータの数が少なく、インストールをカスタマイズする必要がない場合に適しています。
- **基本配布の実行**
管理者用インストールを行って、Reflection for Secure IT のファイルを管理者用インストールポイントにコピーします。この作業を、管理者用インストールイメージを作る、と呼ぶこともあります。次に、配布ツールを使用してこのファイルにアクセスしてワークステーションに配布するパッケージを作成します。基本配布は、インストールのカスタマイズを必要とせず、Reflection for Secure IT を配布するワークステーションの数が多いために適した方法です。
- **カスタム配布の実行**
管理者用インストールを行って、Reflection for Secure IT のファイルを管理者用インストールポイントにコピーします (基本配布の場合と同じです)。次に、インストールの方法、外観、およびエンドユーザのコンピュータの動作をカスタマイズします。カスタム配布では、配布するワークステーションの台数に制限はありません。この場合は、ユーザに、カスタマイズしたファイルを提供することができます (例えば、製品によってはこれらのファイルにワークスペースやセッションドキュメントが含まれます)。

管理者用インストールの実行

管理者用インストールイメージを作成すると、Reflection for Secure IT のイメージがネットワーク上の場所にコピーされ、後に複数のワークステーションへのインストールに使用されます。このネットワーク上の場所は、配布ツールがワークステーションに配布するパッケージにアクセスして作成するのに使用します。また、エンドユーザはこの場所にある `setup.exe` を実行して、インストールを実行します。

警告: この操作手順では、[詳細設定] タブと [ファイルの場所] タブのみを使用してください。ほかのタブで作成された構成は無視されます。

管理者用インストールイメージを作成するには

- 1 Attachmate 設定ウィザード (`setup.exe`) を実行します。
- 2 [詳細設定] タブで、[管理者用インストールイメージをサーバに作成する] をクリックします。

- 3 [続行] をクリックし、管理者用インストールイメージを作成するフォルダを指定します。
- 4 [インストール] をクリックします。

注意: 管理者用インストールイメージは、通常ファイルサーバ上のフォルダに作成されます。ただし、管理者用インストールイメージは、ローカルハードディスクのどのフォルダにでも作成できるので、テストをするのに便利です。

コマンドラインからのインストール

Attachmate 設定ウィザードのコマンドラインを使用して、配布 (または CD) のイメージまたは、管理者用インストールイメージから Reflection for Secure IT をインストールできます。また、バッチファイルにコマンドラインオプションを書き込んで、インストールパラメータを事前に設定しておき、Reflection for Secure IT のインストール時におけるユーザの介入を減らすことができます。サイレントインストールを行うために、インストール時のダイアログボックスを表示しないことも可能です。

また、コマンドラインオプションを使えば、ユーザが Reflection for Secure IT をインストールできるよう準備できます。通常、MSI のどのコマンドラインオプションも、Attachmate 設定ウィザードのコマンドラインで使用することができます。

コマンドラインからインストールするには

- コマンドプロンプト、または [スタート]-[ファイル名を指定して実行] コマンドで `setup.exe` ファイルのあるディレクトリに移動して、以下のいずれかを実行します。

- 管理者用インストールイメージを作成するには、以下を入力します。

```
setup.exe /install /admin TARGETDIR=path
```

`path` の部分には、サーバ上の管理者用インストールイメージへのパスが入りません。

または

- 一般的な設定でワークステーションにインストールするには、以下のように入力します。

```
setup.exe /install INSTALLDIR=path
```

`path` には、インストールディレクトリのパスが入ります (`INSTALLDIR=path` はオプション)。

注意: インストールをカスタマイズするためのコマンドラインオプションの一覧を表示するには、`setup.exe` ファイルのあるディレクトリに移動し、以下のように入力します。

```
setup.exe /?
```

MSI を使用して直接インストールするには

コマンドプロンプトまたは、[スタート]-[ファイル名を指定して実行] コマンドで、`msi` ファイルのあるディレクトリに移動して、以下のように入力します。

```
msiexec.exe /i installation_file_name.msi
```

例えば、Reflection for Secure IT Windows クライアント バージョン 7.0 をインストールする場合、以下のコマンドを使用します。

```
msiexec.exe /i rsshc700.msi
```

インストール記録の開始と終了

インストールログファイルは、インストールに関する詳細情報を提供するもので、ユーザの一時ディレクトリ (%tmp%) に保存され、atm で始まるファイル名が自動生成されます。このフォルダを開くには、[スタート]-[ファイル名を指定して実行] コマンドで、%tmp% を入力します。

インストールログファイルを作成または無効にするには

- 1 Attachmate 設定ウィザードを実行します。

インストール元	手順
ダウンロードサイト	ダウンロードリンクをクリックして、ダウンロードしたプログラムを実行します。インストーラファイルの場所を選択して [次へ] をクリックします。これにより、ファイルが指定の場所に解凍され、Attachmate 設定ウィザードが起動します。
管理者用インストールイメージ	管理者用インストールポイントから、 <code>setup.exe</code> ファイルをダブルクリックします。

- 2 [詳細設定] タブで、[インストールのログファイルを作成する] をオンまたはオフにします。
- 3 [インストール] をクリックします。

インストールのカスタマイズ

Attachmate カスタム設定ツールを使って、Reflection for Secure IT のインストールをカスタマイズできます。以下に示す手順では、このツールの起動方法とツールを使用して設定をエンドユーザに配布する方法について説明します。追加の情報については、Attachmate カスタム設定ツールの [ヘルプ] メニューを参照してください。

Attachmate カスタム設定ツールを開く

Attachmate カスタム設定ツールを実行するためには、最初に管理者用インストールイメージを作成しておく必要があります。

Attachmate カスタム設定ツールを開くには

- コマンドプロンプトまたは [スタート]-[ファイル名を指定して実行] コマンドで、以下のコマンドを入力して Attachmate カスタム設定ツールを開きます。

```
path_to_setup\setup.exe /admin
```

注意: 現在、Attachmate 設定ウィザードを実行中の場合は Attachmate カスタム設定ツールを実行することはできません。`setup.exe` プログラムのインスタンスは一度に 1 つしか実行できません。

カスタム設定の種類を選択

Attachmate カスタム設定ツールを開いたら、トランスフォームやコンパニオンインストールパッケージを作成したり、既存のファイルを開いたりすることができます。

カスタム設定の種類を選択するには

- 1 **[カスタム設定の選択]** ダイアログボックスで、カスタム設定の種類を選択します。

目的	選択する項目
新規トランスフォーム (.mst) を作成する	[以下の製品のセットアップカスタム設定ファイルを新規作成する] (既定)
新規コンパニオンインストールパッケージ (.msi) を作成する	[コンパニオンインストーラを新規作成する]
既存ファイルを開く	[既存のセットアップカスタム設定ファイルまたはコンパニオンインストーラを開く]

- 2 **[OK]** をクリックします。

コンパニオンインストーラによるカスタム設定のインストール

Attachmate カスタム設定ツールを使って、カスタマイズされた Reflection for Secure IT を構成できます。これを行うには、1 つまたは複数のコンパニオンインストーラパッケージを作成して、カスタム設定ファイルをインストールしてから、次に、コンパニオンパッケージをインストールに追加します。ファイルのインストール用にユーザ固有の場所とグローバルな場所の両方を指定できます。

注記: Reflection for Secure IT で使用される構成ファイルの名前を場所については、以下の一覧 (手順に対応しています) を参照してください。

コンパニオンパッケージを作成して、Reflection for Secure IT 設定ファイルをインストールするには

- 1 **管理者用インストールイメージを作成します。** 『81 ページ』。
- 2 コマンドラインから Attachmate カスタム設定ツールを開きます。
`path_to_setup\setup.exe /admin`
- 3 **[カスタム設定の選択]** ダイアログボックスで、**[コンパニオンインストーラを新規作成する]** を選択し (または既存の MSI を開き)、**[OK]** をクリックします。
- 4 ナビゲーション画面で、**[パッケージ情報の指定]** をクリックします。このタブを使用して、パッケージが Windows の **[プログラムの追加と削除]** 一覧で使用するプログラム名を指定します。また、所属名も指定します。
- 5 ナビゲーション画面で、**[インストール場所の指定]** をクリックします。この画面では、すべてのユーザ用にファイルをインストールするか (既定)、または作成するパッケージをインストールするユーザ用にのみファイルをインストールするかを指定します。また、この画面では、既定のインストール場所を指定することもできます。
- 6 ナビゲーション画面で、**[ファイルの追加]** をクリックします。
- 7 **[ファイルの追加先]** で、インストール先の場所を指定します。Reflection for Secure IT で使用されるファイルとファイルの場所の一覧については、以下の一覧 (手順に対応しています) を参照してください。
- 8 (オプション) **[ショートカットを含める]** をクリックして、インストールしたファイルを開くためにユーザが使用できるショートカットをインストールします。例えば、設定ファイル (*.r3w) をインストールする場合は、そのファイル内の設定を使用して Reflection を起動できるショートカットをインストールできます。**[ショートカットの構成]** 画面を使用して、ショートカットをインストールする場所を指定できます。

- 9 [追加] をクリックして、インストールに追加するファイルを参照して指定してから、[開く] をクリックします。
- 10 [ファイル] - [名前を付けて保存] をクリックし、インストーラファイルの名前 (ReflectionSettings.msi など) を入力します。

ユーザ固有のファイルおよび場所

ファイル名	[ファイルの追加先]
*.r3w	[PersonalFolder]\Attachmate\Reflection
	注記: これは、Reflection for Secure IT 設定ファイルの既定の場所です。ほかの場所にある設定ファイルも使用することができます。
config	[PersonalFolder]\Attachmate\Reflection\.ssh
	注記: このファイルの詳細については、「 <i>Secure Shell クライアント構成ファイル</i> 『 19 ページ』」を参照してください。
known_hosts	[PersonalFolder]\Attachmate\Reflection\.ssh
	注記: このファイルの詳細については、「 <i>既知のホストファイル</i> 『 39 ページ』」を参照してください。
pki_config	[PersonalFolder]\Attachmate\Reflection\.pki
	注記: このファイルは、Reflection 証明書マネージャの設定の構成に使用します。
trust_store.p12	[PersonalFolder]\Attachmate\Reflection\.pki
	注記: このファイルは、Reflection の信頼された認証局の構成に使用します。
Settings.rfw	[PersonalFolder]\Attachmate\Reflection
	注記: これは、FTP クライアント設定ファイルの既定の場所です。xml ファイルを使用して FTP クライアントを構成することもできます。xml 設定ファイルを使用するメリットは、ユーザ名やパスワードなどのユーザ固有の情報を含めないで設定を配布できるという点にあります。
rftp.xml	[PersonalFolder]\Attachmate\Reflection
	注記: このファイルは、FTP クライアントの [エクスポート] コマンドを使用して作成できます。この場所に保存された設定は、ユーザが初めて FTP クライアントを実行した時に Settings.rfw ファイルに移行されます。

ファイル名	[ファイルの追加先]
rsckrb5.xml	[AppDataFolder]\Attachmate\Reflection
<p>注記: このファイルは、Reflection Kerberos マネージャの [設定のエクスポート] コマンドを使用して作成できます。この場所に保存された設定は、ユーザが初めて Reflection Kerberos マネージャ、または Reflection Kerberos を使用するよう構成された Reflection クライアントを実行した時に Windows レジストリに移行されます。</p>	

グローバルなファイルおよび場所

ファイル名	[ファイルの追加先]
ssh_config	[CommonAppDataFolder]\Attachmate\Reflection
<p>注記: これは、グローバルな Secure Shell クライアント構成ファイルです。</p>	
ssh_known_hosts	[CommonAppDataFolder]\Attachmate\Reflection
<p>注記: これは、グローバルな既知のホストファイルです。</p>	
pki_config	[CommonAppDataFolder]\Attachmate\Reflection\.pki
trust_store.p12	[CommonAppDataFolder]\Attachmate\Reflection\.pki
rftp.xml	[CommonAppDataFolder]\Attachmate\Reflection
<p>注記: この場所に保存された設定は、ユーザが初めて FTP クライアントを実行した時に各 Windows ユーザの Settings.rfw ファイルに移行されます。</p>	
rsckrb5.xml	[CommonAppDataFolder]\Attachmate\Reflection
<p>注記: この場所に保存された設定は、ユーザが初めて Reflection Kerberos マネージャ、または Reflection Kerberos を使用するよう構成された Reflection クライアントを実行した時に各 Windows ユーザの Windows レジストリに移行されます。</p>	

FTP クライアント設定のインストール

この手順では、Attachmate カスタム設定ツールの [ユーザ設定の変更] オプションを使用して、FTP クライアントのカスタム設定をインストールします。開始する前に、管理者用インストールを作成します。このインストールの `setup.exe` を使用して、Attachmate カスタム設定ツールを起動します。また、FTP クライアントがコンピュータにインストールされている必要があります。

設定を構成するには

- 1 Reflection Secure FTP クライアントを起動します。
- 2 インストールに含めるサイトおよび設定を構成して、設定を保存します。

コンパニオンインストーラを新規作成するには

- 1 コマンドラインから Attachmate カスタム設定ツールを開きます。
`path_to_setup\setup.exe /admin`
- 2 [カスタム設定の選択] ダイアログボックスで、[コンパニオンインストーラを新規作成する] を選択し (または既存の MSI を開き)、[OK] をクリックします。
- 3 ナビゲーション画面で、[パッケージ情報の指定] をクリックします。このタブを使用して、パッケージが Windows の [プログラムの追加と削除] 一覧で使用するプログラム名を指定します。また、所属名も指定します。
- 4 ナビゲーション画面で、[インストール場所の指定] をクリックします。[インストールの種類] で、[インストールするユーザのみにインストールする] をオンにします。
- 5 ナビゲーション画面で、[ユーザ設定の変更] をクリックします。
- 6 Reflection 製品の一覧で Reflection FTP クライアントを選択し、[定義] をクリックします。

注意: FTP クライアントがワークステーションにインストールされていないと、[定義] ボタンは使用できません。

- 7 配布に含める FTP クライアント設定を選択して、[OK] をクリックします。設定がエクスポートされたことを示す確認メッセージが表示されます。[OK] をクリックしてこのメッセージを閉じます。

注意: [ユーザ設定] をオフにする (既定) と、エクスポートされたファイルには、ユーザ名、パスワードなどのユーザ固有の情報は含まれません。詳細については、ダイアログボックスの [ヘルプ] ボタンをクリックしてください。

