Orchestrating a brighter world

NEC

SecureWare/Credential Lifecycle Manager
コマンドリファレンス

2019 年 10 月

日本電気株式会社

はしがき

本書は、SecureWare/Credential Lifecycle Manager(以下、CLM と称します)と、そのクライアントである SecureWare/Credential Lifecycle Agent (以下、CLA と称します)が提供するコマンドについて説明したものです。

本書の構成は以下のとおりです。

章	タイトル	内容
1	提供コマンド一覧	CLM が提供するコマンドの概要説明
2	WebAPI 実行コマンド	WebAPI 実行コマンドについての説明
3	ID 鍵コマンド	ID 鍵コマンドについての説明
4	ID 鍵コマンドの機能拡張	ID 鍵コマンドが提供する拡張機能についての説明
5	共通鍵一括取得コマンド	共通鍵一括取得コマンドについての説明
6	終了コード・エラーコード一覧	ID 鍵コマンドの終了コード､エラーコードについての説明

2019 年 10 月　第四版

目次

1　提供コマンド一覧

CLM、CLA で提供するコマンドは、下表の通りです。

コマンドの詳細は、2 章以降をご覧ください。

項番	コマンド	コマンド名	インストールディレクトリ	説明
1	WebAPI 実行コマンド	[Linux 版] swclmclient [Windows 版] swclmclient.exe	[Linux 版] /opt/nec/pf/swcagent/bin/ [Windows 版] %ProgramFiles%¥NEC¥swcagent	パラメータを指定することにより、各 WebAPI を実行するコマンド。 なお、本コマンドを実行するために必要な環境変数を含めたコマンドを ID 鍵コマンドとして用意していますので、本コマンドを直接実行する必要はありません。
2	ID 鍵コマンド	[Linux 版] SWCLMCLIENT [Windows 版] SWCLMCLIENT.cmd	[Linux 版] /opt/nec/pf/swcagent/bin/ [Windows 版] %ProgramFiles%¥NEC¥swcagent	WebAPI 実行コマンドを機能拡張し、CLM の提供機能をより便利に利用可能にするコマンド。
3	共通鍵一括取得コマンド	[Linux 版] SWCLMKEYGET [Windows 版] SWCLMKEYGET.cmd	[Linux 版] /opt/nec/pf/swcagent/bin/ [Windows 版] %ProgramFiles%¥NEC¥swcagent	デバイスが保有する共通鍵を CLM から一括取得するコマンド。

2 WebAPI 実行コマンド

WebAPI 実行コマンドは、CLM が提供する WebAPI を実行するコマンドです。

本コマンドを実行することで、CLM が提供する WebAPI の機能をコマンドラインから利用し、ID 鍵の管理を行うことができます。

また、本コマンドは、発行・取得・更新した証明書・共通鍵のファイル保存や証明書・共通鍵の改ざん検知など WebAPI 実行の他に付加機能を提供します。

前提

- Linux 版
  - コマンドは、root ユーザもしくは root グループに所属するユーザで実行してください。
  - コマンド実行環境では、事前に以下の環境変数を設定しておく必要があります。環境変数を設定しない場合、コマンド実行時にエラーが発生し、コマンド実行に失敗しますのでご注意ください。

    LD_LIBRARY_PATH=/opt/nec/pf/swcagent/lib:$LD_LIBRARY_PATH

    SWSDKV50_LIBPATH=/opt/nec/pf/swcagent/lib
- Windows 版
  - コマンドは、Administrators 権限を保有したユーザで実行してください。

3　ID 鍵コマンドの基本機能

ID 鍵コマンドは、WebAPI 実行コマンド(swclmclient)の呼び出しと環境変数の設定をラップすることでユーザが WebAPI 実行コマンドを使いやすくすることに加え、WebAPI 実行コマンドを機能拡張することでより便利に CLM の提供機能を使用できるようにしたコマンドです。以下のコマンドをご使用頂くことで WebAPI 実行コマンドを直接実行する必要はありません。拡張機能の詳細は、4 章をご覧ください。

[Linux]
/opt/nec/pf/swcagent/bin/SWCLMCLIENT
[Windows]
%ProgramFiles%¥NEC¥swcagent¥SWCLMCLIENT.cmd

前提

- Linux 版
  - コマンドは、root ユーザもしくは root グループに所属するユーザで実行してください。
- Windows 版
  - コマンドは、Administrators 権限を保有したユーザで実行してください。

基本提供機能

本コマンドの提供している基本機能は以下の通りです。

- ID・パスワード発行
  ID・パスワード発行 API を実行し、エッジ－クラウド間の通信を安全に行うための ID・パスワードを発行します。パスワードの複雑さはサーバ側の設定に従います。
- ID・パスワード取得
  ID・パスワード取得 API を実行し、 ID・パスワード発行 API で発行した ID・パスワードを取得します。
- ID・パスワード照合
  ID・パスワード照合 API を実行し、WebAPI 実行コマンドが所有している ID・パスワードと、CLM に登録されている ID・パスワードが一致するかを照合(認証)します。
- ID・パスワード削除
  ID・パスワード削除 API を実行し、 ID・パスワード発行 API で発行した ID・パスワードを削除します。

- 証明書発行
  証明書発行 API を実行し、CA 証明書、公開鍵証明書を発行・取得します。また、発行・取得した証明書から証明書改ざん検知用のハッシュ値を生成し、ファイルに保存します。

- 証明書取得

  証明書取得 API を実行し、CA 証明書、公開鍵証明書を取得します。また、取得した証明書から証明書改ざん検知用のハッシュ値を生成し、ファイルに保存します。
- 証明書更新

  証明書更新 API を実行し、公開鍵証明書を更新します。更新した証明書から証明書改ざん検知用のハッシュ値を生成し、ファイルに保存します。
- 証明書失効

  証明書失効 API を実行し、公開鍵証明書を失効します。

- 共通鍵発行

  共通鍵発行 API を実行し、共通鍵を発行します。また、発行した共通鍵から共通鍵改ざん検知用のハッシュ値を生成し、ファイルに保存します。
- 共通鍵取得

  共通鍵取得 API を実行し、共通鍵を取得します。また、取得した共通鍵から共通鍵改ざん検知用のハッシュ値を生成し、ファイルに保存します。
- 共通鍵更新

  共通鍵更新 API を実行し、共通鍵を更新します。更新した共通鍵から共通鍵改ざん検知用のハッシュ値を生成し、ファイルに保存します。
- 共通鍵削除

  共通鍵削除 API を実行し、共通鍵を削除します。
- 共通鍵照合

  共通鍵照合 API を実行し、指定共通鍵が CLM に登録されている共通鍵と一致するかを照合(認証)します。

実行形式

実行時に以下のオプション、および、オプション値を指定します。指定できるオプションは、機能によって異なります。

- コマンドを利用する上での留意事項
  - 証明書発行・取得・更新・失効と、共通鍵発行・取得・更新・削除・照合コマンドを実行する場合、事前に ID・パスワード照合コマンド(3 章 ID・パスワード照合(SWCLMCLIENT idverify)をご覧ください)を実行し、CLM と ID・パスワードによる認証を行っておく必要があります。
  - ID・パスワード取得、証明書取得・更新、共通鍵取得・更新コマンドでは、有効期限内の ID・パスワード、証明書、共通鍵を操作することが可能です。有効期限が切れた ID・パスワード、証明書、共通鍵は、更新できません。
  - コマンドで使用しないオプションを指定した場合、そのオプションは無視されます。
    例) ID・パスワード発行コマンドで-mode オプションを指定すると、-mode オプションは無視してコマンド実行されます。
  - コマンドで使用しない拡張情報を指定した場合、その拡張情報は無視されます。
    例) ID・パスワード発行コマンドで-ext-data オプションに certsubject="/CN=A"や desc="ABC"と指定すると、certsubject="/CN=A"と desc="ABC"は無視してコマンド実行されます。
  - コマンドのオプション、拡張情報に記号を含む文字列を指定する場合、記号はエスケープ文字によりエスケープした状態で指定する必要があります。エスケープ文字およびエスケープの方法については、Linux 版の場合は使用するシェルの仕様に、Windows 版の場合はコマンドプロンプトの仕様に従います。
  - Windows 版のコマンドを実行する際は、必ず拡張子(.cmd)まで指定して実行する必要があります。

● ID・パスワード発行

> SWCLMCLIENT idcreate	-host <hostname> [-port <port >] [-p-host <hostname >] [-p-port <port >] [-timeout <timeout>] -ext-data <extension data>

オプション	必須/任意	説明
-host	必須	CLM のホスト名または IP アドレスを指定します。
-port	任意	CLM の待ち受けポート番号を指定します。既定値は「8443」です。
-p-host	任意	プロキシサーバの IP アドレスを指定します。 コマンドから CLM への通信時に Web プロキシサーバを経由しなければ通信できないネットワーク構成である場合は、本項目を設定してください。
-p-port	任意	プロキシサーバのポート番号を指定します。 コマンドから CLM への通信時に Web プロキシサーバを経由しなければ通信できないネットワーク構成である場合は、本項目を設定してください。
-timeout	任意	タイムアウト時間を指定します。単位は秒です。既定値は 60 秒です。指定可能な値は 1～0x7FFFFFFF です。
-ext-data	必須	拡張情報を指定します。 key=value の形式で指定してください。 value は、クォートで囲む必要があります。また、複数指定する場合はカンマで区切る必要があります。 指定する拡張情報は、後述の「拡張情報」を参照してください。

拡張情報

ID・パスワード発行機能では、-ext-data に以下の項目を指定することができます。

項目	必須/任意	説明
tenantid	必須	テナント ID を指定します。「TenantA」固定です。
alias	必須	ID・パスワードに付与する Alias を指定します。 最大文字列長は 64byte です。 使用可能文字種は、英数記号(但し、「;」「|」「&」「`」「(」「)」「$」「<」

		「>」「*」「?」「{」「}」「[」「]」「!」を除く)です。

実行例

```
#SWCLMCLIENT idcreate -host "192.168.0.1" -port "8443" -ext-data tenantid="TenantA", alias="iotgatewaytest"
```

● ID・パスワード取得

> SWCLMCLIENT idget	-host <hostname> [-port <port >] [-p-host <hostname >] [-p-port <port >] [-timeout <timeout>] -ext-data <extension data>

オプション	必須/任意	説明
-host	必須	CLM のホスト名または IP アドレスを指定します。
-port	任意	CLM の待ち受けポート番号を指定します。既定値は「8443」です。
-p-host	任意	プロキシサーバの IP アドレスを指定します。 コマンドから CLM への通信時に Web プロキシサーバを経由しなければ通信できないネットワーク構成である場合は、本項目を設定してください。
-p-port	任意	プロキシサーバのポート番号を指定します。 コマンドから CLM への通信時に Web プロキシサーバを経由しなければ通信できないネットワーク構成である場合は、本項目を設定してください。
-timeout	任意	タイムアウト時間を指定します。単位は秒です。既定値は 60 秒です。指定可能な値は 1～0x7FFFFFFF です。
-ext-data	必須	拡張情報を指定します。 key=value の形式で指定してください。 value は、クォートで囲む必要があります。また、複数指定する場合はカンマで区切る必要があります。 指定する拡張情報は、後述の「拡張情報」を参照してください。

拡張情報

ID・パスワード取得機能では、-ext-data に以下の項目を指定することができます

項目	必須/任意	説明
keyid	必須	ID・パスワード発行コマンドで発行した ID を指定します。

実行例

```
# SWCLMCLIENT idget -host "192.168.0.1" -port "8443" -ext-data keyid="0000000019"
```

● ID・パスワード照合

> SWCLMCLIENT idverify	-host <hostname> [-port <port >] [-p-host <hostname >] [-p-port <port >] [-timeout <timeout>] -ext-data <extension data>

オプション	必須/任意	説明
-host	必須	CLM のホスト名または IP アドレスを指定します。
-port	任意	CLM の待ち受けポート番号を指定します。既定値は「8443」です。
-p-host	任意	プロキシサーバの IP アドレスを指定します。 コマンドから CLM への通信時に Web プロキシサーバを経由しなければ通信できないネットワーク構成である場合は、本項目を設定してください。
-p-port	任意	プロキシサーバのポート番号を指定します。 コマンドから CLM への通信時に Web プロキシサーバを経由しなければ通信できないネットワーク構成である場合は、本項目を設定してください。
-timeout	任意	タイムアウト時間を指定します。単位は秒です。既定値は 60 秒です。指定可能な値は 1～0x7FFFFFFF です。
-ext-data	必須	拡張情報を指定します。 key=value の形式で指定してください。 value は、クォートで囲む必要があります。また、複数指定する場合はカンマで区切る必要があります。 指定する拡張情報は、後述の「拡張情報」を参照してください。

