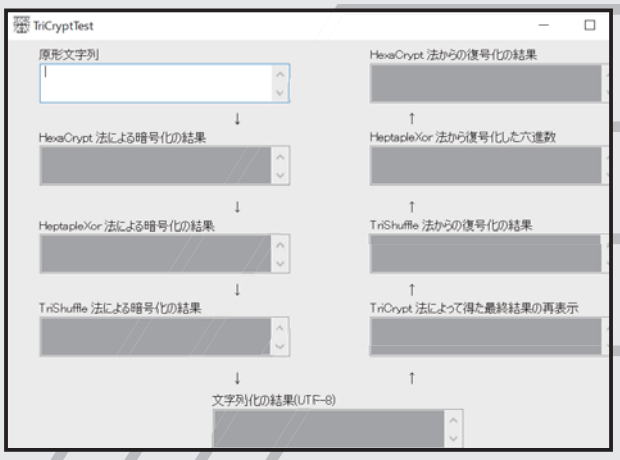
# TriCryptTest

## "TriCryptTest" とは？




### 0. 概要

"TriCryptTest" とは、筆者が考案したバイナリデータの暗号化技術 "TriCrypt" 法による暗号化・復号化のテストを行う目的で作成したアプリケーションです。 Visual Basic により実装されています。

### 1. TriCrypt 法の概要

TriCrypt 法とは、筆者が考案した三種類の暗号化方法を併用し、バイナリデータを三重に暗号化する技術です。いずれの暗号化方法もデータを不可逆破壊しないため、逆順での実行によりデータの復号化が可能です。内部でバイナリデータに適用される順に、それぞれ以下のような暗号化アルゴリズムが使われています。

#### HexaCrypt 法

プログラム内部で疑似的な六面体のサイコロを作り出し、バイナリデータをそのサイコロの出目に変換する暗号化法です。取得したバイナリ配列を 1 byte ごとに見てゆき、それをサイコロ 4 つの出目で表します。六面体のサイコロの出目は 1 ~ 6 なので、実質的には、1 byte のデータを 6 進法で表すことになります ( この際、サイコロの出目を、-1 して読み替えることになります。1 の目は 6 進法の 0, 5 の目は 6 進法の 4 とみなす、など )。

1 byte がとりうる最大値は 255 なので、$6^3 = 216 < 255 < 6^4 = 1296$ より、最大で 255 を取りうる数値を 6 進法で表すには、4 桁の桁数が必要になります。よって、サイコロを 4 つ使うことになります。

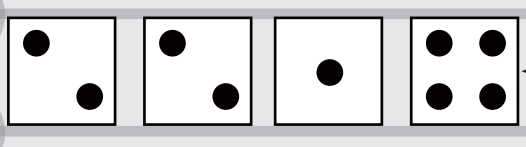
以下は、 1 byte の "255" という数値を、4 個のサイコロの出目に変換する際のイメージ図です。

255

$255 = 6^3 * 1 + 6^2 * 1 + 6^1 * 0 + 6^0 * 3$ 。

よって、6 進数で表現すれば $1103_{(6)}$ となる。

サイコロで表せる出目は 1 ~ 6 なので、各桁の数字にそれぞれ +1 して読み替える。これより 1103 → 2214 と読み替える。
この結果より、"255" という数字を、以下のサイコロ 4 つの並びで表す。