- 8 [ファイル] - [名前を付けて保存] をクリックし、インストーラファイルの名前 (例えば、FTPClientSettings.msi) を入力します。

カスタム設定したコンパニオンインストーラを配布するには

- 1 コンパニオンインストーラを Reflection インストールに追加 『84 ページ』 します。
- 2 ユーザに `Setup.exe` を使用してインストールするように指示します。

コンパニオンパッケージによって、ユーザの Application Data フォルダの `Attachmate\Reflection` フォルダに XML 設定ファイルがインストールされます。この場所に `rftp.xml` ファイルがインストールされると、このファイルに構成された設定は、ユーザが初めて FTP クライアントを実行した時に `Settings.rfw` ファイルに移行されます。

既存インストールへのコンパニオンインストーラの追加

以下の手順を使用して、1 つまたは複数のカスタムインストーラパッケージを製品のインストールに追加できます。

コンパニオンパッケージを既存のインストールに追加するには

- 1 コマンドラインから `Attachmate` カスタム設定ツールを開きます。
`path_to_setup\setup.exe /admin`
- 2 [以下の製品のセットアップカスタム設定ファイルを新規作成する] を選択し、[OK] をクリックします。
- 3 ナビゲーション画面で、[インストールの追加とプログラムの実行] をクリックします。
- 4 [追加] をクリックします。
[プログラムエントリの追加と変更] ダイアログボックスが開きます。
- 5 [ターゲット] 一覧で、コンパニオン `.msi` ファイルを参照して選択します。
- 6 [このプログラムをベース製品のインストール後に実行する] を選択します。
- 7 [OK] をクリックします。
- 8 [ファイル] - [名前を付けて保存] をクリックして、トランスフォームを保存します。

注記: トランスフォームを保存すると、カスタム設定ツールでは、自動的に `Setup.ini` ファイルが更新され、コンパニオンパッケージのインストール用に [RunPrograms] セクションと手順が追加されます。

- 9 `Setup.exe` を使ってインストールするようユーザに指示します。
インストールが完了すると、コンパニオンパッケージは自動的にインストールされま
す。

Secure Shell コマンドラインユーティリティ

Reflection Secure Shell クライアントは、コマンドラインユーティリティを実装しています。これらユーティリティの実行ファイルは、プログラムと同一のインストール先フォルダに存在します。

- `ssh` 『90 ページ』
- `ssh-keygen` 『95 ページ』
- `sftp`
- `scp` 『104 ページ』

F-Secure から移行し、F-Secure コマンドラインユーティリティ用に作成したスクリプトを維持する必要がある場合は、以下のユーティリティを使用します。これらのユーティリティでは、F-Secure のユーティリティと同じスイッチセットを使用できます。

注記: F-Secure コマンドラインユーティリティ用に作成したスクリプトがない場合は、上記のユーティリティを使用することをお勧めします。

- `ssh2` 『94 ページ』
- `sftp2` 『103 ページ』

- `scp2` 『[107](#) ページ』

ssh コマンドラインユーティリティ

構文: ssh [options] [user@]hostname [host command]

ssh コマンドユーティリティは、Windows コマンドラインから Secure Shell 接続を確立するために使用できます。

注記:

- Reflection では ssh2 『[94](#) ページ』ユーティリティも使用できます。ssh と ssh2 の両方を使用して Secure Shell 接続を確立できますが、これらの 2 つのユーティリティで対応している一部のオプションが異なります。ssh オプションは、Reflection クライアントのみが対応している一部の追加オプションを含む OpenSSH Secure Shell 実装に基づいています。ssh2 オプションは、Reflection for Secure IT UNIX クライアントおよび F-Secure クライアントと互換性があります。
- 既存の SecureShell 接続を再利用できます。ただし、そのためには各コマンドラインでこれを明示的に有効にするか、SSHConnectionReUse 環境変数を「Yes」に設定する必要があります。詳細については、「*Secure Shell セッションにおける接続の再利用* 『[28](#) ページ』」を参照してください。

オプション

-A

認証エージェント転送を有効にします。この設定は、*構成ファイル* 『[19](#) ページ』でホストごとに指定することもできます。エージェント転送を有効にする場合は注意が必要です。リモートホストでファイル権限を回避できるユーザは、転送された接続を介してローカルエージェントにアクセスできます。攻撃者は鍵の情報を取得できませんが、エージェントに読み込まれた識別情報を使用して、その鍵で操作を実行して認証を有効にすることができます。

-a

認証エージェントの転送を無効にします。(これは既定の動作です)。

-b *bind_address*

複数のインタフェースまたは別名を付けられたアドレスを持つマシン上の、送信元となるインタフェースを指定します。

-c *cipher_spec*

優先順に指定した Cipher のカンマ区切りの一覧。既定値は、「aes128-ctr,aes128-cbc,aes192-ctr,aes192-cbc,aes256-ctr,aes256-cbc,3des-cbc,blowfish-cbc,cast128-cbc,arcfour128,arcfour256,arcfour」です。接続が *FIPS モード* 『[21](#) ページ』で動作するように設定されている場合、既定値は「aes128-ctr,aes128-cbc,aes192-ctr,aes192-cbc,aes256-ctr,aes256-cbc,3des-cbc」になります。

プロトコルバージョン 1(使用は現在推奨されない) では、Cipher を 1 つ指定できません。指定できる値は「3des」、「blowfish」、「des」です。

-C

すべての送信データの圧縮を有効にします。圧縮はモデム回線やほかの低速接続に適していますが、高速のネットワークでは応答速度の低下を招くだけです。

-e *escape_character*

端末セッションのエスケープ文字を設定します。既定の文字はチルダ (~) です。エスケープ文字を「none」に設定すると、使用できるエスケープ文字はなくなり、チルダは他の文字と同様に機能します。以下のエスケープシーケンスを使用できません(チルダを、*escape_character* に指定した文字に置き換えてください)。

- ~. 接続を終了します。
- ~R *rekey* を要求します (SSH プロトコル 2 のみ)。
- ~# 転送された接続を一覧表示します。
- ~? 使用可能なエスケープシーケンスを表示します。
- ~~ エスケープ文字をホストに送信します。

-E *provider*

外部の鍵プロバイダとして指定のプロバイダを使用します。

-f

コマンドの実行前にクライアントをバックグラウンドで動作するようにします。

-F *config_file*

この接続に使用する代替構成ファイルを指定します。構成ファイルをコマンドラインで指定した場合、ほかの**構成ファイル**『19 ページ』は無視されます。

-g

ゲートウェイポートを有効にします。リモートホストは、ローカル転送ポートへの接続を許可されます。

-H *scheme*

この接続に使用する *SSH 構成セクション*『112 ページ』を指定します。

-i *key_file*

鍵認証に使用する秘密鍵を指定します。鍵ファイルは、**構成ファイル**『19 ページ』でホストごとに指定することもできます。複数の **-i** オプションを指定できます (構成ファイルで複数の鍵を指定できます)。ファイルまたはパスが空白を含む場合、引用符を使用します。

-k *directory*

構成ファイル、ホスト鍵ファイル、ユーザ鍵ファイルの保存先に別の場所を指定します。

-l *login_name*

リモートコンピュータでのログインに使用する名前を指定します。この設定は**構成ファイル**『19 ページ』で指定することもできます。

-L *localport:remotehost:hostport*

指定されたローカルポートからのデータを、安全なトンネルを介して、指定された宛先ホストおよびポートにリダイレクトします。詳細については、「ローカルポート転送」を参照してください。ポート転送を構成ファイルに指定することもできます。管理者としてログインしないかぎり、権限ポート (ポート番号 1024 以下) を転送できません。IPv6 アドレスは、別の構文 *port/host/hostport* を使用して指定できます。

-m *mac_spec*

この接続に使用する 1 つまたは複数のカンマ区切り MAC (メッセージ認証コード) アルゴリズムを指定します。優先順にアルゴリズムを指定します。既定値は「`hmac-sha1,hmac-sha256,hmac-sha512,hmac-md5,hmac-ripemd160,hmac-sha1-96,hmac-md5-96`」です。接続が *FIPS モード* 『21 ページ』で動作するように設定されている場合、既定値は「`hmac-sha1,hmac-sha256,hmac-sha512`」です。

-N

リモートコマンドを実行しません。これはポート転送だけを構成する場合に役立ちます(プロトコルバージョン 2 のみ)。

-o *option*

構成ファイル 『114 ページ』で対応するオプションを指定します。例えば、次のように入力します。

```
ssh "-o FIPSMode=yes" myuser@myhost
```

-p *port*

サーバ上の接続先ポートを指定します。既定は 22 で、これは Secure Shell 接続の標準ポートです。この設定は、**構成ファイル** 『19 ページ』でホストごとに指定できます。

-q

クワイエットモードを有効にします。このモードでは、バナーを含むすべての警告および診断メッセージが表示されません。

-R *localport:remotehost:hostport*

(Secure Shell サーバを実行するコンピュータ上の) 指定されたりポートからのデータを、安全なトンネルを介して、指定された宛先ホストおよびポートにリダイレクトします。詳細については、「リモートポート転送」を参照してください。ポート転送を構成ファイルに指定することもできます。管理者としてログインしないかぎり、権限ポート (ポート番号 1024 以下) を転送できません。IPv6 アドレスは、別の構文 `port/host/hostport` を使用して指定できます。

-T

シェルを実行しません。

-S

シェルを実行しません。

-t

コマンドが指定されている場合も TTY を強制的に割り当てます。

-T

仮想端末の割り当てを無効にします。

-v

デバッグレベルを冗長モードに設定します。これは、デバッグレベルを 2 に設定することと同じです。

-V

製品名およびバージョン情報を表示して終了します。コマンドラインで他のオプションが指定された場合、それらは無視されます。

-x

X11 接続の転送を無効にします。

-X

X11 接続の転送を有効にし、X11 クライアントを信頼されないものとして扱います。ゲストのリモート X11 クライアントは、信頼される X11 クライアントに属するデータを不正に変更できません。

X11 転送を有効にする場合は注意が必要です。ユーザの X 認可データベースのリモートホストでファイル権限を回避できるユーザは、転送された接続を介してローカル X11 ディスプレイにアクセスできます。攻撃者は、キーストローク監視などの行動を実行できる可能性があります。

-Y

X11 接続の転送を有効にし、X11 クライアントを信頼関係があるクライアントとして扱います。

X11 転送を有効にする場合は注意が必要です。ユーザの X 認可データベースのリモートホストでファイル権限を回避できるユーザは、転送された接続を介してローカル X11 ディスプレイにアクセスできます。攻撃者は、キーストローク監視などの行動を実行できる可能性があります。

-1

ssh がプロトコルバージョン 1 のみを試行するようにします。プロトコルバージョン 1 の使用は現在推奨されません。

-2

ssh がプロトコルバージョン 2 のみを試行するようにします。

-4

IPv4 アドレスのみを使用して接続させます。

-6

IPv6 アドレスのみを使用して接続させます。

ssh2 コマンドラインユーティリティ

構文: ssh2 [options] [user@]hostname [host command]

ssh2 コマンドラインユーティリティでは、Secure Shell の F-Secure 実装と互換性のあるスイッチを使用できます。使用可能なスイッチを確認するには、コマンドウィンドウに以下のコマンドを入力します。

```
ssh2 -h
```

注記: ssh2 を使用して確立された接続では、既定のクライアント構成ファイル『[19](#) ページ』は使用されません。これらの接続では、存在する場合、F-Secure 構成ファイルが使用されます。

ssh-keygen コマンドラインユーティリティ

ssh-keygen - クライアント/サーバ認証に使用される鍵を作成、管理、および変換します。

書式

```
ssh-keygen [-b bits] -t type [-N [passphrase]] [-C comment] [-f output_keyfile]
ssh-keygen -B [-f input_keyfile]
ssh-keygen -c [-P passphrase] [-C comment] [-f keyfile]
ssh-keygen -e [-f input_keyfile]
ssh-keygen -p [-P old_passphrase] [-N new_passphrase] [-f keyfile]
ssh-keygen -i [-f input_keyfile]
ssh-keygen -y [-f input_keyfile]
ssh-keygen -l [-f input_keyfile]
```

説明

ssh-keygen コマンドラインユーティリティは、公開鍵認証用に RSA 鍵および DSA 鍵を作成したり、既存の鍵のプロパティを編集したり、ファイル形式を変換したりするために使用できます。オプションが何も指定されないと、ssh-keygen によって 2048 ビットの RSA 鍵のペアが生成され、秘密鍵を保護するために鍵の名前とパスフレーズの入力を求められます。公開鍵は、秘密鍵と同じ名前をベースに .pub 拡張子を付けて作成されます。鍵の生成が完了すると、鍵の場所が表示されます。

オプション

-b *bits*

鍵のサイズを指定します。鍵のサイズを大きくすると、ある程度までセキュリティは向上します。鍵のサイズを大きくすると最初の接続が遅くなりますが、正常に接続した後は、鍵のサイズはデータストリームの暗号化や解読の速度に影響しません。使用する鍵の長さは、多くの要素に依存します。その要素には、鍵の種類、鍵の有効期間、保護するデータの値、潜在的な攻撃者にとって利用可能なリソース、この非対称鍵とともに使用する対称鍵のサイズなどがあります。ニーズに合った最適な選択をするには、セキュリティ管理者にお問い合わせください。鍵のサイズが、64 ビットの倍数となる次の値に切り上げられます。既定では、DSA 鍵は 1024 ビット、RSA 鍵は 2048 ビットになります。

-B

指定された鍵の指紋を SHA-1 Bubble Babble 形式で表示します。-f を使用して鍵ファイルを指定できます。ファイル名を指定しないと、ファイル名に関する問い合わせが行われます。秘密鍵または公開鍵の名前を指定できますが、いずれの場合も公開鍵が使用可能でなければなりません。

-c

秘密鍵ファイルおよび公開鍵ファイルのコメントの変更を要求します。この操作は、RSA1 鍵の場合のみ可能です。プログラムによって、秘密鍵を含むファイル、鍵にパスフレーズがある場合はそのパスフレーズ、および新しいコメントの入力が要求されます。

-C *comment*

鍵ファイル内のコマンドフィールドの情報を指定します。文字列に空白が含まれる場合は引用符を使用します。鍵の作成時にコメントを指定しない場合は、鍵の種類、作成者、作成日時を含む既定のコメントが作成されます。

-e

指定の OpenSSH 公開鍵または秘密鍵を使用して、Reflection 形式の公開鍵を生成します。-f を使用して鍵ファイルを指定できます。ファイル名を指定しないと、ファイル名に関する問い合わせが行われます。

