CryptOneX Business

for MacOSX

Ver. 01.0e 2021

増田和行

(SRI)構造改革研究所

「本ソフトウェアは技術的制限手段を違法に回避するものではありません。」ここに明記致します。

**目次**

はじめに

インストール

ユーザ名とパスワード

環境変数

コードブック

スケジュールファイル

実行形態のインストール

CryptOneXの実行

ログ

暗号通信と公開鍵

アルゴリズム

連絡先

**はじめに**

**はじめに、には変更情報など重要情報を記述しています。必ず読んで下さい。/etc/servicesと/etc/hostsの登録は必ず行ってください。登録には配布tarファイルの展開ディレクトリ下のSample.services、Sample.hostsを参照してください。LogServXの正しいインストール、環境設定が済めばあとはログへのエラーメッセージから全ての正しい方向への修正ができるでしょう。LogServXおよびClientx.bashのインストール中はTRC\_\*.traceファイルを参照してください。**

**CryptOneX 01.0eでは環境変数CRYPTONEX\_DATA\_SIZEの入力で少数点数の利用が可能となりました。また都合により01.0dからの上位互換性は保証できていません。上位互換性の保証は再度再開します。**

**CryptOneX 01.0dは01.0cからの上位互換性を保証します。旧バージョン(01.0c)の暗号文は01.0dで平文に復号可能です。CRYPTONEX\_DATA\_SIZEを設けユーザデータサイズのチェックを可能としました。Mbyte単位で指定して下さい。**

　CryptOneX 01.0cでは**乱数のアリゴリズムに変更がありました。従来のCryptOneXでは復号ができません。バージョンアップには注意して下さい。そのためパスワードファイルの作り変えが必須です。本バージョンからは./INSTALLの必要はありません。**また**コードブックの有効期間を指定するために環境変数CRYPTONEX\_DATE\_LIMTを設け、「開始日/有効期間」を指定する**ことにしました。**PKeyRM.bashを追加しました。CRYPTONEX\_PKEY\_PATHに指定されたディレクトリ下のコマンド引数に指定した有効期日を過ぎた公開鍵は自動的に削除**します。PKGen.bashと合わせてご利用下さい。

　CryptOneX 01.0bではエラーメッセージの修正を行いました。完璧な修正ではありませんが、パドランプの利用を考えた修正です。パドランプの点灯にはLogServerXで管理する文言”Error”のあるメッセージの検出が便利です。またクライアント／サーバ間のプロトコルを変更しました。**本バージョンでは./INSTALLの必要はありません。同一LogServX下で100セットのCryptOneXが動作可能です。それ以上の利用はお控えください、動作の保障ができません。CryptRouterXは環境設定（スケジュールファイル）さえ間違わなければ何段でも利用可能です。**

　CryptOneX 01.0aではアルゴリズムに変更がありました。実行コマンドPKGen.bashに変更を加え、公開鍵の性能を高め、管理ファイルFRTOMAN.datを生成することになりました。また、RGNRouterXを作成しました。ClientX.bash、およびRGNRouterX.bashの変更および追加が発生しています。スケジュールファイルRGNSched.datも追加になりました。/etc/services、ならびに/etc/hostsにRGNRouterX用の情報の追加が必須です。RGNRouterXの利用には環境変数CRYPTONEX\_RGN\_NAME(ClientX.bash)、CRYPTONE\_SERV\_TYPE(RGNRouterX.bash)の設定が必須です。

　CryptOneX 01.08ではパスワードコマンドにチーム名が追加になりました。

　CryptOneX 01.07ではLogServXに変更があります。インストール時刻が同一なCryptOneXサーバは同時には動かなくなりました。Trial版では、インストール時刻が異なれば1ログサーバ当たり最大3 CryptOneXサーバまでの管理が可能です。更に01.06の配布済みソフトを廃止しました。Transposition版では暗号アルゴリズムは転置の１段のみとなりました。

　CryptOneX 01.06ではIOLogとInspec.bashの配布を始めました。Inspec.bashを起動するとLogServXで管理するマシングループでの起動CryptOneXプロセス数が表示できます。crontabコマンいードにてCron.datを登録しInspec.msgをご覧下さい。

　尚、cronの停止にはrootユーザにてログインし/usr/lib/cron/tabs下の ”ユーザ名” ファイルを消去せねばなりません。

　CryptOneX 01.05ではサーバログの修正がありました。クライアントマシンでアクセスしたファイル名をログ出力できるようにしました。また、クライアントのサーバ接続を環境変数CRYPTONEX\_LOGONの値(0 or 1)に応じて出力できるようにしました（起動シェル参照の事）。

　CryptOneX 01.04ではテストシェルの変更、および乱数生成アルゴリズムを変更しました。

　CryptOneX 01.03 ではスケジュールファイルの簡単な構文チェックコマンドを添付しました。また、暗号アルゴリズムも4段としました。一段と解読が難しくなることでしょう。

