暗号アルゴリズムの詳細評価

報告書

ストリーム暗号

MULTI-S01 編

1. 概要

提案方式は疑似乱数生成器 PANAMA の乱数生成器としての安全性に大きく依存します。そこで、PANAMA については、提案方式からの利用に限定して、別途評価を行いました。詳細は、「暗号アルゴリズムの詳細評価報告書　疑似乱数生成　PANAMA(MULTI-S01 内部)編」を参照してください。評価者は PANAMA については疑似乱数生成器としてのテストを合格したと判断し、提案方式の評価を行いました。

以下、テスト項目の概要と、テスト結果の概要を示します。提案方式は統計的性質に対して以下に述べるような性質があるものの、致命的な欠陥とは考えにくいと判断します。PANAMA の疑似乱数としての性質が優れていることから、評価者は提案方式を合格と判断します。

1.1. 統計的性質に関する評価について

テストデータを生成し、下記のテスト項目に対するテストプログラムを作成後、評価を行いました。テストデータ取得には「特殊なデータの生成」と「ランダムサンプリング」を併用しました。

ストリーム暗号の統計的性質（0/1 等頻度性テストなど）の評価方式として、様々な方法(提案方式の場合は内部で使用する PANAMA 単体の統計的性質に関するテスト)が考えられます。しかし、同一の評価尺度で各暗号の方式の特徴を抽出するためには、グローバルな評価方法を採用する必要があります。そこで、全てのストリーム暗号方式に適用可能な方式として、「平文 M と暗号文 C の排他的論理和」が乱数性をもつかどうかを評価することとしました。すなわち、

$$M \oplus C$$

の乱数性の評価を行いました。「乱数加算型ストリーム暗号方式」では、疑似乱数生成器の統計的テストを行えば評価が可能となり、安全性に関する議論がしやすくなるからでです。

従って以下に述べるテスト項目の結果は、提案方式で使用する PANAMA に対する統計的性質に対する評価ではないことに注意してください。MULTI-S01 での使用方法に限定した PANAMA についての統計的性質に対する評価は、「暗号アルゴリズムの詳細評価報告書　疑似乱数生成　PANAMA(MULTI-S01 内部)編」を参照してください。

また、このテストに不合格だからといって、乱数積和型ではないストリーム暗号方式である、本提案方式が安全ではないとは言い切れません。このことにも注意が必要です。暗号文（または、平文との差分）に疑似乱数と同レベルの統計的性質が必要とは言えません。疑似乱数の評価尺度を若干緩めて評価しました。

なお、当該項目に対するテストに合格したとしても、その項目に対する安全性を保障（証明）するものではありません。

1.1.1. 0/1 等頻度性テスト

提案方式は、本テストに合格したと判断します。詳細は、別冊（ストリーム暗号の評価　0/1 等頻度性テスト MULTI-S01 編）を参照してください。テストに不合格となる件数は許容範囲内と判断します。2 章も参照してください。

1.1.2. 連性テスト

提案方式は、本テストに合格したと判断します。詳細は、別冊（ストリーム暗号の評価　連性テスト　MULTI-S01 編）を参照してください。テストに不合格となる件数は許容範囲内と判断します。2 章も参照してください。

1.1.3. 長周期連性テスト

提案方式は、本テストに合格したと判断します。詳細は、別冊（ストリーム暗号の評価　長周期連性テスト MULTI-S01 編）を参照してください。テストに不合格となる件数は許容範囲内と判断します。2 章も参照してください。

1.1.4. 一様性テスト

提案方式は、本テストに合格したと判断します。詳細は、別冊（ストリーム暗号の評価　一様性テスト　MULTI-S01 編）を参照してください。テストに不合格となる件数は許容範囲内と判断します。2 章も参照してください。

1.1.5. 線形複雑度テスト

提案方式は、本テストに合格したと判断します。詳細は、別冊（ストリーム暗号の評価　線形複雑度テスト MULTI-S01 編）を参照してください。

1.1.6. 相互情報量テスト

提案方式は、本テストに合格したと判断します。詳細は、別冊（ストリーム暗号の評価　相互情報量テスト MULTI-S01 編）を参照してください。2 章も参照してください。

1.2. 用途に対する適合性

次の各項目について、机上もしくはマシンテストにより評価を行いました。

1.2.1. 周期(出力系列の再現性)の評価について

同一鍵を使用した場合、周期が設計値より小さくなることはないか、机上考察を行い、提案方式は本テストに合格したと判断します。詳細は、3 章を参照してください。

1.2.2. 同等出力系列発生条件の評価について

異なる鍵を使用して、同等（まったく同一ではない）出力系列が発生する条件があるか机上考察を行い、提案方式は本テストに合格したと判断します。詳細は、3 章を参照してください。

1.3. 出力系列に対する入力空間の大きさについて

鍵及び平文の入力空間が設計値より小さくなることはないか、机上考察を行い、提案方式は本テストに合格したと判断します。詳細は、4 章を参照してください。

1.4. ユーザの立場からの評価について

仕様書を用いて実装する状況を想定して、評価を行いました。5 章を参照してください

1.5. 暗号解析の立場からの評価について

統計的性質を利用するのではなく、アルゴリズム構造から解析を行う状況を想定して、評価を行いました。6 章を参照してください。

1.6. ドキュメントについて

ドキュメントに関する誤植、ドキュメントの内容について、評価を行いました。7 章を参照してください。

2. 統計的性質に関する評価について

2.1. テストに合格しなかった項目について

テスト項目に合格しなかった「0/1 等頻度性テスト」、「連性テスト」、「長周期連性テスト」および「一様性テスト」について言及します。詳細は、各別冊を参照してください。