By Yuichi Watanabe
as of April, 2016

# TriCryptTest

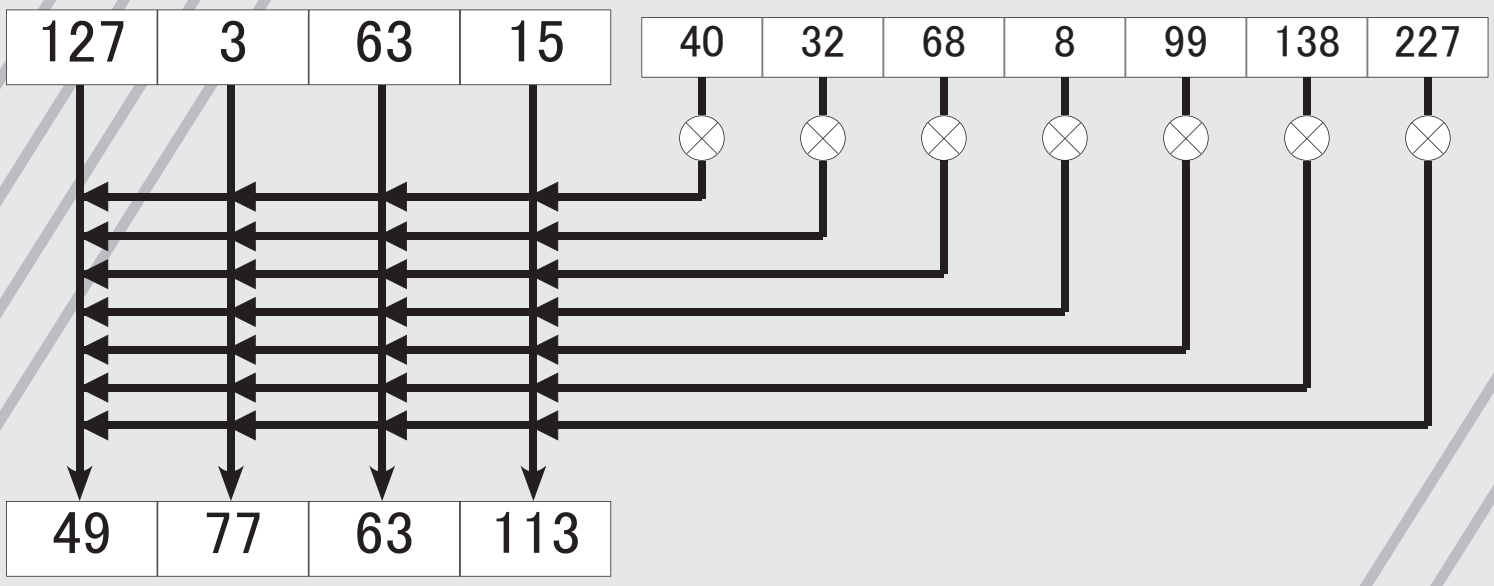
## "TriCryptTest" とは？

### HeptapleXor 法

「七重排他論理和演算」という意味合いを込めて命名した暗号化方法です。入力されたバイト配列内の 1 byte ごとに、7 つの byte 数で xor 演算を行い、その結果をバイト配列で出力します。

なお、 xor 演算に使う 7 つの byte 数は、プログラム実行時のタイムスタンプから、特定のアルゴリズムにより作成されます。実行時のタイムスタンプにより暗号化結果が変化するため、原則として理論上同一の暗号化パターンが二回以上利用される可能性は低くなります。これにより、クラッカーの暗号解読を困難にさせることを意図しています。

以下は、4 byte の配列を、HeptapleXor 法で暗号化・復号化する際のイメージ図です。利用している ″40, 32, 68, 8, 99, 138, 227″ の 7 つの byte 数は、「2016 年 4 月 1 日 12 時 34 分 56 秒」というタイムスタンプから作成された一例となります。




### TriShuffle 法

バイト配列 = トランプやカードゲームのデッキ ( 山札 )、バイト配列内の 1 byte = カード 1 枚と見立て、バイト配列を「シャッフル」する暗号化方法です。内部的には、以下に述べる 3 種類のシャッフル方法を、特定の順番で行っています。シャッフル方法の実行順番は、入力されたキーナンバーから決定されます。

以下、順番に ″1, 2, 3, 4, 5, 6, 7, 8, 9, 10″ という数値を持つ、長さ 10 のバイト配列を例として、シャッフル結果を例示しています。