　CryptOneX 01.02からはCryptOneXのインストールが必須になりました。./INSTALLコマンドを利用してtarファイルを展開後、展開ディレクトリのインストール(./INSTALL)を行ってください。また、簡単なVernam onetime padによる回線暗号を導入に成功しました。通信中の秘密が一応守れるでしょう。

　サーバシステムの起動にはStart.bashを利用し、停止にはStop.bashを利用してください。/etc/hosts、/etc/services、への情報登録もお忘れなく。

　CryptOneX 01.01はCryptOne ver.7.02の拡張中に作成されたバージョンで、アルゴリズムの詳細に変更がありました。暗号化、復号アルゴリズムは2段となり、冗長性を持たせられなくなりました。

　実行環境にも大きな変更がありました。共用メモリの利用を止め、/tmpのサイズや多重度の指定が不要となりました。

　環境変数CRYPTONEX\_CODE\_PATH下の全てのディレクトリ下の全てのコードブックを自動的に利用することも可能となりました。詳しくはコマンド使用法(CryptOneXの実行)を参照して下さい。

　以下、目次に沿って解説して行きます。

**インストール**

　CryptOneXをインストールするには CryptOneX.tar.Zを解凍します。

1. uncompressコマンドにてファイルを伸張しtarファイルを作成します。
2. tar xvfにて当該ディレクトリ下にデータを展開します。
3. ./INSTALLコマンドの実行（実行形態のインストール参照)。

全コマンドがカレントディレクトリに展開されます。以下のコマンド及びディレクトリがカレントに展開されているのを確認してください。CryptOneXは環境変数さえ正しく設定できればルータ無しでも完全な動作が可能です。

16\_digit　 鍵ファイル（コードブック）を作るコマンド

CCCDPSSched PSSched.dat、CDSched.datの構文チェックコマンド

CCRGNSched RGNSched.datの構文チェックコマンド

CCRouterX Schedule.dat の構文チェックコマンド

CODE　 コードブックの格納ディレクトリ

CDServX コードブックサーバのコマンド

CDServX.bash コードブックサーバの起動コマンド

ClientX クラアントの起動コマンド

ClientX.bash クライアント機能の起動コマンド

CryptOneX　 CryptOneXのコマンド

CryptOneX.bash　CryptOneXの起動コマンド

CryptRouterX CryptOneXルータコマンド

CryptRouterX.bash CryptOneX ルータの起動コマンド

DOC CryptOneX説明書格納ディレクトリ

DragonAll.bash　コードブック（.CODE/g0下の）生成コマンド

DragonZ.bash　 コードブック（.CODE/g0/Z下の）生成コマンド

Kill シグナル送信コマンド(Stop.bashで利用する)

LosServX ログサーバのコマンド

LosgServX.bash ログサーバの起動コマンド

PASS　 パスワードファイルの格納ディレクトリ

PKey 公開鍵の格納ディレクトリ

PSServX パスワードサーバのコマンド

PSServX.bash パスワードサーバの起動コマンド

Passwd　 パスワード作成コマンド

Passwd.bash　 パスワード作成シェル記述

RGNRouterX リージョンルータのコマンド

RGNRouterX.bash リージョンルータの起動コマンド

SCHD スケジュールファイルの格納ディレクトリ

Start.bash CryptOneXシステムの起動コマンド

Stop.bash CryptOneXシステムの停止コマンド

dumdum　 Hexダンプコマンド

gtest.dat　 テスト用データファイル

PKGen.bash　 公開鍵生成コマンド

test1.sh　 テスト用コマンド

test2.sh テスト用コマンド

以上のコマンド/ディレクトリ群が展開されることを確認してください。

**ユーザ名とパスワード**

　システムを正常に動作させるにはユーザ名とパスワードの登録が必要です。Passwd.bashを起動してユーザ名とパスワードを設定しましょう。パスワードとユーザ名がPASSディレクトリ下に生成されます。

使用法は

Passwd.bash ＜チーム名＞　＜ユーザ名＞　＜パスワード＞　{Creat | Check}

です。Creat はパスワードの作成を Checkはパスワードの確認を行います。

　多数のユーザ名とパスワードを登録するには上に示したコマンドのリストを記述してシェルファイルを作成してください。リスト順にユーザ名とパスワードのデータが生成されます。この際、重複のチェックは一切やってないので前もって重複のチェックを行ってください。UNIXのソートコマンドも利用すると便利でしょう。このパスワードは簡単に破られるかもしれません。

**環境変数**

　現在使用可能な環境変数については配布コマンド(\*.bash)の環境定義を参照してください。以下に簡単な説明を行います。

環境変数の説明

Start.bash起動コマンドの環境変数

CRYPTONEX\_CODE\_HOST=コードブックサーバのホスト名

CRYPTONEX\_CODE\_PATH=コード（鍵）ファイルの所在(./CODE)

CRYPTONEX\_CODE\_SERV=コードブックサーバのサービス名

CRYPTONEX\_DATE\_LIMT=開始日/有効日数

CRYPTONEX\_LOG\_HOST=ログサーバのホスト名

CRYPTONEX\_LOGON=1はaccept log オン

CRYPTONEX\_LOG\_PATH=ログファイル格納ディレクトリ(./.)