拡張情報

ID・パスワード照合機能では、-ext-data に以下の項目を指定することができます。

項目	必須/任意	説明
keyid	必須	ID・パスワード発行コマンドで発行した ID を指定します。
key	必須	keyid に指定した ID のパスワードを指定します。

実行例

```
# SWCLMCLIENT idverify -host "192.168.0.1" -port "8443" -ext-data keyid="0000000001
",key="UM9Po1uJ"
```

● ID・パスワード削除

> SWCLMCLIENT iddelete	-host <hostname> [-port <port >] [-p-host <hostname >] [-p-port <port >] [-timeout <timeout>] -ext-data <extension data>

オプション	必須/任意	説明
-host	必須	CLM のホスト名または IP アドレスを指定します。
-port	任意	CLM の待ち受けポート番号を指定します。既定値は「8443」です。
-p-host	任意	プロキシサーバの IP アドレスを指定します。 コマンドから CLM への通信時に Web プロキシサーバを経由しなければ通信できないネットワーク構成である場合は、本項目を設定してください。
-p-port	任意	プロキシサーバのポート番号を指定します。 コマンドから CLM への通信時に Web プロキシサーバを経由しなければ通信できないネットワーク構成である場合は、本項目を設定してください。
-timeout	任意	タイムアウト時間を指定します。単位は秒です。既定値は 60 秒です。指定可能な値は 1～0x7FFFFFFF です。
-ext-data	必須	拡張情報を指定します。 key=value の形式で指定してください。 value は、クォートで囲む必要があります。また、複数指定する場合はカンマで区切る必要があります。 指定する拡張情報は、後述の「拡張情報」を参照してください。

拡張情報

ID・パスワード削除機能では、-ext-data に以下の項目を指定することができます。

項目	必須/任意	説明
keyid	必須	削除対象 ID を指定します。 ID・パスワード発行コマンドで発行した ID を指定します。
key	必須	削除対象 ID のパスワードを指定します。 keyid に指定した ID のパスワードを指定します。

実行例

```
# SWCLMCLIENT iddelete -host "192.168.0.1" -port "8443" -ext-data keyid="0000000001",key="UM9Po1uJ"
```

● 証明書発行

> SWCLMCLIENT certcreate	-host <hostname> [-port <port >] [-p-host <hostname >] [-p-port <port >] [-timeout <timeout>] -ext-data <extension data> -mode <mode> [-cacerttype <type>] [-certtype <type>] [-passphrase <passphrase>] -outpath <output directory> [-cacertname <filename>] [-certname <filename>] [-certkeyname <filename>]

オプション	必須/任意	説明
-host	必須	CLM のホスト名または IP アドレスを指定します。
-port	任意	CLM の待ち受けポート番号を指定します。既定値は「8443」です。
-p-host	任意	プロキシサーバの IP アドレスを指定します。 コマンドから CLM への通信時に Web プロキシサーバを経由しなければ通信できないネットワーク構成である場合は、本項目を設定してください。
-p-port	任意	プロキシサーバのポート番号を指定します。 コマンドから CLM への通信時に Web プロキシサーバを経由しなければ通信できないネットワーク構成である場合は、本項目を設定してください。
-timeout	任意	タイムアウト時間を指定します。単位は秒です。既定値は 60 秒です。指定可能な値は 1～0x7FFFFFFF です。
-ext-data	必須	拡張情報を指定します。 key=value の形式で指定してください。 value は、クォートで囲む必要があります。また、複数指定する場合はカンマで区切る必要があります。 指定する拡張情報は、後述の「拡張情報」を参照してください。

-mode	必須	取得する証明書の種別を指定します。 以下のいずれかを指定してください。 caonly: CA 証明書を取得する client: クライアント証明書を取得する server: サーバ証明書を取得する
-cacerttype	任意	CA 証明書の種別を指定します。 以下のいずれかを指定してください。 pem: pem 形式の証明書 (本項目省略時の既定値) der：der 形式の証明書
-certtype	任意	サーバ証明書、クライアント証明書の種別を指定します。 以下のいずれかを指定してください。 pem: pem 形式の証明書 (本項目省略時の既定値) der：der 形式の証明書 p12：PKCS#12 形式の証明書(chain なし)
-passphrase	任意 ※certtype が p12 の場合は、必須です。	サーバ証明書、クライアント証明書の秘密鍵のパスフレーズを指定します。 クライアント証明書の種別に「p12」が指定された場合、PKCS#12 形式に変換する際のパスフレーズにも利用します。 最大文字列長は、50byte です。 使用可能文字種は、ASCII 0x21～0x7E(但し、「;」「|」「&」「`」「(」「)」「$」「<」「>」「*」「?」「{」「}」「[」「]」「!」を除く)です。
-outpath	必須	証明書出力ディレクトリを絶対パスで指定します。 最大文字列長は、1009 －(certname、cacertname、certkeyname に指定したファイル名の最大文字列長) byte です。 使用可能文字種は、ASCII 0x21～0x7E(但し、「;」「|」「&」「`」「(」「)」「$」「<」「>」「*」「?」「{」「}」「[」「]」「!」を除く)です。
-cacertname	任意	取得した CA 証明書を保存するファイル名を指定します。 拡張子は不要です。既定値は「cacert」です。 最大文字列長は、242byte です。 使用可能文字種は、ASCII 0x21～0x7E(但し、「/」を除く。「`」「"」を指定する場合は、バックスラッシュでエスケープすること) です。

-certname	任意	取得したサーバ証明書、クライアント証明書を保存するファイル名を指定します。 拡張子は不要です。既定値は以下の通りです。 mode が「client」である場合 ：「clcert」 mode が「server」である場合: 「svcert」 最大文字列長は、242byte です。 使用可能文字種は、ASCII 0x21～0x7E(但し、「/」を除く。「`」「"」を指定する場合は、バックスラッシュでエスケープすること)です。
-certkeyname	任意	取得したサーバ証明書、クライアント証明書の秘密鍵を保存するファイル名を指定します。 拡張子は不要です。既定値は以下の通りです。 mode が「client」である場合 ：「clkey」 mode が「server」である場合: 「svkey」 最大文字列長は、242byte です。 使用可能文字種は、ASCII 0x21～0x7E(但し、「/」を除く。「`」「"」を指定する場合は、バックスラッシュでエスケープすること)です。
-passphrasetype	任意	取得したサーバ証明書、クライアント証明書の秘密鍵のパスフレーズを stdout に出力します。 以下のいずれかを指定します。省略時は、パスフレーズを出力しません。 base64：BASE64 でスクランブルをかけて出力 plaintext: 平文で出力

拡張情報

証明書発行機能では、-ext-data に以下の項目を指定することができます。

項目	必須/任意	説明
devicename	必須	どの機器(デバイス・エッジ・サーバなど)に発行するかを CLM が管理・判別するために指定します。 キッティング済みエッジ GW では、既に devicename は CLM にエッジ ID で登録されていますので、エッジ ID を指定します。 最大文字列長は、64byte です。 使用可能文字種は、英数記号(但し、「;」「|」「&」「`」「(」「)」「$」「<」「>」「*」「?」「{」「}」「[」「]」「!」を除く)です。
catype	必須	証明書の認証局名称を指定します。 ・CA 証明書を取得したい場合、「CA」を指定してください。

		・公開鍵証明書を発行・取得・更新・失効したい場合、「ca1」を指定してください。
certsubject	必須	証明書のサブジェクトを指定します。 「/C=国別コード/ST=都道府県名/L=市区町村名/O=部門名/OU=組織名/CN=コモンネーム」の形式で指定してください。「/CN=コモンネーム」以外は省略可能です。

実行例

[CA 証明書発行]

```
# SWCLMCLIENT certcreate -host "192.168.0.1" -port "8443" -mode "caonly" -cacerttype
  "pem" -outpath "/home/iotgateway" -cacertname "iotgatewaytest_ca" -ext-data devic
  ename="iotgatewaytest",catype="CA"
```

[クライアント証明書(PEM)]

```
# SWCLMCLIENT certcreate -host "192.168.0.1" -port "8443" -mode "client" -cacerttype
  "pem" -certtype "pem" -outpath "/home/iotgateway" -cacertname "iotgatewaytest_ca"
  -certname "iotgatewaytest_clcert" -certkeyname "iotgatewaytest_clientkey" -ext-data c
  ertsubject="/C=JP/O=NEC/CN=iotgatewaytest",devicename="iotgatewaytest",catype="ca
  1"
```

[クライアント証明書発行(PKCS#12)]

```
# SWCLMCLIENT certcreate -host "192.168.0.1" -port "8443" -mode "client" -certtype "p
12" -outpath "/home/iotgateway" -certname "iotgatewaytest_clcert" -passphrase "passwor
d" -ext-data certsubject="/C=JP/O=NEC/CN=iotgatewaytest",devicename="iotgatewaytest",
catype="ca1"
```

[サーバ証明書発行(DER)]

```
# SWCLMCLIENT certcreate -host "192.168.0.1" -port "8443" -mode "server" -cacerttype
  "der" –certtype "der" -outpath "/home/iotgateway" –cacertname "iotgatewaytest_ca"
  -certname "iotgatewaytest_svcert" -certkeyname "iotgatewaytest_svkey" -ext-data cert
  subject="/C=JP/O=NEC/CN=iotgatewaytest",devicename="iotgatewaytest",catype="ca1"
```

● 証明書取得

> SWCLMCLIENT certget	-host <hostname> [-port <port >] [-p-host <hostname >] [-p-port <port >] [-timeout <timeout>] -ext-data <extension data> [-certtype <type>] [-passphrase <passphrase>] -outpath <output directory> [-certname <filename>] [-certkeyname <filename>]

オプション	必須/任意	説明
-host	必須	CLM のホスト名または IP アドレスを指定します。
-port	任意	CLM の待ち受けポート番号を指定します。既定値は「8443」です。
-p-host	任意	プロキシサーバの IP アドレスを指定します。 コマンドから CLM への通信時に Web プロキシサーバを経由しなければ通信できないネットワーク構成である場合は、本項目を設定してください。
-p-port	任意	プロキシサーバのポート番号を指定します。 コマンドから CLM への通信時に Web プロキシサーバを経由しなければ通信できないネットワーク構成である場合は、本項目を設定してください。
-timeout	任意	タイムアウト時間を指定します。単位は秒です。既定値は 60 秒です。指定可能な値は 1～0x7FFFFFFF です。
-ext-data	必須	拡張情報を指定します。 key=value の形式で指定してください。 value は、クォートで囲む必要があります。また、複数指定する場合はカンマで区切る必要があります。 指定する拡張情報は、後述の「拡張情報」を参照してください。
-certtype	任意	取得対象証明書の種別を指定します。 pem: pem 形式の証明書（本項目省略時の既定値) der：der 形式の証明書 p12：PKCS#12 形式の証明書(chain なし)

-passphrase	任意 ※certtype が p12 の場合、必須です。	取得対象証明書の秘密鍵のパスフレーズを指定します。 取得対象証明書の種別に「p12」が指定された場合、PKCS#12 形式に変換する際のパスフレーズにも利用します。 最大文字列長は、50byte です。 使用可能文字種は、ASCII 0x21～0x7E(但し、「;」「|」「&」「`」「(」「)」「$」「<」「>」「*」「?」「{」「}」「[」「]」「!」を除く)です。
-outpath	必須	取得対象証明書出力ディレクトリを絶対パスで指定します。 最大文字列長は、1009 – (certname、certkeyname に指定したファイル名の最大文字列長) byte です。 使用可能文字種は、ASCII 0x21～0x7E(但し、「;」「|」「&」「`」「(」「)」「$」「<」「>」「*」「?」「{」「}」「[」「]」「!」を除く)です。
-certname	任意	取得対象証明書を保存するファイル名を指定します。 拡張子は不要です。既定値は「cert」です。 最大文字列長は、242byte です。 使用可能文字種は、ASCII 0x21～0x7E(但し、「/」を除く。「`」「"」を指定する場合は、バックスラッシュでエスケープすること)です。
-certkeyname	任意	取得対象証明書の秘密鍵を保存するファイル名を指定します。 拡張子は不要です。既定値は「certkey」です。 最大文字列長は、242byte です。 使用可能文字種は、ASCII 0x21～0x7E(但し、「/」を除く。「`」「"」を指定する場合は、バックスラッシュでエスケープすること)です。