-f filename

生成された秘密鍵のファイル名を指定します(公開鍵もまた作成されます。公開鍵の名前は、常に、秘密鍵と同じ名前に .pub ファイル拡張子を付けたものになります)。このオプションを-e、-i、-l、-p、-y、および -B と組み合わせることで、入力ファイル名を指定することもできます。

-i

指定の Reflection 公開鍵または秘密鍵を使用して、OpenSSH 形式の公開鍵または秘密鍵を生成します。-f を使用して鍵ファイルを指定できます。ファイル名を指定しないと、ファイル名に関する問い合わせが行われます。

-l

MD5 ハッシュを使用して、指定した公開鍵ファイルの指紋を表示します。-f を使用して鍵ファイルを指定できます。ファイル名を指定しないと、ファイル名に関する問い合わせが行われます。秘密鍵を指定している場合、ssh-keygen は一致する公開鍵ファイルを探してそのファイルの指紋を出力しようと試みます。

-N passphrase

パスフレーズを設定します。例えば、新しい鍵のパスフレーズを指定するには

```
ssh-keygen -N mypassphrase -f keyfile
```

パスフレーズで保護されていない新規の鍵を作成するには

```
ssh-keygen -N -f keyfile
```

また、-N を -p および -P と組み合わせることで、既存の鍵のパスフレーズを変更することもできます。

-P

このオプションは、既存の秘密鍵のパスフレーズを変更するために使用します。このオプションを単独で使用すると、プログラムによって、秘密鍵を含むファイル、古いパスフレーズ、および新しいパスフレーズの入力 (2 回) が要求されます。このオプションを -f、-P、および -N と組み合わせて使用すると、パスフレーズを非対話的に変更できます。例えば、次のようになります。

```
ssh-keygen -p -f keyfile -P oldpassphrase -N newpassphrase
```

-P passphrase

(古い) パスフレーズを指定します。

-q

メッセージ出力を控えます。

-t type

鍵の生成に使用する鍵のアルゴリズムを指定します。プロトコルバージョン 2 で使用可能な値は「rsa」または「dsa」です。

-y

指定された秘密鍵を使用して、公開鍵の新しいコピーを派生させます。-f を使用して鍵ファイルを指定できます。ファイル名を指定しないと、ファイル名に関する問い合わせが行われます。

戻り値

`ssh-keygen` は、コマンドが正常に実行されると 0 (ゼロ) を戻します。0 以外の値は、コマンドが失敗したことを示しています。

sftp コマンドラインユーティリティ

構文: sftp [options] [user@]host[#port]:source_file
[user@]host[#port][:destination_file]

注記: 既存の SecureShell 接続を再利用できます。ただし、そのためには各コマンドラインでこれを明示的に有効にするか、SSHConnectionReUse 環境変数を「Yes」に設定する必要があります。詳細については、「*Secure Shell セッションにおける接続の再利用*」『[28 ページ](#)』を参照してください。

コマンドラインオプション

-a

ASCII モードでファイルを転送します。

-b *buffersize*

1 回の要求の最大バッファサイズを設定します。有効な値は 1024 ~ 32768 です。

-B *batchfile*

ログイン成功後、指定したバッチファイルで各コマンドを実行し、接続を終了します。例えば、次のコマンドは、*myname* を使用して *myhost* に接続し、*myfile* のコマンドを実行します。ファイルのすべてのコマンドを実行した後、接続は終了されます。

```
sftp -B c:\mypath\myfile myhost.com myname
```

バッチファイルでは、以下に説明する任意の双方向コマンドを使用できます。

注意: セミコロンは、**-B** オプションを使って **sftp** コマンドに提供されたスクリプト内のコメントには使用できません。これらのバッチファイル内のコメントに印を付けるには、番号記号 (#) を使用します。

-c *cipher*

優先順に指定した Cipher のカンマ区切りの一覧。既定値は、

「aes128-ctr,aes128-cbc,aes192-ctr,aes192-cbc,aes256-ctr,aes256-cbc,3des-cbc,blowfish-cbc,cast128-cbc,arcfour128,arcfour256,arcfour」です。接続が *FIPS モード* 『[21 ページ](#)』で動作するように設定されている場合、既定値は

「aes128-ctr,aes128-cbc,aes192-ctr,aes192-cbc,aes256-ctr,aes256-cbc,3des-cbc」になります。

プロトコルバージョン 1(使用は現在推奨されない) では、Cipher を 1 つ指定できません。指定できる値は「3des」、「blowfish」、「des」です。

-C

すべての送信データの圧縮を有効にします。圧縮はモデム回線やほかの低速接続に適していますが、高速のネットワークでは応答速度の低下を招くだけです。

-d

対象をディレクトリにします。

-F *config_file*

この接続に使用する代替構成ファイルを指定します。構成ファイルをコマンドラインで指定した場合、ほかの構成ファイル 『[19 ページ](#)』は無視されます。

-H *scheme*

この接続に使用する *SSH 構成セクション* 『[112](#) ページ』を指定します。

-i *key_file*

鍵認証に使用する秘密鍵を指定します。鍵ファイルは、*構成ファイル* 『[19](#) ページ』でホストごとに指定することもできます。複数の **-i** オプションを指定できます (構成ファイルで複数の鍵を指定できます)。ファイルまたはパスが空白を含む場合、引用符を使用します。

-k *directory*

構成ファイル、ホスト鍵ファイル、ユーザ鍵ファイルの保存先に別の場所を指定します。

-m *mac_spec*

この接続に使用する 1 つまたは複数のカンマ区切り MAC (メッセージ認証コード) アルゴリズムを指定します。優先順にアルゴリズムを指定します。既定値は「`hmac-sha1,hmac-sha256,hmac-sha512,hmac-md5,hmac-ripemd160,hmac-sha1-96,hmac-md5-96`」です。接続が *FIPS モード* 『[21](#) ページ』で動作するように設定されている場合、既定値は「`hmac-sha1,hmac-sha256,hmac-sha512`」です。

-o *option*

構成ファイル 『[114](#) ページ』で対応するオプションを指定します。例えば、次のように入力します。

```
ssh "-o FIPSMode=yes" myuser@myhost
```

-P

タイムスタンプとファイルファイル属性を保存します。

-P *port*

リモートホスト上で接続するポート。

-q

クワイエットモードを有効にします。このモードでは、バナーを含むすべての警告および診断メッセージが表示されません。

-Q

進行状況インジケータの表示をオフにします。

-R *maximum_requests*

同時要求の最大数を指定します。大きい値を指定すると、ファイル転送速度が上がりますが、メモリ消費量が増えます。既定値は、未処理要求が 16 です。

-s *subsystem*

ssh サブシステムを指定します。

-S *program*

暗号化された接続に使用するプログラム。

-u

コピー後にソースファイルを削除します。

-v

デバッグレベルを冗長モードに設定します。これは、デバッグレベルを 2 に設定することと同じです。

-V

製品名およびバージョン情報を表示して終了します。コマンドラインで他のオプションが指定された場合、それらは無視されます。

-1

ssh がプロトコルバージョン 1 のみを試行するようにします。プロトコルバージョン 1 の使用は現在推奨されません。

-2

ssh がプロトコルバージョン 2 のみを試行するようにします。

-4

接続で IPv4 アドレスのみを使用するようにします。

-6

接続で IPV6 アドレスのみを使用するようにします。

双方向モード

ascii

転送の種類を ASCII に設定します。

binary

転送の種類をバイナリに設定します。

bye

sftp を終了します。

cd path

リモートディレクトリを path に変更します。

chmod path

path に関連付けられているファイル権限を変更します。mode を使用して、3 桁の数字のファイル権限を指定します。

lcd path

ローカルディレクトリを path に変更します。

exit

sftp を終了します。

get remote-path [local-path]

remote-path を取得して、ローカルマシンに格納します。ローカルパス名が指定されていない場合、リモートマシン上での同じ名前がローカルパス名として与えられます。

gettext [*extension,extension...*]

ascii 転送を使用するファイル拡張子を表示します。この一覧を変更するには、**settext** を使用します。

help

ヘルプテキストを表示します。

lls [*ls-options path*]

path または、*path* が指定されていない場合は現在のディレクトリのローカルディレクトリ一覧を表示します。

mkdir *path*

path で指定されたローカルディレクトリを作成します。

lpwd

ローカル作業ディレクトリを印刷します。

ls [*path*]

path または、*path* が指定されていない場合は現在のディレクトリのリモートディレクトリ一覧を表示します。

mkdir *path*

path で指定されたリモートディレクトリを作成します。

put *local-path [local-path]*

pwd

リモート作業ディレクトリを表示します。

quit

sftp を終了します。

reget *remote-file [local-file]*

指定された転送を再開します。これは **get** コマンドのように機能しますが、部分的に書き込まれたローカルファイルの存在を確認し、見つかった場合は最後に試行が中止された場所から転送を開始します。

rename *oldpath newpath*

リモートファイルの名前を *oldpath* から *newpath* に変更します。

rmdir *path*

path で指定されたリモートディレクトリを削除します。

rm *paths*

path で指定されたリモートファイルを削除します。

settext [*extension,extension...*]

ascii 転送を使用するファイル拡張子を設定します。ワイルドカード文字を使用できません。引数が指定されないと、どのファイル拡張子でも ASCII 転送は使用されません。

`version`

`sftp` のバージョンを表示します。

`?`

`help` と同義です。

sftp2 コマンドラインユーティリティ

構文: `sftp [options] [user@]host[#port]:source_file`
`[user@]host[#port]:destination_file`

sftp2 コマンドラインユーティリティでは、Secure Shell の F-Secure 実装と互換性のあるスイッチを使用できます。使用可能なスイッチを確認するには、コマンドウィンドウに以下のコマンドを入力します。

```
sftp2 -h
```

使用できる sftp コマンドの一覧については、「sftp」を参照してください。

scp コマンドラインユーティリティ

構文: `scp [options] [user@host:]file1 [user@host:]file2`

scp コマンドラインユーティリティは、ネットワークのホスト間でファイルを安全にコピーします。データ転送には Secure Shell `sftp` サブシステムを使用し、Secure Shell と同じ認証を使用して同じセキュリティを提供します。認証に必要な場合、`scp` はパスワードまたはパスフレーズを要求します。ファイル名には、ホストとユーザの指定を含めて、ファイルのコピー先/コピー元のホストを示すことができます。

例

次のコマンドラインは、ファイル `f1` をホストからローカルマシンにコピーして、名前を `f2` にします。

```
scp user@host:f1 f2
```

次のコマンドは、ローカルファイル `f1` をリモートホスト上の `f2` にコピーします。

```
scp f1 user@host:f2
```

注記: 既存の SecureShell 接続を再利用できます。ただし、そのためには各コマンドラインでこれを明示的に有効にするか、`SSHConnectionReUse` 環境変数を「Yes」に設定する必要があります。詳細については、「*Secure Shell セッションにおける接続の再利用*」『[28 ページ](#)』を参照してください。

オプション

使用できるオプションは、次のとおりです。

-a

ASCII モードでファイルを転送します。

-b *buffersize*

1 回の要求の最大バッファサイズを設定します。

-B

バッチモードを有効にします。バッチモードでは、パスワードまたはパスフレーズを要求されません。認証にはパスフレーズなしユーザ鍵を使用します。

-c *cipher*

優先順に指定した Cipher のカンマ区切りの一覧。既定値は、

「`aes128-ctr,aes128-cbc,aes192-ctr,aes192-cbc,aes256-ctr,aes256-cbc,3des-cbc,blowfish-cbc,cast128-cbc,arcfour128,arcfour256,arcfour`」です。接続が *FIPS モード* 『[21 ページ](#)』で動作するように設定されている場合、既定値は

「`aes128-ctr,aes128-cbc,aes192-ctr,aes192-cbc,aes256-ctr,aes256-cbc,3des-cbc`」になります。

プロトコルバージョン 1(使用は現在推奨されない) では、Cipher を 1 つ指定できません。指定できる値は「`3des`」、「`blowfish`」、「`des`」です。

-C

圧縮を有効にします。

-d

対象をディレクトリにします。

-D *level*

デバッグレベルを設定します。使用可能な値は 1、2、および 3 です。

-F *configfile*

新しくユーザ別の構成ファイル『19 ページ』を指定します。構成ファイルがコマンドラインで指定されると、システム全体の構成ファイルは無視されます。

-H *scheme*

この接続に使用する SSH 構成セクション『112 ページ』を指定します。

-i *keyfile*

RSA または DSA 認証の識別情報 (秘密鍵) を読み取るファイルを選択します。識別情報ファイルは、構成ファイルでホストごとに指定することもできます。複数の **-i** オプションを指定できます (構成ファイル『19 ページ』で複数の識別情報を指定できます)。空白を含むパス名は、二重引用符で囲む必要があります。

-k *directory*

構成ファイル、ホスト鍵ファイル、ユーザ鍵ファイルの保存先に別の場所を指定します。

-l *limit*

帯域幅を指定の値 (KB) に制限します。

-o *option*

構成ファイル『19 ページ』で使用される形式でオプションを指定する場合があります。これは、独立したコマンドラインフラグがないオプションを指定する場合に便利です。使用できるオプションの一覧については、「構成のキーワードのリファレンス『114 ページ』」を参照してください。

--overwrite

既存のコピー先ファイルを上書きするかどうかを指定します。使用可能な値は「yes」と「no」です。既定値は「yes」です。

-P

タイムスタンプとファイル属性を保持します。

-P *port*

リモートホスト上で接続するポート

-q

静的モード。警告および診断メッセージ (バナーを含む) がすべて表示されなくなります。

-Q

進行状況インジケータの表示をオフにします。

-r

ディレクトリ (すべてのサブディレクトリを含む) を再帰コピーします。

-u

コピー後にソースファイルを削除します。

-v

冗長モード。ssh が進捗状況に関するデバッグメッセージを表示するようになります。これは、デバッグ接続、認証、および構成の問題が発生した場合に役立ちます。複数の -v オプションを指定すると、冗長性が増します。最大値は 3 (-vvv) です。

-V

バージョン番号とアプリケーション情報を表示します。

-1

プロトコルバージョン 1 のみを使用します。また、このオプションを使用すると、ssh トンネルを介して rcp を使用する OpenSSH サーバにファイルを転送できます。

-2

プロトコルバージョン 2 のみを使用します。

-4

IPv4 アドレスのみを使用します。

-6

IPv6 アドレスのみを使用します。

scp2 コマンドラインユーティリティ

scp2 ユーティリティは、F-Secure から移行した場合に使用します。scp2 ユーティリティでは、Secure Shell の F-Secure 実装と互換性のあるスイッチを使用できます。使用可能なスイッチを確認するには、コマンドウィンドウに以下のコマンドを入力します。

```
scp2 -h
```