CRYPTONEX\_LOG\_SERV=ログサーバのサービス名

CRYPTONEX\_SERV\_NAME=CryptOneXのサービス名

CRYPTONEX\_PASS\_HOST=パスワードサーバのホスト名

CRYPTONEX\_PASS\_PATH=パスワードファイルの所在(./PASS)

CRYPTONEX\_PASS\_SERV=パスワードサーバのサービス名

CRYPTONEX\_PKEY\_PATH=CDServX で利用するコードプックの所在

CRYPTONEX\_ROUT\_TYPE=RGNRouterXを指定するとRGNRouterXの機能を実現(RGNRouterX.bash)

CRYPTONEX\_SCHD\_PATH=スケジュールファイルの所在(./SCHD)

CRYPTONEX\_ROUT\_SERV=CryptOneXルータのサービス名

Client.bashの環境変数

CRYPTONEX\_HOST\_NAME=CryptOneX、あるいはルータのホスト名

CRYPTONEX\_RGN\_NAME=リージョン名

CRYPTONEX\_SERV\_NAME=CryptOneX、あるいはルータのサービス名

の定義が必須です。各コマンドシェルを参照し必要に応じて変更してください。リージョンルータを利用しない時には環境変数をコメントにして下さい。メッセージが出てきませんので変更には注意して下さい。

**コードブック**

　鍵データファイル（コードブック）は./CODE/g0、g1下に26 ボリューム（A,…,Z）、10ブック、各ブックは0~9999の一万ライン用意しました。起動コマンドの-vパラメータにピリオド「.」の一文字を入力する(-v.)と全コードブックのデータを有効に利用できます。ディレクトリ（ボリューム）名の追加には注意してください。全く新しいコードブックとなってしまいます。DragonZ.bashの改造で鍵情報の世代管理も可能なので試してみましょう(DragonAll.bashも配布中)。

　ボリューム名は最大63文字まで利用できます。世代管理のため、および個人の特殊な利用法のためにこの制限としました。十分な考慮の上ご利用ください。尚コードブックファイルを新しいものに変えてしまうと、前世代のデータを保存してない時には復号が困難になりますのでコードブックの書き換えには十分な注意を払いましょう。コードブックはフォーマットさえ合致すれば、ユーザが個人用に自由に作成可能です。お試しください。

スケジュールファイル

　スケジュールファイルはルータでの負荷分散のためのSchedule.datと多数のユーザのパスワードを管理するためのPSched.datm、、コードブック用のCDSched.datなどを用意しました。以下にファイルフォーマットを例示して説明に変えることにしましょう。CryptOneでのスケジュールファイルから発展して２個のファイルが必要になりました。

ルータマシン上でのスケジュールファイル(Schedule.dat)

#

# サンプルデータ

#

begin;

    machine =   localhost,  SimpServ;

    machine =   localhost,  SimpServ;

    machine =   localhost,  SimpServ;

end;

#

#

#

　LANを使用する場合、localhostにはCryptOneXサーバのホスト名を、SimpServにはCryptOneXサーバのサービス名を記述してください。何行でも記述できます。

　CryptOneXサーバでのパスワード用スケジュールファイル(PSSched.dat)

#

#   Sample file

#

b e g i n ;

    machine =Team1, localhost,  PassServ;

e n d ;

#

#   end of file

　このファイルではコメントも#に続けて記入できます。記述の意味はTeam1のパスワード検索用のパスワードサーバのホスト名をlocalhostとしサービス名称をPassServとする、です。ホスト名とサービス名をカンマで続けて記入してください。同様にコードブックサーバ、リージョンルータのスケジュールファイル(CDSched.dat、RGNSched.dat)も記述して下さい。

**実行形態のインストール**

**実行に先立ってインストールを行わねばなりません。Tarファイルを展開したディレクトリにある INSTALLコマンドを実行しましょう。OSXが用意したコマンドとは違うので ./INSTALLと入力して実行しましょう。**

* **./INSTALL**

**./を忘れるとOSXコマンドのINSTALLの実行が始まります。必ず注意 ./ をつけて下さい。して行ってください。**

**CryptOneX実行**

　コマンドは2種類です。CryptOneXシステムの起動コマンドはStart.bashで、停止はStop.bashで行いましょう。クライアント起動はClientX.bashです。公開鍵を取得するためと暗号化の本体機能を起動するコマンドがClientX.bashです。

