テキスト暗号化ツール

Chanel

利用ガイド

第 1 版

タシャカネルラボ

目次

■免責事項・注意事項（必ずお読みください）■

●免責事項

Chanel は暗号化プログラムですが、HSP でのアルゴリズム実装が主目的の実験的プログラムの単純な集積体でしかなく、HSP で本格的暗号アルゴリズムが実装できることを紹介したものに過ぎません。**製品化された有料の暗号化プログラムのようには安全性が考慮されていない事をご理解**ください。

例えば攻撃に対する安全性の観点では、ウィルスや悪意のあるコードを埋め込まれたアプリなどによるメモリ内読み取り、意図しない画面キャプチャなどへの対策は講じられていません。

品質の安全性の観点では暗号化したデータが解読できないといったことが考えられます。そういった事例はテスト中発生していませんが、あらゆる条件を考慮したテストデータを用いたわけではないため、暗号化・解読が絶対に可能であるとは限りません。そのため、**文書データの消失が起こりうる前提で、暗号化した文書が解読できることを確認した上で暗号化前の文書を削除するか、削除せずに保持しておくなどしてご利用**ください。

商用利用への制限は設けませんが、**機密漏洩・暗号文の解読不能などの事故・不利益が発生する可能性があることをご認識の上でご利用**ください。また商用利用を考えた場合、処理速度が非常に遅く、実用性がないことをご認識ください。

商用利用・個人利用などの利用形態を問わず、**Chanel の利用にあたって生じたすべての不利益において、作者は一切の責任を負いません。**あくまでも**自己責任**でのご利用となります。

●注意事項

暗号技術は戦略物資の扱いとなります。現在日本では、日本で独自開発された一定の強度を超える暗号技術の海外持ち出しは「外国為替および外国貿易法」に基づく輸出許可又は役務取引許可が必要になる

ことを除いて暗号技術を制限なく利用・配布することができますが、今後の法改正により暗号技術の取り扱いが制限されることもあり得ます。

そのような場合、法律に従った扱いをしてください。

Chanel は日本で独自開発された暗号技術を使っていませんので、2022 年 8 月現在規制対象外です。

本項を読んだか読まなかったかを問わず、Chanel を実行した時点で、利用者は上記に同意したとみなされます。

1.はじめに

暗号化ツール「Chanel」はテキストの暗号化と解読を行うツールです。いくつかのセキュリティアルゴリズムを組み合わせた暗号化スイーツとなっており、PGP の仕組みを取り入れています。暗号化といっても適当に考えた独自アルゴリズムとか単純 XOR ではなく、ソビエト・ロシアで開発された本格的なアルゴリズムを実装しています。

開発中のコードは Rusalka (RUssian Security Algorithm LinKAge)といいました。Rusalka とはロシアの民間伝承に出てくる水の精霊のことですが、今回 Rusalka のイメージでフィリピン人トランスジェンダーの妻（といっても正式な結婚はできませんが）Chanel のちょっとセクシーな画像の使用許可と名前の使用許可が出たこともあり、Chanel（CryptgrapH Algorithm NExus Latch)を正式名称としました。プログラムを見てお分かりと思いますが、プログラムアイコンのちょっとセクシーなポーズを取っている女性が私の妻 Chanel です。

1） 本書の使い方

余計な説明はいいから、とにかくすぐ使いたい：

「2．Chanel 利用の準備」と「3．一人で Chanel を利用する」をお読みください。他の人と暗号文でメッセージをやりとりする場合は「4．他の人と暗号文をやりとりする」もお読みください。

登録したパスワードを変えたい：

「5．パスワードを変更するには」をお読みください。

登録したパスワードを忘れてしまった：

「6．パスワードを忘れてしまったら」をお読みください。

全体的にどんな機能があるのか知りたい、使ってみたい：

「2．Chanel 利用の準備」から「9．その他の機能」までをお読みください。

Chanel で使われている暗号技術や仕組みを簡単に知りたい：

「１０．Chanel で使われている用語について」と「１１．Chanel の仕組みについて」をお読みください。

暗号技術全般についてもっと知りたい：

申し訳ありませんが、本書の範囲を超えますので、暗号技術や情報セキュリティの本を探してください。
暗号技術に限って言えば、『暗号技術入門　秘密の国のアリス』結城浩著 SB クリエイティブ発行がおすすめです。現在最新版の第 3 版が出ています。
第 3 版でも 2015 年と古いのですが、基礎的なところは大きく変わっていないので問題はないでしょう。
あ、別に宣伝広告費をもらっているわけではありませんよ。

２）　Chanel をつくったきっかけ

2007 年か 2008 年の頃でしたが、ED という暗号化ツールのヘルプを見て、ED が使っている暗号化アルゴリズムの一つがソ連で開発された GOST 24187-89 である事を知りました。
元々ミリタリー系のボードゲームでよくソ連軍やロシア軍をプレイしていた私は、なんとなくあるだろうと思ってはいたソ連・ロシアの暗号化アルゴリズムの具体的な名前に触れて興味をいだき、その仕様を色々調べ始めた事がきっかけです。

PGP についてはその途上『暗号技術入門　秘密の国のアリス』という書籍で知りました。
当初の考えでは GOST 24187-89 が HSP3 で実装できたら万歳で終わるつもりだったのですが、その後色々と調べたアルゴリズムは HSP3 で実装できるのではないかと思いたち、試行錯誤を繰り返して（数学が超不得意だったので、それこそ逆数ってなに？　どうやって実装したらいいの？　といったレベルで調べて）一通りのアルゴリズムが HSP3 で実装可能だと分かり、実際に実装してみました。
そうなると、一連の実装をつかって何かツールができないかと欲を出して、思いついたのが PGP の仕組みを使ったツールを HSP3 で作り上げる、ということでした。
ただ実際には PGP の知識はこの『暗号技術入門　秘密の国のアリス』から得た知識だけで、本物を使

い倒したわけではなく細かい仕様は分からないので、Chanel は HSP3 による、あくまでも PGP の仕組みを使ったツールです。ユーザインターフェースも全く異なりますし、もちろん PGP そのものとの互換性もありません。

それから十数年、一通りの形はかなり以前の 2012 年か 2013 年頃にはできあがって、たまにちょっとずつ手を入れたりして使っていたのですが、ある時事件が……

実は私、かなり遅くまで 32 ビット版 Windows XP を使っていたのですが、2015 年に PC を買い替えて 64 ビット版 Windows10 にしたら、Chanel（当時は Rusalka）が全く動かなくなったのです。アイコンをダブルクリックしても何も出ず。再コンパイルすれば動くかと試したけどダメ。起動した直後に落ちているような現象でした。買い替えた PC に仮想環境を作って試したところ、Windows XP では旧 PC でも仮想環境でも動くのに、Windows7 から動かない事がわかりました。

しょっちゅう使っていたものでもなく、大切なファイルを暗号化していたわけでもなく、特に困らなかったので、きっと Win32api の仕様が Windows7 から変わったんだろうな、面倒くさいからいいやと、いい加減な推測だけしていったんは諦めそのまま放置していたのですが、今年 2022 年に入ってどうしても原因を見つけたいと突然思い立ち、stop 命令を入れながらどこでおかしくなっているのか探し、ようやくみつけたのが、たった一行の return を書き忘れるというバグでした……

コントロールに色をつけるためにブラシを返すところの return が一か所だけ抜けていたのです。

逆にどうして Windows XP では動いていたのかが疑問でしたが、ともかく動くようになったのでめでたしめでたし、となりました。

そこでせっかく直ったのであれば、毎年開催されている HSP コンテストで、HSP3 でも本格的な暗号化アルゴリズムが開発可能だと知ってもらいたいと思い、この度コンテストに応募することにし、色々動作を見直したり手直しした結果がいまお手元にある Chanel です。

色々Google で探してみたのですが、理論上可能であることを示唆しているサイトはありますが、本格的な暗号化アルゴリズムを HSP で実装している例は無さそうでした。

3） Chanel でできること

・自分の「安全な連絡先」を作成します（基本は初回のみ）。

・「安全な連絡先」を利用して、暗号化したテキストを解読するパスワードを直接受け渡しする必要がない暗号文を作成できます。

・解読した文章が信頼できるかどうか判定します。

・「安全な連絡先」（暗号化・解読するための重要な情報）を管理できます。

・他の人の「安全な連絡先」を追加できます。

・自分と他人の「安全な連絡先」をファイルに出力できます。

・他人の安全な連絡先に「保証」を与えることができます。

・他人の「安全な連絡先」を削除することができます。

・Chanel を使うためのパスワード変更が可能です。

・Chanel を使うためのパスワードを忘れた場合、パスワードや Chanel で使っている重要なデータが漏れたと思われる場合に再度パスワードを登録しなおす事ができます。

※ただし、それ以前の暗号文を解読することはできなくなります。

4） Chanel でできないこと

・テキストエディタ風ですが、簡易的なテキストエディタ機能しかありません。

あくまでも文章の暗号化と解読を行い、文章を読んだり編集したりする場合はメールツールや、他のエディタにコピー・ペーストして行う前提です。

画面サイズを大きくすることもできません。

・テキストファイルを開いたり、保存したりすることはできません。

・ファイル単位の暗号化・解読はできません。あくまでも画面に入力されたテキストだけです。

・非常に長い文章の暗号化・解読は、できないわけではないですが非常に時間がかかります。

・メーラーではないので、暗号文の送信機能はありません。

暗号文を送る場合は、メーラーにコピー・ペーストして送ってください。

・「安全な連絡先」や暗号文に「保証」を別々に付与する事はできません。

・「安全な連絡先」の無効化証明書を発行する機能はありません。

・　「安全な連絡先」の信頼度をユーザが設定する機能はありません。

・

5）　Chanel で利用している暗号技術

GOST 24187-89：

旧ソビエト連邦時代に開発された共通鍵暗号アルゴリズム。キー長 256 ビット、ブロックサイズ 64 ビットでラウンド回数 32 回のファイステル構造体。Chanel では CBC モードで利用。

GOST R 34.11-94：

ロシア連邦で開発された 256 ビットの不可逆変換関数(ハッシュ関数)アルゴリズム。

Production Ready の CryptPro S-BOX で実装

GOST R 34.10-2001：

ロシア連邦で開発された、楕円曲線を用いるデジタル署名アルゴリズム

楕円 Elgamal 公開鍵暗号：

公開鍵暗号の Elgamal を楕円曲線上で計算するアルゴリズム

桁上がりなし Range Coder 圧縮アルゴリズム：

Michael Schindler により考案された圧縮アルゴリズム「Range Coder」をロシア人の Dmitry Subbotin が、桁上がりが発生しないように修正したアルゴリズム

別名「ロシア人民のための Range Coder」

Ansi X 9.17 疑似乱数生成アルゴリズム：

暗号用の疑似乱数を生成するためのアルゴリズムで、独自にミリ秒単位の日時とマウス位置で初期値を設定し、より真正乱数に近くしている。

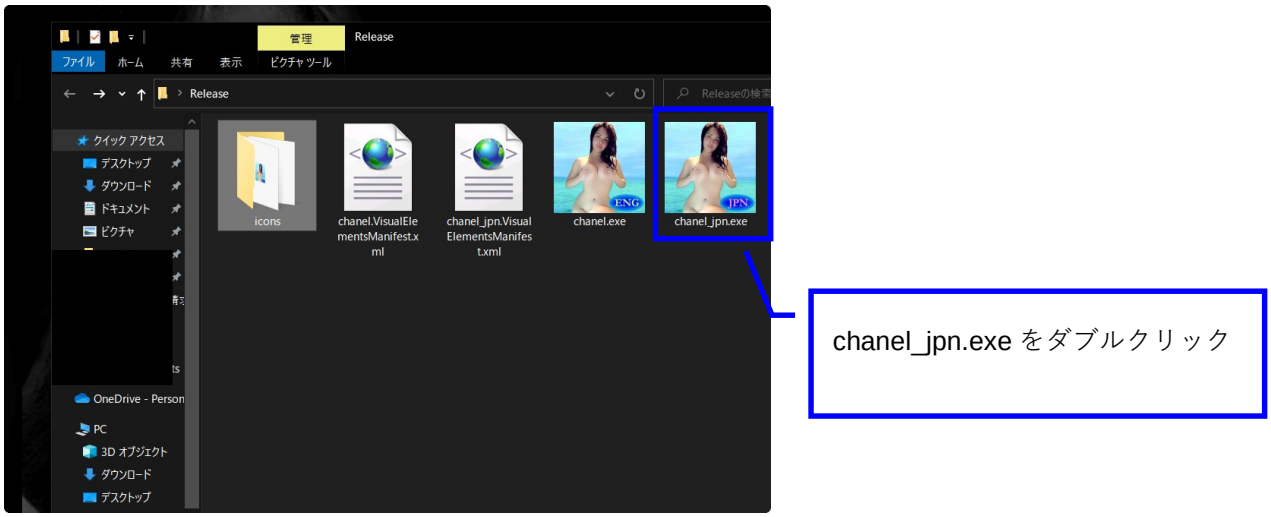
2.Chanel 利用の準備

1） Chanel 利用ユーザの「安全な連絡先」作成

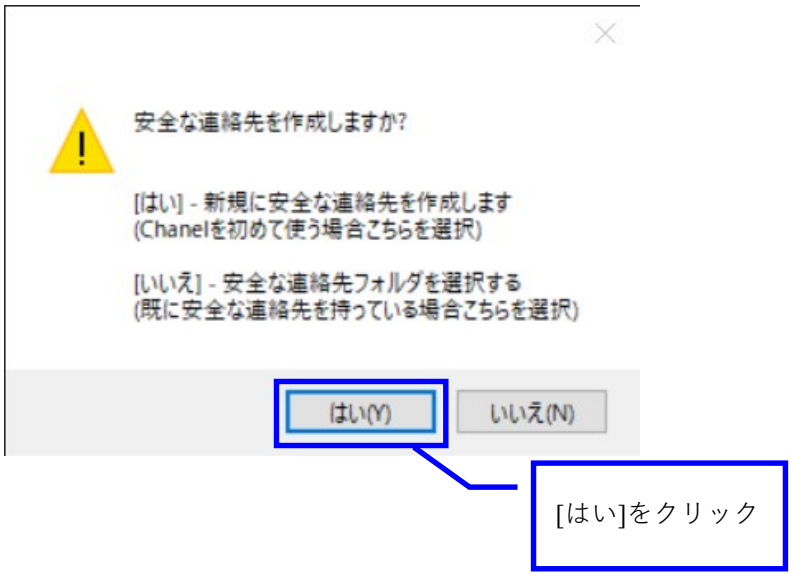
初回起動時、必ず「安全な連絡先」を作成します。

プログラムを解凍したフォルダで chanel_jpn.exe をダブルクリックします。

※Chanel.exe もありますが、英語版となりますので chanel_jpn.exe の使用をおすすめします。



[安全な連絡先を作成しますか？]ダイアログの[はい]をクリックします。



[安全な連絡先を作成]ポップ画面で必要な情報を入力します。

あなたの E メール：

本当のメルアドでなく、適当にでっち上げたメルアドでも構いません。

※ただし他の人との間で暗号をやりとりする場合は他の人と重複しないようにしてください。

あなたの名前：

これも本名である必要はありません。

※ただし他の人との間で暗号をやりとりする場合は、あなただと分かるニックネームなどの方がよいでしょう。

パスワード：

特に制限はないですが、他の人が推測しにくい方がよいでしょう。

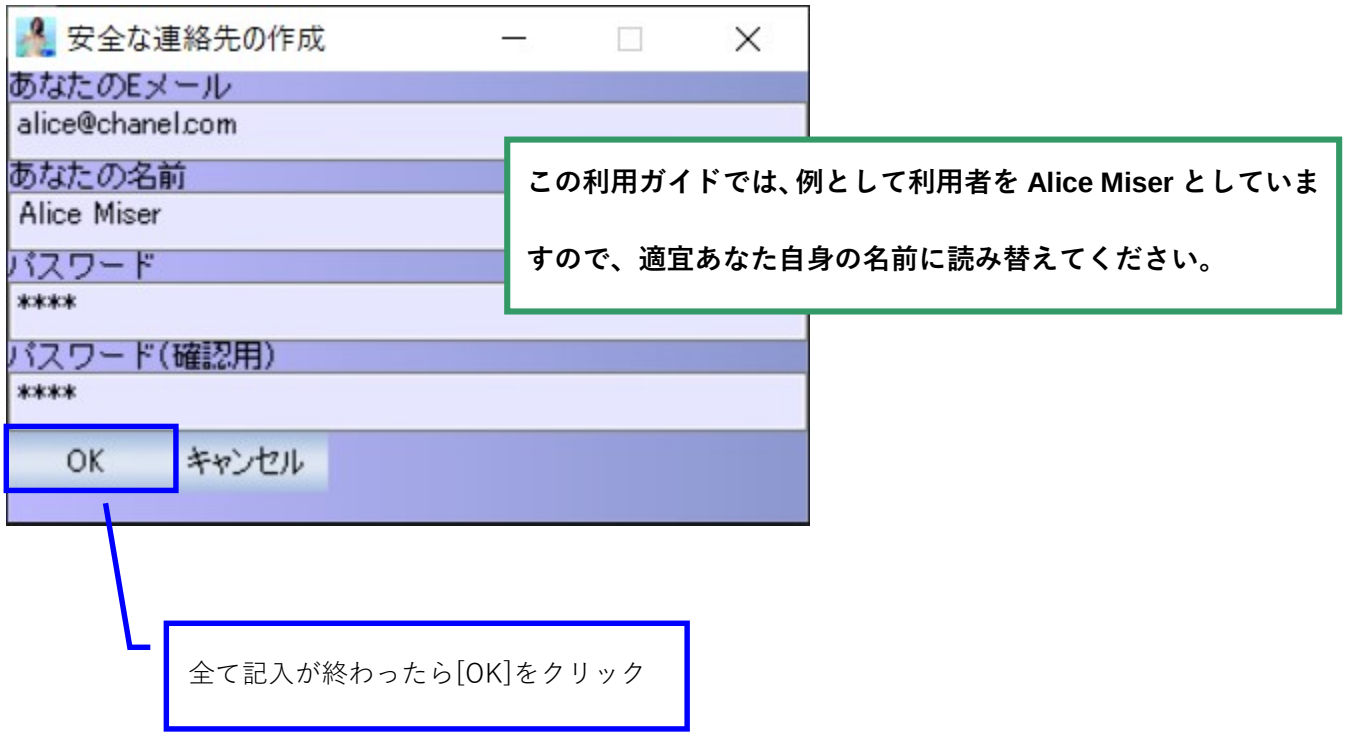
※パスワードは絶対に忘れないように管理してください。

パスワードを忘れた場合、パスワードを再度設定できますが、それ以前に暗号化した文章がすべて解読不能となり二度と読む事ができなくなります。

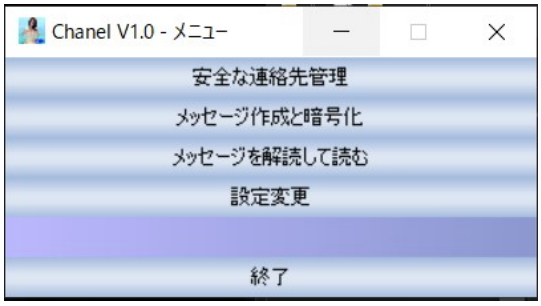
パスワード（確認用）：

パスワード欄と同じパスワードをもう一度記入してください。

全ての記入が終わったら[OK]ボタンをクリックしてください。



Chanel のメニュー画面が開きます。



以上で Chanel 利用ユーザの「安全な連絡先」作成は完了です。

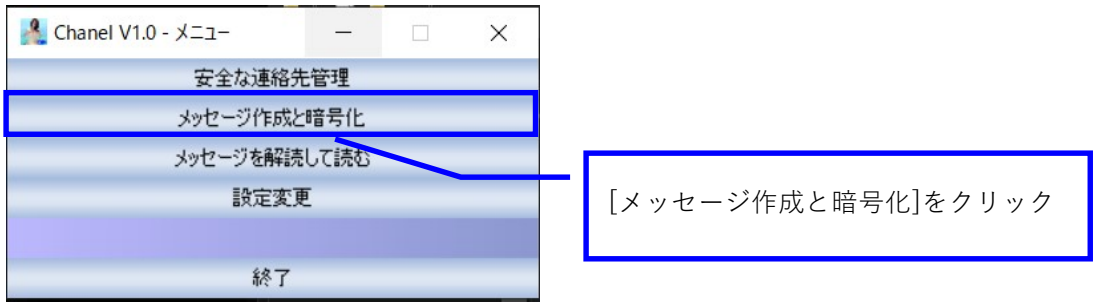
3.一人で Chanel を利用する

ここでは、テキストの暗号化・解読を自分だけでおこなう方法を説明します。例えば秘密の日記を書いて家族にみられないようにする、家族に内緒でこっそりと小説を書いて応募するので誰にも読まれたくない、などといった使い方です。

