CryptOneT

for MacOSX

Ver. 1.01 2016/10/11

Ver. 2.0a 2018

増田和行

(SRI)構造改革研究所

**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

CryptOneTの実行

暗号通信と公開鍵

アルゴリズム

連絡先

はじめに

　CryptOneT ver.02.0aでは実行コマンド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。

　CryptOneT ver.02.08では実行コマンドにチーム名が追加になりました。パスワードコマンドの変更です。

　LogServX.bash にてログサーバを起動してください。/etc/hosts、/etc/services、への情報登録をお忘れなく。

　CryptOneT ver. 01.01はCryptOne ver.7.02の拡張バージョンCryptOneZのサブセットで、アルゴリズムの詳細に変更がありました。応答速度はCryptOneに比して速く、0.45sec/MB以上の応答速度を実現しています。この性能はMacBookによるものですがディスクオリエンテッドなiMacよりも速いようです。暗号化、復号アルゴリズムは1段で、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。

　ver.01.02では環境変数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(CryptOneT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**

　CryptOneTをインストールするには CryptOneT.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ファイルを生成します。
2. tar xvfにて当該ディレクトリ下にデータを展開します。

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。

16\_digit　鍵ファイル（コードブック）を作るコマンド

CODE　コードブックの格納ディレクトリ

CryptOneT　CryptOneTのコマンド

CryptOneT.bash　CryptOneTの起動コマンド

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

PASS　パスワードファイルの格納ディレクトリ

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

PKey　公開鍵の格納ディレクトリ

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。その際/tmp/TRC\_\*.traceへの出力メッセージに注意してください。

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。この暗号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

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

CRYPTONEX\_PASS\_PATH=パスワードファイルのディレクトリ(./PASS)

の定義が必須です。各コマンドシェルを必要に応じて変更してください。

**コードブック**

　コードブックは暗号化のための鍵情報です。A~Zまでの26ボリューム、0~9の10ブック、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。ディレクトリ名の追加には注意してください。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。

**CryptOneTの実行**

　コマンドは１種類です。起動コマンドは公開鍵を取得するためと暗号化の本体機能を起動するCryptOneT.bashです。障害発生時にはエラーメッセージが/tmp/TRC\_\*.traceに出力されます。

　以下で使用するユーザ名、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。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、CryptOneT.bashの使用法は

CryptOneT.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

CryptOneT.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

CryptOneT.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

CryptOneT.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。こちら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。

**暗号通信と公開鍵**

　CryptOneT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CryptOneT -gを使ってCryptOneTコマンドで生成します。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. CryptOneT.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 CryptOneT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。次にリモートの宛先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. CryptOneT.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。

　リモートの宛先局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CryptOneT.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CryptOneTのアルゴリズムは以下の通りです。**

1. **Transposition.**

**バイト単位のTransposition**

**１レコード長は最大１０２４バイトです。**

**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)