　まずStart.bashはCryptOneXシステムの起動を行います。各サーバの起動コマンドは実行時のコマンドプロンプトをみてください。コマンド名とソケットバッファのサイズをKbyteで指定します。内部の各起動コマンドはバックグラウンドで動作します。

　ClientX.bashで使用するリージョン名（環境変数にて指定）、チーム名、ユーザ名、パスワードは63文字まで入力可能です。メールアドレスなど一意に決定できるユーザ名を入力しましょう。チーム名、ユーザ名、パスワードは前もって登録が必要です（ユーザ名とパスワード、Passwd.bashを参照）。

　また、ClientX.bashの使用法は

ClientX.bash –e –t<チーム名> –n<ユーザ名> -s<パスワード>

-v<ボリューム> -b<ブック> -l<行番号> <ファイル名>

あるいは

ClientX.bash –d –t<チーム名> –n<ユーザ名> -s<パスワード>

-v<ボリューム> -b<ブック> -l<行番号> <ファイル名>

あるいは

ClientX.bash –g –t<チーム名> –n<ユーザ名> -s<パスワード>

-v<ボリュームc> -b<ブック> -l<行番号>

<公開鍵ファイル名>

あるいは

ClientX.bash –p –t<チーム名> –n<ユーザ名> -s<パスワード>

-k<公開鍵ファイル名>.pk <ファイル名>

の4種類が利用可能です。-e は暗号化のコマンドで、-dは復号のためのコマンドです。-gは公開鍵を取得するためのコマンドで、ここで指定した公開鍵ファイル名には拡張子 (.pk)が自動的に付加されます。-pは公開鍵を利用した暗号化のためのコマンドです。ClientX.bash -pでの公開鍵ファイル名には明示的に拡張子(.pk)を付けてください。-vアーギュメントの値<ボリューム>に一文字ピリオド「.」を指定すると環境変数CRYPTONEX\_CODE\_PATH下の全てのディレクトリの全コードブックを自動的に利用することができます。<ボリューム>に一致する文字を入力すると先頭から一致するディレクトリ名下のコードブック(各10冊づつ)を利用することができます。

**ログ**

　実行中に発生したエラー情報は一部LosServerXとsocketによるconnect、スケジュールファイルの構文チェック、ClientXのエラーメッセージに限って /tmp/TRC\*.traceに出力されます。その他の情報はCRYPTONEX\_LOG\_PATH下のディレクトリにCryptOneX.log.yymmddの名称のファイルで出力します。ほとんどの障害はこの情報で修復できるはずです。障害を回避できないときは下記連絡先にご連絡下さい。

**暗号通信と公開鍵**

　CryptOneXは暗号ツールですが、その目的は公開鍵を利用した暗号通信にあります。CryptOneと異なり、公開鍵もClientX.bash -gを使ってCryptOneXが生成します。以下手短に公開鍵を利用した暗号通信の実現手順を記述します。

　ローカルな送信元では以下の手順で公開鍵ファイルを送信します。

1. ClientX.bash -gを起動し公開鍵ファイルを生成します。
2. メールツールを起動し、暗号通信で返信をいただく準備をします。
3. メール本文に暗号ソフト CryptOneXを利用してこれから送信する公開鍵で必要な暗号文書を作成し、返信するようお願いを書きましょう。
4. 公開鍵ファイルを添付し、宛先に送信しましょう。

　これで公開鍵が宛先に送信されました。次にリモートでは公開鍵を利用して要求のファイルを暗号化しましょう。

1. ClientX.bash –p を利用して返信する添付文書ファイルを暗号化しましょう。
2. 暗号化文書をメールに添付し返信しましょう。

　リモートから返信されたメールを開き、添付文書を取り出し復号します。

1. ローカルの送信元では暗号化されている添付文書をダウンロードします。
2. 暗号文書をClientX.bash –dを利用して復号しましょう。

　これで暗号通信ができたことになります。ツールの開発が可能になればこれらの手順はソフトウェアに実装されることでしょう。

**アルゴリズム**

**現在使用しているCryptOneXのアルゴリズムは以下の通りです。**

1. **Transposition**

**バイト単位のTransposition**

**可変長ビット単位のTransposition**

**１レコード長は最大１０２４バイト（Force版では 2048バイト）です。**

1. Vernam one time pad

　ワンタイムパッドによる**ストリーム暗号**

1. コード変換

　単純なコード変換

**最大１０２４バイト読み込みの度にTrasposition、コード変換テーブルを生成し直します。**

**連絡先**

　連絡は下記までお願いいたします。

[k.masuda.52431@gmail.com](mailto:k.masuda.52431@gmail.com)