1） 文章の暗号化

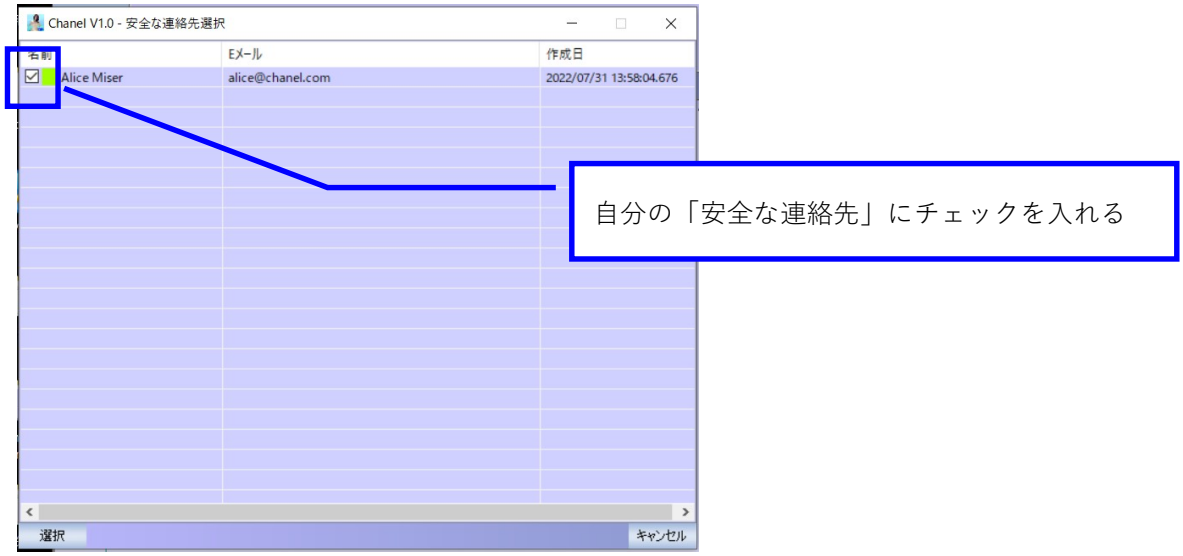
ここでは文章を作成して暗号化する手順を説明します。

メニューの[メッセージ作成と暗号化]ボタンをクリックします。



暗号化画面が開くので、[宛先]ボタンをクリックします。

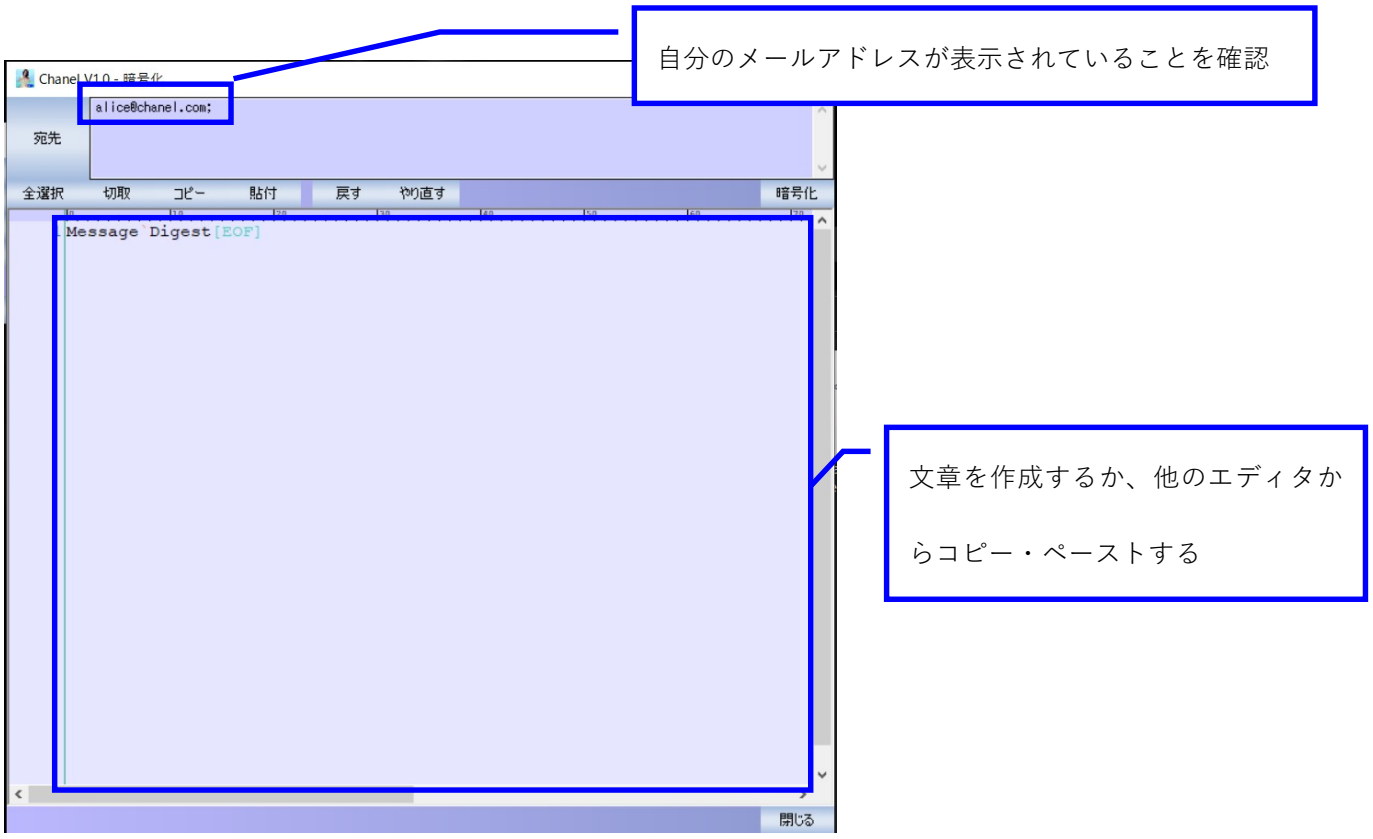
[安全な連絡先選択]画面が開くので、チェックボックスをクリックして自分（黄緑色の四角がついている連絡先）を宛先として選びます。



自分の連絡先にチェックを入れたら[選択]ボタンをクリックします。

[選択]ボタンをクリックすると、[安全な連絡先選択]画面は閉じられます。

[暗号化]画面の宛先表示部に自分のメールアドレスが表示されていることを確認したらテキスト編集部に文章を作成するか、他のエディタからコピー・ペーストします。



※簡単な編集機能は上部のボタンを使う事ができ、また下記の一般的なショートカットキーが使えます。

全選択：Ctrl + A

切り取り：Ctrl + X

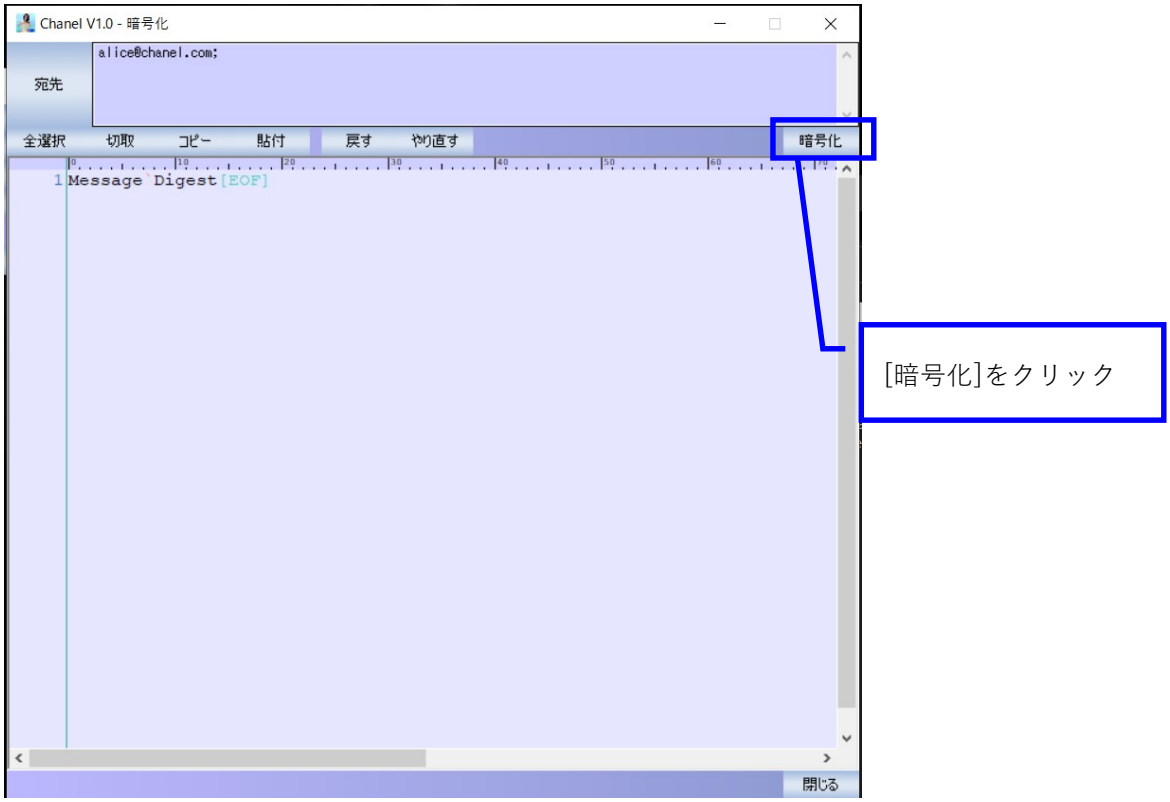
コピー：Ctrl + C

貼り付け：Ctrl + V

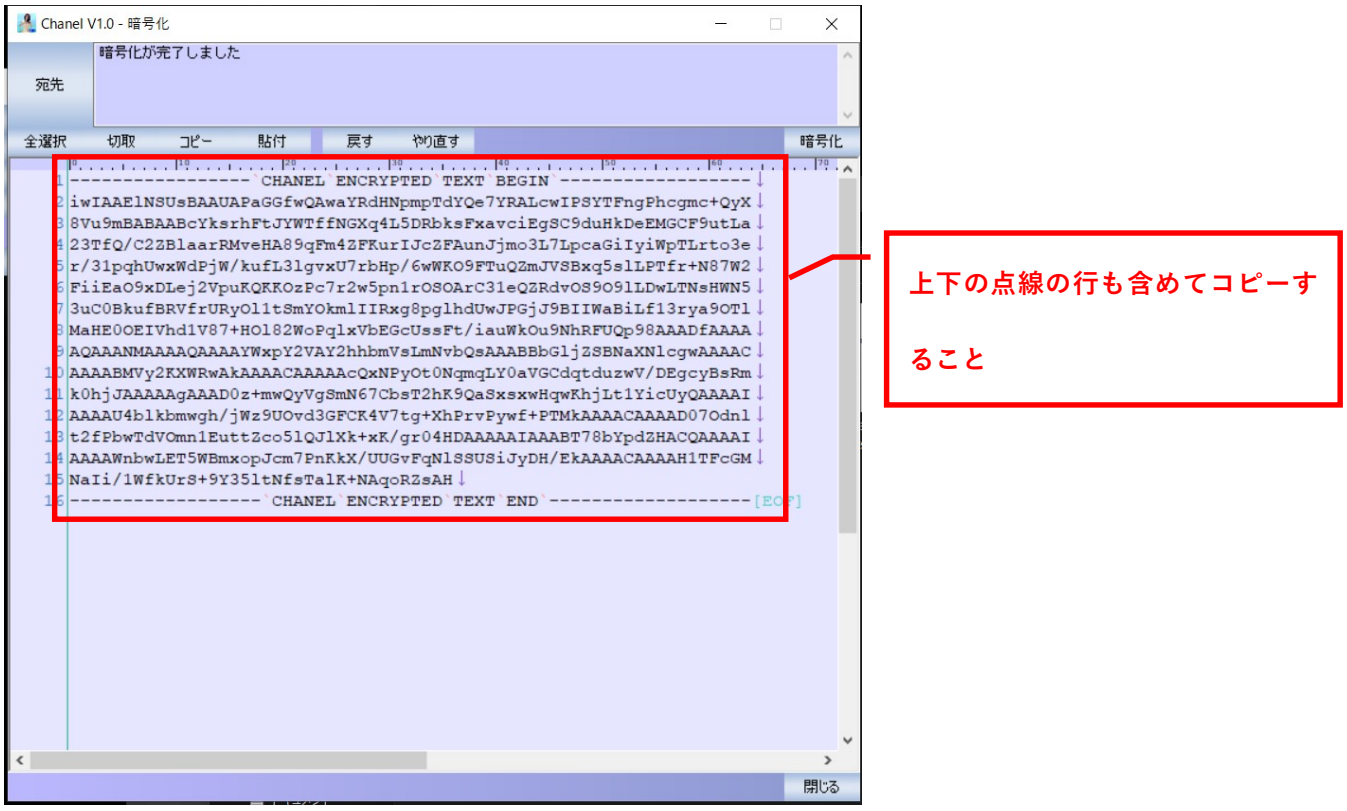
戻す：Ctrl + Z

やり直す：Ctrl + Y

[暗号化]ボタンを押して作成した文章を暗号化します。



文章が暗号化されます。



暗号化された文章はコピーして、テキストエディタなどに貼り付けて保存してください。

コピーするとき、暗号文の上下にある、点線付の「CHANEL ENCRYPTED TEXT BEGIN」と「CHANEL ENCRYPTED TEXT END」と記載されている行も含めてコピーしてください。

この２行が無いと解読できなくなります。

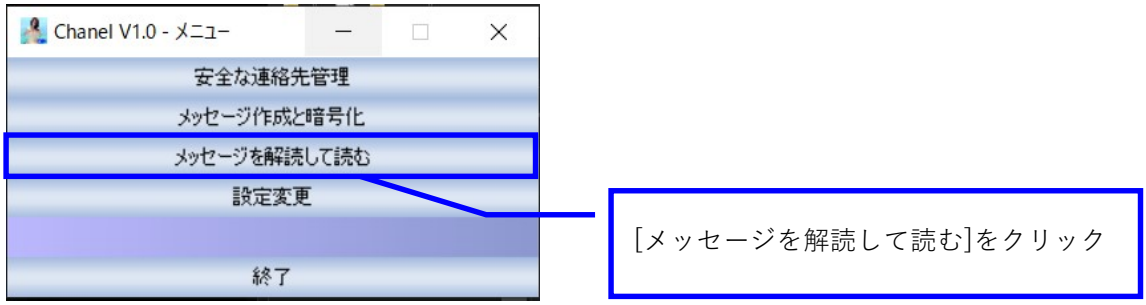
2） 暗号文の解読

ここでは、暗号化された文章を解読する手順を説明します。

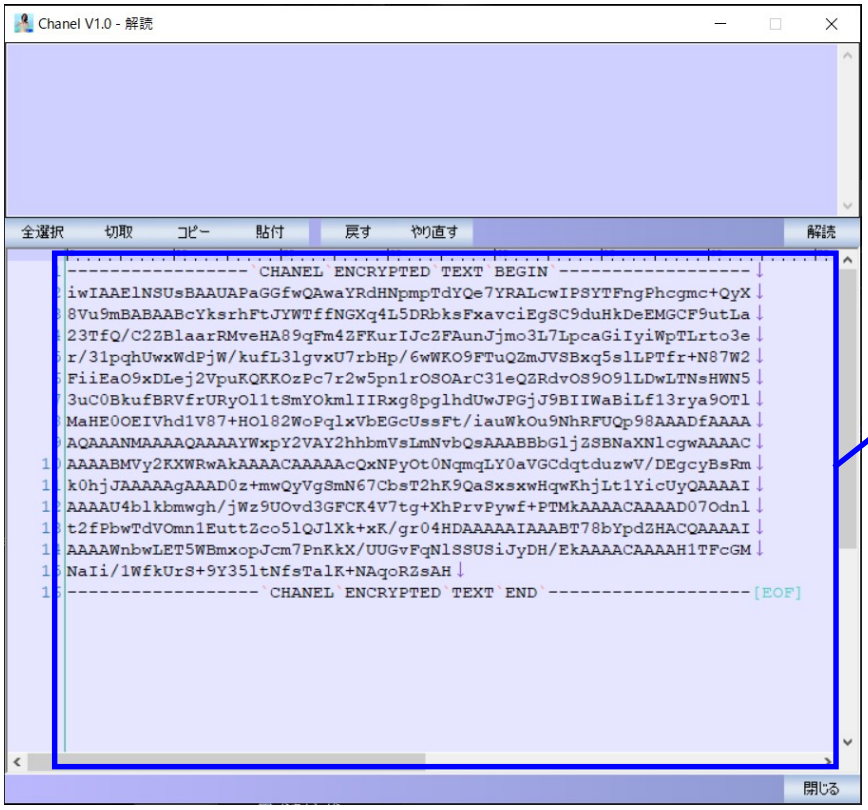
※ 正しくは暗号文をパスワードを使って元の文章に戻す事を「復号」といい、「解読」とはパスワード等を知らない場合に様々な方法でアタックして無理やり解読する事をいいます。ただ一般的に「復号」は使われず「暗号」といえば「解読」なので Chanel では「解読」を使います。