付録

このセクション内

Secure Shell クライアントが使用するファイル.....	110
SSH 構成セクション	112
サンプル構成ファイル.....	113
構成ファイルのキーワードのリファレンス - Secure Shell の設定.....	114
構成ファイルのキーワード参照 - 端末エミュレーション設定.....	125
DOD PKI 情報.....	130

付 録 A

Secure Shell クライアントが使用するファイル

以下のファイルが、Reflection for Secure IT Secure Shell クライアントによって使用されません。

ユーザ固有の Secure Shell ファイル

Windows に現在ログインしているユーザの Secure Shell 接続に影響するファイルです。これらのファイルはユーザの *.ssh* フォルダ『[135](#) ページ』にあります。

config

ユーザの構成ファイル。このファイルには、SSH 構成セクション『[112](#) ページ』によって管理された Secure Shell 設定が含まれています。このファイルの内容は、[Reflection Secure Shell の設定]『[16](#) ページ』ダイアログボックスを使用して設定を変更するたびに更新されます。また、任意のテキストエディタでこのファイルを直接編集することもできます。構成ファイルのキーワードのリファレンスでは、Reflection Secure Shell クライアントが対応しているキーワードの一覧が示されています。

known_hosts

[Reflection Secure Shell の設定] ダイアログボックスの [ホスト鍵] タブの [信頼されているホストキー]『[40](#) ページ』の一覧を更新した場合、または不明なホストに接続して、[ホスト鍵の信頼性] ダイアログボックスで [常時]『[41](#) ページ』を選択した場合、このファイルは自動的に更新されます。

システム全体で使用する Secure Shell ファイル

このコンピュータの全ユーザの Secure Shell 接続に影響するファイルです。これらのファイルは、手動で作成して Reflection アプリケーションデータフォルダ『[136](#) ページ』に保存する必要があります。

ssh_config

システム全体の構成ファイル。このファイルは、ユーザの構成ファイルで指定されていない値に対してマシン全体の既定を提供します。

ssh_known_hosts

システム全体の既知のホスト鍵の一覧。このファイルには、組織内のすべてのコンピュータの公開ホスト鍵を含める必要があります。このファイルには、システム名、公開鍵、オプションのコメントフィールド、という形式 (各フィールドは空白で区切られる) で公開鍵が 1 行に 1 つずつ含まれています。同じコンピュータに対して異なる名前が使用されている場合、それらの名前をすべて一覧に入れ、カンマで区切る必要があります。正式なシステム名 (ネームサーバによって返される名前) は、ユーザのログイン時に接続先ホストを確認するために使用されます。Secure Shell では、鍵の確認前にユーザが指定した名前が正式な名前に変換されることはないため、その他の名前が必要になります。これは、ネームサーバへのアクセス権を持つユーザがホスト認証を偽装できないようにするためです。[Reflection Secure Shell の設定] ダイアログボックスの [ホスト鍵] タブの [グローバルホスト鍵]『[40](#) ページ』にある鍵は、表示はできますが、編集はできません。

注記: 構成ファイルで、GlobalKnownHostsFile キーワード『[114](#) ページ』を設定して、ホスト鍵データベースの格納先に別の場所を指定できます。

PKI 対応の Reflection で使用されるファイル

Reflection で認証に PKI (Public Key Infrastructure) を使用するように設定している場合に使用されるファイルです。これらのファイルは、ユーザの `.pki` フォルダにあります。

pki_config

Reflection の証明書マネージャ『[49](#) ページ』を使用して構成した設定が保存されます。この設定は、すべての *Reflection* セッションで使用されます。

trust_store.p12

PKCS#12『[135](#) ページ』形式のファイル。*Reflection* 証明書マネージャ『[49](#) ページ』に追加された信頼されたルート証明書が含まれます。

identity_store.p12

PKCS#12『[135](#) ページ』形式のファイル。*Reflection* の証明書マネージャ『[49](#) ページ』に追加された秘密鍵と証明書が含まれます。

cert_cache

中間ルート証明書のキャッシュ。このファイルを削除すると、キャッシュをクリアできます。

crl_cache

CRL (Certificate Revocation List) のキャッシュ。このファイルを削除すると、キャッシュをクリアできます。

付 録 B

SSH 構成セクション

すべての Reflection Secure Shell 構成情報は、SSH 構成セクションを使用して *Secure Shell 構成ファイル* 『19 ページ』に保存されます。Secure Shell 接続時、Reflection では現在の SSH 構成セクションを使用して、接続方法を決定します。また、設定を変更すると、Reflection では変更を現在の SSH 構成セクションに保存します。

特定のホストに固有の Secure Shell 設定を構成したい場合、SSH 構成セクション名をホスト名と同じにする必要があります。

注意: セクションを指定せずに [Reflection Secure Shell の設定] ダイアログボックスを開いた場合、Secure Shell 設定のいずれかを変更するとすぐに、現在指定されているホスト名を使用して、Reflection により新しい SSH 構成セクションが自動的に作成されます。

複数のホストに対する接続に同じ Secure Shell 設定を使用したい場合、[Reflection Secure Shell の設定] ダイアログボックスを開く前に、SSH 構成セクションに内容を表す名前を入力し、このセクションに保存したい設定を構成します。セクションを作成および構成すると、以降のホストセッションの構成時にこのセクションを指定できます。

SSH 構成セクションの名前は、大文字と小文字が区別されます。

SSH 構成セクションの保存方法

Reflection Secure Shell 構成情報は、*Secure Shell 構成ファイル* 『19 ページ』に保存されます。SSH 構成セクション名は、キーワード Host を使用して識別されます。構成ファイルは、[Reflection Secure Shell の設定] ダイアログボックスを閉じると更新されます。構成する既定以外のすべての設定は、現在のセクションに保存されます。

例については、「*サンプル構成ファイル* 『113 ページ』」を参照してください。

付 録 C

サンプル構成ファイル

このサンプル Secure Shell 構成ファイルには、2 つの SSH 構成セクション
MyHost.Demo.com および GeneralSSH があります。

MyHost.Demo.com の設定では、実際のホスト名を使用する一連の Secure Shell 設定を指定
します。この設定は SSH 構成セクションとして MyHost.Demo.com を指定するすべての
接続に使用され、SSH 構成セクションが指定されていない場合にそのホストへの接続に
も使用されます。

GeneralSSH では実際のホストアドレスを指定しないため、この設定はセッション構成時
にこの SSH 構成セクションを指定した場合にのみ使用されます。

この config ファイルで、新しいホスト (MyHost.Demo.com 以外) への接続を構成し、
GeneralSSH セクションを指定しない場合、Reflection では既定の Secure Shell 設定を使用
して接続します。

```
Host MyHost.Demo.Com
  Protocol 2
  KbdInteractiveAuthentication no
  ChallengeResponseAuthentication no
  PasswordAuthentication no
  RSAAuthentication no
  IdentityFile "C:\SSHusers\Joe\.ssh\mykey"
  LogLevel VERBOSE
#EndHost

Host GeneralSSH
  StrictHostKeyChecking yes
  ServerAlive yes
#EndHost
```

付 録 D

構成ファイルのキーワードのリファレンス - Secure Shell の設定

Secure Shell *構成ファイル* 『[19](#) ページ』を直接編集する場合は、このリファレンスを使用してください。構成ファイルはセクションに分かれており、各セクションは Host キーワードで識別されます。各セクションでは、指定のホストまたは *SSH 構成セクション* 『[112](#) ページ』を使用して確立されるすべての接続用に Secure Shell 設定を指定します

構成ファイルは、キーワードの後に値が続きます。構成オプションは、空白またはオプションの空白と 1 つの等号 (=) で区切ることができます。キーワードは大文字と小文字を区別しませんが、引数は大文字と小文字を区別します。

番号記号 (#) で始まる行はコメントです。空の行は無視されます。

注記: この一覧の項目は、Secure Shell 接続に関する機能を設定します。追加のキーワードを使用して、ssh コマンドラインセッションの端末エミュレーションを構成できます。これらのキーワードの詳細については、「*構成ファイルのキーワードのリファレンス - 端末エミュレーションの設定* 『[125](#) ページ』」を参照してください。

BatchMode

パスワードやパスフレーズの入力画面を含め、ユーザ入力に関するすべての問い合わせを無効にするかどうかを指定します。このキーワードは、スクリプトおよびバッチジョブの場合に役立ちます。指定可能な値は「yes」および「no」です。既定値は「no」です。

注記: キーボード対話型認証が構成されている場合は、このキーワードでユーザ入力に関する問い合わせを無効にできません。ただし、BatchMode が有効な場合、キーボード対話型認証を使用する接続は失敗します。

BindAddress

複数のインタフェースまたは別名を付けられたアドレスを持つコンピュータ上の、送信元となるインタフェースを指定します。

ChallengeResponseAuthentication

試行/応答認証を使用するかどうかを指定します。引数は、「yes」または「no」です。サーバからのプロンプトとユーザからの応答を必要とする SecurID や PAM 認証などの外部認証方式を使用している場合は、この認証方式をお勧めします。既定値は「yes」です。これは、対応している SSH プロトコル 1 のみに適用されますが、推奨されません。SSH プロトコルバージョン 2 には KbdInteractiveAuthentication を使用してください。

CheckHostIP

このフラグが「yes」に設定されている場合、Reflection Secure Shell クライアントは、ホストの公開鍵の確認に加えて、known_hosts ファイルでホスト IP アドレスを確認します。既知のホスト一覧内のホストの IP が、接続に使用している IP アドレスに一致する場合にかぎり、接続が許可されます。既定値は「no」です。注記: StrictHostKeyChecking が no に設定されている場合、この設定は適用されません。

CheckHostPort

このフラグが「yes」に設定されている場合、Reflection Secure Shell クライアントは、ホストの公開鍵の確認に加えて、`known_hosts` ファイルでホストのポートを確認します。既知のホスト一覧内のホストのポートが、接続に使用しているポートに一致する場合にかぎり、接続が許可されます。既定値は「no」です。注記:
`StrictHostKeyChecking` が no に設定されている場合、この設定は適用されません。

Cipher

プロトコルバージョン 1 のセッションの暗号化に使用する Cipher を指定します。現在、「blowfish」、「3des」、および「des」に対応しています。des は、3des の Cipher に対応していない旧プロトコル 1 実装との相互運用性を確保するために、Secure Shell クライアントでのみ使用できます。des は、暗号が脆弱であるため、ほとんど使用されなくなりました。既定値は「3des」です。

Ciphers

プロトコルバージョン 2 で使用できる Cipher を優先順に指定します。複数の Cipher を指定する場合は、カンマで区切る必要があります。既定値は、「aes128-ctr,aes128-cbc,aes192-ctr,aes192-cbc,aes256-ctr,aes256-cbc,3des-cbc,blowfish-cbc,cast128-cbc,arcfour128,arcfour256,arcfour」です。接続が FIPS モードで動作するように設定されている場合、既定値は「aes128-ctr,aes128-cbc,aes192-ctr,aes192-cbc,aes256-ctr,aes256-cbc,3des-cbc」になります。

ClearAllForwardings

ローカル転送、リモート転送、または動的転送されたポートのうち、既に処理されたすべてのポートを構成ファイルまたはコマンドラインからクリアします。注記: scp と sftp 利用の場合は、この設定の値に関係なく、転送されたすべてのポートが自動的にクリアされます。指定可能な値は「yes」および「no」です。既定値は「no」です。

圧縮

圧縮を有効にするかどうかを指定します。圧縮は、モデム回線などの低速接続には向いていますが、高速ネットワークでは応答速度を低下させます。また、圧縮はパケットをより不規則にするため、悪意のある人物がパケットを解読することが難しくなります。指定可能な値は「yes」および「no」です。既定値は「no」です。

CompressionLevel

圧縮が有効な場合に使用する圧縮レベルを指定します。このオプションは、プロトコルバージョン 1 にのみ適用されます。引数には、1 (高速) ~ 9 (低速、高圧縮) の整数を指定する必要があります。既定値は 6 で、このレベルはほとんどのアプリケーションに適しています。値の意味は gzip と同じです。

ConnectionAttempts

終了する前に接続を試行する回数 (1 秒に 1 回) を指定します。引数には、整数を指定する必要があります。接続が時々失敗する場合、スクリプトでこれを使用すると便利です。既定値は 1 です。

ConnectionReuse

再認証が不要になるように、同じホストへの複数のセッションで元の Secure Shell 接続を再使用するかどうかを指定します。引数は、「yes」または「no」です。「yes」に設定すると、新しい接続でホスト名、ユーザ名、SSH 構成セクション (使用する場合) がすべて一致した時に既存のトンネルが再使用されます。「no」に設定にすると、各セッションで新しい接続が確立されます。つまり、新しい接続ごとに認証処理が繰り返され、接続固有の設定 (転送や暗号化など) が変更されている場合はそれらが適用されます。Reflection のウィンドウを使用して接続を確立する場合の既定値は「yes」です。コマンドラインユーティリティ『[88](#) ページ』を使用して接続を確立する場合の既定値は「no」です。詳細については、「*Secure Shell セッションにおける接続の再利用*『[28](#) ページ』」を参照してください。

DisableCRL

ホストの証明書の検証時に CRL (Certificate Revocation List) を確認するかどうかを指定します。「yes」に設定すると、CRL が確認されなくなります。この設定の既定値は、現在のシステムで CRL の確認がどのように設定されているかによって決まります。システムの設定を確認および編集するには、Internet Explorer を起動し、[ツール] - [インターネットオプション] - [詳細設定] を選択します。[セキュリティ] の [サーバ証明書の取り消しを確認する] チェックボックスがオンになっているかどうかを確認します。

DynamicForward

安全なチャネルを介してローカルマシン上の TCP/IP ポートを転送し、アプリケーションプロトコルを使用してリモートマシンからの接続先を決定するように指定します。引数には、ポート番号を指定する必要があります。現在、SOCKS4 プロトコルに対応しており、Reflection Secure Shell は SOCKS4 サーバとして動作します。複数の転送の指定が可能で、追加の転送は、コマンドラインで指定できます。管理者権限を持つユーザのみが、権限対象のポートを転送できます。

EscapeChar

エスケープ文字を設定します (既定値は「~」)。エスケープ文字は、コマンドライン上でも設定できます。引数には、1 文字、または「^」の後に 1 文字を指定するか、「none」を指定してエスケープ文字全体を無効にします (バイナリデータに対して接続を透過的にします)。