拡張情報

証明書取得機能では、-ext-data に以下の項目を指定することができます。

項目	必須/任意	説明
devicename	必須	どの機器(デバイス・エッジ・サーバなど)から取得されるかを CLM が判別するために指定します。 キッティング済みエッジ GW では、既に devicename は CLM にエッジ ID で登録されていますので、エッジ ID を指定します。 最大文字列長は、64byte です。 使用可能文字種は、英数記号(但し、「;」「|」「&」「`」「(」「)」「$」「<」「>」「*」「?」「{」「}」「[」「]」「!」を除く)です。

catype	必須	証明書の認証局名称を指定します。 ・CA 証明書を取得したい場合、「CA」を指定してください。 ・公開鍵証明書を発行・取得・更新・失効したい場合、「ca1」を指定してください。
keynumber	必須 ※ certserial を指定しない場合、必須です。	対象証明書の鍵番号を指定します。
certserial	必須 ※ keynumber を指定しない場合、必須です。	対象証明書のシリアル番号を指定します。

実行例

[クライアント証明書取得 (PEM 形式、シリアル番号指定)]

```
# SWCLMCLIENT certget -host "192.168.0.1" -port "8443" -certtype "pem" -outpath "/home/iotgateway" -certname "iotgatewaytest_clcert" -certkeyname "iotgatewaytest_clientkey" -ext-data devicename="iotgatewaytest",certserial="CEC798689CF69040",catype="ca1"
```

[クライアント証明書取得 (PKCS#12 形式、鍵番号指定)]

```
# SWCLMCLIENT certget -host "192.168.0.1" -port "8443" -certtype "p12" -outpath "/home/iotgateway" -certname "iotgatewaytest_clcert" -passphrase "password" -ext-data devicename="iotgatewaytest",keynumber="39",catype="ca1"
```

[サーバ証明書取得 (DER 形式、シリアル番号指定)]

```
# SWCLMCLIENT certget -host "192.168.0.1" -port "8443" -certtype "der" -outpath "/home/iotgateway" -certname "iotgatewaytest_svcert" -certkeyname "iotgatewaytest_svkey" -ext-data devicename="iotgatewaytest",certserial="DEBAB6D4A8257CA5",catype="ca1"
```

● 証明書更新

> SWCLMCLIENT certupdate	-host <hostname> [-port <port >] [-p-host <hostname >] [-p-port <port >] [-timeout <timeout>] -ext-data <extension data> [-certtype <type>] [-passphrase <passphrase>] -outpath <output directory> [-certname <filename>]

オプション	必須/任意	説明
-host	必須	CLM のホスト名または IP アドレスを指定します。
-port	任意	CLM の待ち受けポート番号を指定します。既定値は「8443」です。
-p-host	任意	プロキシサーバの IP アドレスを指定します。 コマンドから CLM への通信時に Web プロキシサーバを経由しなければ通信できないネットワーク構成である場合は、本項目を設定してください。
-p-port	任意	プロキシサーバのポート番号を指定します。 コマンドから CLM への通信時に Web プロキシサーバを経由しなければ通信できないネットワーク構成である場合は、本項目を設定してください。
-timeout	任意	タイムアウト時間を指定します。単位は秒です。既定値は 60 秒です。指定可能な値は 1～0x7FFFFFFF です。
-ext-data	必須	拡張情報を指定します。 key=value の形式で指定してください。 value は、クォートで囲む必要があります。また、複数指定する場合はカンマで区切る必要があります。 指定する拡張情報は、後述の「拡張情報」を参照してください。
-certtype	任意	取得対象証明書の種別を指定します。 pem: pem 形式の証明書 (本項目省略時の既定値) der：der 形式の証明書 p12：PKCS#12 形式の証明書(chain なし)

-passphrase	任意 ※certtype が p12 の場合、必須です。	更新後の証明書の秘密鍵のパスフレーズを指定します。 取得対象証明書の種別に「p12」が指定された場合、PKCS#12 形式に変換する際のパスフレーズにも利用します。 最大文字列長は、50byte です。 使用可能文字種は、ASCII 0x21～0x7E(但し、「;」「|」「&」「`」「(」「)」「$」「<」「>」「*」「?」「{」「}」「[」「]」「!」を除く)です。
-outpath	必須	更新対象証明書が保存されているディレクトリを絶対パスで指定します。 最大文字列長は、1009 – certname に指定したファイル名の文字列長 byte です。 使用可能文字種は、ASCII 0x21～0x7E(但し、「;」「|」「&」「`」「(」「)」「$」「<」「>」「*」「?」「{」「}」「[」「]」「!」を除く)です。
-certname	任意	更新対象証明書ファイル名を指定します。 拡張子は不要です。既定値は「cert」です。 最大文字列長は、242byte です。 使用可能文字種は、ASCII 0x21～0x7E(但し、「/」を除く。「`」「"」を指定する場合は、バックスラッシュでエスケープすること)です。

拡張情報

証明書更新機能では、-ext-data に以下の項目を指定することができます。

項目	必須/任意	説明
devicename	必須	どの機器(デバイス・エッジ・サーバなど)から更新されるかを CLM が判別するために指定します。 キッティング済みエッジ GW では、既に devicename は CLM にエッジ ID で登録されていますので、エッジ ID を指定します。 最大文字列長は、64byte です。 使用可能文字種は、英数記号(但し、「;」「|」「&」「`」「(」「)」「$」「<」「>」「*」「?」「{」「}」「[」「]」「!」を除く)です。
catype	必須	証明書の認証局名称を指定します。「ca1」固定です。
keynumber	必須 ※certserial を指定しない場	対象証明書の鍵番号を指定します。

	合、必須です。	
certserial	必須 ※　keynumber を指定しない場合、必須です。	対象証明書のシリアル番号を指定します。
newkeynumber	任意	更新後の証明書の鍵番号を指定します。 既存の証明書を更新後証明書としたい場合に指定してください。
newcertserial	任意	更新後の証明書のシリアル番号を指定します。 既存の証明書を更新後証明書としたい場合に指定してください。
newcerttype	任意	更新後の証明書の種別を指定します。 以下のいずれかを指定してください。 pem: pem 形式の証明書（本項目省略時の既定値) der：der 形式の証明書 p12：PKCS#12 形式の証明書(chain なし)
newpath	任意	更新後の証明書出力ディレクトリを絶対パスで指定します。 -outpath に指定したディレクトリと異なるディレクトリに出力する場合に指定してください。 最大文字列長は、1009 – (newfilename に指定したファイル名の文字列長) byte です。 使用可能文字種は、ASCII 0x21～0x7E(但し、「;」「|」「&」「`」「(」「)」「$」「<」「>」「*」「?」「{」「}」「[」「]」「!」を除く)です。
newfilename	任意	更新した証明書を保存するファイル名を指定します。 拡張子は不要です。 -certname に指定したファイル名と異なるファイル名を付与する場合に指定してください。 最大文字列長は、242byte です。 使用可能文字種は、ASCII 0x21～0x7E(但し、「/」を除く。「`」「"」を指定する場合は、バックスラッシュでエスケープすること)です。

実行例

[証明書更新（PEM 形式、シリアル番号指定、更新後証明書を新規発行)]

```
# SWCLMCLIENT certupdate -host "192.168.0.1" -port "8443" -certtype "pem" -outpath
  "/home/iotgateway" -certname "iotgatewaytest_clcert" -ext-data devicename="iotgate
  waytest",catype="ca1",certserial="CEC798689CF69040",newpath="/home/iotgateway",ne
```

```
    wfilename="iotgatewaytest_clcert _new",newcerttype="pem"
```

[証明書更新 (PEM 形式、鍵番号指定、既存の証明書で更新)]

```
# SWCLMCLIENT certupdate -host "192.168.0.1" -port "8443" -certtype "pem" -outpath
  "/home/iotgateway" -certname "iotgatewaytest_clcert" -ext-data devicename="iotgatew
  aytest",catype="ca1",keynumber="64",newkeynumber="25",newpath="/home/iotgateway
  ",newfilename="iotgatewaytest_clcert_new",newcerttype="pem"
```

● 証明書失効

> SWCLMCLIENT certrevoke	-host <hostname> [-port <port >] [-p-host <hostname >] [-p-port <port >] [-timeout <timeout>] -ext-data <extension data> [-certtype <type>] [-certname <filename>]

オプション	必須/任意	説明
-host	必須	CLM のホスト名または IP アドレスを指定します。
-port	任意	CLM の待ち受けポート番号を指定します。既定値は「8443」です。
-p-host	任意	プロキシサーバの IP アドレスを指定します。 コマンドから CLM への通信時に Web プロキシサーバを経由しなければ通信できないネットワーク構成である場合は、本項目を設定してください。
-p-port	任意	プロキシサーバのポート番号を指定します。 コマンドから CLM への通信時に Web プロキシサーバを経由しなければ通信できないネットワーク構成である場合は、本項目を設定してください。
-timeout	任意	タイムアウト時間を指定します。単位は秒です。既定値は 60 秒です。指定可能な値は 1～0x7FFFFFFF です。
-ext-data	必須	拡張情報を指定します。 key=value の形式で指定してください。 value は、クォートで囲む必要があります。また、複数指定する場合はカンマで区切る必要があります。 指定する拡張情報は、後述の「拡張情報」を参照してください。
-certtype	任意	失効対象証明書の種別を指定します。 pem: pem 形式の証明書 (本項目省略時の既定値) der：der 形式の証明書 p12：PKCS#12 形式の証明書(chain なし)
-certname	必須	失効対象証明書ファイル名を指定します。拡張子は不要です。 最大文字列長は、242byte です。

		使用可能文字種は、ASCII 0x21～0x7E(但し、「/」を除く。「`」「"」を指定する場合は、バックスラッシュでエスケープすること)です。

拡張情報

証明書失効機能では、-ext-data に以下の項目を指定することができます。

項目	必須/任意	説明
devicename	必須	どの機器(デバイス・エッジ・サーバなど)から失効されるかを CLM が判別するために指定します。 キッティング済みエッジ GW では、既に devicename は CLM にエッジ ID で登録されていますので、エッジ ID を指定します。 最大文字列長は、64byte です。 使用可能文字種は、英数記号(但し、「;」「|」「&」「`」「(」「)」「$」「<」「>」「*」「?」「{」「}」「[」「]」「!」を除く)です。
catype	必須	証明書の認証局名称を指定します。「ca1」固定です。
keynumber	必須 ※ certserial を指定しない場合、必須です。	対象証明書の鍵番号を指定します。
certserial	必須 ※ keynumber を指定しない場合、必須です。	対象証明書のシリアル番号を指定します。
inpath	必須	失効対象証明書が保存されているディレクトリを絶対パスで指定します。 最大文字列長は、1009 – certname に指定したファイル名の文字列長 byte です。 使用可能文字種は、ASCII 0x21～0x7E(但し、「;」「|」「&」「`」「(」「)」「$」「<」「>」「*」「?」「{」「}」「[」「]」「!」を除く)です。

実行例

[証明書失効（鍵番号指定)]

```
# SWCLMCLIENT certrevoke -host "192.168.0.1" -port "8443" -certtype "pem" -certname
  ="iotgatewaytest_clcert" -ext-data devicename="iotgatewaytest",catype="ca1",keynumbe
  r="65",inpath="/home/iotgateway"
```

[証明書失効 (シリアル番号指定)]

```
# SWCLMCLIENT certrevoke -host "192.168.0.1" -port "8443" -certtype "pem" -certname
   ="iotgatewaytest_clcert" -ext-data devicename="iotgatewaytest",catype="ca1",certserial
   ="CEC798689CF69040",inpath="/home/iotgateway"
```

● 共通鍵発行

> SWCLMCLIENT cmkeycreate	-host <hostname> [-port <port >] [-p-host <hostname >] [-p-port <port >] [-timeout <timeout>] -ext-data <extension data> -outpath <output directory>