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|

・ディールシャッフル

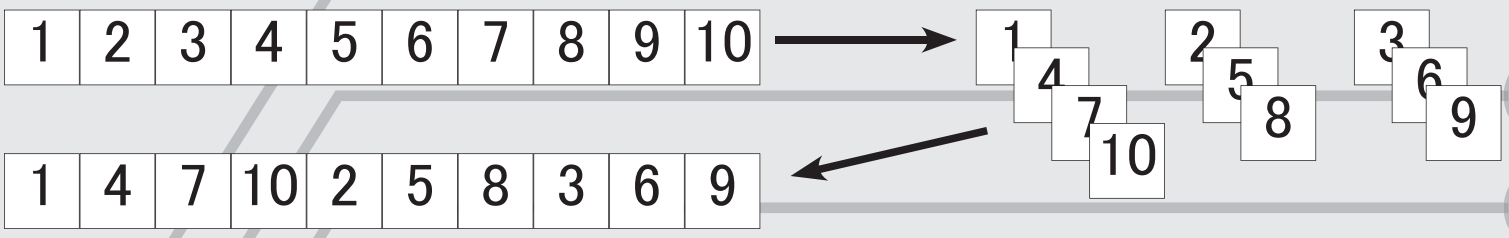
バイト配列をいくつかの小さい配列にいったん小分けし、それらを連結して再度一つのバイト配列を作り出すシャッフル方法です。バイト配列は、入力されたキーナンバーに応じて、2 ~ 6 個の山に分割され、再度まとめられます。