FipsMode

「yes」に設定すると、接続を確立する際に米国政府の連邦情報処理規格 (FIPS) 140-2 に適合したセキュリティプロトコルおよびアルゴリズムが使用されます。この規格に適合しないオプションは、[暗号化] タブで使用できません。

注記: この設定は、Host キーワードで指定される SSH 構成セクションに適用されます。同じ SSH 構成セクション (またはホスト名) を使用するように後続の Secure Shell セッションを設定する場合を除いて、この設定は後続のセッションには影響を与えません。

ForwardAgent

このオプションを「yes」に設定すると、Reflection 鍵エージェント接続の転送が有効になります。エージェント転送を有効にする場合は注意が必要です。エージェントの UNIX ドメインソケットのリモートホストでファイル権限を回避できるユーザは、転送された接続を介してローカルエージェントにアクセスできます。攻撃者は鍵の情報を取得できませんが、エージェントに読み込まれた識別情報を使用して、その鍵で操作を実行して認証を有効にすることができます。場合によっては、サーバでもエージェント転送を有効にする必要があります。既定値は「no」です。

ForwardX11

X11 接続を安全なチャネルを介して自動的にリダイレクトし、DISPLAY を設定するかどうかを指定します。引数は、「yes」または「no」です。既定値は「no」です。

注記: Reflection X を使用して Secure Shell を構成する場合、ForwardX11 は自動的に「yes」に設定されます。

GatewayPorts

転送されたローカルポートへの接続をリモートホストに許可するかどうかを指定します。既定では、Reflection Secure Shell はローカルポート転送をループバックアドレスにバインドします。これにより、ほかのリモートホストが転送されたポートに接続することを防ぎます。GatewayPorts を使用すると、Reflection Secure Shell がローカルポート転送をワイルドカードアドレスにバインドし、転送されたポートにリモートホストが接続できるように指定できます。この設定を有効にする場合は注意が必要です。このキーワードを使用すると、システムに転送されたポートを認証なしでリモートホストから使用できるので、ネットワークと接続のセキュリティが低下する可能性があります。引数は、「yes」または「no」です。既定値は「no」です。

GlobalKnownHostsFile

Reflection アプリケーションデータフォルダ『[136](#) ページ』内の既定ファイル `ssh_known_hosts` の代わりに、グローバルホスト鍵データベースに使用するファイルを指定します。

注記: パス名またはファイル名に空白が含まれている場合は、ファイル名を引用符で囲みます。

GssapiAuthentication

Kerberos KDC に対する認証に GSSAPI 認証を使用するかどうかを指定します。この設定は、使用しているプロトコルがプロトコルバージョン 2 の場合のみ該当します (プロトコルバージョン 1 での同等の設定は、KerberosAuthentication です)。指定可能な値は「yes」および「no」です。既定値は「no」です。

GssapiDelegateCredentials

ホストへの発券許可チケット (krbtgt) の転送に GSSAPI を使用するかどうかを指定します。この設定は、使用しているプロトコルがプロトコルバージョン 2 の場合のみ該当します (プロトコルバージョン 1 での同等の設定は、KerberosTgtPassing です)。指定可能な値は「yes」および「no」です。既定値は「yes」です。

GssapiUseSSPI

Microsoft の SSPI (Security Support Provider Interface) を GSSAPI 認証に使用するかどうかを指定します。この設定は、Kerberos/GSSAPI 認証が有効になっている (プロトコルバージョン 2 では GssapiAuthentication を、プロトコルバージョン 1 では KerberosAuthentication を使用) 場合のみ適用できます。このキーワードの引数は、「yes」または「no」です。「no」に設定した場合は、Reflection Secure Shell クライアントで GSSAPI 認証に Reflection Kerberos クライアントが使用されます。「yes」に設定した場合は、Reflection Secure Shell クライアントで Secure Shell サーバへの認証に Windows ドメインのログイン資格情報 (SSPI) が使用されます。SSPI はプロトコルバージョン 2 接続のみに対応しており、サーバが GSSAPI-with-mic 認証方式に対応している必要があります。既定値は「yes」です。

GssServicePrincipal

クライアントが Kerberos の鍵配布センター (KDC) にサービスチケット要求を送信する時に使用する、既定以外のサービスプリンシパル名を指定します。GSSAPI プロバイダに SSPI を選択した場合は、この設定を使用して、Windows ドメインと異なるレルムのサービスプリンシパルを指定できます。完全なホスト名、@、レルム名の順に指定します。例えば、myhost.myrealm.com@MYREALM.COM のようになります(既定で、ホスト名の値は接続する Secure Shell サーバの名前になり、レルムは GssapiUseSSPI の値によって異なります。GSSapiUseSSPI が「no」の場合は、レルム名は既定のプリンシパルプロファイルで指定した名前になります。GSSapiUseSSPI が「yes」の場合は、レルム名は Windows ドメイン名です)。

Host

後続 (次の Host キーワードまで) の宣言の対象を、指定した SSH 構成セクション『[112](#) ページ』に属しているものと識別します。「*」と「?」の文字は、ワイルドカードとして使用できます。パターンとしての単独の「*」は、全ホストに共通の既定を指定するために使用できません。Reflection 接続は、Host 文字列 (ワイルドカード文字を含む) と一致する最初の項目を使用します。以降の一致は無視されます。

注記: [Reflection Secure Shell の設定] ダイアログボックスを閉じる時は、既定の設定を使用する値は構成ファイルに保存されません。既定値が手動でファイルに追加されている場合、その値は、このダイアログボックスを閉じると削除されます。特定のホスト名を使用するスタンザと組み合わせてワイルドカード使用のホストスタンザを使用する場合、この状況はホストの作成上の制約になります。ワイルドカード使用のスタンザに構成されている値を無効にする目的で、特定のホストスタンザ内に既定値を手動で構成している場合、ホスト固有の SSH 構成セクションの設定を表示するために [Secure Shell の設定] ダイアログボックスを開くと、この既定の設定が削除されます。このような状況は、グローバル構成ファイルを使用することで適切に対処できます。グローバル構成ファイルは、[Reflection Secure Shell の設定] ダイアログボックスを開いたり閉じたりしても更新されません。

HostKeyAlgorithms

クライアントが使用するプロトコルバージョン 2 ホスト鍵アルゴリズムを優先順に指定します。このオプションの既定値は、「x509v3-sign-rsa,x509v3-sign-dss,ssh-rsa,ssh-dss」です。この設定は、サーバが証明書認証と標準のホスト鍵認証の両方を行うように設定されている場合に有効です。SSH プロトコルでは、ホスト認証は 1 回だけ試行できます。ホストで証明書が提示され、クライアントが証明書を使用したホスト認証を行うように設定されていない場合は、接続に失敗します(複数回の認証試行に対応しているユーザ認証とは異なります)。

HostKeyAlias

ホスト鍵データベースファイルでホスト鍵を検索したり保存するために、実際のホスト名の代わりに使用する別名を指定します。このオプションは、ssh 接続をトンネリングしたり、単一ホストで複数のサーバを実行する場合に役立ちます。

IdentityFile

鍵認証に使用する秘密鍵を指定します。これらのファイルは、ユーザの .ssh フォルダ『[135](#) ページ』にあります。IdentityFile の項目は、[Secure Shell の設定] ダイアログボックスの [ユーザ鍵] タブにある一覧から鍵または証明書を選択すると追加されます。構成ファイルに複数の識別情報ファイルを指定することができます。これらすべての識別情報は、順番に試行されます。

注記: パス名に空白が含まれている場合は、パス名全体を引用符で囲みます。

KbdInteractiveAuthentication

キーボード対話型認証を使用するかどうかを指定します。指定可能な値は「yes」および「no」です。既定値は「yes」です。サーバからのメッセージとユーザからの応答を必要とする SecurID や PAM 認証などの外部認証方式を使用している場合は、この認証方式をお勧めします。また、パスワードの有効期限切れや初回ログインパスワードの変更が有効になっているホストでのパスワード認証では、この方式のほうが PasswordAuthentication 方式よりも適切に機能することがあります。このキーワードは、認証を正しく行うために有効期限が切れたパスワードをリセットする必要がある場合のパスワード認証でも必要になります。これは SSH プロトコル 2 のみに適用されます。SSH プロトコルバージョン 1 には ChallengeResponseAuthentication を使用してください。

KeepAlive

システムが TCP キープアライブメッセージをほかのサイトに送信する必要があるかどうかを指定します。キープアライブメッセージを送信すると、接続の切断やいずれかのマシンのクラッシュが検出されます。既定値は「yes」(キープアライブの送信)です。これによりクライアントは、ネットワークのダウンやリモートホストからの切断を検出できます。これはスクリプトでは重要で、ユーザにとって役立ちます。ただし、ルートが一時的にダウンした場合でも接続が切断されることになるので、ユーザによっては好ましくないことがあります。キープアライブを無効にするには、値を「no」に設定します。このキーワードは、Windows TCP キープアライブ設定を有効にします。既定では、2 時間ごとにキープアライブメッセージが送信されます。TCP/IP キープアライブの設定には、Windows レジストリに通常は存在しない、KeepAliveTime と KeepAliveInterval という 2 つのオプションパラメータを使用できます。これらのパラメータは、以下の場所にある HKEY_LOCAL_MACHINE レジストリサブツリーに構成します。

```
SYSTEM\CurrentControlSet\Services\Tcpip\Parameters
```

これらのパラメータを設定する方法については、Microsoft サポート技術情報 120642 を参照してください。

KerberosAuthentication

プロトコルバージョン 1 の接続に Kerberos 認証を使用するかどうかを指定します(プロトコルバージョン 2 での同等の設定は GssapiAuthentication です)。このキーワードの引数は、「yes」または「no」です。

KerberosTgtPassing

Kerberos TGT をサーバに転送するかどうかを指定します。これは、Kerberos サーバが実際に AFS kaserver である場合のみ機能します。この設定は、プロトコルバージョン 1 にも適用されます(プロトコルバージョン 2 での同等の設定は GssapiDelegateCredentials です)。このキーワードの引数は、「yes」または「no」です。

KexAlgorithms

クライアントが対応する鍵交換アルゴリズムと、その優先順位を指定します。指定できる値は「diffie-hellman-group1-sha1」、「diffie-hellman-group-exchange-sha1」、および「diffie-hellman-group14-sha1」です。既定値は「diffie-hellman-group1-sha1,diffie-hellman-group-exchange-sha1,diffie-hellman-group14-sha1」です。

注記: Reflection Kerberos クライアントを使用する GSSAPI 認証が有効になっている場合は、追加の鍵交換アルゴリズム gss-group1-sha1 および gss-gex-sha1 が一覧に自動的に追加されます。

LocalForward

ローカルマシン上の TCP/IP ポートが、安全なチャネルを介してリモートマシンの指定したホストおよびポートに転送されるように指定します。複数の転送の指定が可能です。管理者権限を持つユーザのみが、権限対象のポートを転送できます。また接続確立後に、FTP の転送、リモートデスクトップの構成、実行ファイル (*.exe) の自動起動を行うための引数を、必要に応じて設定することもできます。このキーワードの構文は以下のとおりです。

```
LocalForward localport host:hostport [FTP=0|1] [RDP=0|1] [<実行ファイル名> [args]]
```

ファイル名の変換方法は次の 3 種類です。

<i>localport</i>	ローカルポート番号
<i>host:hostport</i>	リモートホストとそのホスト上のポート(<i>localhost</i> を指定して、すでに Secure Shell 接続を確立した同じリモートホストの異なるポートにデータを転送できます)。IPv6 アドレスは、 <i>host/port</i> というもう 1 つの構文を使用して指定できます。
FTP	FTP ファイル転送をトンネリングする場合は、1 に設定します。
RDP	リモートデスクトップセッションをトンネリングする場合は、1 に設定します。
<実行ファイル名>	Secure Shell 接続の確立直後にアプリケーションを起動するには、実行ファイルを指定します。必要に応じて完全なパス情報を含めます。安全なトンネルを介してデータを転送するには、指定の <i>localport</i> を使用して、 <i>localhost</i> (またはループバック IP アドレス 127.0.0.1) に接続するようにこのアプリケーションを設定する必要があります。

Logfile

デバッグ用にログファイルを指定します。すべてのセッションの入出力がこのファイルに書き込まれます。以下に示すように、このキーワードは `-o` コマンドラインユーティリティオプションとともに使用します。

```
-o Logfile=%path%logfile_name
```

注記: パスまたはファイル名に空白が含まれている場合は、パス/ファイル名を引用符で囲みます。

LogLevel

Reflection Secure Shell クライアントからのメッセージを記録する時に使用する冗長レベルを指定します。可能な値は、QUIET、FATAL、ERROR、INFO、VERBOSE、DEBUG、DEBUG1、DEBUG2、および DEBUG3 です。既定値は INFO です。DEBUG と DEBUG1 は同等です。DEBUG2 と DEBUG3 はそれぞれ、高位レベルの冗長出力を指定します。

Mac

MAC (メッセージ認証コード) アルゴリズムを優先順に指定します。MAC アルゴリズムは、データの整合性を保護するためにプロトコルバージョン 2 で使用されます。複数のアルゴリズムを指定する場合は、カンマで区切る必要があります。既定値は「hmac-sha1,hmac-sha256,hmac-sha512,hmac-md5,hmac-ripemd160,hmac-sha1-96,hmac-md5-96」です。接続が FIPS モードで動作するように設定されている場合、既定値は「hmac-sha1,hmac-sha256,hmac-sha512」です。

MatchHostName

ホストの証明書の検証時にホスト名の一致を確認するかどうかを指定します。この設定を「yes」にした場合 (既定)、Reflection で構成するホスト名は、証明書の CommonName フィールドまたは SubjectAltName フィールドに入力されたホスト名と完全に一致している必要があります。

Multihop

マルチホップ『67 ページ』接続を設定します。マルチホップ接続は、複数の SSH サーバを介した安全な接続の確立に使用できます。これは、ネットワーク構成でリモートサーバへの直接アクセスは許可されていないが中間サーバへのアクセスは許可されている場合に役立ちます。

このキーワードの構文は以下のとおりです。

```
Multihop localport host:hostport
```

サーバごとに新しい Multihop 行を追加します。一覧の各接続は、その上の接続で確立されたトンネルを介して送信されます。

以下の例では、ServerC に対して設定された SSH 接続は、最初に ServerA に接続し、次に ServerB、最後に ServerC に接続します。

```
Host ServerC
    Multihop 2022 ServerA:22
    Multihop 3022 ServerB:22
```