オプション	必須/任意	説明
-host	必須	CLM のホスト名または IP アドレスを指定します。
-port	任意	CLM の待ち受けポート番号を指定します。既定値は「8443」です。
-p-host	任意	プロキシサーバの IP アドレスを指定します。 コマンドから CLM への通信時に Web プロキシサーバを経由しなければ通信できないネットワーク構成である場合は、本項目を設定してください。
-p-port	任意	プロキシサーバのポート番号を指定します。 コマンドから CLM への通信時に Web プロキシサーバを経由しなければ通信できないネットワーク構成である場合は、本項目を設定してください。
-timeout	任意	タイムアウト時間を指定します。単位は秒です。既定値は 60 秒です。指定可能な値は 1～0x7FFFFFFF です。
-ext-data	必須	拡張情報を指定します。 key=value の形式で指定してください。 value は、クォートで囲む必要があります。また、複数指定する場合はカンマで区切る必要があります。 指定する拡張情報は、後述の「拡張情報」を参照してください。
-outpath	必須	共通鍵を出力するディレクトリを絶対パスで指定します。 最大文字列長は、1009 – (cmkeyname の文字列長) byte です。 使用可能文字種は、ASCII 0x21～0x7E(但し、「;」「|」「&」「`」「(」「)」「$」「<」「>」「*」「?」「{」「}」「[」「]」「!」を除く)です。

拡張情報

共通鍵発行機能では、 -ext-data に以下の項目を指定することができます。

項目	必須/任意	説明
devicename	必須	どの機器(デバイス・エッジ・サーバなど)に発行するかを CLM が管理・判別するために指定します。 キッティング済みエッジ GW では、既に devicename は CLM にエッジ ID で登録されていますので、エッジ ID を指定します。 最大文字列長は、64byte です。 使用可能文字種は、英数記号(但し、「;」「|」「&」「`」「(」「)」「$」「<」「>」「*」「?」「{」「}」「[」「]」「!」を除く)です。
alias	必須	共通鍵に付与する Alias を指定します。 最大文字列長は、64byte です。 使用可能文字種は、英数記号(但し、「;」「|」「&」「`」「(」「)」「$」「<」「>」「*」「?」「{」「}」「[」「]」「!」を除く)です。
cmkeyname	任意	発行した共通鍵を保存するファイル名を指定します。 省略時の既定値は「cmkey」です。 最大文字列長は、242byte です。 使用可能文字種は、ASCII 0x21～0x7E(但し、「/」を除く。「`」「"」を指定する場合は、バックスラッシュでエスケープすること。)
keylength	任意	発行する共通鍵の長さ(単位: bit)を指定します。 省略時の既定値は 128(bit)です。 keytype に「AES」を指定する場合、「128」または「256」のみ指定可能です。 keytype に「TWINE」を指定する場合、「80」または「128」のみ指定可能です。
keytype	必須	鍵を利用する暗号化方式を指定します。 以下のいずれかを指定します。 AES　：AES で利用する TWINE：TWINE で利用する
use-lightcipher	任意	TWINE と連携し、発行した共通鍵を TWINE で暗号化して出力する(TWINE の共通鍵ファイルとして出力する)場合に指定します。 値は、「1」固定です。 本項目を使用するには、ID 鍵コマンドを実行する環境に軽量暗号 開発キットのインストールが必要です。 また、本項目は、keytype に「TWINE」を指定し、keylength に

		「128」を指定した場合のみ指定可能です。
twinecmd	任意	TWINE の lightcipher コマンドまでのパスを指定します。 use-lightcipher と同時指定した場合に、有効になります。 省略時は、次のパスを使用します。 [Linux 版] /opt/nec/twine-otr/bin/lightcipher64 上記が存在しない場合は、 /data/twine-otr/bin/lightcipher [Windows 版] C:¥Program Files¥NEC¥twine-otr¥lightcipher.exe 上記が存在しない場合は、 C:¥Program Files (x86)¥NEC¥twine-otr¥lightcipher.exe

実行例

[共通鍵発行(AES 用 鍵長 128bit)]

```
# SWCLMCLIENT cmkeycreate -host "192.168.0.1" -port "8443" -ext-data devicename="iotgatewaytest",alias="aeskey",cmkeyname="iotgatewaytest_cmkey_aes",keylength="128",keytype="AES" -outpath "/home/iotgateway"
```

[共通鍵発行(TWINE 用 鍵長 128bit TWINE と連携)]

```
# SWCLMCLIENT cmkeycreate -host "192.168.0.1" -port "8443" -ext-data devicename="iotgatewaytest",alias="twinekey",cmkeyname="iotgatewaytest_cmkey_twine",keylength="128",keytype="TWINE",use-lightcipher=1 -outpath "/home/iotgateway"
```

● 共通鍵取得

> SWCLMCLIENT cmkeyget	-host <hostname> [-port <port >] [-p-host <hostname >] [-p-port <port >] [-timeout <timeout>] -ext-data <extension data> -outpath <output directory>

オプション	必須/任意	説明
-host	必須	CLM のホスト名または IP アドレスを指定します。
-port	任意	CLM の待ち受けポート番号を指定します。既定値は「8443」です。
-p-host	任意	プロキシサーバの IP アドレスを指定します。 コマンドから CLM への通信時に Web プロキシサーバを経由しなければ通信できないネットワーク構成である場合は、本項目を設定してください。
-p-port	任意	プロキシサーバのポート番号を指定します。 コマンドから CLM への通信時に Web プロキシサーバを経由しなければ通信できないネットワーク構成である場合は、本項目を設定してください。
-timeout	任意	タイムアウト時間を指定します。単位は秒です。既定値は 60 秒です。指定可能な値は 1～0x7FFFFFFF です。
-ext-data	必須	拡張情報を指定します。 key=value の形式で指定してください。 value は、クォートで囲む必要があります。また、複数指定する場合はカンマで区切る必要があります。 指定する拡張情報は、後述の「拡張情報」を参照してください。
-outpath	必須	共通鍵を出力するディレクトリを絶対パスで指定します。 最大文字列長は、1009 – (cmkeyname の文字列長) byte です。 使用可能文字種は、ASCII 0x21～0x7E(但し、「;」「|」「&」「`」「(」「)」「$」「<」「>」「*」「?」「{」「}」「[」「]」「!」を除く)です。

拡張情報

共通鍵取得機能では、-ext-data に以下の項目を指定することができます。

項目	必須/任意	説明
devicename	必須	どの機器(デバイス・エッジ・サーバなど)から取得されるかを CLM が判別するために指定します。 キッティング済みエッジ GW では、既に devicename は CLM にエッジ ID で登録されていますので、エッジ ID を指定します。 最大文字列長は、64byte です。 使用可能文字種は、英数記号(但し、「;」「|」「&」「`」「(」「)」「$」「<」「>」「*」「?」「{」「}」「[」「]」「!」を除く)です。
cmkeyname	任意	取得した共通鍵を保存するファイル名を指定します。 省略時の既定値は「cmkey」です。 最大文字列長は、242byte です。 使用可能文字種は、ASCII 0x21～0x7E(但し、「/」を除く。「`」「"」を指定する場合は、バックスラッシュでエスケープすること。)
keynumber	必須	取得対象共通鍵の鍵番号。 鍵番号については、後述「鍵番号について」をご覧ください。
use-lightcipher	任意	TWINE と連携し、取得した共通鍵を TWINE で暗号化して出力する(TWINE の共通鍵ファイルとして出力する)場合に指定します。 値は、「1」固定です。 本項目を使用するには、ID 鍵コマンドを実行する環境に軽量暗号 開発キットのインストールが必要です。 また、本項目は、取得対象共通鍵の keylength が「128」の場合にのみ指定可能です。
twinecmd	任意	TWINE の lightcipher コマンドまでのパスを指定します。 use-lightcipher と同時指定した場合に、有効になります。 省略時は、次のパスを使用します。 [Linux 版] /opt/nec/twine-otr/bin/lightcipher64 上記が存在しない場合は、 /data/twine-otr/bin/lightcipher [Windows 版] C:¥Program Files¥NEC¥twine-otr¥lightcipher.exe 上記が存在しない場合は、 C:¥Program Files (x86)¥NEC¥twine-otr¥lightcipher.exe

実行例

[共通鍵取得(AES 用 鍵長 128bit)]

```
# SWCLMCLIENT cmkeyget -host "192.168.0.1" -port "8443" -ext-data devicename="iotg
  atewaytest",cmkeyname="iotgatewaytest_cmkey_aes",keynumber="147" -outpath "/ho
  me/iotgateway"
```

● 共通鍵更新

> SWCLMCLIENT cmkeyupdate	-host <hostname> [-port <port >] [-p-host <hostname >] [-p-port <port >] [-timeout <timeout>] -ext-data <extension data> -outpath <output directory>

オプション	必須/任意	説明
-host	必須	CLM のホスト名または IP アドレスを指定します。
-port	任意	CLM の待ち受けポート番号を指定します。既定値は「8443」です。
-p-host	任意	プロキシサーバの IP アドレスを指定します。 コマンドから CLM への通信時に Web プロキシサーバを経由しなければ通信できないネットワーク構成である場合は、本項目を設定してください。
-p-port	任意	プロキシサーバのポート番号を指定します。 コマンドから CLM への通信時に Web プロキシサーバを経由しなければ通信できないネットワーク構成である場合は、本項目を設定してください。
-timeout	任意	タイムアウト時間を指定します。単位は秒です。既定値は 60 秒です。指定可能な値は 1～0x7FFFFFFF です。
-ext-data	必須	拡張情報を指定します。 key=value の形式で指定してください。 value は、クォートで囲む必要があります。また、複数指定する場合はカンマで区切る必要があります。 指定する拡張情報は、後述の「拡張情報」を参照してください。
-outpath	必須	更新対象共通鍵が保存されているディレクトリを絶対パスで指定します。 最大文字列長は、1009 – (cmkeyname の文字列長) byte です。 使用可能文字種は、ASCII 0x21～0x7E(但し、「;」「|」「&」「`」「(」「)」「$」「<」「>」「*」「?」「{」「}」「[」「]」「!」を除く)です。

拡張情報

共通鍵更新機能では、-ext-data に以下の項目を指定することができます。

項目	必須/任意	説明
devicename	必須	どの機器(デバイス・エッジ・サーバなど)から更新されるかを CLM が判別するために指定します。 キッティング済みエッジ GW では、既に devicename は CLM にエッジ ID で登録されていますので、エッジ ID を指定します。 最大文字列長は、64byte です。 使用可能文字種は、英数記号(但し、「;」「|」「&」「`」「(」「)」「$」「<」「>」「*」「?」「{」「}」「[」「]」「!」を除く)です。
alias	任意	更新後の共通鍵に付与する Alias を指定します。 最大文字列長は、64byte です。 使用可能文字種は、英数記号(但し、「;」「|」「&」「`」「(」「)」「$」「<」「>」「*」「?」「{」「}」「[」「]」「!」を除く)です。
cmkeyname	任意	更新対象共通鍵ファイル名を指定します。 省略時の既定値は「cmkey」です。 最大文字列長は、242byte です。 使用可能文字種は、ASCII 0x21～0x7E(但し、「/」を除く。「`」「"」を指定する場合は、バックスラッシュでエスケープすること。)
keynumber	必須	対象共通鍵の鍵番号を指定します。 鍵番号については、後述「鍵番号について」をご覧ください。
newkeynumber	任意	更新後の共通鍵の鍵番号を指定します。 既存の共通鍵を更新後共通鍵としたい場合に指定してください。
keylength	任意	更新後の共通鍵の長さ(単位: bit)を指定します。 省略時の既定値は 128(bit)です。 更新対象共通鍵の keytype が「AES」である場合、「128」または「256」を指定します。 更新対象共通鍵の keytype が「TWINE」である場合、「80」または「128」を指定します。
newpath	任意	更新後の共通鍵出力ディレクトリを絶対パスで指定します。 -outpath に指定したディレクトリと異なるディレクトリに出力する場合に指定してください。 最大文字列長は、1009 – (newfilename に指定したファイル名の文字列長) byte です。 使用可能文字種は、ASCII 0x21～0x7E(但し、「;」「|」「&」「`」