なお、「暗号化」との文字上の対称性のため「復号化」という表現を用いる場合もあります。

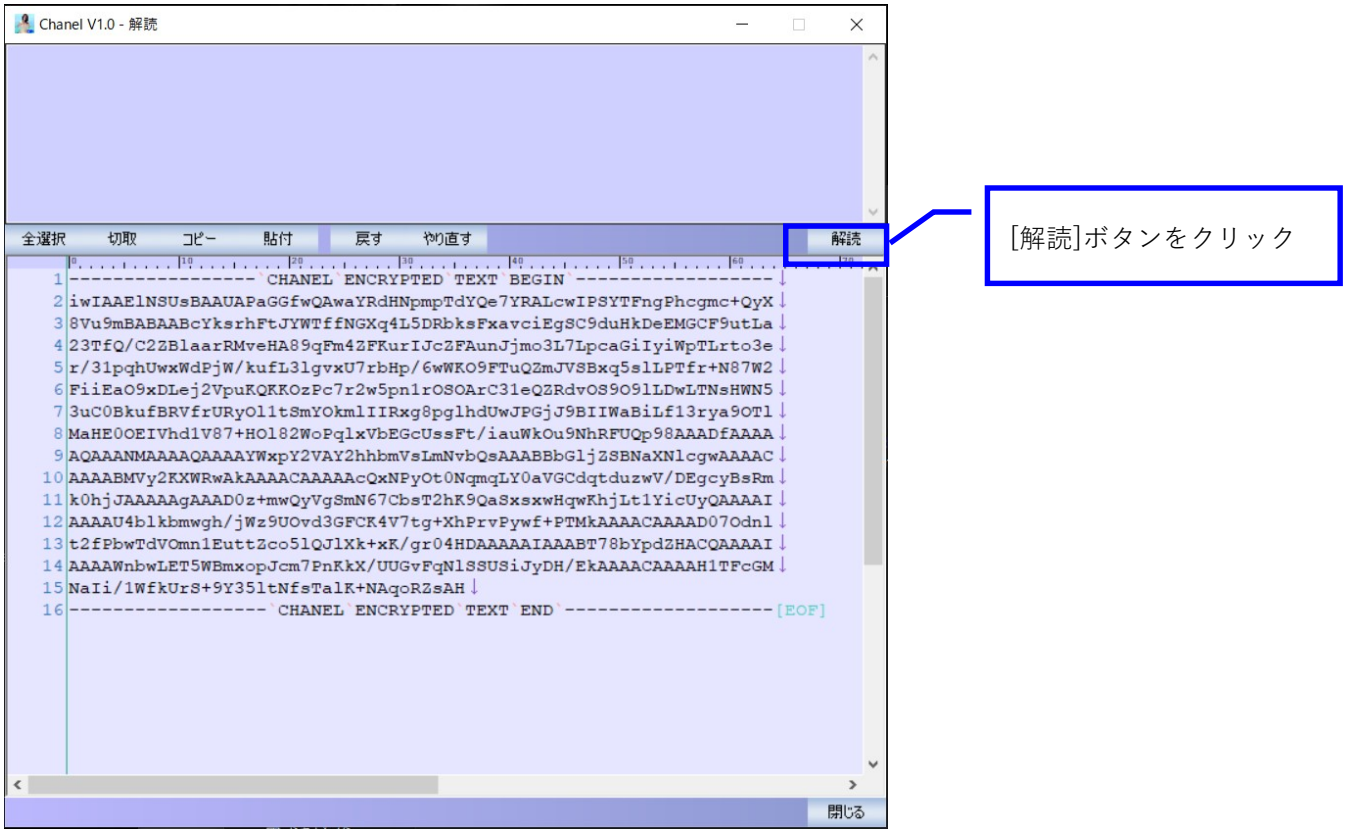
メニューの[メッセージを解読して読む]をクリックします。



[解読]画面が開くので、テキスト編集部分に暗号文を貼り付けます。

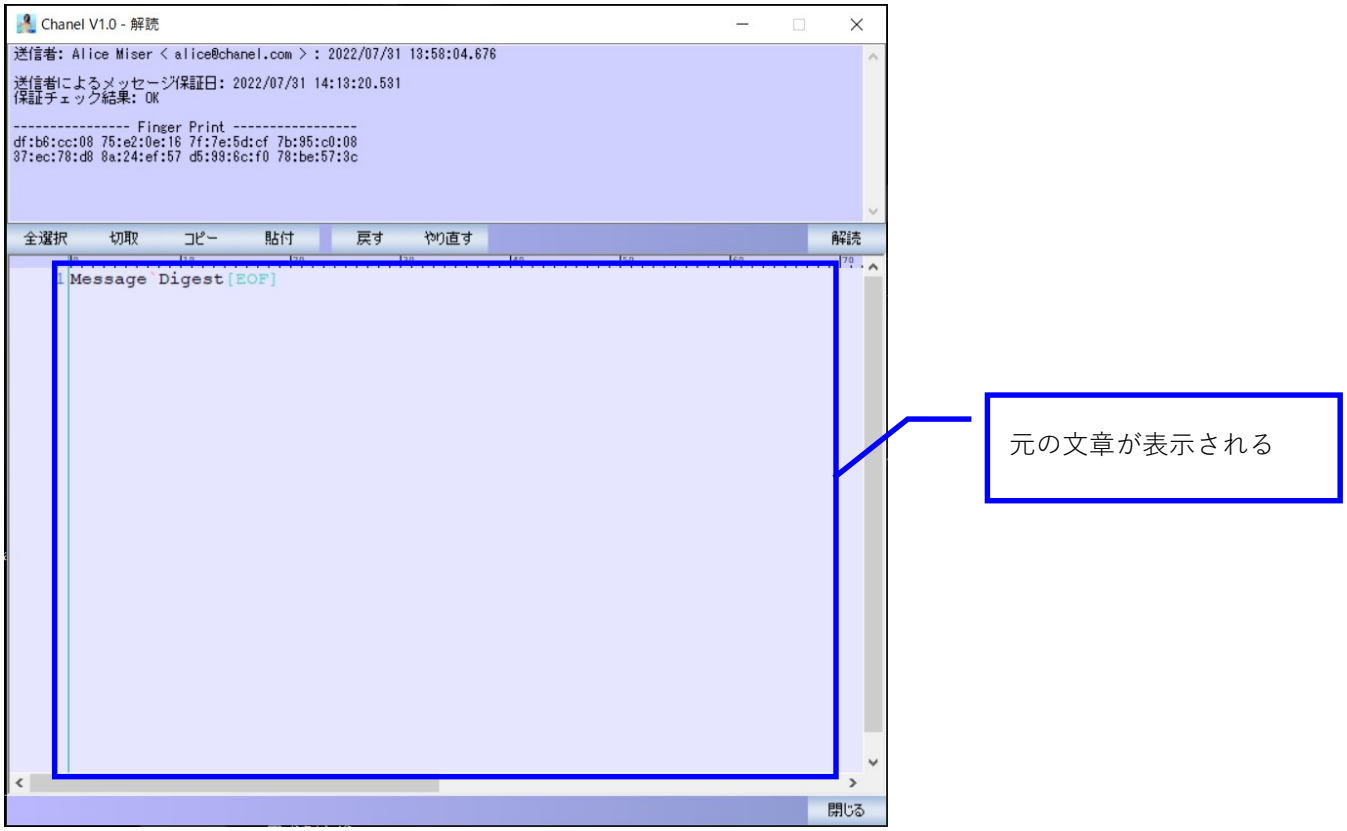


[解読]ボタンをクリックします。



[解読]ボタンをクリック

暗号文が解読されて、元の文章が表示されます。



一人で Chanel を利用する方法は以上です。

4.他の人と暗号文をやりとりする

ここでは他の人と暗号文をやりとりする方法を説明します。

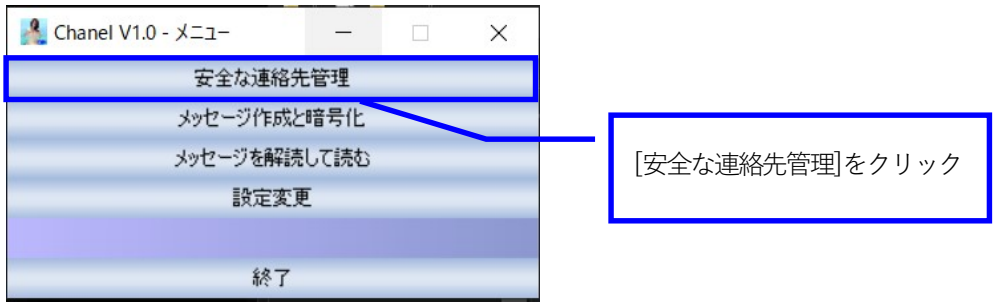
例えば友人や恋人との間で秘密のメッセージを暗号化して送りあうような使い方です。

言うまでもなく、Chanel を使って暗号化したメッセージを他の人とやりとりするためには、相手も Chanel を持っている必要があります。

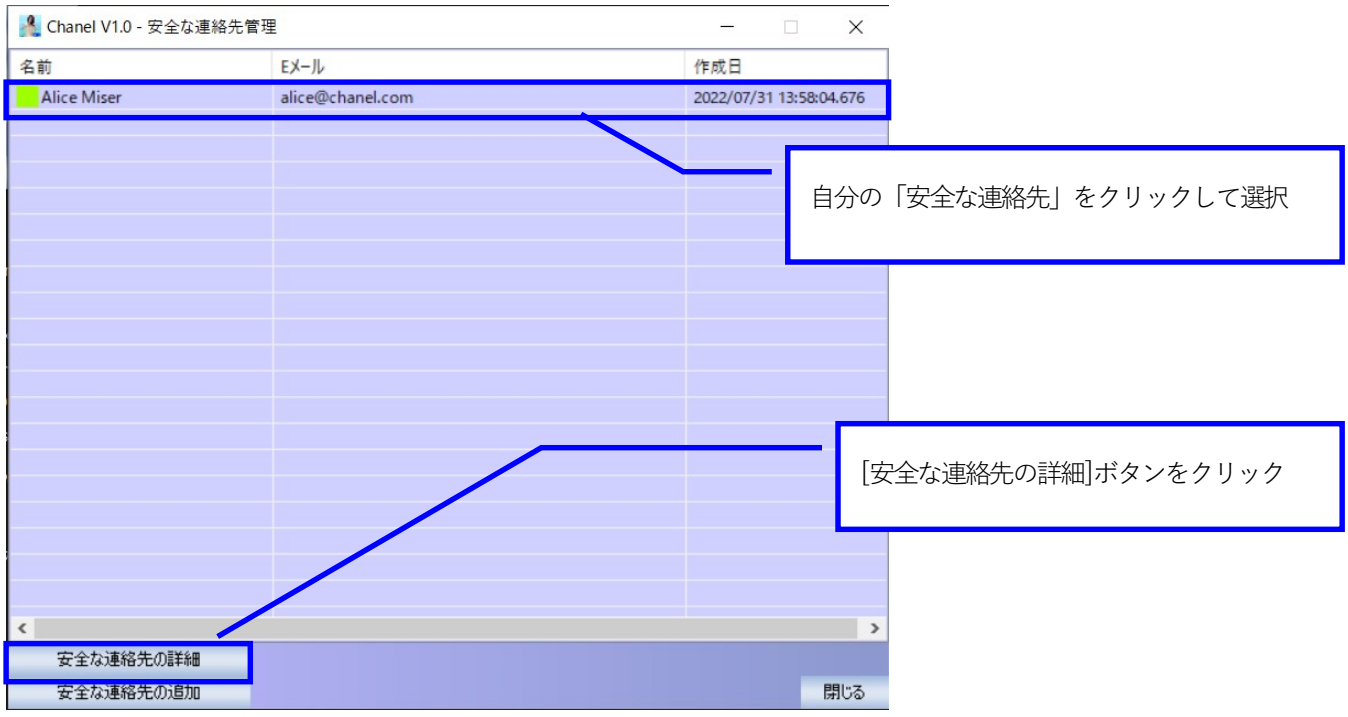
1） 他の人と「安全な連絡先」を交換する

まずはあなたの「安全な連絡先」を相手に渡すために、ファイルに出力します。

メニューの[安全な連絡先管理]をクリックします。



「安全な連絡先管理」画面が開くので、自分の「安全な連絡先」をクリックして選択し、「安全な連絡先の詳細」ボタンをクリックします。ここでもまた自分を Alice とします。

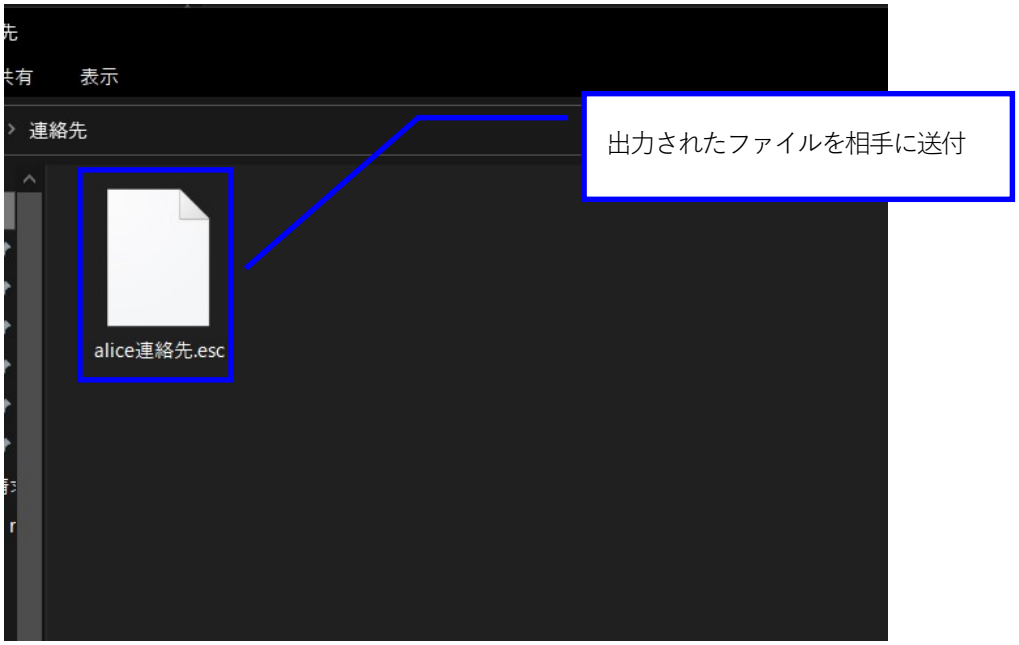


[安全な連絡先の詳細]画面が開くので、[安全な連絡先をファイルに出力]ボタンをクリックします。

[名前を付けて保存]ダイアログが開くので、任意のファイル名(拡張子は「.esc」)でファイルを保存します。

例ではファイル名を Alice 連絡先.esc としています。

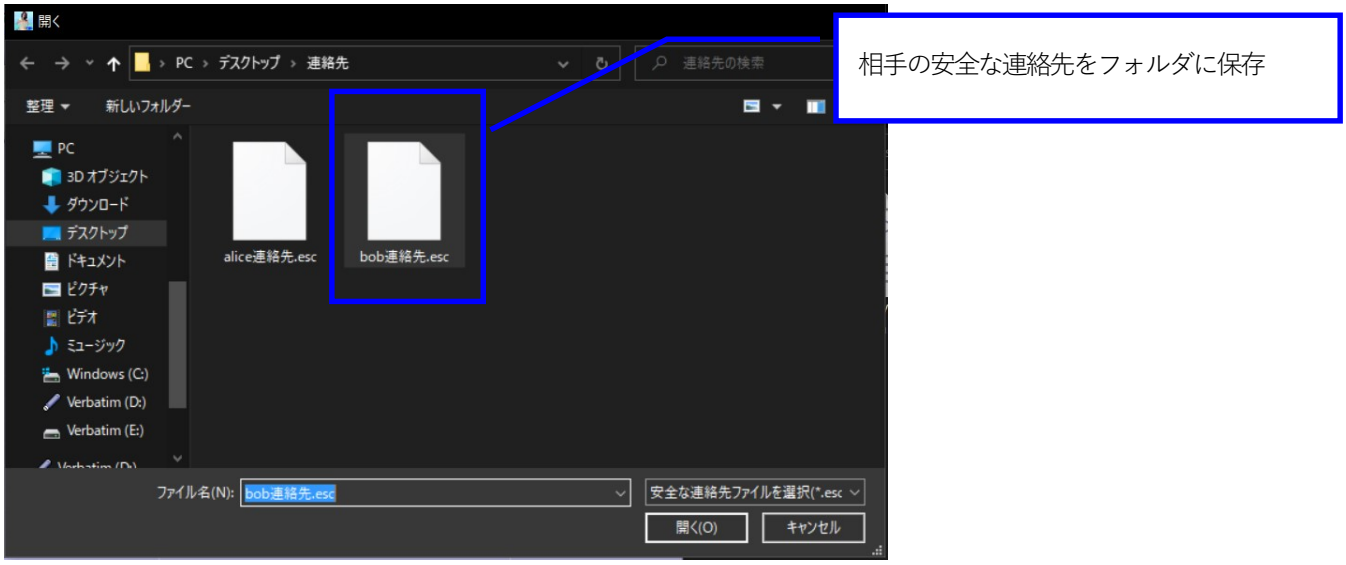
「安全な連絡先」がファイルとして出力されます。このファイルを暗号でメッセージをやりとりする相手に USB メモリに入れて渡すなり、E メールに添付するなどして送ってください。



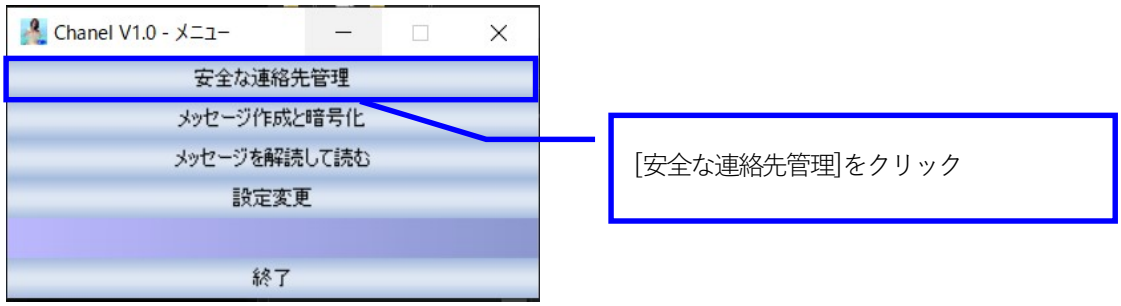
次に、相手の「安全な連絡先」を Chanel に登録します。

ここでは例として、Alice が Bob と暗号でメッセージをやりとりする事とします。

まず Bob の「安全な連絡先」を入手して、任意のフォルダに保存します。

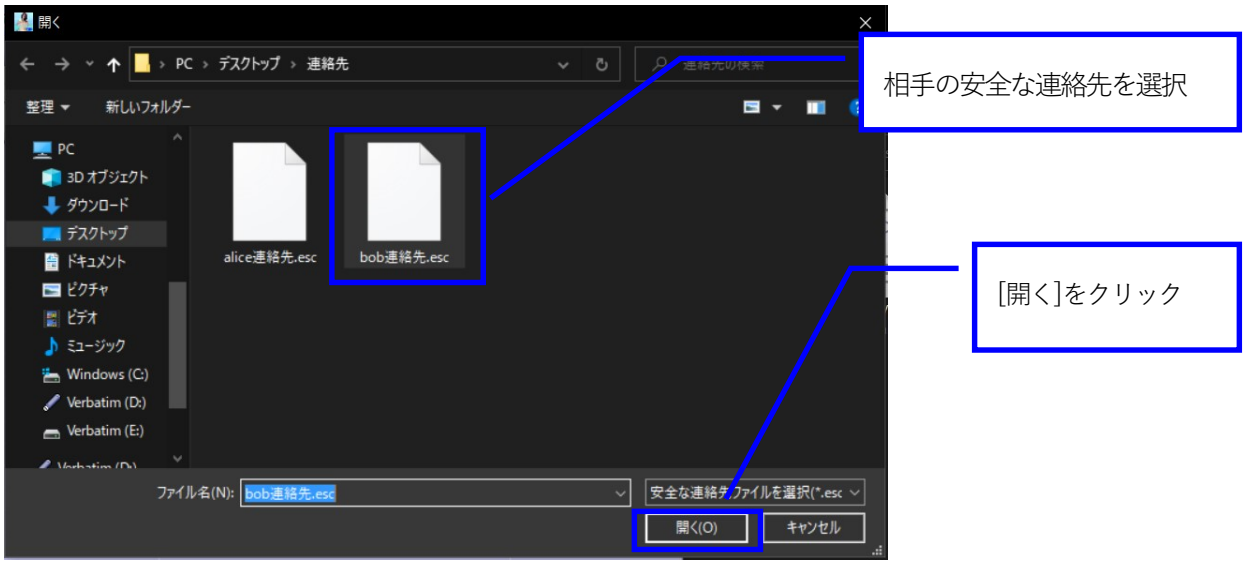


メニューの[安全な連絡先管理]をクリックします。



[安全な連絡先管理]画面が開くので、「安全な連絡先の追加」ボタンをクリックします。

ファイルの選択ダイアログが開くので、追加する「安全な連絡先」（ここでは Bob の安全な連絡先）を選択して[開く]ボタンをクリックします。



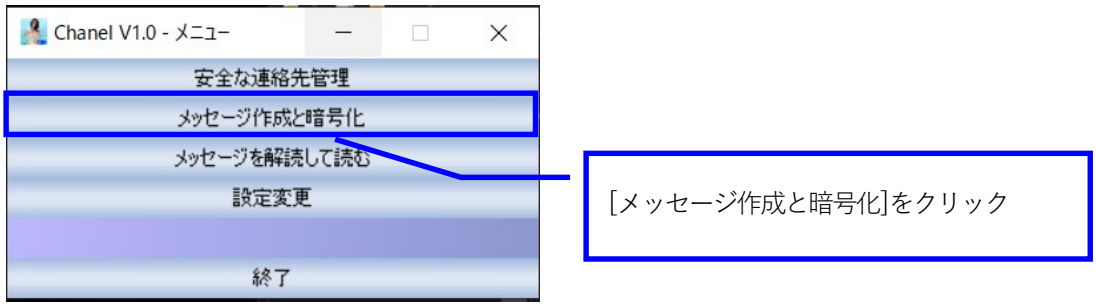
相手の「安全な連絡先」が追加されます。

これで暗号化したメッセージを相手とやりとりする準備ができました。

2） 他の人の連絡先を含めた文章の暗号化

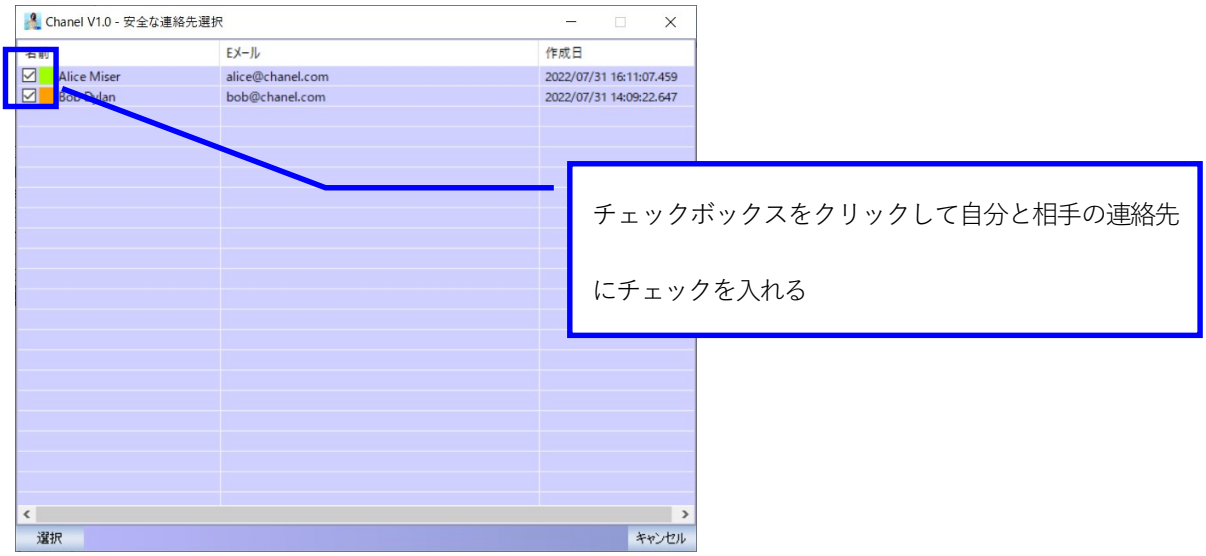
ここでは文章を作成し、メッセージをやりとりする相手を宛先に含めた暗号化する手順を説明します。手順としては、一人で Chanel を利用する場合の暗号化の手順とほとんど同じです。