以下は、例示した 10 バイトの配列が、3 つの山に分割され、ディールシャッフルされる際の模式図になります。



By Yuichi Watanabe
as of April, 2016

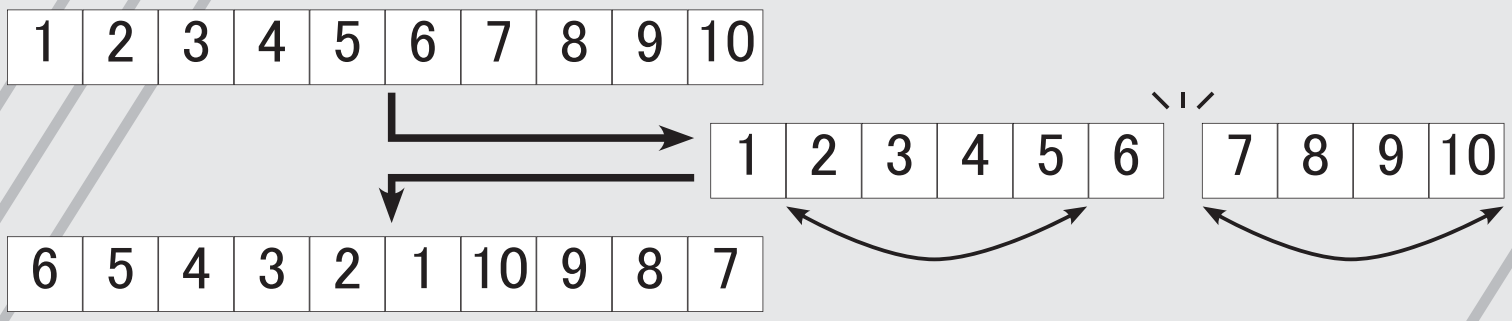
## "TriCryptTest" とは？




・フラップシャッフル

バイト配列をある 1 点で前半と後半に二分割します。その後、前半と後半の配列について並び順をそれぞれ反転させ、再連結して並べ直すシャッフル方法です。

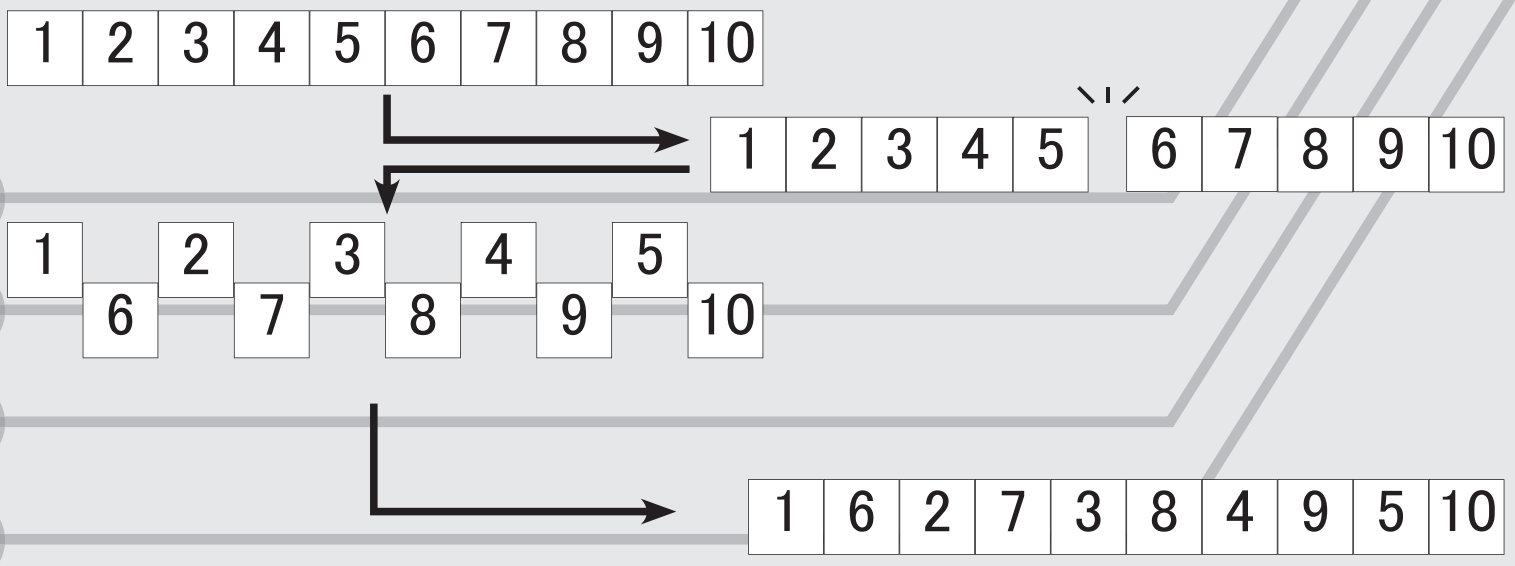
以下は、例示した 10 バイトの配列が、添え字数 = 5 の要素までを前半の配列とするよう二分割され、フラップシャッフルされる際の模式図になります。




・リフルシャッフル

バイト配列を配列の中央で二分割します。その後、前半と後半の配列要素を、先頭から順に互い違いに加えていくことによるシャッフル方法です ( バイト配列の長さが奇数である場合は、前半の配列の長さが、後半の配列の長さ + 1 になるよう分割されます )。入力されたキーナンバーにより、この作業を 1 ~ 3 回繰り返します。

以下は、例示した 10 バイトの配列が、1 回される際の模式図になります。






By Yuichi Watanabe
as of April, 2016

## "TriCryptTest" とは？

### 2. 起動方法

Windows がインストールされたマシンを用意し、以下の手順に従ってください(Windows 7 以降を推奨)。

i). .NET Framework 4 もしくはそれ以降のバージョンがインストール済みであることを確認してください。もし未インストールである場合、以下のリンク先からダウンロードしてください。