NoShell

NoShell が「Yes」に設定されている場合、クライアントは、端末セッションを開かずにトンネルを作成します。このオプションは、ほかの ssh 接続で再利用可能なトンネルを作成するために ConnectionReuse と組み合わせて使用できます。注記: このオプションは、コマンドラインユーティリティを使用して確立された接続に適用され、Reflection for Secure IT ユーザインタフェースで使用するものではありません。

NumberOfPasswordPrompts

パスワード入力を求める回数を指定します。このキーワードの引数には、整数を指定する必要があります。既定値は 3 です。

PasswordAuthentication

パスワード認証を使用するかどうかを指定します。指定可能な値は「yes」および「no」です。既定値は「yes」です。

Port

リモートホスト上で接続するポート番号を指定します。既定値は 22 です。

PreferredAuthentications

クライアントがプロトコル 2 認証方式を試行する順序を指定します。これは、[Reflection Secure Shell の設定] ダイアログボックスの [一般] タブにある [ユーザ認証] 一覧に表示される方式の順序（上から下）に対応します。この設定を使用して、クライアントは 1 つの方式（キーボード対話型など）をほかの方式（パスワードなど）よりも優先できるようになります。既定では、Reflection で試行される認証の順序は、公開鍵、キーボード対話型、パスワードです。GSSAPI 認証が有効になっている場合、既定の順序は、GSSAPI-WITH MIC、EXTERNAL-KEYEX、GSSAPI、公開鍵、キーボード対話型、パスワードに変わります。

注記:

- 構成ファイルに PreferredAuthentications を含める場合は、指定する一覧に、試行する各認証方式を含める必要があります。PreferredAuthentications が存在しても認証方式が何も指定されていないと、認証方式を有効にするキーワードが正しく設定されている場合でも、Reflection でその認証方式は使用されません。
 - 認証方式を PreferredAuthentications の一覧に含めても、その方式を使用する認証は有効になりません。既定で使用されない認証方式を有効にするには、その認証方式のキーワードも正しく構成する必要があります。例えば、GSSAPI 認証を有効にするには、GssapiAuthentication を「yes」に設定します。
-

[プロトコル]

Secure Shell クライアントが対応するプロトコルバージョンを優先順に指定します。可能な値は「1」と「2」です。複数の値を指定する場合は、カンマで区切る必要があります。既定値は「2,1」です。これは、Reflection がまずバージョン 2 を試行し、バージョン 2 を使用できない場合はバージョン 1 にフォールバックすることを意味します。

Proxy

Secure Shell 接続で使用するプロキシの種類を指定します。指定できる値は「SOCKS」と「HTTP」です。

注記: この設定を使用して構成ファイルの Host セクションごとにプロキシの使用を有効にできます。プロキシサーバアドレスは、Windows レジストリにユーザ別に格納されます。

PubkeyAuthentication

公開鍵認証を試行するかどうかを指定します。このオプションは、プロトコルバージョン 2 にのみ適用されます。指定可能な値は「yes」および「no」です。既定値は「yes」です。

RemoteCommand

リモートサーバで実行する 1 つまたは複数のコマンドを指定します。セミコロン (;) を使用して、複数のコマンドを区切ります。接続の確立後、サーバは指定したコマンドを実行（または実行を試行）し、セッションが終了します。サーバは、クライアントから受信したコマンドの実行を許可するよう構成されている必要があります。

RemoteForward

リモートマシン上の TCP/IP ポートが、安全なチャネルを介してローカルマシンの指定したホストおよびポートに転送されるように指定します。最初の引数にはポート番号を、2 番目の引数には *host:port* を指定します。IPv6 アドレスは、*host/port* というもう 1 つの構文を使用して指定できます。複数の転送の指定が可能です。管理者権限を持つユーザのみが、権限対象のポートを転送できます。

RSAAuthentication

RSA 認証を試行するかどうかを指定します。このオプションは、プロトコルバージョン 1 にのみ適用されます。RSA 認証は、識別情報ファイルがある場合にのみ試行されます。指定可能な値は「yes」および「no」です。既定値は「yes」です。

SendEnv

シェルまたはコマンドの実行前にサーバ上に設定する環境変数を指定します。値の形式は `VAR val` にする必要があります。サーバが、指定した変数に対応し、また、これらの環境変数を使用するように構成されている必要があります。

ServerAlive

サーバアライブメッセージを、`ServerAliveInterval` によって指定された間隔で、SSH サーバに送信するかどうかを指定します。Secure Shell の `ServerAlive` 設定では、指定された間隔で SSH プロトコルメッセージをサーバに送信して、サーバが動作していることを確認します。この設定が無効の場合、サーバがダウンしたり、ネットワーク接続が切断されても、SSH 接続は終了しません。この設定を使用して、TCP セッションのみを転送する接続がサーバで時間切れになるのを防ぐこともできます。この設定を使用しないと、サーバで SSH トラフィックが検出されない場合、これらの接続は時間切れになります。指定可能な値は「yes」および「no」です。既定値は「no」です。

注記: Windows のレジストリに TCP キープアライブ設定 (KeepAlive) というのがありますが、これはファイアウォールによってすべての TCP/IP 接続が時間切れにならないようにするものです。Secure Shell の `ServerAlive` 設定は、この TCP キープアライブ設定とは関係ありません。TCP/IP のキープアライブの動作を変更するには、Windows のレジストリを編集する必要があります。

ServerAliveInterval

`ServerAlive` が「yes」の場合に使用する間隔 (秒単位) を指定します。1 以上の整数値を使用します。既定値は 30 です。

SftpBufferLen

SFTP 転送中に各パケットで要求されるバイト数を指定します。既定値は 32768 です。この値を調整して、転送速度を上げることができます。最適な値は、ネットワークとサーバの設定によって異なります。この値を変更すると、転送をキャンセルできるまでの時間に影響する場合があります。

SftpMaxRequests

SFTP 転送中にクライアントに許可される未処理のデータ要求数を指定します。既定値は 10 です。この値を調整して、転送速度を上げることができます。最適な値は、ネットワークとサーバの設定によって異なります。この値を変更すると、転送をキャンセルできるまでの時間に影響する場合があります。

StrictHostKeyChecking

引数は、「yes」、「no」、または「ask」です。既定値は「ask」です。このオプションを「yes」に設定すると、Reflection Secure Shell クライアントはホスト鍵を `known_hosts` ファイル (ユーザの `.ssh` フォルダ『135 ページ』内) に自動的に追加しなくなり、ホスト鍵が変更されているホストへの接続を拒否します。このオプションは、ユーザが直接すべての新規ホストを追加できるようにします。このフラグを「no」に設定すると、Reflection は確認のダイアログボックスを表示せずにホストに接続します。ホスト鍵は信頼されている鍵の一覧に追加されません。このフラグを「ask」に設定すると、ユーザが確認した場合のみ、新しいホスト鍵がユーザの既知のホストファイルに追加されます。既知のホストのホスト鍵は、常に自動的に確認されます。

注記: ホストが x509 証明書を使用して認証を行うように設定されている場合は、この設定は適用されません。ホストでホスト認証用の証明書が提示されたが、必要なその CA 証明書を信頼アンカーとして構成していない場合は、接続に失敗します。

TryEmptyPassword

このフラグを「yes」に設定すると、クライアントは、空のパスワードの入力を試行してパスワード認証を開始します。この試行は、大多数のシステムでログイン試行としてカウントされることに注意してください。

User

ログインに使用するユーザ名を指定します。これは、別のマシンで別のユーザ名を使用する場合に便利です。

UseOCSP

ホストの証明書の検証にクライアントで OCSP (Online Certificate Status Protocol) を使用するかどうかを指定します。指定可能な値は「yes」および「no」です。既定値は「no」です。

UserKnownHostsFile

`known_hosts` ファイル (ユーザの `.ssh` フォルダ『135 ページ』内) ファイルまたはパスが空白を含む場合、引用符を使用します。の代わりにユーザホスト鍵データベースに使用するファイルを指定します。

x509dsasigtype

DSA 秘密鍵の所有を確認する過程でクライアントが使用するハッシュアルゴリズムを指定します。使用可能な値は「sha1raw」(既定) と「sha1asn1」です。

x509rsasigtype

RSA 秘密鍵の所有を確認する過程でクライアントが使用するハッシュアルゴリズムを指定します。使用可能な値は「md5」と「sha1」(既定) です。

付 録 E

構成ファイルのキーワード参照 - 端末エミュレーション設定

この一覧の項目は、Reflection *ssb* 『[90](#) ページ』 コマンドラインセッションと *ssb2* 『[94](#) ページ』 コマンドラインセッションの端末エミュレーション設定を構成します。これらの設定は、Secure Shell **構成ファイル** 『[19](#) ページ』に手動で追加するか、コマンドラインで `-o` スイッチを使用して実行できます。

構成ファイルは、キーワード `Host` によってそれぞれ識別されるセクションに分かれています。各セクションでは、指定したホストまたは *SSH 構成セクション* 『[112](#) ページ』を使用する、すべての接続に使用される設定を指定します。

構成ファイルは、キーワードの後に値が続きます。構成オプションは、空白またはオプションの空白と 1 つの等号 (=) で区切ることができます。キーワードは大文字と小文字を区別しませんが、引数は大文字と小文字を区別します。

番号記号 (#) で始まる行はコメントです。空の行は無視されます。

引用符は、空白を含む文字列引数の前後に必要になります。端末エミュレーションのキーワードおよび引数は、大文字と小文字を区別しません。

注意: Secure Shell 接続を構成するためのキーワードは、別個の一覧に示されています。「**構成ファイルのキーワード参照 - Secure Shell 設定** 『[114](#) ページ』」を参照してください。

AnswerBackMessage

`AutoAnswerback` を「yes」に設定すると、`AnswerBackMessage` でアンサバック要求に応じてホストに送信される文字列を指定します。

指定可能な値は、30 字までの文字列値です。

既定値は "" (ヌル文字列) です。

サンプル構文は次のとおりです。

```
AutoAnswerback yes
AnswerbackMessage "My answer back string"
```

AutoAnswerback

`AutoAnswerback` を「yes」に設定すると、キーワード `AnswerBackMessage` を使用して指定したメッセージ文字列が接続後にホストに自動的に送信されます。

指定可能な値は「yes」または「no」です。

既定値は「no」です。

サンプル構文は次のとおりです。

```
AutoAnswerback yes
AnswerbackMessage "My answer back string"
```

AutoWrap

カーソルが右余白に到達した時の動作を指定します。「yes」に設定すると、カーソルが端末ウィンドウの右余白に到達した時に、文字が次の行に自動的に折り返されます。「no」に設定すると、カーソルが右余白に到達した時に自動的に進みません。追加の文字を入力すると、各文字はカーソルを移動するまで前の文字を上書きします。

指定可能な値は「yes」または「no」です。

既定値は「no」です。

サンプル構文は次のとおりです。

```
AutoWrap yes
```

BackspaceKeyIsDel

バックスペースキーの動作を指定します。「no」に設定すると、バックスペースキーはバックスペース (ASCII 8) 文字を送信します。「yes」に設定すると、バックスペースキーは削除 (ASCII 127) 文字を送信します。

指定可能な値は「yes」または「no」です。

既定値は「no」です。

サンプル構文は次のとおりです。

```
BackspaceKeyIsDel yes
```

CursorKeyMode

クライアントがカーソルキーパッドのキーをどのように処理するかを指定します。「no」に設定すると、カーソルキーパッドは通常モードに設定されます。つまり、カーソルキーパッドのキーはカーソルエスケープシーケンスを送信します。「yes」に設定すると、カーソルキーパッドはアプリケーションモードに設定されます。つまり、カーソルキーパッドのキーはアプリケーションエスケープシーケンスを送信します。

指定可能な値は「yes」または「no」です。

既定値は「no」です。

サンプル構文は次のとおりです。

```
CursorKeyMode yes
```

CursorStyle

カーソルスタイルを指定します。

指定可能な値は「Block」、「Blockblink」、「Line」、「Lineblink」です。

既定値は「Lineblink」です。

サンプル構文は次のとおりです。

```
CursorStyle Block
```

CursorVisible

カーソルを表示するかどうかを指定します。「no」に設定すると、カーソルは端末ウィンドウに表示されません。

指定可能な値は「yes」または「no」です。

既定値は「yes」です。

サンプル構文は次のとおりです。

```
CursorVisible no
```

DisplayCols

端末ウィンドウの桁数を設定します。

指定可能な値の最小値は 80 です。最大値に使用可能な値は、モニタサイズと表示設定によって異なります。

既定値は現在のコマンドウィンドウのサイズによって決まります。

サンプル構文は次のとおりです。

```
DisplayCols 120
```

DisplayRows

端末ウィンドウの行数を設定します。

指定可能な値の最小値は 24 です。最大値に使用可能な値は、モニタサイズと表示設定によって異なります。

既定値は現在のコマンドウィンドウのサイズによって決まります。

サンプル構文は次のとおりです。

```
DisplayRows 30
```

HostCharacterSet

既定以外のホスト文字セットを指定します。

指定可能な文字列値は次のとおりです。

PC437_English	Windows1256
PC737_Greek	Windows1257
PC775_Baltic	Windows1258
PC850_Multilingual	Korean_Johab
PC852_Slavic	ISOLatin_1
PC855_Cyrillic	ISOLatin_2
PC857_Turkish	ISOLatin_3
PC858_Multilingual_Euro	ISO_Baltic
PC860_Portuguese	ISO_Cyrillic
PC861_Icelandic	ISO_Arabic
PC862_Hebrew	ISO_Greek
PC863_CanadianFrench	ISO_Hebrew
PC864_Arabic	ISOLatin_5
PC865_Nordic	ISOLatin_9
PC866_Cyrillic	ISO2022_JIS
PC869_ModernGreek	ISO2022_JIS-Allow
PC932_Shift_JIS	ISO2022_JIS-X0201_1989
PC936_SimplifiedChinese	ISO2022_Korean
PC949_Korean	ISO2022_SimplifiedChinese
PC950_TraditionalChinese	ISO2022_TraditionalChinese
DECMultinational	EUC_Japanese
UCS2	EUC_SimplifiedChinese
Windows1250	EUC_Korean
Windows1251	EUC_TraditionalChinese
Windows1252	GB2312_SimplifiedChinese
Windows1253	GB18030_SimplifiedChinese
Windows1254	UTF7
Windows1255	UTF8

既定値は「PC437_English」です。

サンプル構文は次のとおりです。

```
HostCharacterSet EUC_Japanese
```


InsertMode

入力を挿入モードにするか置換モードにするかを指定します。「no」に設定すると、入力によりカーソル位置の既存の文字が置換されます。「yes」に設定すると、新しい文字がカーソル位置に挿入され、既存の文字は右に移動します。