		「(」「)」「$」「<」「>」「*」「?」「{」「}」「[」「]」「!」を除く)です。
newfilename	任意	更新した共通鍵を保存するファイル名を指定します。 cmkeyname に指定したファイル名と異なるファイル名を付与する場合に指定してください。 最大文字列長は、242byte です。 使用可能文字種は、ASCII 0x21～0x7E(但し、「/」を除く。「`」「"」を指定する場合は、バックスラッシュでエスケープすること。)
use-lightcipher	任意	TWINE と連携し、更新した共通鍵を TWINE で暗号化して出力する(TWINE の共通鍵ファイルとして出力する)場合に指定します。 値は、「1」固定です。 本項目を使用するには、ID 鍵コマンドを実行する環境に軽量暗号 開発キットのインストールが必要です。 また、本項目は、更新対象共通鍵の keytype が「TWINE」であり、keylength が「128」である場合にのみ指定可能です。
twinecmd	任意	TWINE の lightcipher コマンドまでのパスを指定します。 use-lightcipher と同時指定した場合に、有効になります。 省略時は、次のパスを使用します。 [Linux 版] /opt/nec/twine-otr/bin/lightcipher64 上記が存在しない場合は、 /data/twine-otr/bin/lightcipher [Windows 版] C:¥Program Files¥NEC¥twine-otr¥lightcipher.exe 上記が存在しない場合は、 C:¥Program Files (x86)¥NEC¥twine-otr¥lightcipher.exe

実行例

[共通鍵更新 (AES 用 鍵長 128bit、更新後共通鍵を新規発行)]

```
# SWCLMCLIENT cmkeyupdate -host "192.168.0.1" -port "8443" -ext-data devicename="
  iotgatewaytest",alias="newkey",cmkeyname="iotgatewaytest_cmkey_aes",keynumber="1
  47",keylength="128",newpath="/home/iotgateway",newfilename="iotgatewaytest_cmkey
  _aes_new" -outpath "/home/iotgateway"
```

[共通鍵更新 (AES 用 鍵長 128bit、既存の共通鍵で更新)]

```
# SWCLMCLIENT cmkeyupdate -host "192.168.0.1" -port "8443" -outpath "/home/iotgate
  way" -ext-data devicename="iotgatewaytest",cmkeyname="iotgatewaytest_cmkey_aes",
  keynumber="148",newkeynumber="147",newpath="/home/iotgateway",newfilename="io
  tgatewaytest_cmkey_aes_new"
```

● 共通鍵削除

> SWCLMCLIENT cmkeydelete	-host <hostname> [-port <port >] [-p-host <hostname >] [-p-port <port >] [-timeout <timeout>] -ext-data <extension data>

オプション	必須/任意	説明
-host	必須	CLM のホスト名または IP アドレスを指定します。
-port	任意	CLM の待ち受けポート番号を指定します。既定値は「8443」です。
-p-host	任意	プロキシサーバの IP アドレスを指定します。 コマンドから CLM への通信時に Web プロキシサーバを経由しなければ通信できないネットワーク構成である場合は、本項目を設定してください。
-p-port	任意	プロキシサーバのポート番号を指定します。 コマンドから CLM への通信時に Web プロキシサーバを経由しなければ通信できないネットワーク構成である場合は、本項目を設定してください。
-timeout	任意	タイムアウト時間を指定します。単位は秒です。既定値は 60 秒です。指定可能な値は 1～0x7FFFFFFF です。
-ext-data	必須	拡張情報を指定します。 key=value の形式で指定してください。 value は、クォートで囲む必要があります。また、複数指定する場合はカンマで区切る必要があります。 指定する拡張情報は、後述の「拡張情報」を参照してください。

拡張情報

共通鍵削除機能では、-ext-data に以下の項目を指定することができます。

項目	必須/任意	説明
devicename	必須	どの機器(デバイス・エッジ・サーバなど)から削除されるかを CLM が判別するために指定します。 キッティング済みエッジ GW では、既に devicename は CLM にエッジ ID で登録されていますので、エッジ ID を指定します。

		最大文字列長は、64byte です。 使用可能文字種は、英数記号(但し、「;」「|」「&」「`」「(」「)」「$」「<」「>」「*」「?」「{」「}」「[」「]」「!」を除く)です。
keynumber	必須	対象共通鍵の鍵番号を指定します。 鍵番号については、後述「鍵番号について」をご覧ください。
cmkeyname	必須	削除対象共通鍵ファイル名を指定します。 最大文字列長は、242byte です。 使用可能文字種は、ASCII 0x21～0x7E(但し、「/」を除く。「`」「"」を指定する場合は、バックスラッシュでエスケープすること。)
inpath	必須	削除対象共通鍵が保存されているディレクトリを絶対パスで指定します。 最大文字列長は、1009 – (cmkeyname に指定したファイル名の最大文字列長) byte です。 使用可能文字種は、ASCII 0x21～0x7E(但し、「;」「|」「&」「`」「(」「)」「$」「<」「>」「*」「?」「{」「}」「[」「]」「!」を除く)です。
use-lightcipher	任意	TWINE と連携し、TWINE で暗号化して出力(TWINE の共通鍵ファイルとして出力)した共通鍵を削除する場合、本項目を指定します。 値は、「1」固定です。 本項目を使用するには、ID 鍵コマンドを実行する環境に軽量暗号 開発キットのインストールが必要です。
twinecmd	任意	TWINE の lightcipher コマンドまでのパスを指定します。 use-lightcipher と同時指定した場合に、有効になります。 省略時は、次のパスを使用します。 [Linux 版] /opt/nec/twine-otr/bin/lightcipher64 上記が存在しない場合は、 /data/twine-otr/bin/lightcipher [Windows 版] C:¥Program Files¥NEC¥twine-otr¥lightcipher.exe 上記が存在しない場合は、 C:¥Program Files (x86)¥NEC¥twine-otr¥lightcipher.exe

実行例

[共通鍵削除]

```
# SWCLMCLIENT cmkeydelete -host "192.168.0.1" -port "8443" -ext-data devicename="iotgatewaytest",keynumber="66",cmkeyname="iotgatewaytest_cmkey_aes_new",inpath="/home/iotgateway"
```

● 共通鍵照合

> SWCLMCLIENT cmkeyverify	-host <hostname> [-port <port >] [-p-host <hostname >] [-p-port <port >] [-timeout <timeout>] -ext-data <extension data>

オプション	必須/任意	説明
-host	必須	CLM のホスト名または IP アドレスを指定します。
-port	任意	CLM の待ち受けポート番号を指定します。既定値は「8443」です。
-p-host	任意	プロキシサーバの IP アドレスを指定します。 コマンドから CLM への通信時に Web プロキシサーバを経由しなければ通信できないネットワーク構成である場合は、本項目を設定してください。
-p-port	任意	プロキシサーバのポート番号を指定します。 コマンドから CLM への通信時に Web プロキシサーバを経由しなければ通信できないネットワーク構成である場合は、本項目を設定してください。
-timeout	任意	タイムアウト時間を指定します。単位は秒です。既定値は 60 秒です。指定可能な値は 1～0x7FFFFFFF です。
-ext-data	必須	拡張情報を指定します。 key=value の形式で指定してください。 value は、クォートで囲む必要があります。また、複数指定する場合はカンマで区切る必要があります。 指定する拡張情報は、後述の「拡張情報」を参照してください。

拡張情報

共通鍵削除機能では、-ext-data に以下の項目を指定することができます。

項目	必須/任意	説明
devicename	必須	どの機器(デバイス・エッジ・サーバなど)からの照合かを CLM が判別するために指定します。 キッティング済みエッジ GW では、既に devicename は CLM に

		エッジ ID で登録されていますので、エッジ ID を指定します。 最大文字列長は、64byte です。 使用可能文字種は、英数記号(但し、「;」「|」「&」「`」「(」「)」「$」「<」「>」「*」「?」「{」「}」「[」「]」「!」を除く)です。
keynumber	必須	照合対象共通鍵の鍵番号を指定します。 鍵番号については、後述「鍵番号について」をご覧ください。
cmkeyname	必須	照合対象共通鍵ファイル名を指定します。 最大文字列長は、242byte です。 使用可能文字種は、ASCII 0x21～0x7E(但し、「/」を除く。「`」「"」を指定する場合は、バックスラッシュでエスケープすること。)
inpath	必須	削除対象共通鍵が保存されているディレクトリを絶対パスで指定します。 最大文字列長は、1009 – (cmkeyname に指定したファイル名の最大文字列長) byte です。 使用可能文字種は、ASCII 0x21～0x7E(但し、「;」「|」「&」「`」「(」「)」「$」「<」「>」「*」「?」「{」「}」「[」「]」「!」を除く)です。
use-lightcipher	任意	TWINE と連携し、TWINE で暗号化して出力(TWINE の共通鍵ファイルとして出力)した共通鍵を照合する場合、本項目を指定します。 値は、「1」固定です。 本項目を使用するには、ID 鍵コマンドを実行する環境に軽量暗号 開発キットのインストールが必要です。
twinecmd	任意	TWINE の lightcipher コマンドまでのパスを指定します。 use-lightcipher と同時指定した場合に、有効になります。 省略時は、次のパスを使用します。 [Linux 版] /opt/nec/twine-otr/bin/lightcipher64 上記が存在しない場合は、 /data/twine-otr/bin/lightcipher [Windows 版] C:¥Program Files¥NEC¥twine-otr¥lightcipher.exe 上記が存在しない場合は、 C:¥Program Files (x86)¥NEC¥twine-otr¥lightcipher.exe

[共通鍵照合（AES 用鍵)]

```
# SWCLMCLIENT cmkeyverify -host "192.168.0.1" -port "8443" -ext-data devicename="io
  tgatewaytest",keynumber="148",cmkeyname="iotgatewaytest_cmkey_aes",inpath="/hom
  e/iotgateway"
```

[共通鍵照合（TWINE 用鍵 TWINE と連携)]

```
# SWCLMCLIENT cmkeyverify -host "192.168.0.1" -port "8443" -ext-data devicename="io
  tgatewaytest",keynumber="61",cmkeyname="iotgatewaytest_cmkey_twine",inpath="/hom
  e/iotgateway",use-lightcipher=1
```

実行結果

実行結果は、ID 鍵コマンドの終了コードと CLM からのレスポンスで確認します。

ID 鍵コマンド 終了コード

ID 鍵コマンドは、終了コードと詳細な情報を標準出力(一部、標準エラー出力)に以下のフォーマットで出力します。

HTTP ステータスコード
ID 鍵コマンドが出力するメッセージ
CLM から返却されたエラーコード
ID 鍵コマンドの終了コード

ID 鍵コマンドが出力するメッセージは、通常、標準出力に出力しますが、エラー発生時は標準エラー出力に出力します。

HTTP ステータスコードと CLM から返却されたエラーコードの詳細、および ID 鍵コマンドの終了コードについては、6 章をご覧ください。

CLM からのレスポンス

ID 鍵コマンドを実行すると、CLM からのレスポンスをファイルに出力し、以下に保存します。

[Linux]
/tmp/swclm-result.json
[Windows]
%TEMP%¥swclm-result.json

CLM からのレスポンスは JSON 形式です。レスポンスの詳細は、別紙「WebAPI リファレンス」記載の各 API レスポンスパラメータをご覧ください。

「証明書発行・取得・更新」コマンドで発行・取得・更新した証明書について

CLM で発行した証明書は、ID 鍵コマンド 実行環境上に保存します。保存先は、次の通りです。

- 証明書発行、証明書取得時

  パラメータ「-outpath」に指定したディレクトリです。

- 証明書更新時

  パラメータ「-outpath」に指定したディレクトリです。ただし、拡張情報「newpath」を指定している場合は、拡張情報「newpath」に指定したディレクトリに保存します。

保存する証明書のファイル名は、次の通りです。

- CA 証明書

  パラメータ「-cacertname」に指定した値がファイル名となります。

- 公開鍵証明書

  パラメータ「-certname」に指定した値がファイル名となります。

  ただし、証明書更新時、拡張情報「newfilename」を指定している場合は、拡張情報「newfilename」に指定した値がファイル名となります。
- 公開鍵証明書の秘密鍵

  パラメータ「-certkeyname」にした値がファイル名となります。

保存する証明書の拡張子は、次の通りです。

- CA 証明書

  パラメータ「-cacerttype」に指定した値に従い付与されます。
- 公開鍵証明書

  パラメータ「-certtype」に指定した値に従い付与されます。

  ただし、証明書更新時、拡張情報「newcerttype」を指定している場合は、拡張情報「newcerttype」に指定した値に従い付与されます。
- 公開鍵証明書の秘密鍵