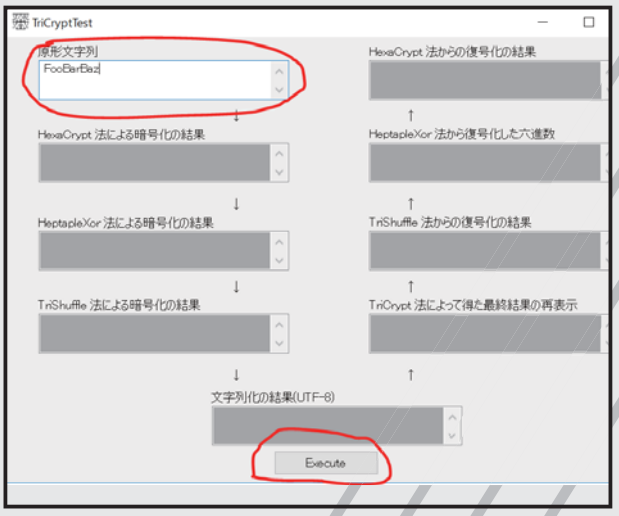
https://msdn.microsoft.com/ja-jp/vstudio/aa496123.aspx

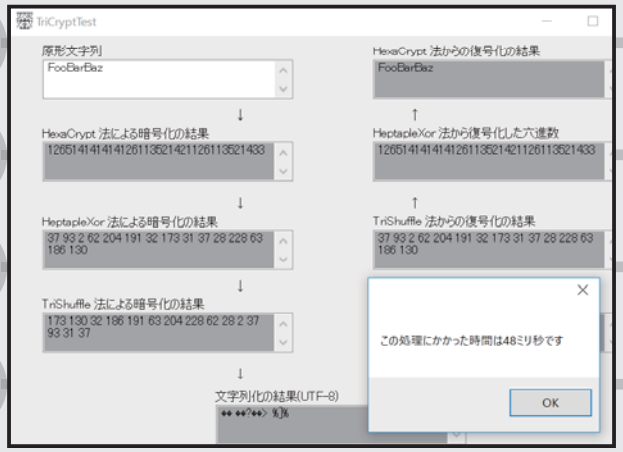
ii). 付属の HexaCryptTest.exe を実行します。

### 3. 利用方法

以下の手順に従ってください。

i). フォーム左上の「原型文字列」のテキストボックスに任意の文字列を入力してください。その後、フォーム下部の "Execute" ボタンをクリックしてください。



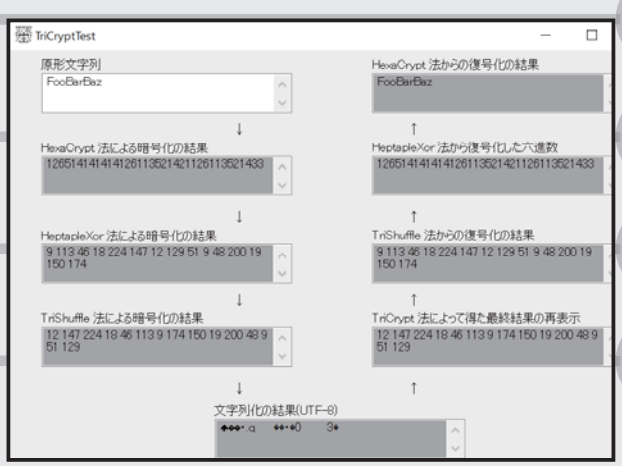

ii). 入力された文字が暗号化され、その後復号化されます。最後に、文字列の暗号化および復号化にかかった総計時間がメッセージに表示されます。



By Yuichi Watanabe
as of April, 2016

## "TriCryptTest" とは？

iii). ii). の作業を行った後、1 秒以上の時間をおいて、再度 "Execute" ボタンをクリックしてください。プログラムを実行する際のタイムスタンプが変わったことで、内部的に利用されている暗号化パターンが変化します ("HeptapleXor 法による暗号化の結果" 以降の出力が変化します)。
しかし、最終的に復号化される文字の内容は変わりません。




## CAUTION!

"TriCrypt.exe" 、および TriCrypt 法は、筆者が独自にバイナリデータ暗号化技術を研究し、その成果を発表する目的で作成されています。その範囲を超える意図をもって、この暗号化方法を利用することはご遠慮ください。また、この暗号化技術を利用してバイナリデータを暗号化した場合でも、そのバイナリデータのセキュリティ上の安全を保証することはできかねます。



By Yuichi Watanabe
as of April, 2016