指定可能な値は「yes」または「no」です。

既定値は「no」です。

サンプル構文は次のとおりです。

```
InsertMode yes
```

InverseVideo

端末ウィンドウが反転表示を使用するかどうかを指定します。「yes」に設定すると、すべての画面属性の前景色および背景色が反転されます。

指定可能な値は「yes」または「no」です。

既定値は「no」です。

サンプル構文は次のとおりです。

```
InverseVideo yes
```

KeyBoardActionMode

キーボードを使用可能にするかどうかを指定します。「yes」に設定すると、キーボードがロックされ使用できません。

指定可能な値は「yes」または「no」です。

既定値は「no」です。

サンプル構文は次のとおりです。

```
KeyBoardActionMode yes
```

MarginBell

マージンベルを鳴らすかどうかを指定します。「yes」に設定すると、カーソルが右余白から 8 文字の時にベルが鳴ります。この設定を「no」に設定すると、マージンベルが鳴りません。

指定可能な値は「yes」または「no」です。

既定値は「yes」です。

サンプル構文は次のとおりです。

```
MarginBell no
```

NewLine

クライアントを行送りモードにするか改行モードにするかを指定します。「no」(行送りモード)に設定すると、[Enter] キーを押すと復帰のみが送信されます。行送り、改ページ、垂直タブを受信すると、カーソルが現在の桁の 1 つ下の行に移動します。「yes」(改行モード)に設定すると、[Enter] キーを押した時に復帰と行送りの両方が送信されます。改ページ、垂直タブを受信すると、カーソルが次の行の先頭桁に移動します。

指定可能な値は「yes」または「no」です。

既定値は「no」です。

サンプル構文は次のとおりです。

```
NewLine yes
```

NRCSet

対応する文字列値のいずれかを使用して、異なる国別文字セットを指定します。これを有効にするには、キーワード `UseNRC` も「yes」に設定する必要があります。

指定可能な文字列値は次のとおりです。

British	Norwegian
Finnish	Portuguese
French	EuropeanSpanish
CanadianFrench	Swedish
German	SwissGerman
Italian	

既定値は「ASCII」です。

サンプル構文は次のとおりです。

```
UseNRC yes
NRCSet British
```

NumericKeyPadMode

クライアントが数字キーパッドのキーをどのように処理するかを指定します。

「yes」に設定すると、キーパッドは数字モードに設定されます。つまり、キーパッドのキーを押すと数値が送信されます。「no」に設定すると、キーパッドはアプリケーションモードに設定されます。つまり、キーパッドのキーはホーム、上、右などのアプリケーションエスケープシーケンスを送信します。

指定可能な値は「yes」または「no」です。

既定値は「no」です。

サンプル構文は次のとおりです。

```
NumericKeyPadMode no
```

OriginMode

カーソルのホーム位置を指定します。「no」に設定すると、カーソルのホーム位置は端末ウィンドウの左上隅になります。「yes」に設定すると、カーソルのホーム位置は端末ウィンドウの余白設定に関連して決まります。

指定可能な値は「yes」または「no」です。

既定値は「no」です。

サンプル構文は次のとおりです。

```
OriginMode yes
```

SevenBitControls

8 ビット C1 制御コードを送信するかどうかを指定します。「yes」に設定すると、8 ビット C1 制御コードに相当する 7 ビット制御コードが送信されます。「no」に設定すると、8 ビット C1 制御コードが送信されます。

注意: ssh コマンドラインクライアントの `HostCharacterSet` の既定値は `PC437_English` です。C1 制御を送信したい場合、`HostCharacterSet` を `DECMultinational` または `ISOLatin` 文字セットのいずれかに設定する必要があります。

指定可能な値は「yes」または「no」です。

既定値は「yes」です。

サンプル構文は次のとおりです。

```
SevenBitControls no
```

TerminalModel

クライアントがエミュレートする端末の種類を指定します。

指定可能な文字列値は「vt52」、「vt102」、「vt220」です。

既定値は「vt220」です。

サンプル構文は次のとおりです。

```
TerminalModel vt102
```

UseNRC

これを「yes」に設定すると、キーワード `NRCSet` を使用して国別文字セットを指定できます。

指定可能な値は「yes」または「no」です。

既定値は「no」です。

サンプル構文は次のとおりです。

```
UseNRC yes
NRCSet British
```

UseANSIColor

「yes」に設定すると、ANSI 色エスケープシーケンスに対応します。

指定可能な値は「yes」または「no」です。

既定値は「yes」です。

サンプル構文は次のとおりです。

```
UseANSIColor no
```

WarningBell

警告ベルを鳴らすかどうかを指定します。「yes」に設定すると、ホストからベル文字 (ASCII 7) を受信したり、キーボードから入力されたりすると、ベルが鳴ります。この設定を「no」に設定すると、警告ベルは鳴りません。

指定可能な値は「yes」または「no」です。

既定値は「yes」です。

サンプル構文は次のとおりです。

```
WarningBell no
```

DOD PKI 情報

このセクションでは、Reflection をインストール、構成、使用して DOD (Department of Defense) 環境またはその他の PKI (Public Key Infrastructure) 環境内で操作する方法について解説します。PKI 構成は、Secure Shell 接続および SSL/TLS 接続の両方に影響します。

DOD PKI モードでの Reflection の実行

既定では、Reflection アプリケーションにより DOD PKI 要件を満たさない構成が一部許可されます。管理者は、Reflection グループポリシーを使用して DOD PKI 要件を満たすようにすべての Reflection セッションを構成できます。

DOD PKI モードを構成するには

- 1 次のいずれかの方法で、グループポリシーエディタを実行します。
 - コマンドラインで `Gpedit.msc` と入力します。
 - [Active Directory のユーザとコンピュータコンソール] で所属ユニットのプロパティを開き、[グループポリシー] タブをクリックして新規ポリシーオブジェクトを編集または作成します。
- 2 Reflection テンプレート (ReflectionPolicy.adm) がインストールされていない場合はインストールします。
- 3 [ローカルコンピュータポリシー] > [ユーザの構成] > [管理用テンプレート] > [Reflection の設定] で、[DoD PKI 以外のモードを許可する] を無効にします。

DOD PKI モードの構成には、以下の影響があります。

- CRL 『[135](#) ページ』を確認したり、OCSP レスポンドを使用するように Reflection を構成する必要があります。DOD PKI モードでは、いずれの確認形式も使用しないオプションは無効です (SSH 接続の場合、[Reflection Secure Shell の設定] ダイアログボックスの [PKI] タブを使用して、証明書の取り消しを構成します。SSL/TLS 接続の場合、[PKI の構成] ダイアログボックスを使用して構成します)。
- Reflection では、FIPS 承認の暗号化アルゴリズムを実行します。SSH 接続の場合、[Reflection Secure Shell の設定] ダイアログボックスの [暗号化] タブでは FIPS 承認オプションのみ使用できます。SSL/TLS 接続の場合、[暗号化レベル] を 40 ビットまたは 56 ビットに設定できません。
- 接続を確立するためには、証明書のホスト名が Reflection 接続に指定したホスト名と完全に一致している必要があります。つまり、[証明書のホスト名と対象ホスト名が一致するかどうかを確認する] が自動的にオンになり、変更することはできません (SSH 接続の場合、[Reflection Secure Shell の設定] ダイアログボックスの [PKI] 『[48](#) ページ』 タブを使用して、この設定を構成します。SSL/TLS 接続の場合、[PKI の構成] ダイアログボックスを使用して構成します)。

トラストポイントのインストールおよび削除

トラストポイントは、信頼チェーン内の任意の CA 『[138](#) ページ』 証明書です。

トラストポイントを Reflection 証明書格納場所に追加するには

- 1 [Reflection 証明書マネージャ] を開きます。 『[49](#) ページ』
- 2 [信頼された認証局] タブをクリックします。
- 3 [インポート] をクリックしてから、証明書 (通常、*.cer または *.crt) を検索して指定します。

トラストポイントを Reflection 証明書格納場所から削除するには

- 1 [Reflection 証明書マネージャ] を開きます。 『[49](#) ページ』
- 2 [信頼された認証局] タブをクリックします。

- 3 証明書を選択して、**[削除]** をクリックします。

注意

- 中間 CA トラストポイントは、LDAP サーバまたは HTTP サーバから取得できます。このサーバは、証明書の AIA (Authority Information Access) 拡張に定義されている明示的な URI、または [Reflection 証明書マネージャ] の [LDAP] タブに構成されている LDAP サーバ情報を使用して特定できます。これらの証明書は、<ユーザ名>\My Documents\Attachmate\Reflection\.pki または \All users\Application data\Attachmate\Reflection にある cert_cache ファイルに保存されます。
 - Reflection が DOD PKI モードで実行中の場合、[Reflection 証明書マネージャ] に追加したルート証明書のみが使用されます。Windows の証明書格納場所に存在する可能性がある DOD PKI 以外の証明書を削除する必要はありません。
-

証明書取り消しの確認の構成

Reflection では、証明書取り消しの確認の既定値は現在のシステム設定に基づいて決まります。システムが CRL の確認を行うように構成されている場合は、既定で Reflection セッションにおいて CRL 『135 ページ』を使用して証明書取り消しが確認されます。OCSP レスポンダを使用するように Reflection を構成することもできます。

Reflection では、CRL 確認を無効にする設定にも対応しています。この設定をテストに使用できますが、Reflection が DOD PKI モードで実行中の場合にはこのオプションを使用できません。

警告: CRL 確認を無効にすると、セキュリティ上のリスクが高まります。このオプションはテストにのみ使用します。

中間証明書または CRL を取得する 1 台または複数の LDAP サーバを定義できます。

LDAP サーバを定義するには

- 1 [Reflection 証明書マネージャ] を開きます。 『49 ページ』
- 2 [LDAP] タブをクリックします。
- 3 **[追加]** をクリックし、次の URL 形式を使用してサーバを指定します。

```
ldap://hostname:portnumber
```

例えば、次のように入力します。

```
ldap://ldapservers.myhost.com:389
```

OCSP を構成するには

- 1 証明書の取り消し情報を要求する 1 台または複数の OCSP サーバを定義できます。
- 2 **[証明書失効の確認]** を **[OCSP を使用する]** に設定します (SSH 接続の場合、[Reflection Secure Shell の設定] ダイアログボックスの [PKI] タブを使用します。SSL/TLS 接続の場合、[PKI の構成] ダイアログボックスを使用します)。

証明書に必要な OCSP レスポンダの URL は、証明書の AIA 拡張に指定されます。この情報が証明書で提供されない場合、次の手順を使用して OCSP レスポンダ情報を構成できます。

- 3 [Reflection 証明書マネージャ] を開きます。 『49 ページ』

- 4 [OCSP] タブをクリックします。
- 5 [追加] をクリックし、次の URL 形式を使用してサーバを指定します。

URL:portnumber

例えば、次のように入力します。

https://ocspmachine.host.com:389

DOD PKI サービスの URI の使用

Reflection では、CRL 『[135](#) ページ』の自動更新および取得に URI を使用できます。RFC3280 のセクション 4.2.1.14 に定義されています。

CRL 確認が有効の場合、Reflection では以下のように証明書の取り消しを確認します。

1. `crl_cache` ファイルで有効な取り消し情報を確認します。見つからない場合は、手順 2 に進みます。
2. 証明書の CDP 拡張で HTTP URI または LDAP URI を確認し、指定した順番 (最初に HTTP、次に LDAP) で問い合わせます。取り消し対象の証明書が見つかった場合は、接続を切断します。証明書が見つからない場合は、手順 3 に進みます。
3. 1 台または複数の LDAP サーバが [Reflection 証明書マネージャ] の [LDAP] タブに指定されている場合、証明書の発行者の拡張子に示されている CA の識別名をまとめて、CRL ファイルに問い合わせます。いずれの CRL にも取り消し対象の証明書が見つからない場合は、次の検証手順に進みます。

期限切れの CRL の更新は自動的に処理されるため、管理者の介入または構成の必要はありません。

OCSP 確認が有効の場合、Reflection ではすべての使用可能な OCSP レスポンドを必ず確認します。これは、証明書が取り消されたことをいずれかのレスポンドが把握している場合に、接続が失敗することを確認するためです。接続を確立するためには、少なくとも 1 つの OCSP レスポンドが使用可能であり、認証ステータスに対して値「good」を返す必要があります。Reflection では、以下のように確認を実行します。

1. 証明書の AIA 拡張で 1 つまたは複数の OCSP レスポンドを確認し、各レスポンドに問い合わせます。いずれかのレスポンドからの証明書のステータスが「revoked」に戻った場合、接続を切断します。
2. [Reflection 証明書マネージャ] の [OCSP] タブを使用して指定した 1 つまたは複数のユーザ構成 OCSP レスポンドを確認し、各レスポンドに問い合わせます。いずれかのレスポンドからの証明書のステータスが「revoked」に戻った場合、接続を切断します。
3. すべてのレスポンドが「unknown」を返した場合、接続を切断します。少なくとも 1 つの OCSP レスポンドから「good」応答が返された場合、次の検証手順に進みます。

URI を使用した中間証明書の取得

RFC3280 のセクション 4.2.1.14 に定義されているように、Reflection では以下のように URI を使用して中間 CA 『[138](#) ページ』証明書を取得できます。

1. `cert_cache` ファイルで必要な中間証明書を確認します。見つからない場合は、手順 2 に進みます。
2. HTTP URI または LDAP URI のいずれかが証明書の AIA (Authority Information Access) 拡張に定義されている場合、中間 CA 証明書の取得にこれらの使用を試行します (最初に HTTP、次に LDAP)。

3. 前の試行に失敗した場合、発行している証明書の件名から識別名をまとめて、CACertificate 属性の内容に定義された LDAP サーバに問い合わせます。

Reflection では証明書のセキュリティポリシ拡張を実施するため、セキュリティポリシ構成は不要です。

証明書と秘密鍵の構成と保護

証明書を使用してクライアント認証を構成するには

- 1 [Reflection 証明書マネージャ] を開きます。『49 ページ』
- 2 [個人] タブで [インポート] をクリックしてから、証明書 (通常、*.pfx または *.p12) を検索して指定します。この鍵を使用することに要求されるパスフレーズを作成する画面が表示されます。システムでこの鍵を保護するのに役立つため、パスフレーズの入力が推奨されます。
- 3 Secure Shell 接続の場合、[Reflection Secure Shell の設定] ダイアログボックスを開き、[ユーザ鍵] タブをクリックして、現在指定しているホストへのクライアント認証に使用したい証明書を選択します (このステップは、SSL/TLS 接続には必要ありません)。