メニューの[メッセージ作成と暗号化]ボタンをクリックします。



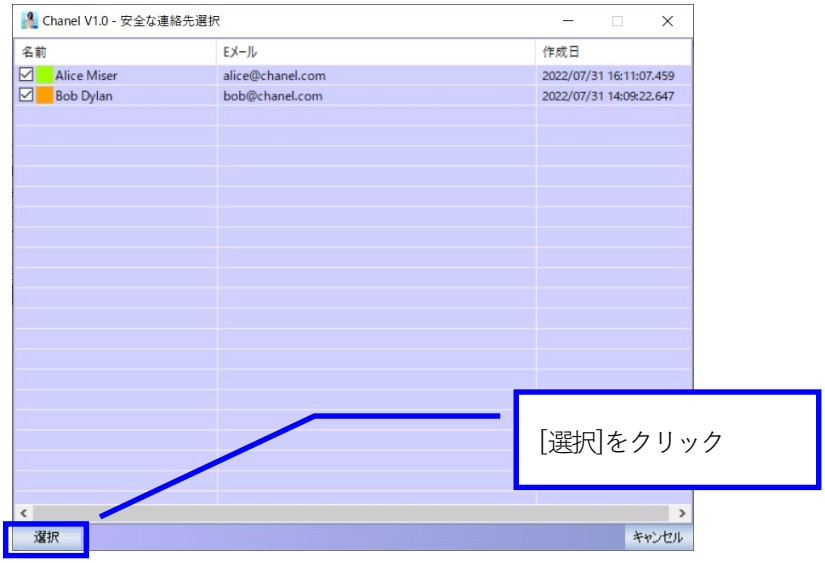
暗号化画面が開くので、[宛先]ボタンをクリックします。

[安全な連絡先選択]画面が開くので、チェックボックスをクリックして自分（黄緑色の四角がついている連絡先）とメッセージを送る相手を宛先として選びます。



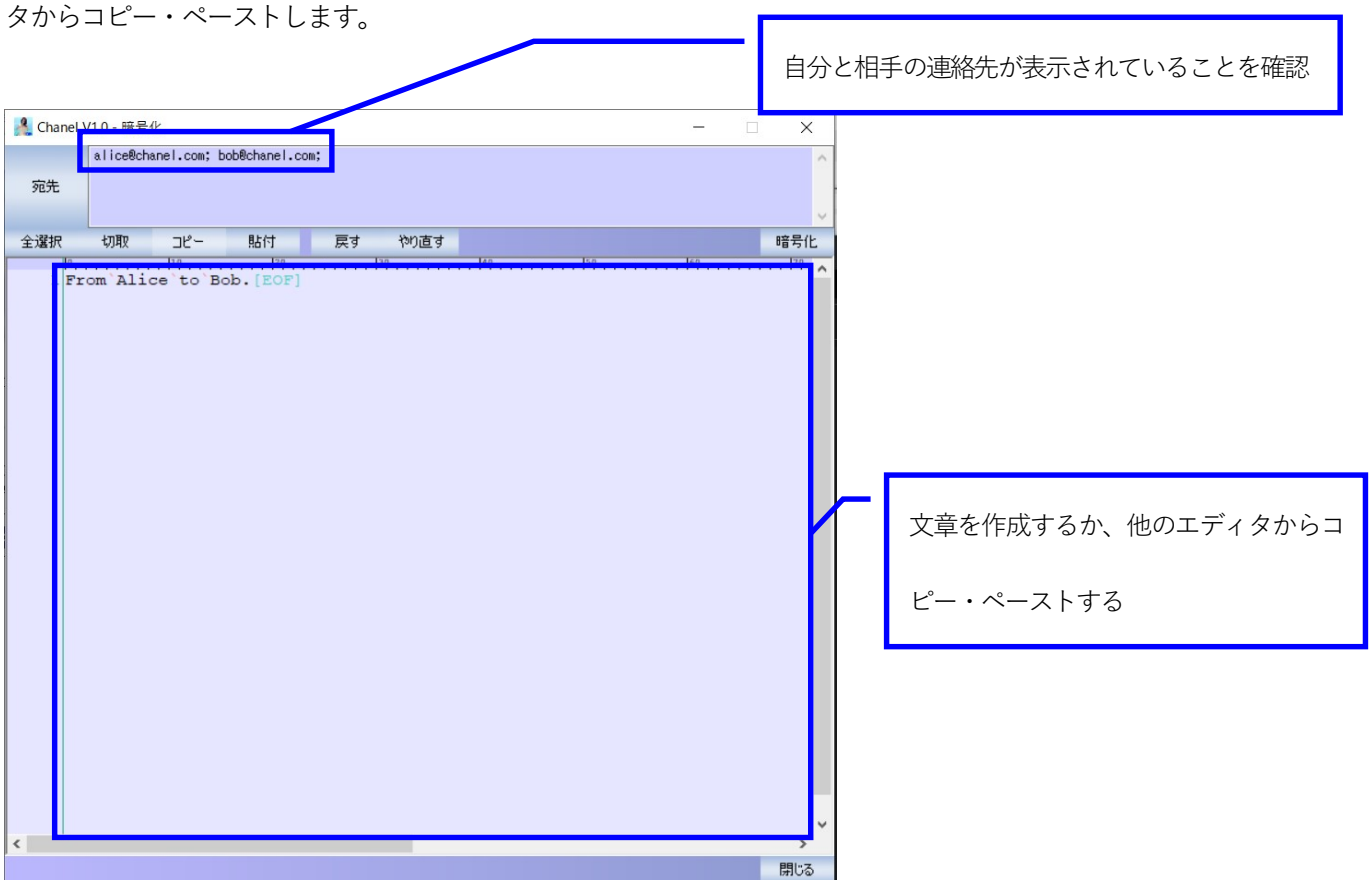
※ここで自分の「安全な連絡先」をチェックしなくてもよいですが、自分では暗号文を解読できなくなります。

自分と相手の「安全な連絡先」にチェックを入れたら[選択]ボタンをクリックします。



宛先表示部に自分と相手の連絡先が表示されたことを確認したらテキスト編集部に文章を作成するか、他のエディ

タからコピー・ペーストします。



※簡単な編集機能は上部のボタンを使う事ができ、また下記の一般的なショートカットキーが使えます。

全選択：Ctrl＋A

切り取り：Ctrl＋X

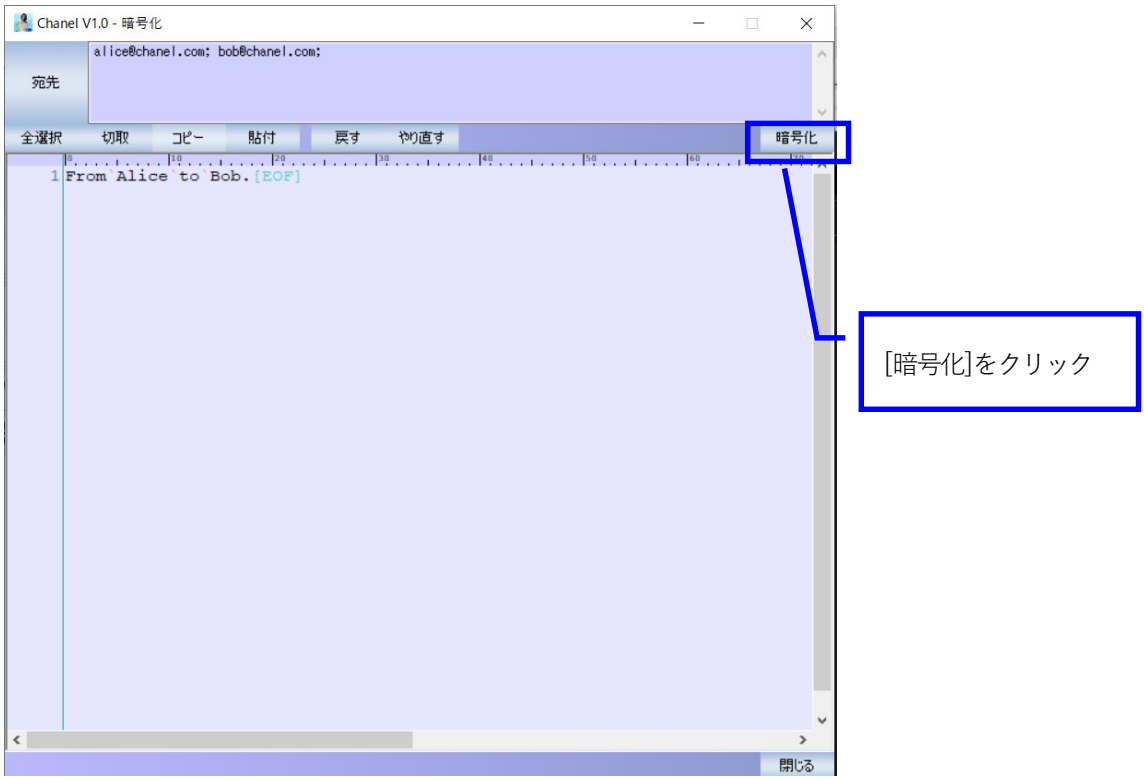
コピー：Ctrl＋C

貼り付け：Ctrl＋V

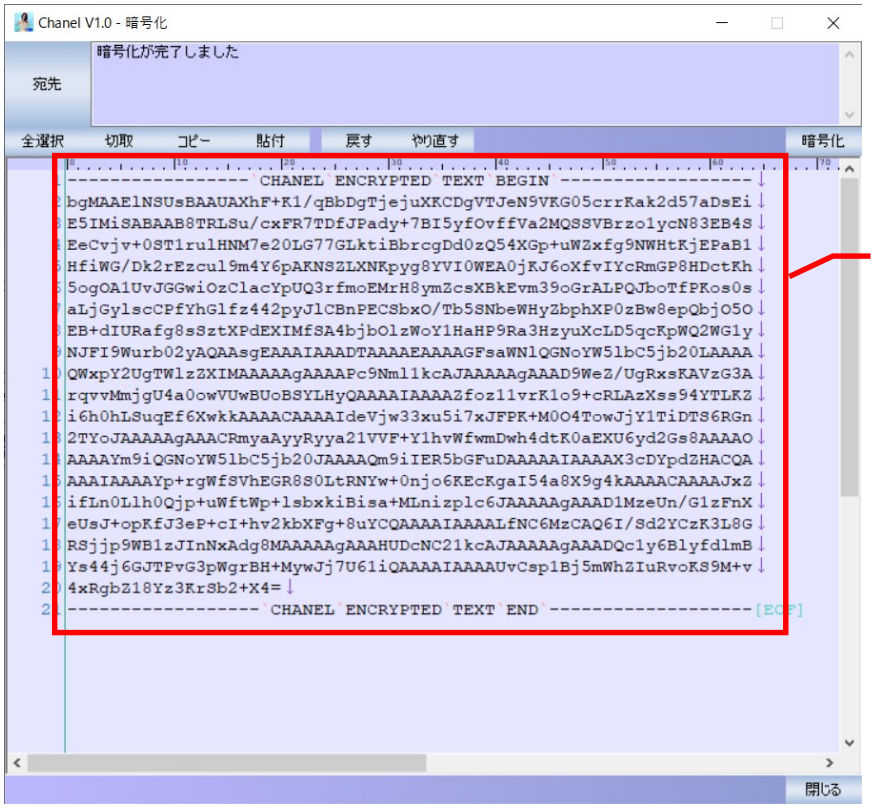
戻す：Ctrl＋Z

やり直す：Ctrl＋Y

[暗号化]ボタンを押して作成した文章を暗号化します。



文章が暗号化されます。



上下の点線の行も含めてコピーすること

暗号化された文章はコピーして、メールなどに貼り付けて宛先の相手に送ってください。

コピーするとき、暗号文の上下にある、点線付の「CHANEL ENCRYPTED TEXT BEGIN」と「CHANEL ENCRYPTED TEXT END」と記載されている行も含めてコピーしてください。

この２行が無いと解読できなくなります。

3）　他の人から受け取った暗号文の解読

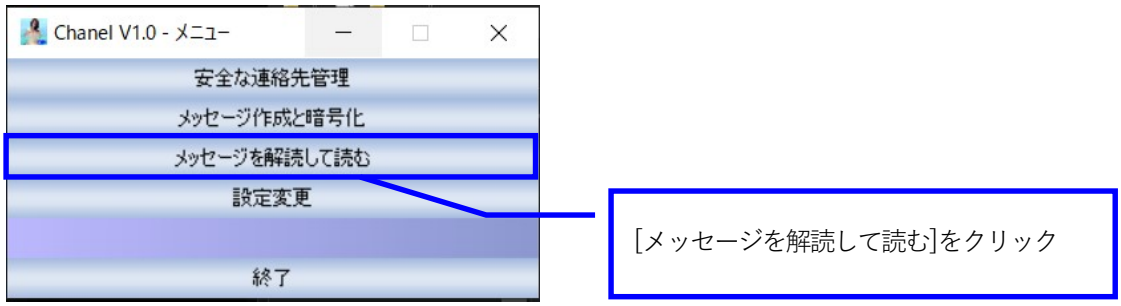
ここでは、他の人から受け取った暗号化した文章を解読する手順を説明します。

手順としては、自分で暗号化した文章を解読する手順と変わりません。

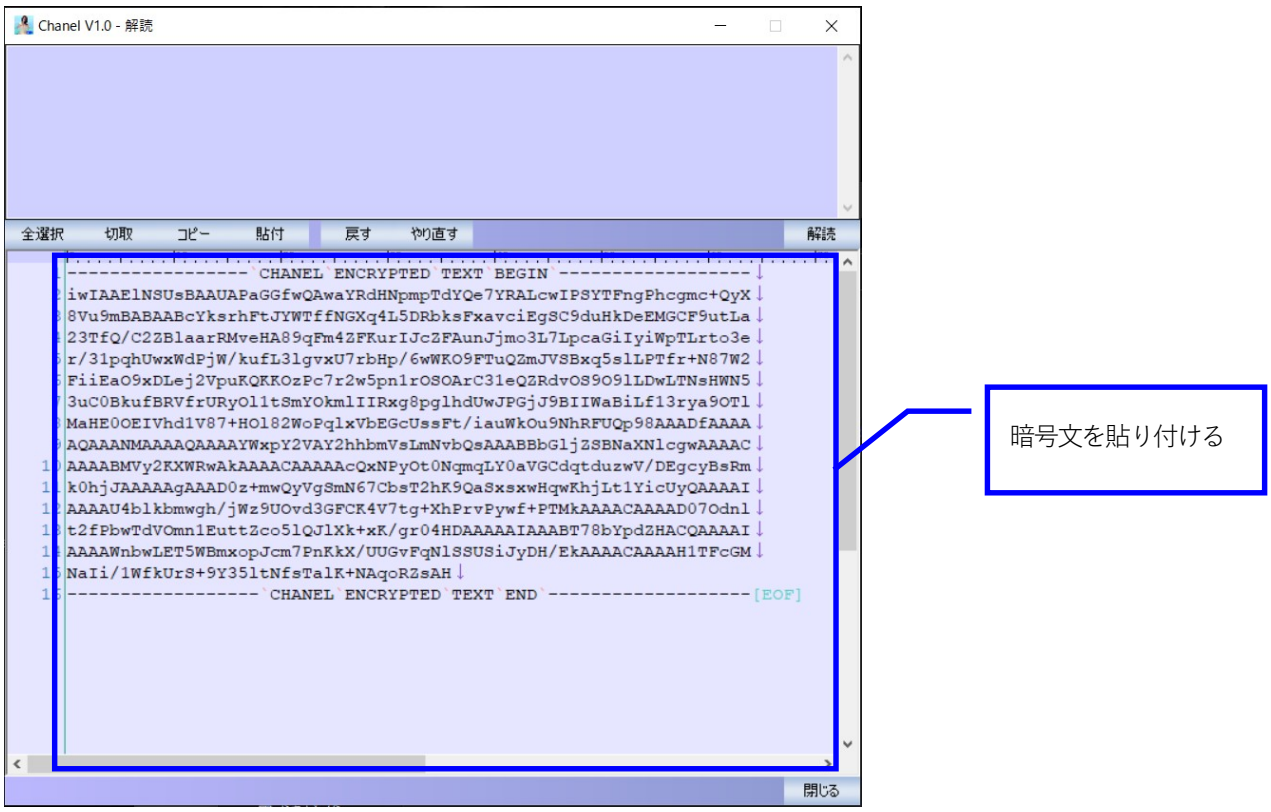
※　正しくは暗号文をパスワードを使って元の文章に戻す事を「復号」といい、「解読」とはパスワード等を知らない場合に様々な方法でアタックして無理やり解読する事をいいます。ただ一般的に「復号」は使われず「暗号」といえば「解読」なので Chanel では「解読」を使います。