  拡張子は「.pem」固定です

「共通鍵発行・取得・更新」機能で発行・取得・更新した共通鍵について

CLM で発行した共通鍵は、ID 鍵コマンド 実行環境上に保存します。保存先は、次の通りです。

- 共通鍵発行、共通鍵取得時

  パラメータ「-outpath」に指定したディレクトリです。
- 共通鍵更新時

  パラメータ「-outpath」に指定したディレクトリです。

  ただし、拡張情報「newpath」を指定している場合は、拡張情報「newpath」に指定したディレクトリに保存します。

保存する共通鍵のファイル名は、次の通りです。

- 共通鍵発行、共通鍵取得時

  拡張情報「cmkeyname」に指定した値がファイル名となります。
- 共通鍵更新

  拡張情報「cmkeyname」に指定した値がファイル名となります。

  ただし、拡張情報「newfilename」を指定している場合は、拡張情報「newfilename」に指定した値がファイル名となります。

鍵番号について

CLM で管理している電子証明書・共通鍵には、CLM 内部の管理番号としてユニークな番号が付与されま

す。この番号を「鍵番号」と呼称します。

電子証明書・共通鍵に付与されている鍵番号は、以下で確認することが可能です。

- ID 鍵コマンドで電子証明書・共通鍵を発行・更新した場合

  CLM サーバから受信したレスポンスを確認します。

  レスポンスは、「/tmp/swclm-result.json（Windows 版は、%TEMP%¥swclm-result.json)」に保存されています(上述「実行結果」もご覧ください)。

  - 証明書発行コマンド、共通鍵発行コマンドを実行して電子証明書・共通鍵を発行した場合

    鍵番号は、レスポンス内のパラメータ「keynumber」に格納しています。
  - 証明書更新コマンド、共通鍵更新コマンドを使用して電子証明書・共通鍵を更新した場合

    更新後の電子証明書・共通鍵の鍵番号は、レスポンス内のパラメータ「keynumber」に格納しています。

ログ出力

ID 鍵コマンドは、以下にログを出力します。

- Linux 版
  - 標準出力・エラー出力
  - /var/log/swcagent.log
- Windows 版
  - 標準出力・エラー出力
  - イベントログ(Application)

    イベントログに出力するログについては、6.5 章をご覧ください。

4　ID 鍵コマンドの機能拡張

ID 鍵コマンドは、WebAPI 実行コマンドを拡張し、CLM の提供機能をより便利に利用可能にするコマンドです。

前提

- Linux 版
  - コマンドは、root ユーザもしくは root グループに所属するユーザで実行してください。
- Windows 版
  - コマンドは、Administrators 権限を保有したユーザで実行してください。

提供機能

本コマンドでは、基本機能に加えて、以下の機能を提供します。

- Alias 指定での最新共通鍵取得

  指定 Alias を付与されている共通鍵のうち、最新の共通鍵を取得します。
- Alias 指定での共通鍵更新

  自サーバ・エッジデバイスで発行・更新した共通鍵のうち、指定 Alias を付与されている共通鍵を更新します。

実行形式

実行時に以下のオプション、および、オプション値を指定します。指定できるオプションは、機能によって異なります。

- 本コマンドを利用する上での留意事項
  - 事前に ID・パスワード照合コマンド(3 章 ID・パスワード照合(SWCLMCLIENT idverify)をご覧ください)を実行し、CLM と ID・パスワードによる認証を行っておく必要があります。
  - 共通鍵取得・更新コマンドでは、有効期限内の共通鍵を操作することが可能です。有効期限が切れた共通鍵は、更新できません。
  - コマンドで使用しないオプションを指定した場合、そのオプションは無視されます。
    例) alias 指定での共通鍵取得コマンドで-mode オプションを指定すると、-mode オプションは無視してコマンド実行されます。
  - コマンドで使用しない拡張情報を指定した場合、その拡張情報は無視されます。
    例) alias 指定での共通鍵取得コマンドで-ext-data オプションに certsubject="/CN=A"や desc="ABC"と指定すると、certsubject="/CN=A"と desc="ABC"は無視してコマンド実行されます。
  - コマンドのオプション、拡張情報に記号を含む文字列を指定する場合、記号はエスケープ文字によりエスケープした状態で指定する必要があります。エスケープ文字およびエスケープの方法については、Linux 版の場合は使用するシェルの仕様に、Windows 版の場合はコマンドプロンプトの仕様に従います。
  - Windows 版のコマンドを実行する際は、必ず拡張子(.cmd)まで指定して実行する必要があります。

- alias 指定での共通鍵取得

> SWCLMCLIENT cmkeyget	-host <hostname> [-port <port >] [-p-host <hostname >] [-p-port <port >] [-timeout <timeout>] -ext-data <extension data> -outpath <output directory>

オプション	必須/任意	説明
-host	必須	CLM のホスト名または IP アドレスを指定します。

-port	任意	CLM の待ち受けポート番号を指定します。既定値は「8443」です。
-p-host	任意	プロキシサーバの IP アドレスを指定します。 コマンドから CLM への通信時に Web プロキシサーバを経由しなければ通信できないネットワーク構成である場合は、本項目を設定してください。
-p-port	任意	プロキシサーバのポート番号を指定します。 コマンドから CLM への通信時に Web プロキシサーバを経由しなければ通信できないネットワーク構成である場合は、本項目を設定してください。
-timeout	任意	タイムアウト時間を指定します。単位は秒です。既定値は 60 秒です。指定可能な値は 1～0x7FFFFFFF です。
-ext-data	必須	拡張情報を指定します。 key=value の形式で指定してください。 value は、クォートで囲む必要があります。また、複数指定する場合はカンマで区切る必要があります。 指定する拡張情報は、後述の「拡張情報」を参照してください。
-outpath	必須	共通鍵を出力するディレクトリを絶対パスで指定します。 最大文字列長は、1009 – (cmkeyname の文字列長) byte です。 使用可能文字種は、ASCII 0x21～0x7E(但し、「;」「|」「&」「`」「(」「)」「$」「<」「>」「*」「?」「{」「}」「[」「]」「!」を除く)です。

拡張情報

共通鍵取得機能では、-ext-data に以下の項目を指定することができます。

項目	必須/任意	説明
devicename	必須	どの機器(デバイス・エッジ・サーバなど)から取得されるかを CLM が判別するために指定します。 キッティング済みエッジ GW では、既に devicename は CLM にエッジ ID で登録されていますので、エッジ ID を指定します。 最大文字列長は、64byte です。 使用可能文字種は、英数記号(但し、「;」「|」「&」「`」「(」「)」「$」「<」「>」「*」「?」「{」「}」「[」「]」「!」を除く)です。
cmkeyname	任意	取得した共通鍵を保存するファイル名を指定します。 省略時の既定値は「cmkey」です。 最大文字列長は、242byte です。

		使用可能文字種は、ASCII 0x21～0x7E(但し、「/」を除く。「`」「"」を指定する場合は、バックスラッシュでエスケープすること。)
alias	必須	取得対象共通鍵に付与されている Alias を指定します。 最大文字列長は、64byte です。 使用可能文字種は、英数記号(但し、「;」「|」「&」「`」「(」「)」「$」「<」「>」「*」「?」「{」「}」「[」「]」「!」を除く)です。
use-lightcipher	任意	TWINE と連携し、取得した共通鍵を TWINE で暗号化して出力する(TWINE の共通鍵ファイルとして出力する)場合に指定します。 値は、「1」固定です。 本項目を使用するには、ID 鍵コマンドを実行する環境に軽量暗号 開発キットのインストールが必要です。 また、本項目は、取得対象共通鍵の keylength が「128」の場合にのみ指定可能です。
twinecmd	任意	TWINE の lightcipher コマンドまでのパスを指定します。 use-lightcipher と同時指定した場合に、有効になります。 省略時は、次のパスを使用します。 [Linux 版] /opt/nec/twine-otr/bin/lightcipher64 上記が存在しない場合は、 /data/twine-otr/bin/lightcipher [Windows 版] C:¥Program Files¥NEC¥twine-otr¥lightcipher.exe 上記が存在しない場合は、 C:¥Program Files (x86)¥NEC¥twine-otr¥lightcipher.exe

実行例

[共通鍵取得]

```
# SWCLMCLIENT cmkeyget -host "192.168.0.1" -port "8443" -ext-data devicename="iotg
  atewaytest",cmkeyname="iotgatewaytest_cmkey_aes",alias="aeskey" -outpath "/home/i
  otgateway"
```

- alias 指定での共通鍵更新

>SWCLMCLIENT cmkeyupdate	-host <hostname> [-port <port >] [-p-host <hostname >] [-p-port <port >] [-timeout <timeout>] -ext-data <extension data> -outpath <output directory>

オプション	必須/任意	説明
-host	必須	CLM のホスト名または IP アドレスを指定します。
-port	任意	CLM の待ち受けポート番号を指定します。既定値は「8443」です。
-p-host	任意	プロキシサーバの IP アドレスを指定します。 コマンドから CLM への通信時に Web プロキシサーバを経由しなければ通信できないネットワーク構成である場合は、本項目を設定してください。
-p-port	任意	プロキシサーバのポート番号を指定します。 コマンドから CLM への通信時に Web プロキシサーバを経由しなければ通信できないネットワーク構成である場合は、本項目を設定してください。
-timeout	任意	タイムアウト時間を指定します。単位は秒です。既定値は 60 秒です。指定可能な値は 1～0x7FFFFFFF です。
-ext-data	必須	拡張情報を指定します。 key=value の形式で指定してください。 value は、クォートで囲む必要があります。また、複数指定する場合はカンマで区切る必要があります。 指定する拡張情報は、後述の「拡張情報」を参照してください。
-outpath	必須	更新対象共通鍵が保存されているディレクトリを絶対パスで指定します。 最大文字列長は、1009 – (cmkeyname の文字列長) byte です。 使用可能文字種は、ASCII 0x21～0x7E(但し、「;」「|」「&」「`」「(」「)」「$」「<」「>」「*」「?」「{」「}」「[」「]」「!」を除く)です。

拡張情報

共通鍵更新機能では、-ext-data に以下の項目を指定することができます。

項目	必須/任意	説明
devicename	必須	どの機器(デバイス・エッジ・サーバなど)から更新されるかを CLM が判別するために指定します。 キッティング済みエッジ GW では、既に devicename は CLM にエッジ ID で登録されていますので、エッジ ID を指定します。 最大文字列長は、64byte です。 使用可能文字種は、英数記号(但し、「;」「|」「&」「`」「(」「)」「$」「<」「>」「*」「?」「{」「}」「[」「]」「!」を除く)です。
alias	必須	更新対象共通鍵に付与されている Alias を指定します。 最大文字列長は、64byte です。 使用可能文字種は、英数記号(但し、「;」「|」「&」「`」「(」「)」「$」「<」「>」「*」「?」「{」「}」「[」「]」「!」を除く)です。
newalias	任意	更新後の共通鍵に付与されている Alias を指定します。 既存の共通鍵を更新後共通鍵としたい場合に指定してください。
cmkeyname	任意	更新対象共通鍵ファイル名を指定します。 省略時の既定値は「cmkey」です。 最大文字列長は、242byte です。 使用可能文字種は、ASCII 0x21～0x7E(但し、「/」を除く。「`」「"」を指定する場合は、バックスラッシュでエスケープすること。)
keylength	任意	更新後の共通鍵の長さ(単位: bit)を指定します。 省略時の既定値は 128(bit)です。 Keytype に「AES」を指定する場合、「128」または「256」を指定します。 Keytype に「TWINE」を指定する場合、「80」または「128」を指定します。
newpath	任意	更新後の共通鍵出力ディレクトリを絶対パスで指定します。 -outpath に指定したディレクトリと異なるディレクトリに出力する場合に指定してください。 最大文字列長は、1009 – (newfilename に指定したファイル名の文字列長) byte です。 使用可能文字種は、ASCII 0x21～0x7E(但し、「;」「|」「&」「`」「(」「)」「$」「<」「>」「*」「?」「{」「}」「[」「]」「!」を除く)です。

newfilename	任意	更新した共通鍵を保存するファイル名を指定します。 cmkeyname に指定したファイル名と異なるファイル名を付与する場合に指定してください。 最大文字列長は、242byte です。 使用可能文字種は、ASCII 0x21～0x7E(但し、「/」を除く。「`」「"」を指定する場合は、バックスラッシュでエスケープすること。)
use-lightcipher	任意	TWINE と連携し、更新した共通鍵を TWINE で暗号化して出力する(TWINE の共通鍵ファイルとして出力する)場合に指定します。 値は、「1」固定です。 本項目を使用するには、ID 鍵コマンドを実行する環境に軽量暗号 開発キットのインストールが必要です。 また、本項目は、更新対象共通鍵の keytype が「TWINE」であり、keylength が「128」である場合にのみ指定可能です。
twinecmd	任意	TWINE の lightcipher コマンドまでのパスを指定します。 use-lightcipher と同時指定した場合に、有効になります。 省略時は、次のパスを使用します。 [Linux 版] /opt/nec/twine-otr/bin/lightcipher64 上記が存在しない場合は、 /data/twine-otr/bin/lightcipher [Windows 版] C:¥Program Files¥NEC¥twine-otr¥lightcipher.exe 上記が存在しない場合は、 C:¥Program Files (x86)¥NEC¥twine-otr¥lightcipher.exe