秘密鍵の保護

クライアントの秘密鍵が盗まれた場合、悪意のあるユーザがそのユーザにアクセス可能な任意のサーバのファイルにアクセスできます。このリスクを最小限にするには、各クライアントユーザがパスフレーズを使用して自分の秘密鍵を必ず保護する必要があります。これによって、パスフレーズを知っている人だけがその鍵で認証できるようになります。ユーザは、組織のセキュリティポリシのパスワード仕様に従ってパスフレーズを作成し、保護する必要があります。

鍵が改ざんされた場合の操作

秘密鍵が不正な人物によって利用可能になった場合、または鍵にアクセスする人物の操作を信用しない理由がある場合、秘密鍵が改ざんされたと見なします。

クライアントの鍵が改ざんされた場合、クライアントの証明書を取り消します。

改ざんされた鍵を置換するには

- 1 新しい秘密鍵と証明書を生成して、[Reflection 証明書マネージャ] に鍵をインポートします。
- 2 識別情報が変更された場合、サーバでこのクライアントの割り当てファイルの行を更新します。

クライアントコンピュータから改ざんされた鍵を削除するには

- 1 [Reflection 証明書マネージャ] の [個人] 『49 ページ』 タブから鍵を削除します。これによって、identity_store.p12 ファイルからこの鍵が削除されます。
- 2 古い鍵と証明書を含む元のファイル (*.pfx or *.p12) がまだクライアントコンピュータにある場合は、DOD 承認ファイル削除ユーティリティを使用してこのファイルを削除します。

用語集

C

CRL (Certificate Revocation List)

認証局によって失効された、電子署名された証明書の一覧。CRL で識別された証明書はすでに有効ではありません。

G

GSSAPI (Generic Security Services アプリケーションプログラムインタフェース)

プログラムにセキュリティサービスへのアクセスを提供するアプリケーションプログラミングインタフェースです。

K

Kerberos

信頼されたサードパーティを使用して TCP/IP ネットワーク上で安全な通信を実現するプロトコル。このプロトコルは、プレーンテキストのパスワードではなく、暗号化されたチケットを使用してより安全にネットワーク認証を行います。

M

MAC (メッセージ認証コード)

データが送信中に変更されていないことの確認に使用されます。MAC は、データと共有秘密鍵が含まれる任意の長さの packets を使用して作成されたハッシュです。送信側と受信側は、共有鍵および合意したアルゴリズムを使用して、転送されたデータの各パケットの MAC を独自に計算します。メッセージが送信中に変更されている場合、別のハッシュ値になり、パケットは拒否されます。

P

PKCS

PKCS (Public Key Cryptography Standards: 公開鍵暗号標準) の略。RSA 研究所によって考案および公布された、公開鍵暗号の実装間の互換性を実現する一連の標準。

各 PKCS 標準では、特定の暗号化用途の仕様が定められています。以下に例を示します。

- PKCS#7 は、メッセージの署名/暗号化、および PKCS#10 メッセージへの応答としての証明書の配布に使用できます。
- PKCS#10 は証明書要求の構文です。
- PKCS#11 暗号化ハードウェアトークンで使われるプログラミングインタフェースです。
- PKCS#12 は、証明書とそれに関連する秘密鍵の格納および送信に使われる個人情報交換構文を定義します。この形式のファイルでは、通常、*.pfx または *.p12 拡張子が使用されます。

R

Reflection ssh フォルダ

Reflection は、個々のユーザの Secure Shell 情報を Windows の個人のドキュメントフォルダの以下の場所に格納します。

Windows XP、Windows Server 2003 の場合:

```
\Documents and Settings\<<ユーザ名>\My Documents\Attachmate\Reflection\.ssh
```

Windows Vista、Windows Server 2008 の場合:

```
\Users\<<ユーザ名>\Documents\Attachmate\Reflection\.ssh
```

類似のファイルは、UNIX システムでは \$HOME ディレクトリに格納されます。

Reflection アプリケーションデータフォルダ

Reflection では、すべてのユーザに使用可能な Secure Shell 情報を以下の場所に保存します。

Windows XP、Windows Server 2003 の場合:

```
\Documents and Settings\all users\Application Data\Attachmate\Reflection
```

Windows Vista、Windows Server 2008 の場合:

```
\ProgramData\Attachmate\Reflection
```

S

Secure Shell

リモートコンピュータへのログインとコマンドの実行を安全に行うためのプロトコル。これは、Telnet、FTP、rlogin、あるいは rsh の代わりとなる安全な方法です。Secure Shell 接続では、ホスト (サーバ) とユーザ (クライアント) の両方の認証が必要です。また、ホスト間の通信はすべて暗号化された通信チャネルを介して行う必要があります。また、Secure Shell では X11 セッションまたは指定の TCP/IP ポートを、安全なトンネルを介して転送することもできます。

U

UTC (Universal Time, Coordinated)

A high-precision time standard. When describing time zones, UTC refers to the time kept on the Greenwich meridian (longitude zero), also known as *Greenwich Mean Time*. UTC times are generally given in terms of a 24-hour clock.

W

Windows ユーザプロファイルフォルダ

ユーザプロファイルフォルダは、Windows システム管理者が構成できます。既定値は以下のとおりです。

- Windows XP、Windows Server 2003 の場合

```
\Documents and Settings\<<ユーザ名>\
```
- Windows Vista、Windows Server 2008 の場合

```
\Users\<<ユーザ名>\
```

Windows 共通アプリケーションデータフォルダ

注意: アプリケーションデータフォルダは、既定では表示されません。

既定の場所は以下のとおりです。

- Windows XP、Windows Server 2003 の場合:
 \Documents and Settings\all users\Application Data\
- Windows Vista、Windows Server 2008 の場合:
 \ProgramData\

データの整合性

データが元のソースから変更されていない保証です。データの整合性を維持する方法は、データが誤って、または悪意を持って変更、改ざん、破壊されていないことを保証するように設計されています。

デジタル署名

送信されたメッセージの信頼性と整合性の確認に使用されます。通常、送信者は公開/秘密鍵のペアのうち秘密鍵を保有し、受信者は公開鍵を保有します。署名を作成するには、送信者はメッセージからハッシュを計算し、この値を自らの秘密鍵で暗号化します。受信者は、送信者の公開鍵を使用して署名を復号化し、受信したメッセージのハッシュを独自に計算します。復号化した値と計算した値が一致した場合、受信者は、送信者が秘密鍵の保有者であり、メッセージが送信中に改ざんされていないことを信頼します。

パスフレーズ

パスフレーズはパスワードに類似していますが、一連の語句、句読点、数字、空白、任意の文字列を組み合わせたフレーズを使用できる点が違います。パスフレーズは、秘密鍵や鍵エージェントなどの保護されたオブジェクトへのアクセスを制限して、セキュリティを向上させます。

ハッシュ

メッセージダイジェストとも呼ばれます。ハッシュ (またはハッシュ値) は可変長のデジタルデータから生成される固定長の数値です。ハッシュは、元のデータより大幅にサイズが小さく、同じハッシュ値がほかのデータで統計的に生成されることのない方法で生成されます。

ポート転送

安全でないトラフィックを安全な SSH トンネルを介してリダイレクトする方法です。ポート転送には、ローカルとリモートの 2 種類があります。ローカルポート転送 (発信ポート転送) は、指定されたローカルポートから送信された発信データを、安全なチャンネルを介して、指定されたリモートポートに送信します。クライアントアプリケーションとサーバとの間でデータを安全に交換するには、関連サーバを実行するコンピュータにクライアントを直接接続するのではなく、リダイレクトされるポートに接続するように構成します。リモートポート転送 (受信ポート転送) は、指定されたリモートポートからの受信データを、安全なチャンネルを介して、指定されたローカルポートに送信します。

漢字

暗号

暗号とは暗号化アルゴリズムのことです。選択した暗号によって、Secure Shell 接続の確立完了後に送信されるデータの暗号化に使用される数学アルゴリズムが決定されます。

暗号化

暗号化とは、暗号すなわち秘密のコードを使用してデータを加工し、許可されたユーザ以外には解読できないようにすることです。暗号化されていないデータに比べ、暗号化されたデータははるかに安全です。

公開鍵と秘密鍵

公開鍵と秘密鍵は、データの暗号化または解読に使用される暗号鍵のペアです。公開鍵で暗号化されたデータは、秘密鍵を使用した場合のみ解読できます。また、秘密鍵で暗号化されたデータは、公開鍵を使用した場合のみ解読できます。

正規表現

正規表現は、1 つ以上の一致する文字列を記述する文字列です。よく *regex* と省略されます。正規表現内では、一部の文字は事前に定義された意味を持ち、何が一致と見なされるかを決定します。たとえば、正規表現「`t.*t`」は、文字 *t* で始まり、かつ終わるすべての単語に一致します。一方、正規表現「`text`」はそれ自体のみに一致します。

電子証明書

PKI (Public Key Infrastructure) の核となる構成要素です。電子証明書は認証局 (CA) から発行され、証明書の情報が有効であることを保証します。各証明書には、証明書の所有者に関する情報、証明書の所有者の公開鍵のコピー (メッセージおよび電子署名の暗号化と解読に使用)、電子署名 (証明書の内容に基づいて認証局が生成) が含まれています。受信者はこの電子署名を使用して、証明書が不正に変更されておらず、信頼できることを確認します。

認証

通信相手の身元を確実に確認する処理。身元の確認は、パスワードなどの既知の情報、または秘密鍵やトークンなど所有しているもの、指紋などの固有の情報を使用して行います。

認証局 (CA)

信頼されている組織内で、電子証明書を発行するサーバ。CA は、新規の証明書の発行を管理し、認証に対して有効でなくなった証明書を取り消します。さらに、CA は、信頼チェーンを形成する 1 つ以上の中間 CA に証明書の発行権限を委任することもあります。最高レベルの CA 証明書は、信頼されたルートと呼ばれます。

索引

C

config ファイル - 19
CRL の確認 - 46

D

DOD PKI 情報 - 130
DOD PKI モード - 130

F

FIPS モード
概要 - 21
FTP クライアント
FTP クライアントの使用 - 14
FTP 設定のインストール - 87
FTP 転送の構成 - 64
設定ファイル - 19

G

GSSAPI
[GSSAPI] タブ - 56
概要 - 55
構成方法 - 55
チケット転送 - 55

H

HMAC - 22
HTTP プロキシ - 73

K

Kerberos (Secure Shell 接続)
[GSSAPI] タブ - 56
概要 - 55
構成方法 - 55
チケット転送 - 55

O

OCSP
OCSP レスポンドの設定 - 52
証明書取り消しの設定 - 46

P

PKI
[Reflection 証明書マネージャ] - 49
DOD PKI モード - 130
Windows の証明書格納場所の無効化 - 45
概要 - 43
クライアント認証 - 44
サーバ認証 - 45
証明書格納場所 - 43
証明書の取り消し確認 - 46

R

[Reflection Secure Shell の設定] ダイアログボックス
[GSSAPI] タブ - 56
[PKI] タブ - 48
[暗号化] タブ - 22
[全般] タブ - 16
[トンネリング] タブ - 65
[ホスト鍵] タブ - 40
[ホストデータ] タブ - 71
[マルチホップ] タブ - 68
[ユーザ鍵] タブ - 35

S

scp コマンドラインユーティリティ - 104
Secure Shell
機能 - 7
基本操作 - 13
[SFTP]
FTP クライアントの使用 - 14
設定ファイル - 19
SOCKS プロキシ - 73
SSH
基本操作 - 13
構成ファイル - 19
SSH 構成セクション - 112
ssh コマンドラインユーティリティ - 90
ssh-keygen コマンドラインユーティリティ - 95

あ

暗号
Secure Shell セッション - 22
暗号化
[暗号化] タブ ([Reflection Secure Shell の設定] ダイアログボックス) - 22
対応している暗号化の規格 - 21
インストール
カスタム - 83
管理者用 - 79
基本 - 10
エクスポート
ユーザ鍵 - 34

か

鍵 (公開鍵認証)
ホストへのユーザ鍵のアップロード - 33
ユーザ鍵の管理 - 32
既知のホストファイル - 39
公開鍵認証
構成 - 32
ホストへのユーザ鍵のアップロード - 33
ユーザ鍵の管理 - 32
構成ファイル
Secure Shell (概要) - 19
SSH 構成セクション - 112
キーワードのリファレンス (Secure Shell の設定) - 114
接続の再利用 - 28

さ

証明書

- [Reflection 証明書マネージャ] - 49
- LDAP を使用した配布 - 47
- Windows の証明書格納場所の無効化 - 45
- 概要 - 43
- クライアント認証 - 44
- サーバ認証 - 45
- 証明書格納場所 - 43
- 証明書の取り消し確認 - 46

接続の再利用 - 28

設定

- カスタム - 83
- 管理者用 - 79
- 基本 - 10

設定ファイル

- Secure Shell 構成ファイル - 19
- インストール - 84
- クライアント設定ファイル - 19

た

端末エミュレーションの設定 - 125

チケット転送 - 55

トンネリング

- [トンネリング] タブ - 65
- FTP 通信の転送 - 64
- TCP 通信の転送 - 64
- マルチホップ - 67
- リモートポート転送 - 62
- ローカルポート転送 - 60

な

認証 (Secure Shell セッション)

- 概要 - 25
- サーバ - 25
- サーバ (証明書) - 26
- 接続の再利用 - 28
- ユーザ - 27

は

パスフレーズ

- ユーザ鍵の変更 - 34

ファイル転送

- FTP クライアントの使用 - 14
- scp コマンドラインユーティリティ - 104

ポート

- Secure Shell の構成 - 16

ポート転送

- [トンネリング] タブ - 65
- FTP 通信の転送 - 64
- TCP 通信の転送 - 64
- マルチホップ - 67
- リモートポート転送 - 62
- ローカルポート転送 - 60

ホスト鍵

- [ホスト鍵] タブ ([Reflection Secure Shell の設定] ダイアログボックス) - 40
- 鍵または証明書の優先 - 39
- 管理 - 38
- 既知のホストファイル - 39
- ホスト鍵の確認の構成 - 38

ホスト変数およびコマンド

- 環境変数 - 71
- ホストコマンドの実行 - 71

ま

問題解決

- 接続のトラブルシュート - 75
- ログファイルの使用 - 76

や

ユーザ鍵

- [ユーザ鍵] タブ ([Reflection Secure Shell の設定] ダイアログボックス) - 35
- [ユーザ鍵の生成] ダイアログボックス - 36
- エクスポート - 34
- 管理 - 32

ら

リモートポート転送 - 62

ローカルポート転送 - 60