なお、「暗号化」との文字上の対称性のため「復号化」という表現を用いる場合もあります。

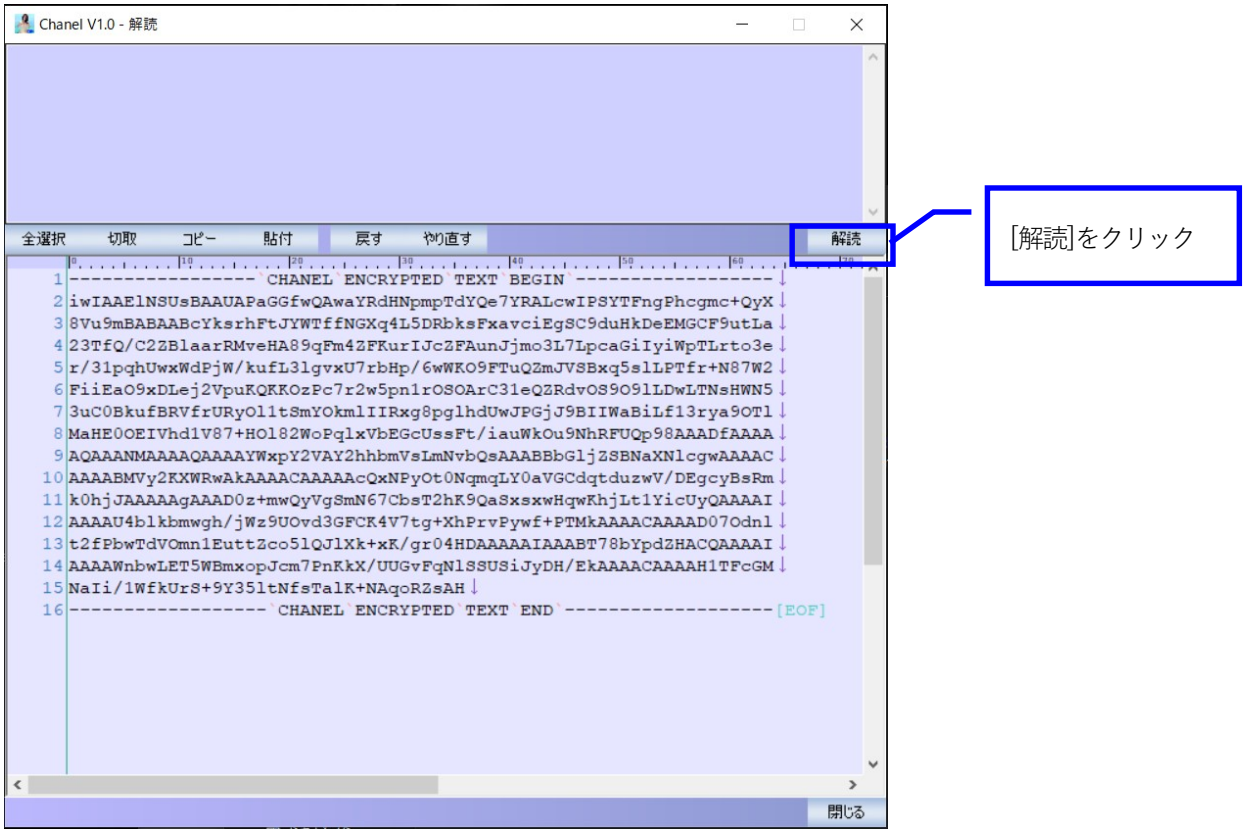
メニューの[メッセージを解読して読む]をクリックします。



[解読]画面が開くので、テキスト編集部分に暗号文を貼り付けます。

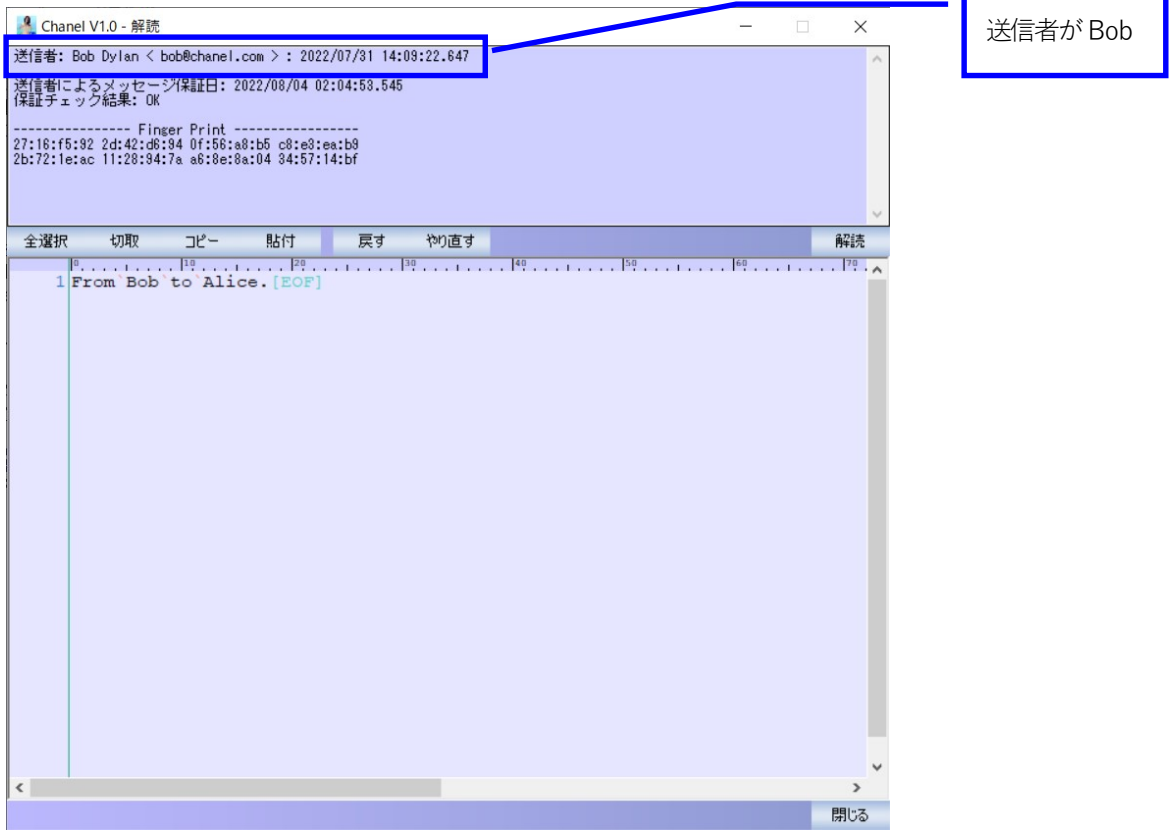


[解読]ボタンをクリックします。



[解読]をクリック

元の文章が表示されます。サンプル図は Alice が Bob から送られた暗号文を解読した状態の図です。



送信者が Bob

他の人と暗号文をやりとりする方法は以上です。

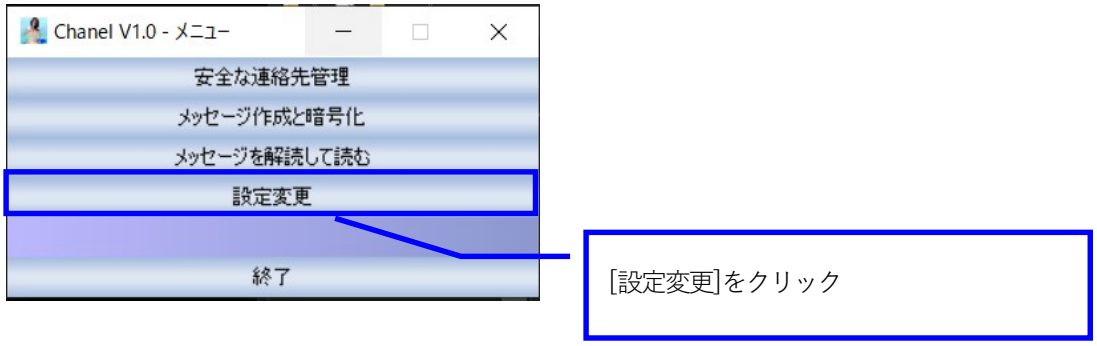
5.パスワードを変更するには

ここで説明するのは、**現在のパスワードが分かっていて、なんらかの理由でパスワードを変更する**方法です。

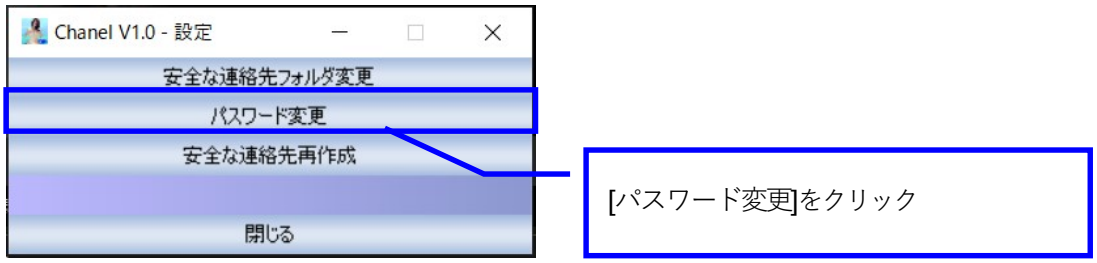
現在のパスワードを忘れてしまった場合、パスワードの再作成となりますので「パスワードを忘れてしまったら」の説明に従ってください。

1）　パスワードを変更する

メニューの[設定変更]ボタンをクリックします。



設定画面が開くので、[パスワード変更]ボタンをクリックします。



[パスワード変更]画面で必要な情報を入力します。

現在のパスワード：

現在の、Chanel を利用するためのパスワードを入力します。

新しいパスワード：

変更後のパスワードを入力します。

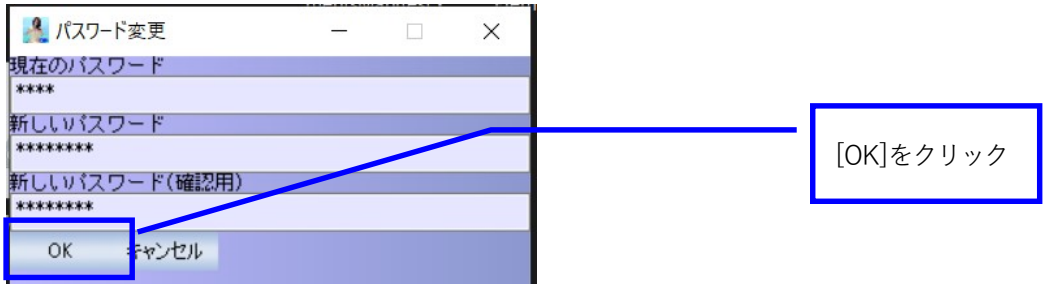
特に制限はないですが、他の人が推測しにくい方がよいでしょう。

※パスワードは絶対に忘れないように管理してください。

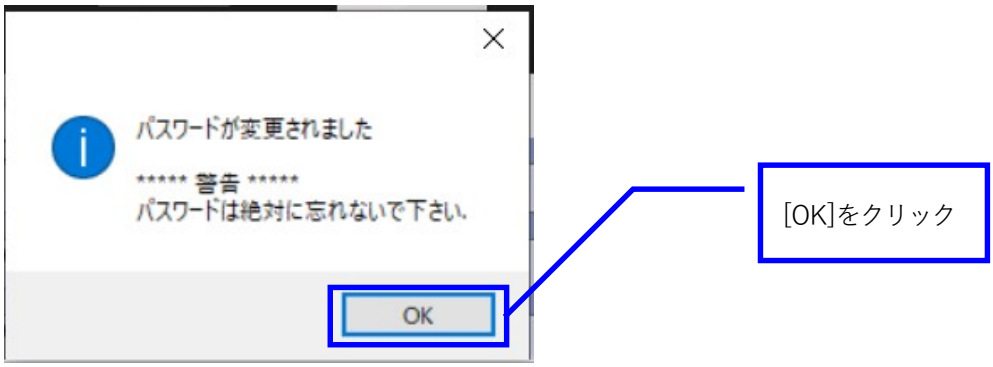
新しいパスワード（確認用）：

新しいパスワード欄と同じ変更後のパスワードを、もう一度入力します。

全ての記入が終わったら[OK]ボタンをクリックして下さい。



ダイアログが開いたら、[OK]ボタンをクリックしてください。



パスワードを変更する方法の説明は以上です。

6.パスワードを忘れてしまったら

1）　【重要】パスワードを忘れた場合どうなるか

Chanel を使うためのパスワードを忘れた場合、Chanel 起動時の[パスワードを入力してください]ポップで[忘れた]ボタンをクリックして再度パスワードを再作成することで Chanel をまた使う事ができるようにはなりますが、**重要な制約があります。**

現在のパスワードが分かっていてパスワードを変更することと、現在のパスワードを忘れたために、パスワードを再作成することは、Chanel が行う処理としては全く異なるため注意してください。

●それまでに暗号化した文書や、送られてきた暗号文が解読できなくなる

Chanel を使うためのパスワードは、実は暗号文の解読に必要な、「重要なデータ」を暗号化するパスワードなので、これを忘れると「重要なデータ」が解読できなくなります。パスワードを再作成するとき、解読不能となった「重要なデータ」は破棄・再作成されてしまいます。過去に暗号化した文書は破棄された古い「重要なデータ」でしか解読できず、再作成した新しい「重要なデータ」では解読できません。そのため過去の暗号文が解読できなくなります。

Chanel は実験的な暗号化アルゴリズムの集積体なので、製品版のような本格的なセキュリティは考慮されていないものの、Chanel が仕組みを参考にした PGP は高い機密性を保つために複雑な仕組みで作られており、その構造を取り入れているためこのような仕組みになっています。

ですので、**パスワードは絶対に忘れないようにしてください**。

●「安全な連絡先」を他の人に送りなおさなくてはならなくなる

パスワードを作り直すにあたっては「解読に必要な重要なデータ」のひとつである「安全な連絡先」も作り直されます。そのため複数の人と暗号文をやりとりしている場合、新しい「安全な連絡先」を全ての相手に送り直さなくてはなりません。

もし相手があなたの古い「安全な連絡先」を使ってあなた宛ての暗号文を作成した場合、あなたはその暗号文を解読することができません。

これも Chanel が PGP の仕組みを取り入れているために起こります。

●本格的な暗号技術が取り入れられているため作者も過去の暗号文を解読できない

Chanel が使っている暗号の仕組みは作者が適当に作ったものではなく、旧ソビエト・現ロシアが国家レベルで採用している暗号化の仕組みです。

安全な暗号化の仕組みとは、どのように文書が暗号化されるか、という仕組みが広く公開されていても、パスワードを知らない人が暗号文を解読することが極めて困難である仕組みです。

そのため、作者である私はもちろん、Chanel が用いている旧ソビエトの暗号化の仕組み GOST 24187-89 の開発者ですらパスワードを割り出すことも、過去の暗号文を解読することもできません。

2）　ではどうすればいいか

まずは、どうにかパスワードを思い出すようにしてみてください。思いつく限りのパスワードを[パスワードを入力して下さい]画面で試してみましょう。

Chanel にロック機能はありませんので、何回パスワードを間違えても使えなくなることはありません。

それでもどうしても思い出せない場合、いよいよパスワードを再作成することになります。

ただしその前に、**後からパスワードを思い出した時のために今使っている Chanel を保存しておく**手があります。

やり方は簡単です。

Chanel を、解凍したフォルダごと名前を変えてコピーして保存しておくだけです。

後からパスワードを思い出す事ができたら、ここで保存しておいた Chanel を使えば過去の暗号文を解読する事ができるようになります。

ただし、**パスワードを作り直した方の Chanel でも過去の暗号文を読めるようにする、という手段はありません。**

全く無いわけではありませんが、過去の暗号文全てを、保存しておいた Chanel で解読してパスワードを作り直した方の Chanel で暗号化し直す、という方法になります。暗号化した文書の量が少なければいいですが、文書が多いと大変な労力をともないます。

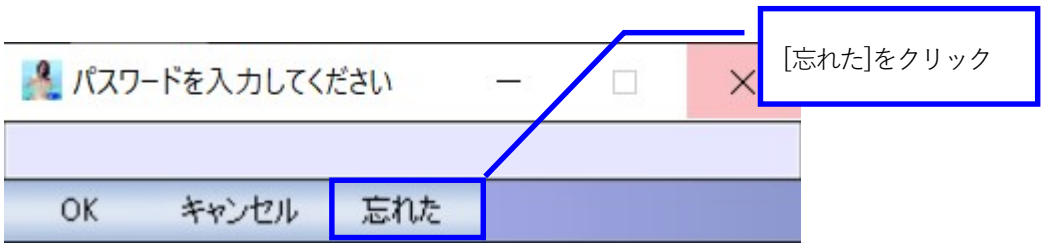
また自分の文書を暗号化したのではなく他の人から送られた暗号文の場合、暗号化し直した時点で送信者情報やデジタル署名は全て自分に置き換わるため失われます。文書を書いたのが他の人であっても、それを改めて暗号化するのは自分だからです。

3） パスワードを再作成する

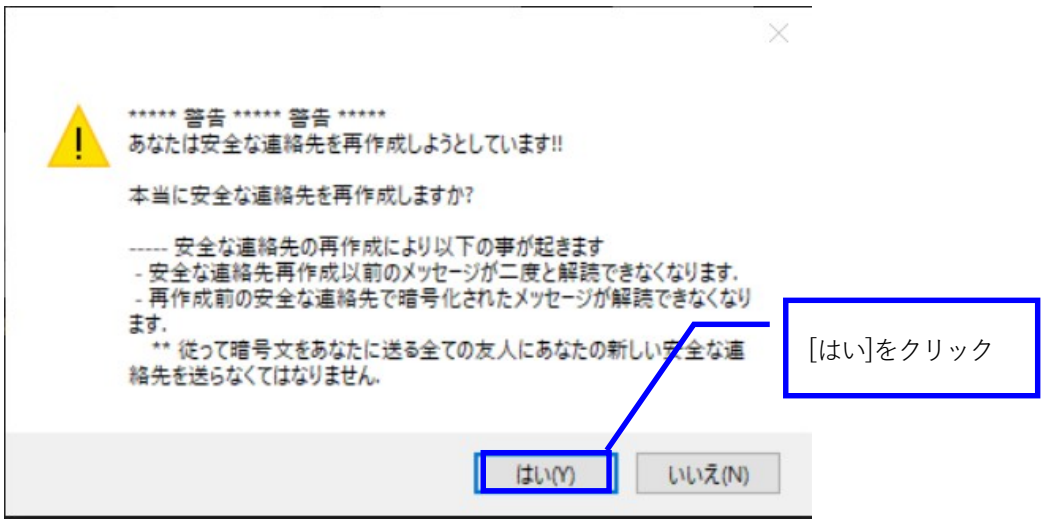
どうしてもパスワードを思い出せない場合、パスワードを再作成するしかありませんが、ここでもう一度。

後からパスワードを思い出した時のために、パスワードを再作成する前に Chanel をフォルダごとコピーして保存しておいてください。「安全な連絡先」のフォルダを変更している場合、「安全な連絡先」フォルダをコピーして保存しておいてください。

Chanel のコピーを終えたら、通常使っている(コピーではない方の)Chanel を起動してください。[パスワードを入力してください]ポップが開いたら、パスワード入力欄には何も入力せずに[忘れた]ボタンをクリックします。



以下のようなダイアログが表示されますので[はい]ボタンをクリックします。「安全な連絡先を再作成」のような事が書いてありますが、問題ありません。パスワード再作成とは、いままでの「安全な連絡先」を破棄して新たに再作成することに他ならないからです。



[パスワード再作成]画面が表示されますので、必要な情報を入力してください。

パスワード：

特に制限はないですが、他の人が推測しにくい方がよいでしょう。

※パスワードは絶対に忘れないように管理してください。

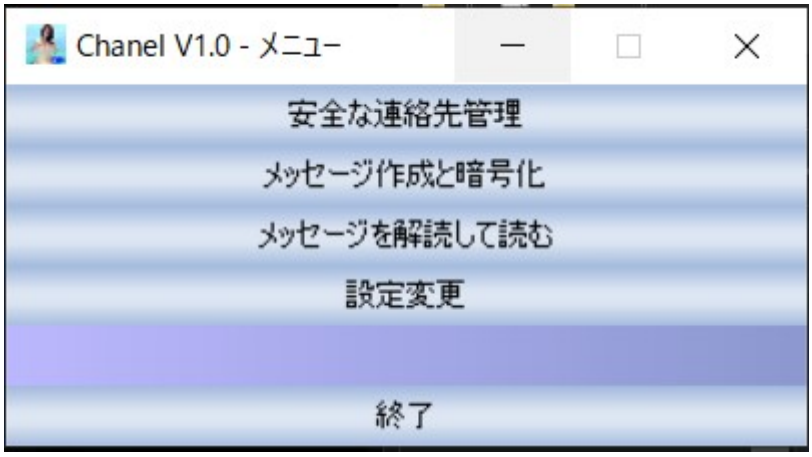
再びパスワードを忘れた場合、パスワードは何度でも再作成できますが、その都度、再作成前に暗号化した文章や、他の人が再作成前のあなたの「安全な連絡先」を使って暗号化したメッセージが解読不能となることは言うまでありません。

パスワード（確認用）：

パスワード欄と同じパスワードをもう一度記入してください。

全ての記入が終わったら[OK]ボタンをクリックしてください。

[メニュー]画面が開いて Chanel が使えるようになります。



【参考】

[安全な連絡先管理]画面を開くと、今まで使っていた「安全な連絡先」の色アイコンがグレーになっていることがわかります。これは「安全な連絡先」が無効になっている事をあらわします。

パスワード再作成する方法は以上です。

7.「安全な連絡先」の管理

ここからは Chanel の少し高度な使い方となります。単に一人で Chanel を使っている場合や、他の人と単に暗号化したメッセージのやり取りを楽しむだけであればこの先で説明する機能が必要になることはほぼありません。

1） [安全な連絡先管理]画面

①安全な連絡先一覧

「安全な連絡先」所有者の名前、E メールアドレス、「安全な連絡先」の作成日が一覧表示されます。クリックすると、「安全な連絡先」が選択状態になります。

Chanel では、**E メールアドレス**と**作成日**がそれぞれ一致する「安全な連絡先」は、同じ「安全な連絡先」として扱われます。

色アイコンについて：

「安全な連絡先」所有者の名前に付加されている色アイコンは「安全な連絡先」の信頼度を表します。

色	信頼度
	自分自身の「安全な連絡先」です。完全に信頼できます。
	自分が「保証」した「安全な連絡先」です。信頼度は非常に高いです。
	他の二名以上の「保証」がある「安全な連絡先」です。やや信頼できます。
	所有者自身の保証しかない「安全な連絡先」です。あまり信頼できません。
	無効化された「安全な連絡先」です。

※注意

PGP では信頼度を利用者が自分で決めて自由に設定できるようですが、Chanel にそういった機能はありません。

また PGP で「安全な連絡先」(公開鍵)を無効化する場合は無効化証明書というものが必要になるようですが、Chanel では単純に作成日が新しい「安全な連絡先」が追加されると、古い「安全な連絡先」は自動的に無効化されます。

Chanel は「安全な連絡先」の信頼度が非常に重要になる状況下での利用を考慮していませんので、あくまでも参考レベルです。

②[安全な連絡先の詳細]ボタン

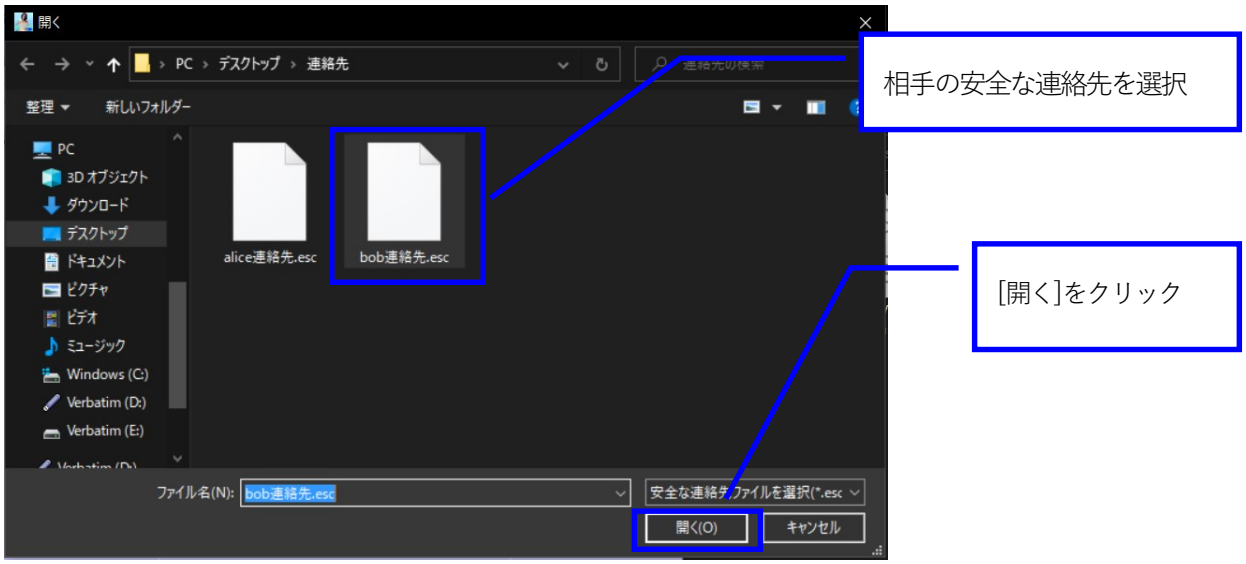
安全な連絡先一覧で選択された「安全な連絡先」の「安全な連絡先の詳細」画面を開きます。