実行例

[共通鍵更新]

```
# SWCLMCLIENT cmkeyupdate -host "192.168.0.1" -port "8443" -ext-data devicename="
  iotgatewaytest",alias="aeskey",cmkeyname="iotgatewaytest_cmkey_aes",keylength="128
  " -outpath "/home/iotgateway"
```

実行結果

実行結果は、ID 鍵コマンドの終了コードと CLM からのレスポンスで確認します。

詳細は、3 章「実行結果」をご覧ください。

「共通鍵発行・取得・更新」機能で発行・取得・更新した共通鍵について

CLM で発行した共通鍵は、ID 鍵コマンド 実行環境上に保存します。

詳細は、3 章「「共通鍵発行・取得・更新」機能で発行・取得・更新した共通鍵について」をご覧ください。

ログ出力

ID 鍵コマンドは、以下にログを出力します。

- Linux 版
  - 標準出力・エラー出力
  - /var/log/swcagent.log
- Windows 版
  - 標準出力・エラー出力
  - イベントログ(Application)

    イベントログに出力するログについては、6.5 章をご覧ください。

5 共通鍵一括取得コマンド

共通鍵一括取得コマンドは、ID 鍵コマンドを機能拡張し、デバイスが保有する共通鍵を CLM から一括取得するコマンドです。

前提

- Linux 版共通鍵一括取得コマンド
  - コマンドは、root ユーザもしくは root グループに所属するユーザで実行してください。
- Windows 版共通鍵一括取得コマンド
  - コマンドは、Administrators 権限を保有したユーザで実行してください。
- Linux 版、Windows 版共通
  - 事前準備として、以下を行っておく必要があります。
    - 共通鍵一括取得コマンド実行環境上で ID・パスワード照合を未実施である場合、ID・パスワード照合コマンド(3 章 ID・パスワード照合(SWCLMCLIENT idverify)をご覧ください)を実行し、CLM と ID・パスワードによる認証を行っておく必要があります。
    - 共通鍵一括取得コマンド実行環境にインストールした CLA の設定ファイルへ、CLM へのアクセスに必要な情報の設定を行います。

      [CLA 設定ファイル]

      Linux 版: /etc/swcagent/swcagent.conf

      Windows 版: c:¥swclm¥conf¥swcagent¥swcagent.conf

      [設定内容]

      swcagent.conf の末尾に、以下の 2 行を追記します。

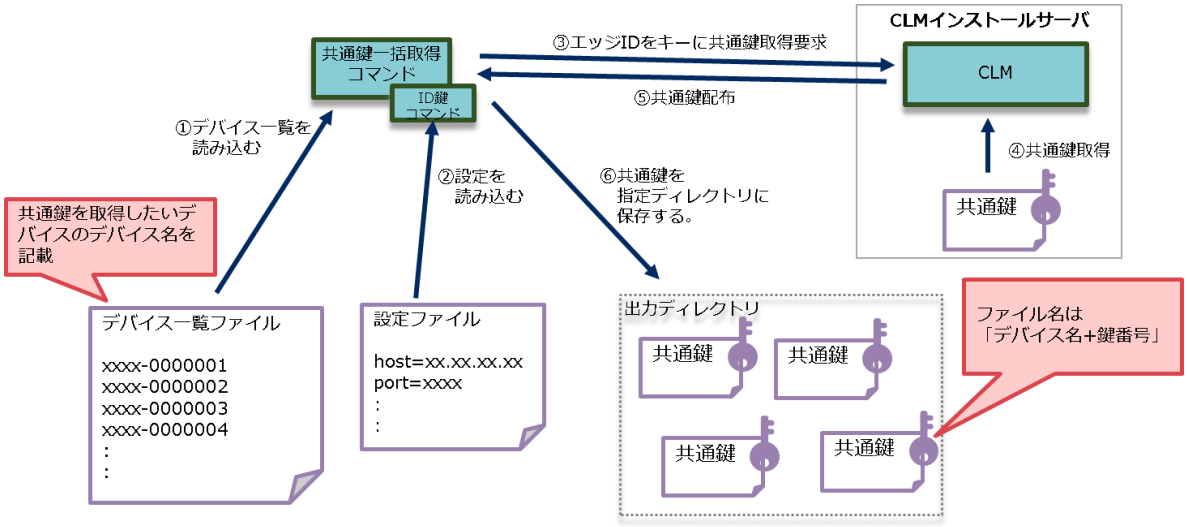
host=CLM インストールサーバの IP アドレスまたは FQDN port=CLM の待ち受けポート番号

提供機能

本コマンドの提供している機能は以下の通りです。

- 共通鍵一括取得

  CLM が管理している共通鍵を、デバイス名をキーに検索・取得し、指定ディレクトリにファイルとして保存します。1 回の実行で、複数のデバイスに紐づいた共通鍵を取得・保存することが可能です。



コマンド実行時にエラーが発生した場合は、エラー内容を標準エラー出力とログファイルに出力します。複数のデバイスを指定して実行し、途中でエラーが発生した場合は、エラーが発生したデバイスの処理を中断・スキップし、後続のデバイスの共通鍵取得を実行します。

実行形式

実行時に以下のオプション、および、オプション値を指定します。

- 共通鍵一括取得

  [Linux 版]

> SWCLMKEYGET	-i <devicename list file> -o <outdir> -e <log file> [-b]

オプション	必須/任意	説明
-i	必須	デバイス名 の一覧を記載したファイルまでの絶対 PATH を指定します。デバイス名は、1 行に 1 デバイス名を記述します。

-o	必須	取得した共通鍵を保存するディレクトリまでの絶対 PATH を指定します。本パラメータに指定するディレクトリは、事前に作成しておく必要があります。
-e	必須	エラー発生時のログを出力するファイルまでの絶対 PATH を指定します。指定したファイルが既に存在する場合は、ファイルを上書きします。
-b	任意	取得した共通鍵のバイナリデータを保存したい場合に指定します。 詳細は、後述「共通鍵一括取得コマンドで取得した共通鍵について」をご覧ください。

[Windows 版]

> SWCLMKEYGET.cmd	<devicename list file> <outdir> <log file> [-b]

引数	必須/任意	説明
<devicename list file>	必須	デバイス名 の一覧を記載したファイルまでの絶対 PATH を指定します。デバイス名は、1 行に 1 デバイス名を記述します。
<outdir>	必須	取得した共通鍵を保存するディレクトリまでの絶対 PATH を指定します。本パラメータに指定するディレクトリは、事前に作成しておく必要があります。
<log file>	必須	エラー発生時のログを出力するファイルまでの絶対 PATH を指定します。指定したファイルが既に存在する場合は、ファイルを上書きします。

オプション	必須/任意	説明
-b	任意	取得した共通鍵のバイナリデータを保存したい場合に指定します。 詳細は、後述「共通鍵一括取得コマンドで取得した共通鍵について」をご覧ください。

実行例

[共通鍵一括取得 (Linux 版)]

```
# SWCLMKEYGET -i /tmp/devicelist -o /tmp/cmkeydir -e /tmp/errorlog
```

[共通鍵一括取得 (Widnows 版)]

```
> SWCLMKEYGET.cmd c:¥tmp¥devicelist c:¥tmp¥cmkeydir c:¥tmp¥errorlog
```

実行結果

実行結果は、共通鍵一括取得コマンドの終了コードで確認します。

正常終了時は、終了コード 0 で終了します。標準出力には以下を出力します。

```
HTTP/1.1 200.
errorcode: 0
retcode: 0
HTTP/1.1 200.
errorcode: 0
retcode: 0
```

実行中にエラーが発生した場合は、終了コード 0 以外で終了します。合わせて、エラーに関する情報を標準出力・標準エラー出力、エラーログに以下のフォーマットで出力します。

```
<HTTP ステータスコード>
<ID 鍵コマンドが出力するメッセージ>
<CLM から返却されたエラーメッセージ (CLM からエラーメッセージを返却された場合)>
<CLM から返却されたエラーコード>
<ID 鍵コマンドの終了コード>
<共通鍵一括取得コマンドが出力するメッセージ>
```

HTTP ステータスコード、CLM から返却されたエラーコード、ID 鍵コマンドの終了コード、共通鍵一括取得コマンドが出力するメッセージの詳細については、6 章をご覧ください。

共通鍵一括取得コマンドで取得した共通鍵について

共通鍵は、引数「-o」(Windows 版の場合は引数<outdir>)で指定したディレクトリに保存します。

ファイル名と保存形式は、コマンド実行時の引数の指定により異なります。

- 引数「-b」を指定しない場合
  - 共通鍵は、バイナリデータを 16 進数変換した状態(以降、16 進数変換済データと表記します)で保存します。
  - ファイル名は、「デバイス名_鍵番号」です。
- 引数「-b」を指定した場合
  - 共通鍵は、16 進数変換済データ、バイナリデータの 2 つの形式で保存します。
    合わせて、バイナリデータの改ざん検知に使用する hmac ファイルも生成し、保存します。
  - ファイル名は、次の通りです。
    - 16 進数変換済データ: 「デバイス名_鍵番号」

- ✧ バイナリデータ　　：「デバイス名_鍵番号.bin」
- ✧ hmac ファイル　　：「デバイス名_鍵番号.bin.hmac」

- 例

  デバイス名が「xxxx0000000001」、鍵番号が「1」である場合

  - ➢ 引数「-b」を指定しない場合

    共通鍵の 16 進数変換済データを「xxxx0000000001_1」というファイル名で保存します。

  - ➢ 引数「-b」を指定した場合

    共通鍵の 16 進数変換済データを「xxxx0000000001_1」というファイル名で保存します。

    共通鍵のバイナリデータを「xxxx0000000001_1.bin」というファイル名で保存します。

    共通鍵の hmac ファイルを「xxxx0000000001_1.bin.hmac」というファイル名で保存します。

ログ出力

共通鍵一括取得コマンドは、以下にログを出力します。

- 標準出力・エラー出力
- 引数「-e」（Windows 版の場合は、引数<log file>）に指定したファイル

6 終了コード・エラーコード一覧

6.1 ID 鍵コマンド 終了コード一覧

終了コードは以下の通りです。

エラーコード	説明
0	正常終了
1	コマンドのパラメータが不正
2	ファイル読み込み失敗
3	conf ファイル内パラメータ不正
4	必須パラメータが設定されていない
5	コマンドの書式が不正
6	既に実行中
7	メモリ確保エラー
8	ファイル書き込み失敗
11	取得した証明書の展開に失敗
12	取得した証明書の復号に失敗
13	証明書ストアへの取得した証明書保存に失敗
14	証明書が既に存在
15	証明書が改ざんされている
20	Socket エラー
31	ID・パスワードの照合が行われていない
32	ファイルフォーマットが不正
33	外部コマンド実行が失敗
40	ネットワークエラー
101	CLM への接続に失敗(Connection timeout)
102	CLM への接続に失敗(Connection refused)
103	暗号鍵の生成に失敗
104	認証プロトコルが不一致
105	プロトコルメッセージの送信に失敗
106	プロトコルメッセージの受信に失敗
107	メッセージの暗号化に失敗
108	メッセージの復号に失敗
129	SIGHUP によりプロセスが終了した（Linux 版 CLA 使用時)
130	SIGINT によりプロセスが終了した（Linux 版 CLA 使用時)