※安全な連絡先一覧で「安全な連絡先」をダブルクリックしても同じ動作をします。

③[安全な連絡先の追加]ボタン

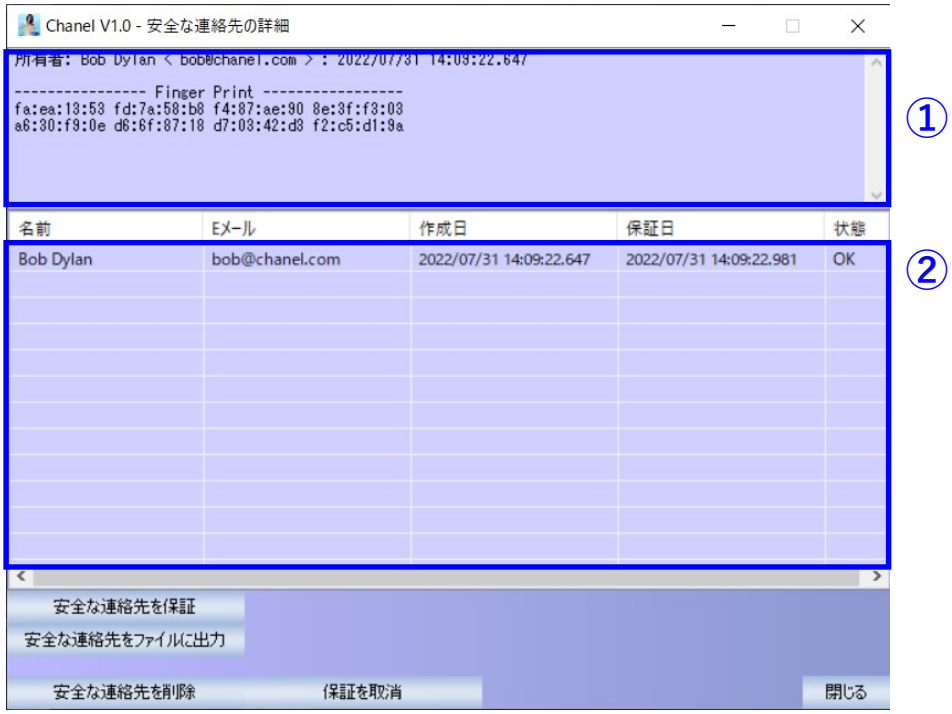
他の人の「安全な連絡先」を追加します。

ファイルの選択ダイアログが開くので、追加する「安全な連絡先」（図では Bob の安全な連絡先を追加するものとします）を選択して[開く]ボタンをクリックします。



相手の「安全な連絡先」が追加されます。

2） [安全な連絡先の詳細]画面



①「安全な連絡先」詳細表示部

「安全な連絡先」の詳細が表示されます。

「安全な連絡先」の所有者名(図では Bob が所有者)、Ｅメールアドレス、作成日が最上段に表示されます。

２段目の Finger Print は「安全な連絡先」の HASH 値です。

「安全な連絡先」が本当に所有者のものか疑わしい場合、所有者と会う・電話をするなど、メールやチャットのように文字だけに頼らない直接的な手段でこの値をお互いに確認しあう、といった使い方をします。

ただ Chanel はそこまでするほどの状況下で使われる事は想定されていませんので、そのための情報も出る、というレベルです。

②「保証」一覧

表示されている「安全な連絡先」に付与された「保証」の一覧です。

「保証」とは、表示されている「安全な連絡先」が間違いなく所有者自身のものだと「保証」した人が請け負いま

す、という意味になります。

「保証」はやはり「安全な連絡先」によって行われますが、一覧に表示されているのは「保証」を行った人の「安全な連絡先」の所有者名と E メールアドレス、「保証」した人の「安全な連絡先」の作成日、「状態」となります。

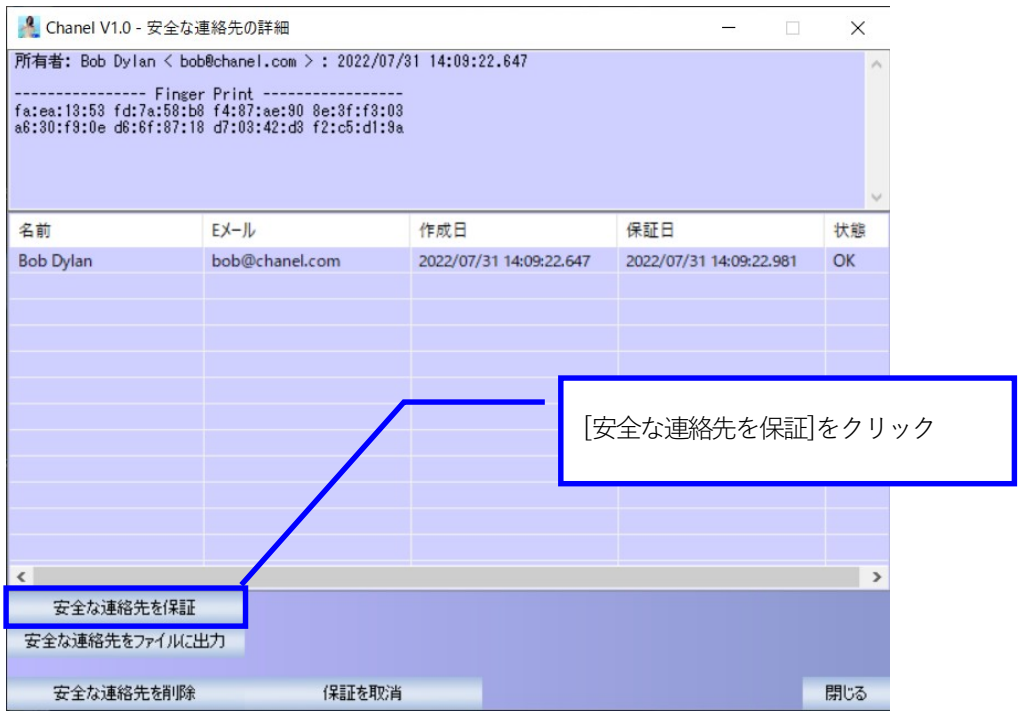
「状態」とは、「保証」を保証者の「安全な連絡先」を使って Chanel が検証した結果で、下記の種類があります。

状態	検証結果
OK	「保証」を検証した結果、問題なかった事を表します。
NG	「保証」を検証した結果、問題ありの結果となったため、「保証」が信用できない事を表します。 「保証」したのは保証者本人ではなく、何者かが「安全な連絡先」を偽造し保証者を装って「保証」した可能性があります。
KD	Key Deleted: 「保証」の検証に使われた保証者の「安全な連絡先」は無効化されている事を表します。 保証者が再作成した新しい「安全な連絡先」が登録されることで、古い「安全な連絡先」が無効化された場合、この状態になります。 ただし、保証者の「安全な連絡先」（既に古くなっているわけですが）を不正に入手した何者かが「保証」した可能性も否定できません。
NK	No Key: 保証者の「安全な連絡先」が Chanel に登録されていないので、「保証」の検証ができないことを表します。検証ができないため信頼度は不明で、不明である以上「保証」は信頼できません。
!!	「保証」検証処理中に何らかのプログラム的なエラーが発生した事を表します。「保証」の信頼度は NK と同じ検証不能により不明なので、不明である以上は信頼できません。 通常は出ません。

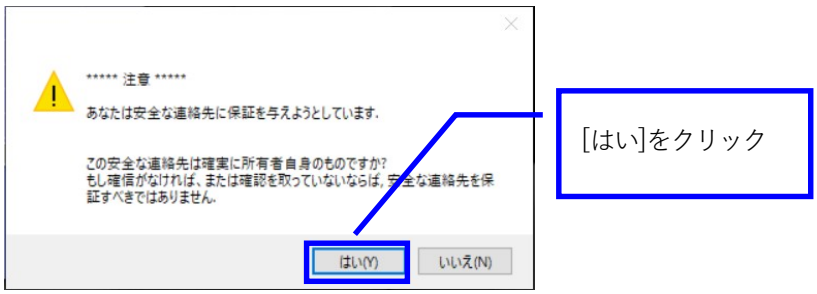
3）　他の人の「安全な連絡先」を「保証」する

図の例は Alice が Bob の「安全な連絡先」を保証する例です。

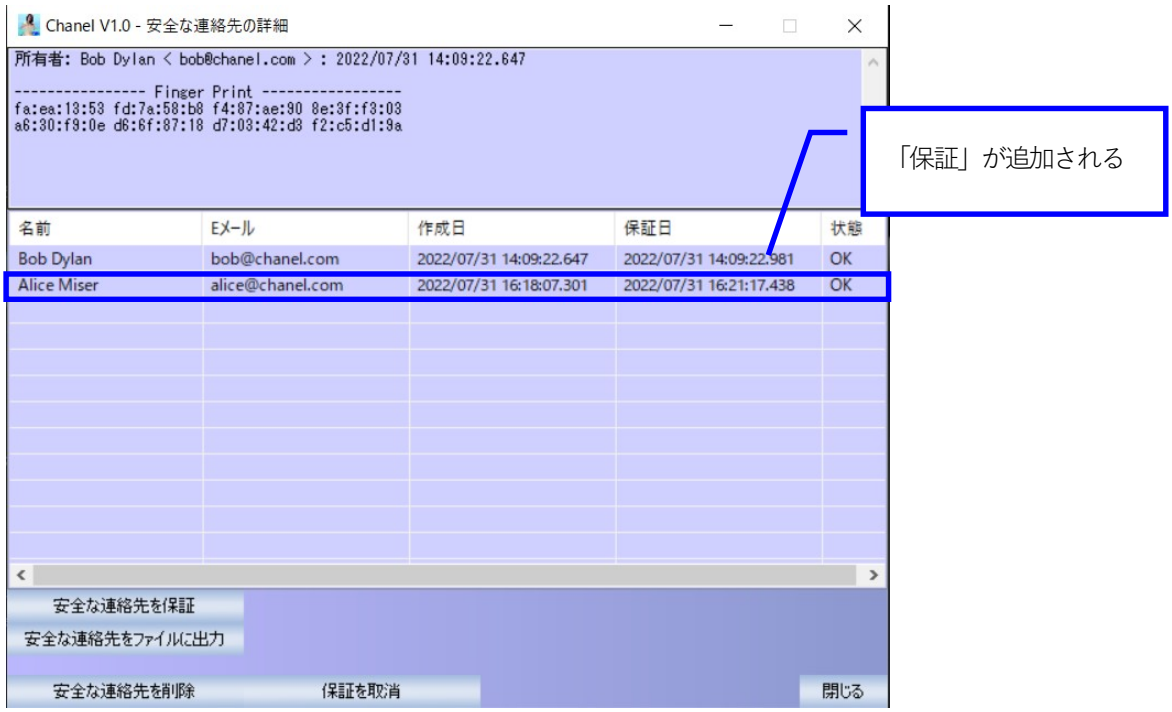
[安全な連絡先を保証]ボタンをクリックします。



ダイアログの[はい]ボタンをクリックします。



あなたの「保証」が追加されます。



※注意

他の人の「安全な連絡先」を保証するということは、二つの意味があります。

1. 「安全な連絡先」の、色アイコンの信頼度を上げる

2.あなたが「この『安全な連絡先』が所有者の連絡先で間違いない」と宣言する

Chanel では、PGP と同様に他の人の「安全な連絡先」をファイルに出力して第三者に配布する事ができます。日常生活だと、例えば Ellen と連絡を取りたがっている人に、あなたがちょうど Ellen の連絡先を知っていたから電話番号とか Line の ID を教えてあげるようなものです。

こういった場合 Chanel では他の人の「安全な連絡先」をファイルに出力して渡すわけですが、この時 2.は、他の人の「安全な連絡先」の確実性をあなたが「保証」して第三者に渡すという意味になります。

Chanel はそこまでしなくてはならない状況での利用を考慮されていません（し、普通の生活をしていて通信相手の信頼性が大きな問題になる状況が発生することはあまりないと思います）が、リスクを承知でそのような状況下で Chanel を利用する場合は注意してください。

もし本格的に信頼度を活用するばあい、所有者の確実性が問題になりますので、例えば一例として下記の場合は「保

証」し、そうでない場合は「保証」しない、といった自己ルールが必要になります。

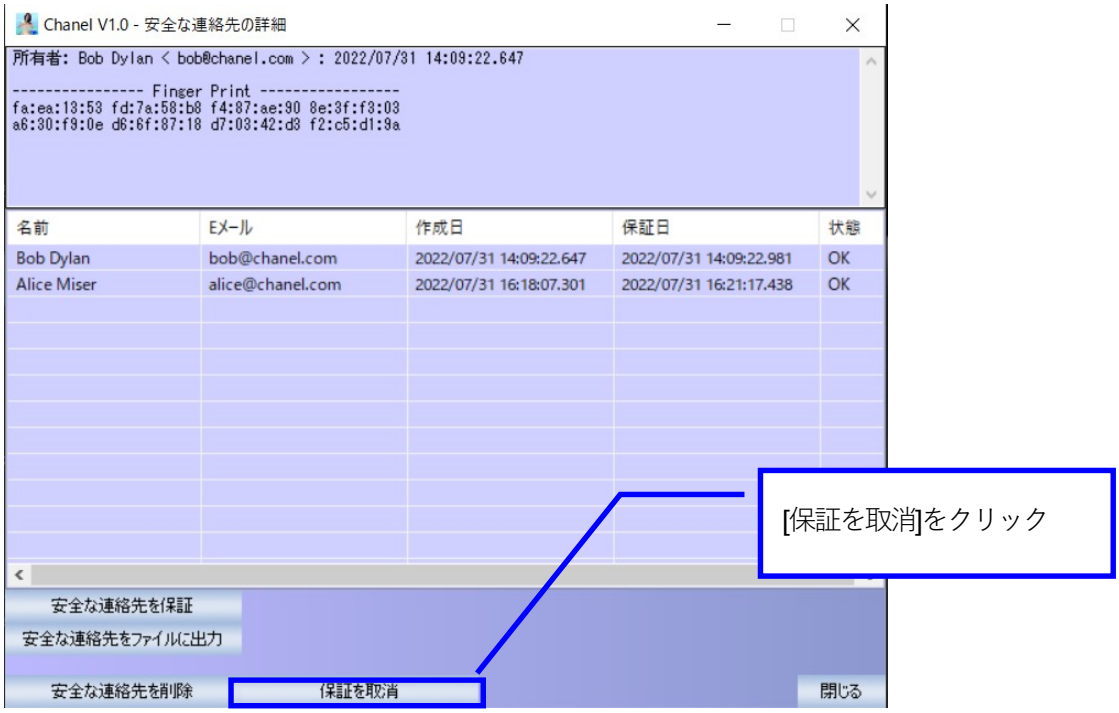
・USB メモリで直接対面して受け渡しを行った「安全な連絡先」

・電話やチャットアプリの音声通話・ビデオ通話で相手とフィンガープリントを確認した、あるいはまさにその場で送られた確実な「安全な連絡先」

・直接受け渡されていないが、ある人の「保証」があって、その人であれば「保証」するにあたっては必ず所有者確認する、安全面の意識が信頼できる人である

4）　他の人の「安全な連絡先」の「保証」を取消す

「保証」を取り消したい「安全な連絡先」の[安全な連絡先の詳細]画面で、[保証を取消]ボタンをクリックすると、「保証」が取り消されます。



※注意

保障を取り消すことができるのは、**他の人の安全な連絡先に付与した自分の保証だけ**です。

他の人が付与した保証を取り消すことはできません。

また、自分の「安全な連絡先」の「保証」（作成時に付与された自己保証）も取り消すことはできません。

5） 他の人の「安全な連絡先」をファイルに出力する

ファイルを出力したい人の「安全な連絡先」の[安全な連絡先の詳細]画面で、[安全な連絡先をファイルに出力]ボタンをクリックします。

[名前を付けて保存]ダイアログが開くので、任意のファイル名(拡張子は「.esc」)でファイルを保存します。

例ではファイル名を Bob 連絡先.esc としています。

「安全な連絡先」がファイルとして出力されます。このファイルを USB メモリに入れて渡すなり、E メールに添付するなどして送ってください。

6） 他の人の「安全な連絡先」を削除する

削除したい「安全な連絡先」の[安全な連絡先の詳細]画面で、[安全な連絡先を削除]ボタンをクリックします。

ダイアログで[はい]ボタンをクリックします。

他の人の「安全な連絡先」を削除する方法は以上です。

※注意

基本的に「安全な連絡先」は削除しないでください。無効化された「安全な連絡先」でも無効化されている事をしめす記録として使う事ができるため、誰かが無効化されている「安全な連絡先」を使った暗号文を送ってきた場合、「保証」を検証した結果「無効化された安全な連絡先」が使われた事がわかるので、所有者を装った第三者からの偽装メッセージの暗号文ではないかを疑う事ができます。

もしこの時、無効化された「安全な連絡先」が削除されてしまっていた場合は「保証」の検証結果が「安全な連絡先が存在しない」ことを示す NK になるため、所有者を装っただれかが無効化された「安全な連絡先」を使って送ってきた暗号文なのか、所有者が「安全な連絡先」を再作成したが、まだその最新版の「安全な連絡先」を入手していないのかがすぐに判別しづらくなるためです（「安全な連絡先」の作成日を確認したり、所有者に直接確認すればすぐにわかる事ではありますが）。

どうしても削除が必要になる場面はそうそう無いですが、**一つ確実に削除が必要になる場合があります。**

それは、**所有者を装った第三者が偽装した「安全な連絡先」を Chanel に登録してしまっていた場合**です。

「安全な連絡先」は、E メールアドレスと作成日が一致していれば同じ「安全な連絡先」とみなされるため、この二つが完全一致した「安全な連絡先」及び作成日が過去となる「安全な連絡先」は登録できません。もし Chanel に登録されている「安全な連絡先」が偽物だと発覚した場合、所有者の正しい「安全な連絡先」と入れ替えなくてはなりませんが、このとき正しい「安全な連絡先」が偽装された「安全な連絡先」と作成日まで一致しているか、それよりも過去の日時となっている場合はそのままでは登録できませんので、偽装された「安全な連絡先」の削除が必要になります。

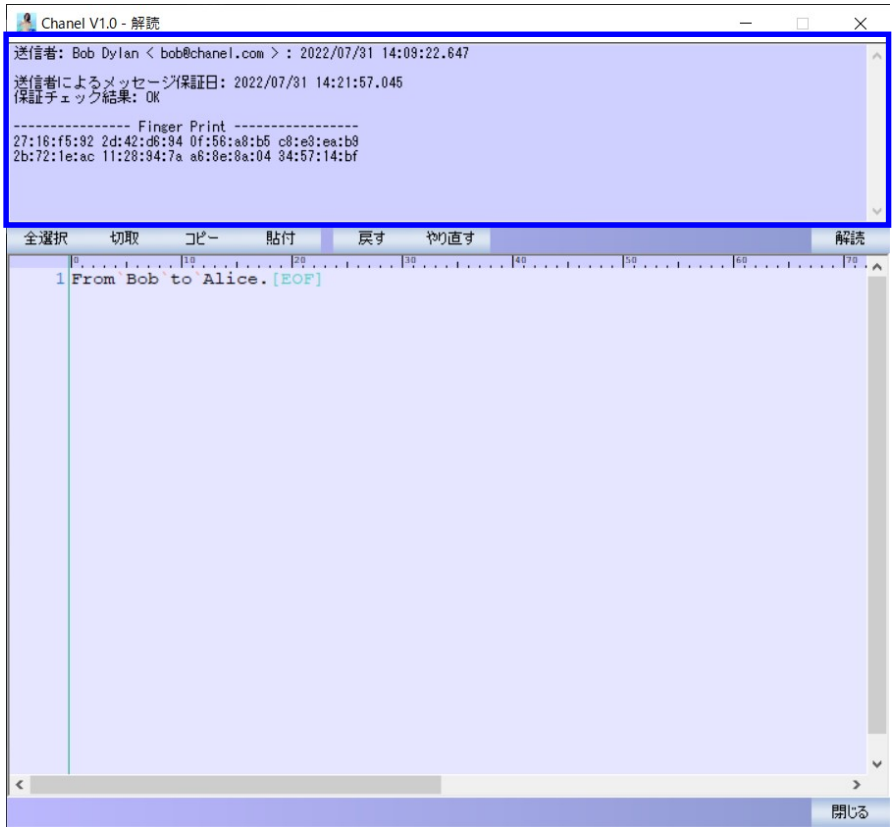
「安全な連絡先」の管理については以上です。

8.[解読]画面の状態表示部について

ここでは、[解読]画面の上部にある状態表示部に表示される項目についてを説明します。

1） [解読]画面の状態表示部の各項目について

図では Bob が Alice に送ったメッセージを解読した結果を例としています。青枠で囲まれた部分が状態表示部です。



最上段には、メッセージを暗号化して送った送信者の「安全な連絡先」の所有者名(図では Bob が所有者)、E メールアドレス、「安全な連絡先」の作成日が表示されます。

2 段目は、送信者がメッセージ内容を「保証」した日時とその検証結果です。日時はメッセージが暗号化された日時と考えてよいです。

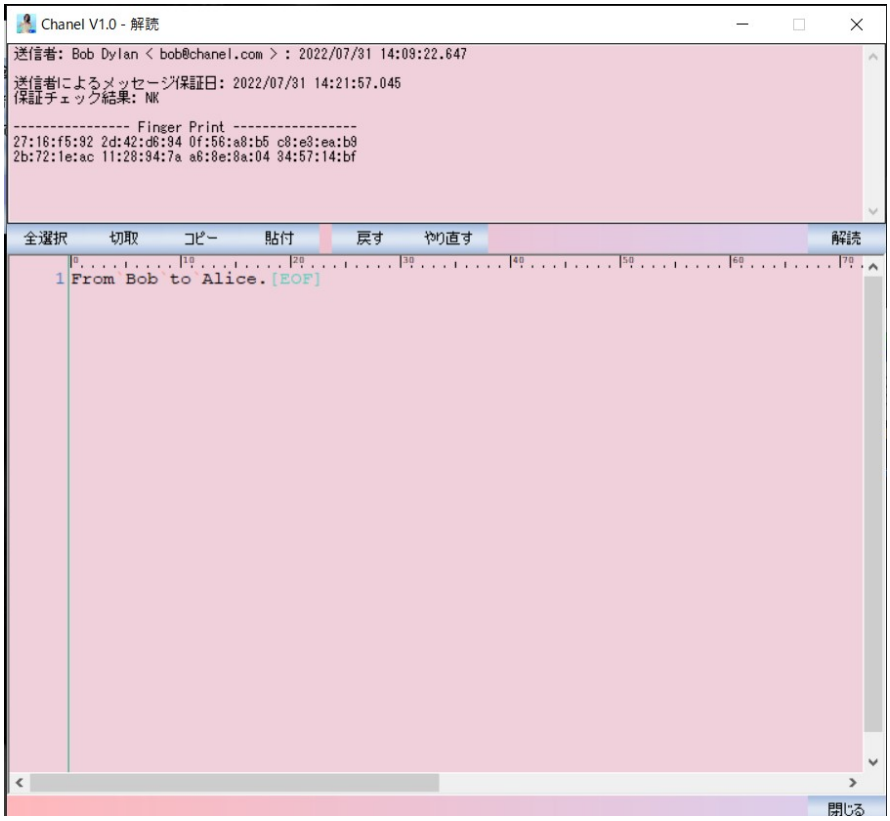
3 段目の Finger Print は、送られた文章の HASH 値です。

送られた文章が本当に送信者によって書かれたものか疑わしい場合、送信者と会う・電話をする・音声通話・ビデオ通話など、メールやチャットのように文字だけに頼らない直接的な手段でこの値をお互いに確認しあう、といった使い方をします。

ただ Chanel はそこまでするほどの状況下で使われる事は想定されていませんので、そのための情報も出る、とい

うだけです。

メッセージ送信者の「保証」の検証が失敗した場合の例



図の例は、Bob が Alice に送ったメッセージが復号されたとき、メッセージの「保証」の検証に失敗した場合です。

「保証」の検証に失敗すると、解読画面全体が赤っぽく変わりますのですぐにわかります。

状態表示部を見ると、2 段目の保証チェック結果が「NK」つまり No Key となっていますので、送信者 Bob の「安全な連絡先」が Alice の Chanel に登録されていないか、削除されている事を表します。

保証チェック結果は[安全な連絡先の詳細]画面の「保証」の「状態」と同じで、下記の表のとおりです。

状態	検証結果
OK	「保証」を検証した結果、問題なかった事を表します。
NG	「保証」を検証した結果、問題ありの結果となったため、「保証」が信用できない事を表します。 「保証」したのは保証者本人ではなく、何者かが「安全な連絡先」を偽造し保証者を装って「保証」した可能性があります。
KD	Key Deleted: 「保証」の検証に使われた保証者の「安全な連絡先」は無効化されている事を表します。 保証者が再作成した新しい「安全な連絡先」が登録されることで、古い「安全な連絡先」が無効化された場合、この状態になります。 ただし、保証者の「安全な連絡先」(既に古くなっているわけですが)を不正に入手した何者かが「保証」した可能性も否定できません。
NK	No Key: 保証者の「安全な連絡先」が Chanel に登録されていないので、「保証」の検証ができないことを表します。検証ができないため信頼度は不明で、不明である以上「保証」は信頼できません。
!!	「保証」検証処理中に何らかのプログラム的なエラーが発生した事を表します。「保証」の信頼度は NK と同じ検証不能により不明なので、不明である以上は信頼できません。 通常は出ません。

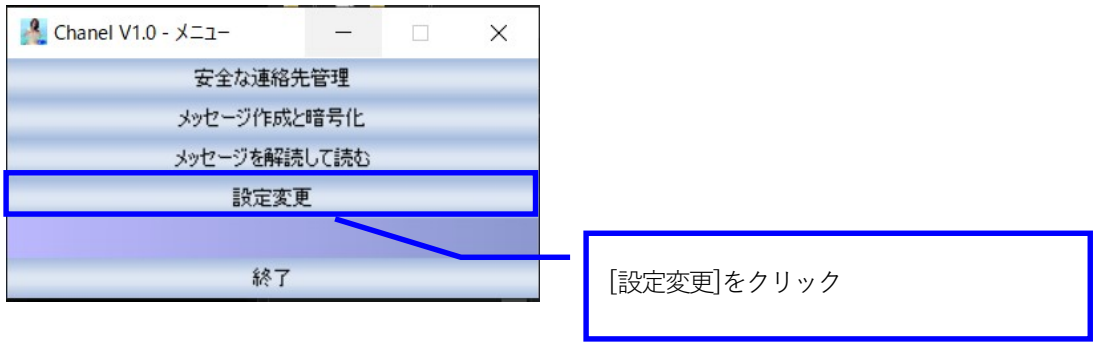
[解読]画面の状態表示部については以上です。

9.その他の機能

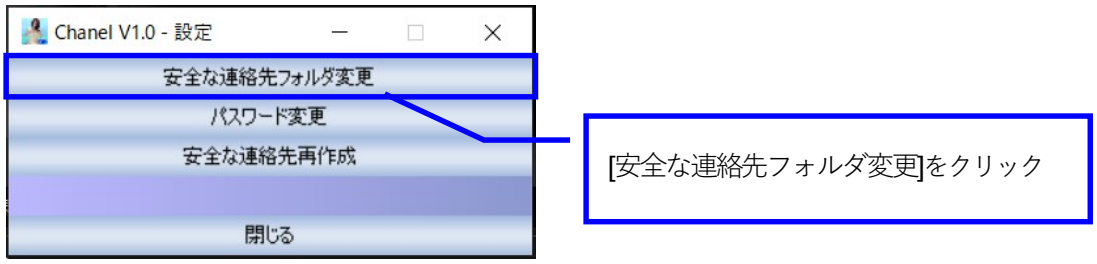
ここでは、いままでで説明されていない「安全な連絡先」のフォルダ変更のやり方と、「安全な連絡先」の再作成のやり方を説明します。

1）　「安全な連絡先」のフォルダを変更する

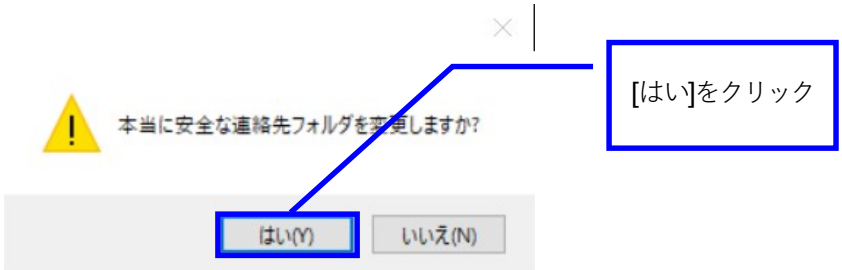
メニューの[設定変更]ボタンをクリックします。



設定画面が開くので、[安全な連絡先フォルダ変更]ボタンをクリックします。



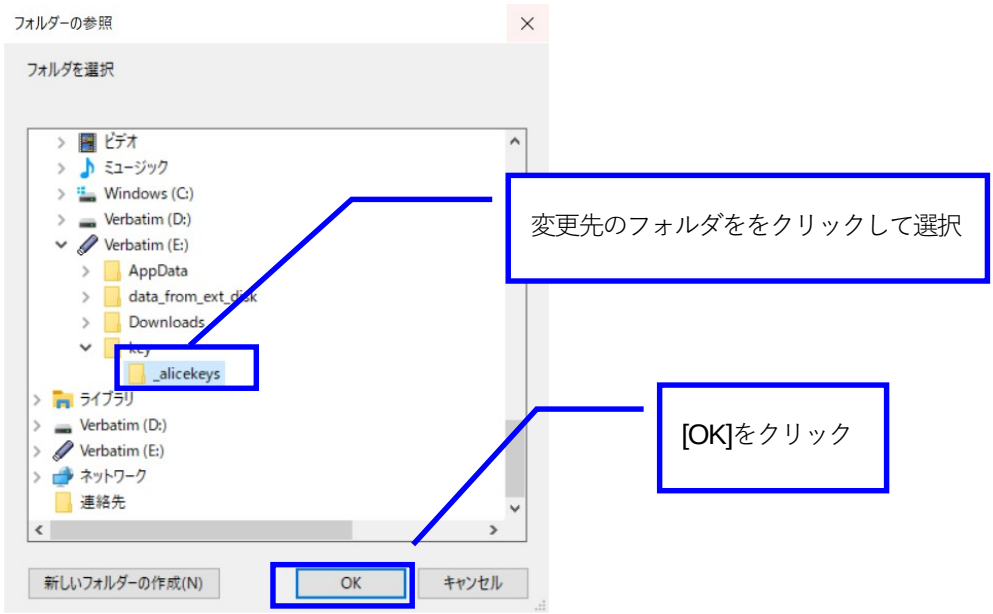
ダイアログで[はい]ボタンをクリックします。



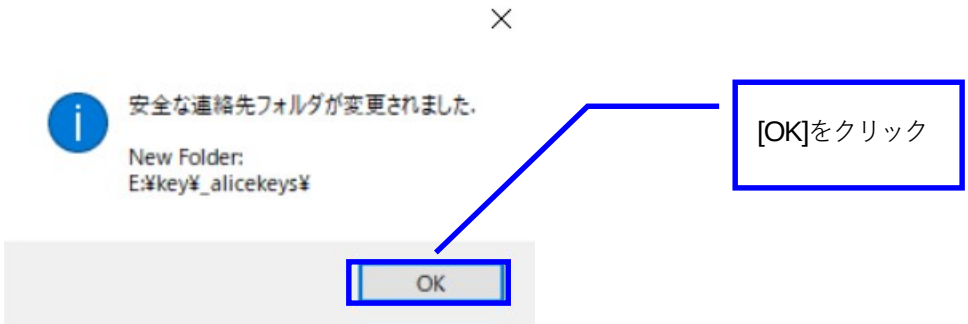
[フォルダの参照]ダイアログで変更先のフォルダを選択して[OK]ボタンをクリックします。

図は、ドライブ E（USB メモリスティック）上に移した_alicekeys フォルダへ変更する例です。

※「安全な連絡先」のフォルダはデフォルトでは_chanelkeys ですが、名前を変更しても構いません。



安全な連絡先フォルダが変更された通知ダイアログで[OK]ボタンをクリックします。



この後、「メニュー」画面でなんらかのボタンをクリックすると「パスワード入力」画面が出ますので Chanel を利用するためのパスワードを入力してください。

「安全な連絡先」のフォルダを変更するケースについて

安全な連絡先のフォルダを変更するのは、図の例のように、「安全な連絡先」を外部メモリなどに移動した場合、移動後のフォルダの「安全な連絡先」を使用するように切り替えるような場合などです。

この機能は Chanel 実行中にフォルダを変更する場合に使います。

変更前に Chanel を閉じていた場合、または Chanel が閉じられている間にフォルダを移動していた場合、次回起動時に「安全な連絡先」を新規に作成するか、既存の「安全な連絡先」を使うかポップが出て聞かれますので、 [いいえ]ボタンをクリックすると「フォルダの参照」ダイアログが開きます。そこで変更後のフォルダを選択しても同じです。

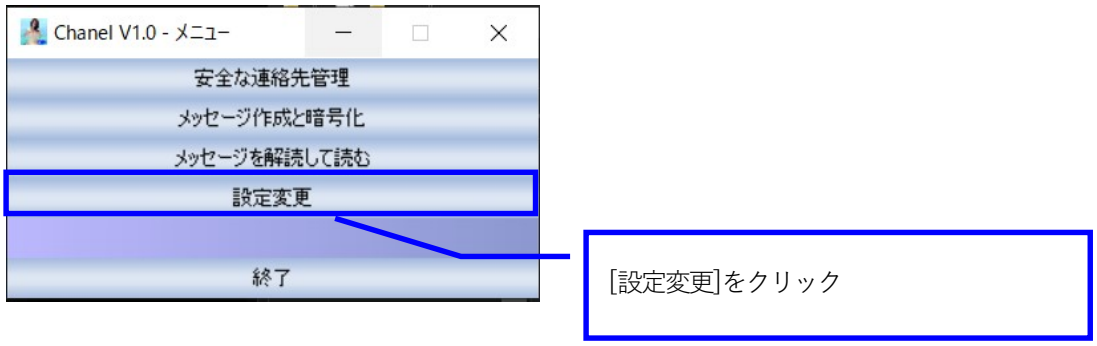
2）　「安全な連絡先」を再作成する

※注意

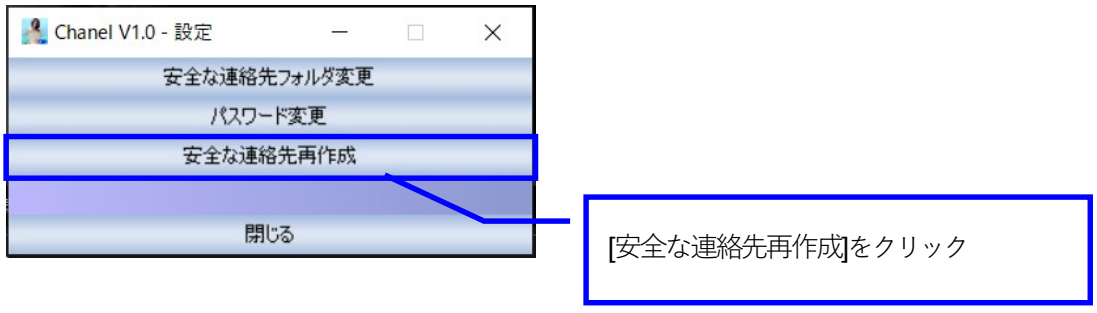
「安全な連絡先」を再作成することで Chanel は解読に必要な情報を全て作り直すため、以前の暗号文の解読ができなくなります。あらかじめ Chanel をフォルダごとコピーして保存しておいてください。

「安全な連絡先」のフォルダを変更している場合、「安全な連絡先」フォルダをコピーして保存しておいてください。

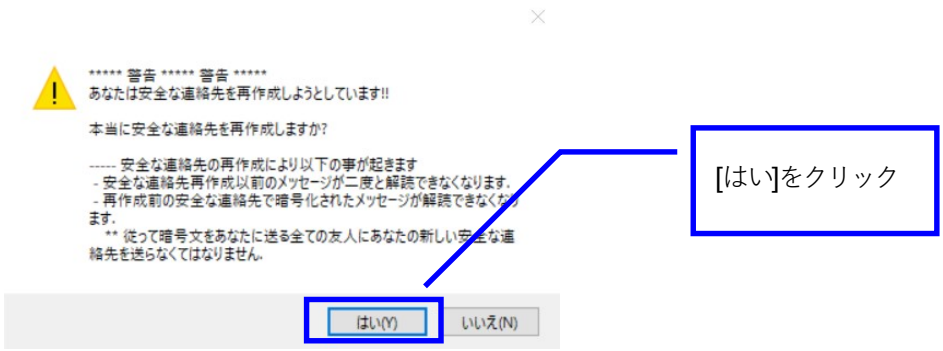
メニューの[設定変更]ボタンをクリックします。



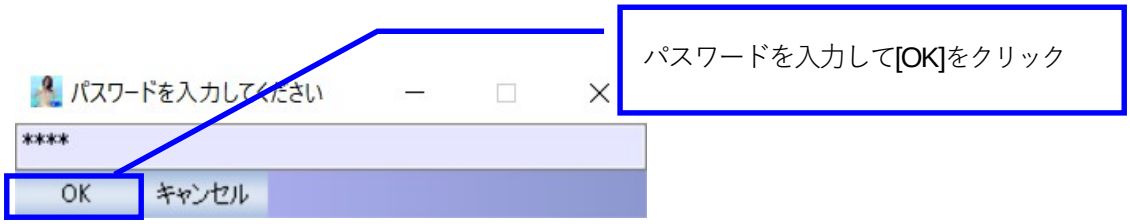
設定画面が開くので、[安全な連絡先再作成]ボタンをクリックします。



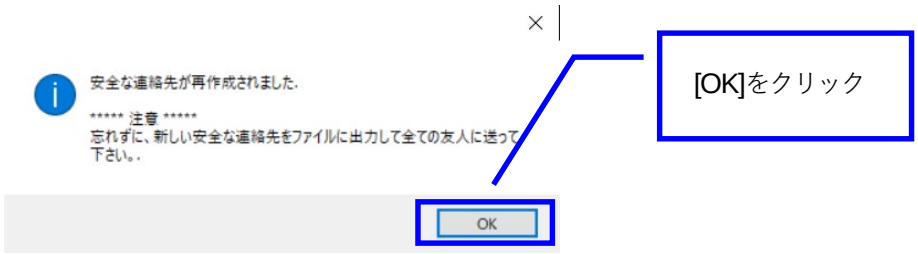
ダイアログで[はい]ボタンをクリックします。



パスワード入力を求められるので Chanel を利用するためのパスワードを入力して[OK]ボタンをクリックします。



ダイアログで[OK]ボタンをクリックします。



「安全な連絡先」を再作成する方法は以上です。

※重要

繰り返しになりますが「安全な連絡先」の再作成をすることで、過去の暗号文の解読が不可能になりますから**実行前に現在の「安全な連絡先」をコピーして保存することを忘れずに行ってください。**

また、そもそも「安全な連絡先」の再作成はパスワードが漏れた、もしくはその疑いが極めて高い、などのケースに限るべきです。

その他の機能については以上です。

10. (Annex 1)Chanel で使用している用語について

Chanel は暗号技術を利用した暗号化ツールですが、作者が友人、つまり暗号技術の用語を知らない人と使う事も想定して作ったため、暗号技術の用語を、Chanel の画面や本書では判りやすく言い換えています。ここでは Chanel で使われている用語と暗号技術の用語の関連を記載します。

1） Chanel で使われている用語と暗号技術の用語

ここまでは暗号技術用語を知らない利用者向けの説明であったため、暗号技術特有の用語をより一般的な言葉に置き換えてきました。

ここで、Chanel で使われている用語と暗号技術用語の対応を表にして示します。

Chanel での用語	暗号技術の用語
安全な連絡先	公開鍵。あるいはパブリックキー
保証	デジタル署名
保証する	デジタル署名を付与する
文章または文書	平文
解読	復号
仕組み	「アルゴリズム」という意味で使っている場合がある
パスワード	Chanel で利用するパスワードの場合、パスフレーズ。 文書を暗号化するパスワードの場合、暗号化キー

秘密鍵、あるいはプライベートキー、初期化ベクトル、ソルトは内部処理で使われるだけで、利用者向けに使用されない用語のため、Chanel での用語はありません。

暗号文、HASH 値、あるいはフィンガープリントはそのまま使用しています。

11. (Annex 2)Chanel の仕組みについて

ここでは Chanel が文書をどのように暗号化して、それを復号するかを説明します。

ここからは Chanel で使用している一般的な用語ではなく暗号技術の用語を用いての説明となりますので、暗号技術を知らない場合は「Chanel で使用している用語について」の表を見ておいてください。

また、内容は IT の用語や技術についてある程度知っている人向けになります。

まず、Chanel の仕組みを説明するには暗号技術の大雑把な内容を把握しておく必要がありますので、最初に説明します。とはいってもその詳細をアルゴリズムや、暗号技術一般論まで説明することは本書の範囲を超えますので、Chanel の仕組みを知る上で最小限の説明にとどめます。

次に、Chanel がそれらの暗号技術をどのように組み合わせて暗号化・復号を行っているかを説明します。

1） Chanel で使われている暗号技術の概要

共通鍵暗号：

一般的な暗号化したファイルのやり取りに使われる、送信者が暗号化キーを使って暗号化し、受信者は暗号化と同じ暗号化キーを使い復号する仕組みです。暗号化 zip ファイルもその一つと言えます。

暗号化キーは、一般には「パスワード」と呼ばれます。

暗号化は、データの内容を一定サイズ毎のブロックに切り出して、ブロック毎に一見無意味なデータの羅列のようになるまでデータを崩し、ビット演算という特殊な計算処理を行うなどして暗号化します。このときに暗号化キーの値を取り込んで処理するため、復号には暗号化したときと同じキーが必要となります。

また、暗号化にはモードというものがあって、Chanel は CBC モードを使っています。

CBC モードとは、直前のブロックの暗号化結果が次のブロックの暗号化データに影響を与える仕組みです。細かい事を言うと、直前のブロックの暗号化結果を、次のブロックの平文にミックス（XOR という演算処理を行います）してから暗号化を行います。

これをやらないと、たとえば 1 ブロックが 5 文字だとして「ABCDE」を暗号化した結果が「85413」だったとすると、「ABCDEABCDE」を暗号化すると「8541385413」と同じ結果が繰り返されてしまいます。この仕組みはモードとして ECB モードという名前がついてはいますが、同じ値の繰り返しは攻撃者に解読のヒントを与えてしまい、暗号強度としては弱くなります。

CBC モードは前のブロックの暗号化の結果が次のブロックの平文に影響を与えるため、元のデータが繰り返していても暗号化の結果は繰り返しになりません。

では、データの先頭のブロックを暗号化する時には前のブロックの暗号化結果が無いためどうするのか。そのためには前のブロックの暗号化結果に相当するデータをランダムな値で作る必要があります。これを「初期化ベクトル」といいます。余談ですが、この「初期化ベクトル」が第三者に知られてしまったとしても、暗号強度に問題はありません。