131	SIGQUIT によりプロセスが終了した（Linux 版 CLA 使用時)
132	SIGILL によりプロセスが終了した（Linux 版 CLA 使用時)
133	SIGTRAP によりプロセスが終了した（Linux 版 CLA 使用時)
134	SIGABRT によりプロセスが終了した（Linux 版 CLA 使用時)
136	SIGFPE によりプロセスが終了した（Linux 版 CLA 使用時)
137	SIGKILL によりプロセスが終了した（Linux 版 CLA 使用時)
139	SIGSEGV によりプロセスが終了した（Linux 版 CLA 使用時)
141	SIGPIPE によりプロセスが終了した（Linux 版 CLA 使用時)
142	SIGALRM によりプロセスが終了した（Linux 版 CLA 使用時)
143	SIGTERM によりプロセスが終了した（Linux 版 CLA 使用時)
200	HTTP エラー
201	CLM から返却された終了コード ・要求応答不正(HTTP エラー) (CLM V1.0 接続時) ・要求応答不正（CLM(限定版) 接続時)
202	CLM から返却された終了コード(認証エラー) (CLM(限定版) 接続時)
203	CLM から返却された終了コード(電文不正(復号エラー)) (CLM(限定版) 接続時)
204	CLM から返却された終了コード(電文不正(形式誤り)) (CLM(限定版) 接続時)
205	CLM から返却された終了コード(証明書要求待ちのタイムアウト) (CLM(限定版) 接続時)
206	CLM から返却された終了コード(最大接続数超過(BUSY)) (CLM(限定版) 接続時)
207	CLM から返却された終了コード(乱数不正) (CLM(限定版) 接続時)
208	CLM から返却された終了コード(製造番号不正) (CLM(限定版) 接続時)
210	CLM から返却された終了コード(CA 未構築) (CLM(限定版) 接続時)
211	CLM から返却された終了コード(クライアント証明書の生成に失敗) (CLM(限定版) 接続時)
212	CLM から返却された終了コード(CA 証明書がない) (CLM(限定版) 接続時)
213	CLM から返却された終了コード(二重発行エラー) (CLM(限定版) 接続時)
214	CLM から返却された終了コード(証明書発行履歴保存失敗) (CLM(限定版) 接続時)
255	内部エラー

6.2 ID 鍵コマンド エラーコード一覧

HTTP ステータスコードと CLM から返却されたエラーコードの詳細は、下表の通りです。

エラーコード	エラーコード返却時 HTTP ステータス	説明
1001	500	CLM の設定不正。CLM の設定を確認してください。
1002	500	設定ファイルの読み込みに失敗。CLM の設定を確認してください。
1101	400	JSON 構文エラー。ネットワークの状態を確認してください。
1102	400	必須パラメータが指定されていない。コマンドで指定したパラメータを確認してください。
1103	400	パラメータの指定方法誤り。コマンドで指定したパラメータを確認してください。
1104	400	パラメータの諸元誤り。コマンドで指定したパラメータを確認してください。
1201	403	パスワードが不一致。指定した ID・パスワードを確認し、再実行してください。
1202	403	パスワードのステータスが有効ではない。指定した ID・パスワードを確認し、再実行してください。
1203	403	パスワードが有効期限切れ。指定した ID・パスワードを確認し、再実行してください。
1204	404	データベース上に ID・パスワード情報が見つからない。指定した ID・パスワードを確認し、再実行してください。
1205	500	keyID 番号が keyID 長の最大値をオーバー。CLM の設定を確認してください。
1301	404	証明書情報が取得できない(keynumber または certserial の値が不正)。コマンドで指定した keynumber または certserial を確認してください。
1302	404	CA 局名称が不正でテーブル情報が取得できない(パラメータ catype の値が不正)。コマンドで指定した catype を確認してください。
1303	404	紐付け情報が取得できない(証明書とデバイス情報の組み合わせが不正)。コマンドで指定した keynumber や

		devicename を確認してください。
1304	403	keynumber または certserial の証明書のステータスが有効ではない。コマンドで指定した keynumber または certserial を確認してください。または、証明書の情報を確認してください。
1305	403	keynumber または certserial の証明書が有効期限切れ。コマンドで指定した keynumber または certserial を確認してください。または、証明書の情報を確認してください。
1306	403	keynumber または certserial の証明書がすでに失効または更新済み。コマンドで指定した keynumber または certserial を確認してください。または、証明書の情報を確認してください。
1307	403	newkeynumber または newcertserial の証明書のステータスが有効ではない。コマンドで指定した newkeynumber または newcertserial を確認してください。または、証明書の情報を確認してください。
1308	403	newkeynumber または newcertserial の証明書が有効期限切れ。コマンドで指定した newkeynumber または newcertserial を確認してください。または、証明書の情報を確認してください。
1310	500	証明書ファイルのアクセスエラー。CLM のログを確認してください。
1309	404	証明書ファイルが見つからない。CLM のログを確認してください。
1401	500	openssl コマンド実行エラー。CLM のログを確認してください。
1402	500	openssl コマンド実行結果の形式異常。CLM のログを確認してください。
1403	500	コマンドのロックタイムアウト。CLM のログを確認してください。
1501	404	共通鍵情報が取得できない(keynumber の値が不正)。コマンドで指定した keynumber を確認してください。または共通鍵の情報を確認してください。
1502	404	共通鍵情報が取得できない(newkeynumber の値が不正)。コマンドで指定した keynumber または newkeynumber

		を確認してください。または共通鍵を更新してください。
1503	404	紐付け情報が取得できない（keynumber と deviceid の組み合わせが不正)。コマンドで指定した keynumber や devicename を確認してください。
1504	403	keynumber の共通鍵がすでに失効または更新済み。コマンドで指定した keynumber を確認してください。または共通鍵の情報を確認してください。
1505	403	newkeynumber の共通鍵の status が有効ではない。コマンドで指定した newkeynumber を確認してください。または共通鍵の情報を確認してください。
1506	403	newkeynumber の共通鍵が有効期限切れ。コマンドで指定した newkeynumber を確認してください。または共通鍵を更新してください。
1507	400	keynumber の暗号化方式と newkeynumber の暗号化方式が異なる。コマンドで指定した newkeynumber を確認してください。または共通鍵の情報を確認してください。
1508	400	リクエスト中の共通鍵値と DB の共通鍵値が一致しない。コマンドで指定した inpath と cmkeyname を確認してください。または共通鍵の情報を確認してください。
1509	403	共通鍵のステータスが有効ではない。コマンドで指定した inpath と cmkeyname を確認してください。または共通鍵の情報を確認してください。
1510	403	共通鍵が有効期限切れ。コマンドで指定した inpath と cmkeyname を確認してください。または共通鍵を更新してください。
1511	404	共通鍵ファイルが見つからない。CLM のログを確認してください。
1512	500	共通鍵ファイルのアクセスエラー。CLM のログを確認してください。
1513	404	更新対象の鍵番号と更新後の鍵番号が一致。コマンドで指定した keynumber と newkeynumber を確認してください。
1601	500	データベース接続エラー。CLM のリポジトリの状態を確認してください。
1602	500	データベースアクセスエラー。CLM のリポジトリの状態を

		確認してください。
1603	500	カラム情報不正。CLM のログを確認してください。
1604	500	データベースのデータ異常。CLM のログを確認してください。
1701	400	deviceid が、登録済みの情報と一致しない。コマンドで指定した devicename を確認してください。
1702	400	デバイス名が既に登録済み。コマンドで指定した devicename を変更し、異なるデバイス名にしてください。
1703	403	デバイスキーが一致しない。コマンドで指定した devicename を確認してください。
1704	403	デバイスのステータスが有効でない。コマンドで指定した devicename を確認してください。
1705	403	デバイスが有効期限切れ。コマンドで指定した devicename を確認してください。
1706	404	デバイス情報が見つからない。コマンドで指定した devicename を確認してください。
1707	500	スクリプト実行エラー。CLM のログを確認してください。
1708	404	デバイスのステータスが既に無効状態。
1709	404	デバイスのステータスが既に有効状態。
1710	404	デバイスのステータスが既に削除状態。
1901	500	ログ初期化失敗。CLM のログを確認してください。
2001	500	初期化処理で異常発生。CLM のログを確認してください。
2002	500	暗号化処理で異常発生。CLM のログを確認してください。
2003	500	復号化処理で異常発生。CLM のログを確認してください。
2004	500	SecureWare/開発キットの API 呼び出しで異常発生。CLM のログを確認してください。
2005	500	軽量暗号 開発キットのコマンド呼び出しで異常発生。CLM のログを確認してください。
2201	404	ファイルが見つからない。CLM のログを確認してください。
2202	500	ファイルの読み込みに失敗。CLM のログを確認してください。
2203	500	ファイルの書き込みに失敗。CLM のログを確認してください。
2204	500	ファイルの削除に失敗。CLM のログを確認してください。

2205	500	日付のフォーマット変換（数値<->文字列）失敗。CLM のログを確認してください。
9901	500	システムエラー。CLM のログを確認してください。

6.3 共通鍵一括取得コマンド 終了コード一覧

終了コードは以下の通りです。

エラーコード	説明
0	正常終了
1	引数エラー
2	デバイス情報の参照でエラー
3	共通鍵情報の参照でエラー
4	証跡管理情報の参照でエラー
5	デバイス情報の取得でエラー
6	共通鍵の取得でエラー

6.4 共通鍵一括取得コマンド メッセージ一覧

共通鍵一括取得コマンドが出力するメッセージは、以下の通りです。

メッセージ	説明
Error: infile not specified	[Linux 版] -i オプションが指定されていません。 指定した引数を確認してください。 [Windows 版] 引数<devicename list file>が指定されていません。 指定した引数を確認してください。
Error: outdir not specified	[Linux 版] -o オプションが指定されていません。 指定した引数を確認してください。 [Windows 版] 引数<outdir>が指定されていません。 指定した引数を確認してください。
Error: errorfile not specified	[Linux 版] -e オプションが指定されていません。 指定した引数を確認してください。 [Windows 版] 引数<log file>が指定されていません。 指定した引数を確認してください。
Error: infile not found	[Linux 版] -i オプションで指定したファイルが見つかりません。 指定した引数を確認してください。 [Windows 版] 引数<log file>で指定したファイルが見つかりません。指定した引数を確認してください。
Error: outdir not found	[Linux 版] -o オプションで指定したディレクトリが見つかりません。指定した引数を確認してください。 [Windows 版] 引数<outdir>で指定したディレクトリが見つかりません。指定した引数を確認してください。
Warn：no commonkey	デバイス名に紐づく共通鍵が見つかりません。 指定したデバイス名を確認してください。

Error : outdir is reserved word. Please specify another directory.	[Linux 版] -o オプションで指定したディレクトリ名は予約語のため利用できません。 別のディレクトリを指定してください。 [Windows 版] 引数<outdir>で指定したディレクトリ名は予約語のため利用できません。 別のディレクトリを指定してください。
Error: devgetkeylist error (<エラーコード>)	デバイス鍵情報の取得に失敗しました。 エラーコードは、ID 鍵コマンドの終了コードです。 ID 鍵コマンドの終了コードと、エラーログを元に原因を取り除き、再実行してください。
Error: cmkeybatget error (<エラーコード>)	共通鍵一括取得 API に失敗しました。 エラーコードは、ID 鍵コマンドの終了コードです。 ID 鍵コマンドの終了コードと、エラーログを元に原因を取り除き、再実行してください。

6.5 イベントログ 出力メッセージ一覧

イベントログに出力するイベント ID とメッセージは、下表の通りです。

[イベントログ(Application)]

- イベントメッセージは、イベントビューアの全般タブに出力します。
- 詳細を確認したい場合は、イベントビューアの詳細タブをご覧ください。

イベント ID	イベントメッセージ		ログレベル	ソース
	日本語	英語		
1001	SecureWare/CLA コマンドは正常に終了しました。	SecureWare/CLA command completed successfully.	Informational	swclmclient
8001	SecureWare/CLA コマンドがエラーを返しました。	SecureWare/CLA command returned an error.	Error	swclmclient