Chanel では共通鍵暗号 GOST 24187-89 仕様のアルゴリズムを使っています。

共通鍵暗号は暗号化処理が高速という利点がありますが、送信者が受信者にどのような方法で暗号化キーを受け渡すかが問題となる欠点があります。

電話で伝えるのはどうでしょう。電話口で伝えては聞き間違いも起こります。聞き間違いを少しでも減らすためのフォネティックコード（大文字でアルファの A, 小文字でベイカーの b, 記号の# のように、聞き間違いを避けて単語で伝える方法）というものがありますが文字数が多いと口頭で伝えるのも大変です。また、盗聴されているかもしれません。

紙に書きだしてセキュリティ便で送るのも絶対ではありません。本当にセキュリティ便を無条件に信頼してよいのでしょうか。あるいは、途中で強奪される惧れは絶対にないと言い切れるでしょうか。

暗号化したデータをメールで送り、別メールで暗号化キーを送るやり方が広く使われているようですが、このやり方は極めてナンセンスです。何者かが暗号化したデータを送ったメールを傍受できるということは、暗号化キーを送ったメールも同様に傍受できるということだから、暗号文もパスワードも両方とも傍受されて何の苦労もなく暗号文を復号されてしまいます。

本格的に安全な方法を考え始めると、色々と懸念が湧いてきます。

公開鍵暗号：

最初に、手元に置いて漏洩しないように厳重に管理すべきプライベートキーと、プライベートキーと対になるパブリックキーの二つのキーを作成します。これをキーペアといいます。

日本語ではプライベートキーを秘密鍵、パブリックキーを公開鍵と言います。

パブリックキーはその名の通り、不特定多数の人が入手しても構わないキーで、暗号文を送信したい人にメールで送るなりネット上に公開するなりして渡すキーとなります。

公開鍵暗号の性質として、**パブリックキーで暗号化したデータは、パブリックキーでは復号できず、キーペアのプライベートキーでしか復号できない**という性質があります。

この性質を使い、受信者のプライベートキーが安全に管理されてさえいれば、送信者が広く公開されているパブリックキーで暗号化しても、対となるプライベートキーを持つ受信者にしか復号ができないため、データの機密性が保ちます。

これなら共通鍵暗号のように暗号化キーをどうやって受け渡すかという問題は生じません。そのかわり暗号化・復号では何十桁もの巨大な桁数の数字の計算を行うために、処理速度が遅いという欠点があります。

Chanel では Elgamal 暗号の計算を楕円曲線上で行う、楕円 Elgamal 暗号のアルゴリズムを使っています。

HASH 関数：

共通鍵暗号と似たような方法でデータの並び変えやビット演算などを使い、データを一定の長さの無意味な数値の羅列に置き換えたもので、その結果を HASH 値といいます。

この HASH 値は元のデータの長さを問わず、必ず HASH 関数で決まっている一定の長さになります。例えば HASH 関数で決まっている長さが 32 文字だとしたら、1 文字のデータの HASH 値をとったとしても、5 万文字のデータの HASH 値を取っても、結果は必ず 32 文字になります。

Chanel で使っている GOST R 34.11-94 は HASH 値の長さは 256 ビット、かなり乱暴な言い方をすると 32 文字です。「安全な連絡先の詳細」画面や、「解読」画面の状態表示部に出てきた「フィンガープリント」がまさしく HASH 値です。32 文字、と言いましたがフィンガープリントのアルファベットと数字が 32 文字ではありません。これは 256 ビット長の HASH 値を 1 バイト（＝8 ビット）毎に 16 進数表記に置き換えたものです。

HASH 関数の性質として、**暗号化と異なり HASH 値から元のデータを復元することはできません。**

また、全く同じデータを処理すると、全く同じ HASH 値が生成されるという性質があります。

例えば「おはよう」というメッセージをある HASH 関数で HASH 値をとった場合「0123456」になったとしたら、同じ HASH 関数を使えば、いつだれがどこで「おはよう」の HASH 値をとっても「0123456」になります。

さらに、「おはよぅ」のようにたった一ヵ所を小文字に変えただけでも、HASH 値は「952557」のように全く違ってしまうように作られています。

そういった性質のため、どのようなデータからどのような HASH 値が生成されるかを予測することはほぼ不可能です。

この仕組みにより平文が途中で改ざんされているかどうかが分かります。

たとえば受信者が平文の内容に疑念を抱いたら、HASH 値を取って送信者に会うなり電話するなりして HASH 値を突き合わせることで改ざんの有無がわかります。

HASH 関数はどれだけのサイズのデータでも決まった長さに変換するため、実際にはあらゆるデータが完全に異なる HASH 値に変換されるわけではなく、稀に全くことなるデータから全く同じ HASH 値が生成されることがあり得ます。これを **HASH の衝突**といいます。

HASH されるデータの方は、理論上長さ 0 から無限までデータの組み合わせのパターンがありますが、HASH 値のサイズが例えば 256 ビットであれば、当然ですが限界があります。サイズが決まっている以上は全てのデータを全て異なる HASH 値に変換する事ができないため、衝突は避けることができません。

そのため、HASH 関数はできるだけ衝突の確率が小さくなるように設計されています。

デジタル署名：

仕組みは公開鍵暗号と同じで、最初にプライベートキーとパブリックキーのキーペアを生成します。

公開鍵暗号のアルゴリズムにはもう一つの性質があり、**プライベートキーで暗号化されたデータはパブリックキーで復号できる**という性質があります。つまり公開鍵暗号の逆です。

デジタル署名ではこの性質を利用して、送信者が**送信者のプライベートキーで暗号化**した暗号文を、受信者が**送信者のパブリックキーで復号**します。

送信者のプライベートキーが安全に管理されているのであれば、受信者が送信者のパブリックキーで正しく復号できたということは、そのデータは間違いなく送信者が暗号化したものだということが分かります。

公開鍵暗号の性質を使ってデータに「これは私が書いた文章です」という宣言の意味で署名するようなものだから、デジタル署名というわけです。

仕組みが公開鍵暗号と同じということで、公開鍵暗号と同じく処理速度が遅いという欠点があります。

そのため、通常は元のデータの HASH 値をとり、その HASH 値に対してデジタル署名を行います。

HASH 値の長さ程度であれば処理に時間もかかりません。

データの内容が保持されているかどうかは HASH 値の性質上もし改ざんされていたとしたら HASH 値が異なってしまうことでわかります。

送信者のパブリックキーは不特定多数に対して公開されていますので、送信者のパブリックキーを持っている誰もがデジタル署名を復号して元の平文データの HASH 値を得ることはできますが、HASH 値から平文を復元すること

はほぼ不可能なので、データの機密性も保持されます。

Chanlel では GOST R 34.10-2001 仕様のアルゴリズムを使っています。これは楕円曲線を用いる、Elgamal とよく似たアルゴリズムになっています。

疑似乱数：

ここで言うのは暗号用の乱数生成であってゲームなどで使われる疑似乱数よりも真正乱数(例えばサイコロを振って出た目とか、石を複数ばらまいてどこに落ちたかなど)に近い、予測不可能な乱数を生成するアルゴリズムを言います。

暗号用の疑似乱数は、ハッシュ関数や暗号化を用いるなどで、予測不能な乱数を得る事ができるようなアルゴリズムになっています。

そのため処理速度が、ゲームなどで使われる乱数生成関数よりは遅くなります

Chanel では Ansi X 9.17 仕様のアルゴリズムを少し改造して使っています。

ちなみに、様々な言語で標準的に実装されているランダム関数は、実際には真の乱数ではなく、実行される都度、一見ランダムに見える数値の配列が定期的に循環して出現するが、ただその周期が非常に長いアルゴリズムとなっています。

たとえば 1, 7, 10, 69, 102, 88, 981…… と一見ランダムに出てきたとする。この後一万回、二万回とランダム関数を実行し続けると、ある時に再び全く同じ 1, 7, 10, 69, 102, 88, 981…… が現れます。もし仮に、この 1, 7, 10, 69, 102, 88, 981…… が現れて、次に現れるまでの間の数字の並びを調べたら、毎回まったく同じ並びで数字が表れていることが分かります。

このようなアルゴリズムで疑似乱数を生成した場合、コンピュータが得意とするビット演算などの処理だけで実装が可能なアルゴリズムが多いため、処理速度が速くなります。

速度が要求されて、乱数のセキュリティを考える必要がないゲームなどではこれで必要十分です。

ただ、ことが暗号技術となると、いくら周期が長く、何億回・何兆回周期になったとしても「予測可能である」ということは致命的な攻撃ポイントとなるため使えません。

圧縮技術：

圧縮技術は暗号技術と直接的な関わりはありませんが、暗号化前に平文を圧縮して、多少なりともデータのサイズを減らすために使われています。

圧縮技術は、データに特殊な操作をしてデータのサイズを圧縮し、逆に元のデータに戻す（「解凍」といいます）ことができるアルゴリズムになっています。一般的には zip、lha などが使われているが、Chanel では桁上がり無しの Range Coder を使っています。

実際には、圧縮データを作成した時に作成される解凍用の辞書データがある上、Chanel では暗号文だけではなく公開鍵暗号で暗号化された初期化ベクトルや共通鍵暗号の暗号化キー、デジタル署名、宛先となる人のパブリックキーの一部などが付与された総体が最終的に生成される暗号文となりますので、非常に短いデータを暗号化した場合、最終的にできあがる暗号文のサイズは元のデータよりもかなり大きくなります。

エンコード技術：

エンコード技術はバイナリデータをテキストデータに変換する仕組みで、これも暗号技術と直接的な関わりはありませんが、Chanel で取り入れているためにここで説明します。

バイナリデータをテキストに変換し（エンコードといいます）、またバイナリデータに戻す（デコードといいます）アルゴリズムです。

バイナリデータとは、人間にとって（ごく一部、16 進数を読んでこれは○○のデータだとか、これは○○命令でアドレス xxx の……　とかが判る特殊な人を除いて）は無意味な数値データで、テキストエディタなどで無理やり開くとところどころに読める文字や単語があるが、ほとんどが変な記号のようなものが表示されたり何も表示されなかったりするようなデータです。プログラムの実行ファイルや、Excel や Word の文書データ、圧縮されたファイルなどがバイナリデータです。

バイナリデータを、と言いましたが、実際にはテキストデータをエンコードしてしまうことも可能です。ただ、無駄に文字数が増えるだけだし、一々エンコードするのもデコードするのも手間だし、暗号化したみたいな無意味な文字の羅列に置き換わるけど、分かる人が見たらパスワードも何も使わずデコードされるから秘密にもならない、といったわけでやる意味が何もないので普通はそういう使い方をすることはまずないでしょう。

Chanel で使用しているエンコードは BASE64 です。

BASE64 はバイナリデータを 6 ビット毎に分解して、0b000000 から 0b111111 までのビット列に大文字小文字の英数字と一部の記号をあてはめた対応表に従って文字に置き換え、バイナリデータをテキストデータ(といってもやはり無意味なアルファベットと数字と記号の羅列だが)として表せるようにした仕組みです。

欠点は、英数字 1 文字はコンピュータが扱うデータとして 8 ビットの長さなので、6 ビットのデータを 8 ビットで置き換える分、エンコードされたサイズは元のデータのサイズより大きくなる点です。

その変わり、暗号化されたデータはまさにバイナリデータですが BASE64 で文字列化することでテキストとして保存できるようになり、メールに貼り付けて送付もすることもできるようになります。

戻すときは逆にテキストの文字を対応するバイナリデータに変換します。

2） Chanel が暗号化する仕組み

Chanel は PGP の仕組みを取り入れているため、いくつかの暗号技術を組み合わせた暗号化スイートとなっています。

重要なデータ作成：

重要なデータとは、公開鍵暗号用のプライベートキーとパブリックキー、デジタル署名用のプライベートキーとパブリックキー、各キーの暗号化に使う初期化ベクトルとソルトです。仕組み上、まずこれが無くては始まりません。

ソルトとは、パスフレーズ(Chanel を利用するためのパスワード)に付加して暗号化キーとするための値です。ソルトを付加する仕組みであれば、万一パスフレーズが漏れたとしてもソルトが無ければ完全な暗号化キーにはならず、当面は安全性が保たれます（とはいえパスフレーズがバレてしまったのであればさっさとパスフレーズを変更した方がよいことは言うまでもありませんが)。そのため本格的にセキュリティを考慮して使うのであれば、ソルトは USB メモリなどに保存しておいて肌身離さず持ち歩くなり安全な場所に隠すなりして、Chanel を使うときだけ接続して使う、という使い方になります。そうしておけば、何者かがあなたのパスフレーズを得たとしてもソルトを入手できないため、あなたの Chanel を利用することができません。

ただし、Chanel 自体がそこまでしなくてはならないほどの状況での使用を考慮していないため、デフォルトでは利用者に保存場所を選ばせる事はなく、Chanel フォルダ配下にある「_chanelkeys」フォルダにまとめて保存されます。

上記の重要なデータがない状態などの場合、公開鍵暗号とデジタル署名のキーペアの作成が行われます。キーペアが作成されるのは下記の場合です。

- Chanel 初回起動時
- 「安全な連絡先」の再作成時（つまり、公開鍵暗号とデジタル署名のキーペア再作成です）
- パスワードを忘れたとき（この場合、公開鍵暗号とデジタル署名のキーペアを再作成せざるを得ません）

キーペアを作成したら、疑似乱数でソルトと初期化ベクトルを生成し、ユーザが入力したパスフレーズにソルトを

付加して暗号化キーとし、初期化ベクトルを使って CBC モードの共通鍵暗号でプライベートキーを暗号化して保存します。

次回以降の Chanel の起動時、あるいは必要な場合にパスワード（パスフレーズ）の入力画面がポップされて入力を求められますが、これにはユーザ確認の意味もありますけれども、メインはこの時入力されたパスフレーズと保存されたソルトと初期化ベクトルを使って利用者のプライベートキーを復号しているのです。

暗号化：

Chanel で平文の暗号化を行うと、まず文書を暗号化するための初期化ベクトルと、共通鍵暗号の暗号化キー（セッションキーと言います）が疑似乱数で生成されます。

次に、平文の HASH 値を取得したあと、Range Coder で平文を圧縮してから暗号化されます。

平文の HASH 値にデジタル署名を付与します。

宛先として選ばれた「安全な連絡先」つまりパブリックキーを使ってセッションキーを暗号化します。これを宛先の数だけ行います。宛先の人数分、セッションキーを暗号化した暗号文ができるわけです。

また復号時の検証用にセッションキーの HASH 値も保存されます。

最後に平文の暗号文、デジタル署名、各宛先毎に暗号化された初期化ベクトルとセッションキー、宛先として選ばれた人のパブリックキーの一部（名前と E メールアドレス、パブリックキー作成日。受信者が復号時に、自分が宛先として含まれているかどうかの判定用）をひとまとめにして BASE64 でエンコードして画面に表示します。

これが、暗号化した結果出力されるアルファベットや数字、記号の無意味な羅列の正体です。無意味な英数字の羅列自体は単純に BASE64 エンコードした結果に過ぎず、これ自体が暗号化の直接的な結果というわけではありません。

復号：

まず BASE64 をデコードして全体をバイナリデータに戻します。

暗号文に付与されているパブリックキーの一部を見て、自分が受信者として含まれているかどうかを E メールとパブリックキー作成日から判定します。

自分が受信者に含まれていたら、受信者、つまり自分のプライベートキーを使ってセッションキーを復号します。

送信者側で取得したセッションキーの HASH 値も保存されているので、復号したセッションキーを受信者側でも HASH 値を取って突合して正しく復号できているかどうか判定します。

初期化ベクトルと復号されたセッションキーを使い、暗号文を復号します。

復号された平文の HASH 値と、デジタル署名を送信者のパブリックキーで復号して得た HASH 値を突合して、一致したらデジタル署名検証 OK になります。

最後に、復号した平文を画面に表示します。

12. (Annex3)Chanel ソースコードについて

1） ファイル一覧

Chanel のソースコードはリリースファイルの Source フォルダに同梱されています。ただし開発中に設計書を作成したわけでもなく、コメントも不十分な状態です。また開発期間が長期にわたっているため、コードの書き方や変数名の付け方も統一されておらず、非常に読みづらいコードとなっていますし、なんでこんな回りくどいコードにしてるの？　という箇所も多々あると思います。

このような形でソースコードを公開するのは非常に心苦しい限りではありますが、興味がある方向けに敢えて公開しております。

いきなりソースファイルだけというのもあんまりですので、せめてファイル一覧と簡単な説明を一覧表にしました。

ファイル名(**太字はフォルダ名**)	説明
¥chanel	
Footy2.as	リッチエディットコントロール"Footy2"使用のためのモジュールです。
imagelist.bmp	[安全な連絡先一覧]で使うイメージアイコンの画像ファイルです。
CryptLexer.cc	HSP 用統合開発環境 LeAntilla のコントロールファイル、コードコンプリート表示用です。
CryptLexer.ct	HSP 用統合開発環境 LeAntilla のコントロールファイル、コードチップ表示用です。
Footy2.dll	リッチエディットコントロール"Footy2"用の dll です。
hscallbk.dll	HSP のコールバック関数プラグインです。
longint.dll	HSP の多倍長整数プラグインです。
_chanel_win10.eng.hrt	英語版プログラムアイコン用のリソースファイルです。
_chanel_win10.jpn.hrt	日本版プログラムアイコン用のリソースファイルです。
Encript.hsh	Chanel のヘッダファイルです。

ファイル名(**太字はフォルダ名**)	説明
ecc.hsp	楕円曲線暗号計算用のソースコードです。
gosthash.hsp	HASH 関数計算用のソースコードです。
hspBASE64.hsp	Base64 エンコード・デコード用のソースコードです。
hspGost.hsp	GOST 暗号化・復号用のソースコードです。
hspXORShiftRandomizer128.hsp	XOR Shift 乱数生成器のソースコードです。これ自体は暗号用乱数として使えないので、Ansi X 9.17 仕様の乱数生成用初期値を作成するために使っています。
_Chanel.hss	Chanel のメインルーチン用ソースコードです。画面生成、プログラムの動きの全体制御をおこなっています。
decTohex.hss	十進数、十六進数の変換用ソースコードです。
deleteQuote.hss	メールの引用符">"を削除するソースコードです。
Elgamal.hss	Elgamal 暗号計算用ソースコードです。
folderTool.hss	フォルダ操作用のソースコードです。
getBitLength2.hss	バイナリデータのビット長を算出するソースコードです。
gostMessageDigest.hss	GOST のデジタル署名計算用ソースコードです。
hspGetBinaryChar.hss	文字列の数字をバイナリの数値に変換するソースコードです。
hspModStringToInt.hss	str 型と int 型の配列の相互変換と、pkcs5 パディングを行うソースコードです。
intArrayToLongInt.hss	int 型の配列を longint 型に変換するソースコードです。
KeyManager.hss	公開鍵・秘密鍵の管理全般と、暗号化・復号処理を行う、暗号化のコアモジュールです。

ファイル名(**太字はフォルダ名**)	説明
modBinaryData.hss	バイナリデータ操作用のモジュール変数です。
randomizerAnsiX9.17.hss	暗号化用の Ansi X 9.17 用ソースコードです。
rangecodernocarry.hss	桁上がり無しレンジコーダ用ソースコードです。
CryptLexer.kw	HSP 用統合開発環境 LeAntilla のコントロールファイル、キーワード色付け用です。
172x24d.png	画面用の画像ファイルです。
172x24o.png	画面用の画像ファイルです。
172x24u.png	画面用の画像ファイルです。
176x24d.png	画面用の画像ファイルです。
176x24o.png	画面用の画像ファイルです。
176x24u.png	画面用の画像ファイルです。
320x24d.png	画面用の画像ファイルです。
320x24o.png	画面用の画像ファイルです。
320x24u.png	画面用の画像ファイルです。
64x24d.png	画面用の画像ファイルです。
64x24o.png	画面用の画像ファイルです。
64x24u.png	画面用の画像ファイルです。
72x72d.png	画面用の画像ファイルです。
72x72o.png	画面用の画像ファイルです。
72x72u.png	画面用の画像ファイルです。
_Chanel.prj	HSP 用統合開発環境 LeAntilla のプロジェクトファイルです。
chanel.VisualElementsManifest.xml	英語版プログラムアイコンのスタート画面ピン留め用ア

ファイル名(**太字はフォルダ名**)	説明
	イコン制御ファイルです。
chanel_jpn.VisualElementsManifest.xml	日本語版プログラムアイコンのスタート画面ピン留め用アイコン制御ファイルです。
¥chanel¥source¥icons	
75x75Chanel_icon_001_eng.jpg	英語版プログラムアイコンのスタート画面ピン留め用スモールアイコン画像です。
75x75Chanel_icon_001_jpn.jpg	英語版プログラムアイコンのスタート画面ピン留め用ラージアイコン画像です。
150x150Chanel_icon_001_eng.jpg	日本語版プログラムアイコンのスタート画面ピン留め用スモールアイコン画像です。
150x150Chanel_icon_001_jpn.jpg	日本語版プログラムアイコンのスタート画面ピン留め用ラージアイコン画像です。
¥chanel¥source¥icons¥program_icon	
256x256Rusalka_icon_001_eng.ico	英語版プログラムのプログラムアイコンです。プログラムアイコン用リソースファイルに埋め込まれた画像の元ネタです。
256x256Rusalka_icon_001_jpn.ico	日本語版プログラムのプログラムアイコンです。プログラムアイコン用リソースファイルに埋め込まれた画像の元ネタです。

タイトル：テキスト暗号化ツール Chanel 利用ガイド

発行日：2022 年 8 月 28 日　第 1 版

発行者：タシャカネルラボ