これらのテスト項目は、「自己評価書 MULTI-S01」の 3.1.1 節における「1 ビットの頻度検定」、「連の検定」、「連の検定(長すぎる連の検出)」、「4 ビットの頻度検定」に相当します。しかし、「自己評価書 MULTI-S01」におけるこれらの評価は、疑似乱数生成器 PANAMA に対する評価です。評価者も（MULTIーS01 の利用方法に限定した）PANAMA の統計的性質については評価を行い、PANAMA 単体ではこれらのテストに合格しています。

本報告ではストリーム暗号に対し、同一の評価尺度で各暗号の方式の特徴を抽出するために、$M \oplus C$ の統計的性質を評価しています。また、暗号文、鍵については、ランダムに選択したものの他に、意図的に単純なものを使っているケースも用意しました。テストデータについては、別冊「MULTI-S01 暗号評価に利用したデータについて」を参照してください。

なお、暗号方式が「平文⊕PANAMA の出力する疑似乱数」の場合、すなわち「乱数加算型」の場合は、PANAMA の統計的性質に対する評価と同等の結果となります。

例えば、連の最大長は「自己評価書」によると、32 であることが期待されます。ところが、評価者のテストでは FIPS140 の基準を満たさないものとして、次のものを得ました。

#	連の長さ	Gaps	Blocks
1	34	1	2
2	35	0	1
3	36	0	1
4	39	0	1

この結果が直ちに MULTI-S01 の安全性に影響することはありませんが、構造が単純な「乱数加算型」ストリーム暗号方式よりも劣っているという事実は否めません。

他の性質(0/1 等頻度、連性、一様性)も、各々の 10 件程度の異常値を得ました。しかし、疑似乱数生成器と違い、暗号生成器の出力結果には乱数としての振る舞いをこのレベルまでは求めていません。上記の異常値が発生する条件(秘密鍵、乱数列番号、冗長度、平文)を考察した結果、評価者は、提案方式は統計的性質に関する評価に合格したと判断します。

2.2. 相互情報量に関する評価について

相互情報量は I(M;C)=H(M)-H(M|C)で定義されます。乱数加算型の場合、C=M⊕K であるから、

I(M;C)=H(M)-H(M|C)=H(M)-H(C⊕K|C)=H(M)-H(K)

となり、H(K)=H(M)なら

上記式=0

となります。また、乱数加算でなくても M=D(K,C)という復号において、C を固定したときの K の分布と D(K,C)の分布が同じならば、同一の結論が得られます。

MULTI-S01 は乱数加算型ではありません。しかし、XOR を複雑にしていますが、C を固定すると B の分布と M の分布が一致します。従って、上記の議論より相互情報量

I(M;C)=0

となり、完全暗号となります。しかし、実際は擬似乱数なので、情報理論的には、H(K)は高々256bit であり、

I(M;C)=H(M)-H(K)>=H(M)-256bit

となります。このいずれかになるかは、乱数が如何に真性乱数に近いかにかかっています。

すると、PANAMA の疑似乱数性を仮定する限り、相互情報量は理想的な値(上限値)になると判断します

3. 用途に対する適合性について

本章では用途に関する評価を行いました。

3.1. 周期（出力系列の再現性）の評価

ある鍵と、平文を使って生成される暗号文の周期がどの程度であるか、机上で評価しました。出力系列は PANAMA が生成する乱数系列の周期に依存します。PANAMA の周期は充分大きいと考えられる(暗号アルゴリズムの詳細評価報告書　疑似乱数生成　PANAMA(MULTI-S01 内部)編を参照)ので、暗号文の周期(出力系列の再現性)は充分大きいと考えます。

3.2. 同等出力系列発生条件の評価

本節では、ある鍵と平文を使って生成される暗号文が、ある鍵(別の鍵でもよい)と平文(別の平文でもよい)を使って生成される別の暗号文と同等の出力系列を発生させる方法が可能であるかどうか評価しました。

まったく同じではないが、その類似である同等の出力系列を、本書では次のように定義します。

定義 1

暗号文系列 $S(t)$ と $\overline{S}(t)$ が同等の出力系列とは、ある定数 i があって、次式が成立することとする。

$$S(t) = \overline{S}(t+i)$$

ここで、t はクロック（時間）である。

つまり、ある系列を i ビットずらした系列を得ることが容易であった場合はこの項目に不合格と判断します。

PANAMA の乱数性を仮定し、しかも平文が後続の暗号文に影響を与える構造であることを考えると、同等の出力系列となるような秘密鍵や平文を容易には発見できないと判断します。従って、本評価も合格と判断します。

4. 出力系列に対する入力空間の大きさ

4.1. 入力空間評価

入力空間のサイズについては、疑似乱数生成器 PANAMA の乱数性(全ての入力鍵に対して、出力系列は乱数となる)を仮定する限り、鍵長(256 ビット)と考えられます。

5. ユーザサイドからの評価

仕様書を用いて実装する状況を想定して、問題点がないか考察しました。

バイトオーダーの問題も含めよく配慮されており、実装に際しては問題はないと判断します。ただ、実際にストリーム暗号として使用した場合の問題点をコメントします。

5.1. 改ざんチェック機能と遅延問題について

提案方式では、改ざんチェック機能が付加されており、しかも、ストリーム暗号に対する平文長に対して改ざんチェック用の冗長ビットは定数サイズなので、非常に効率よく通信を行うことが期待できます。

ただ、仕様上、メッセージの分割単位は 2^{32} ブロック(2^{38} ビット、1GB)であり、巨大なデータのやり取りを行う場合は、遅延問題を無視できなくなります。実際、全データを受信して、はじめてデータに改ざんがあるかどうかを判断できるからです。

分割単位は暗号方式の安全性を根拠に得られた値と思いますが、もう少し小さくてもよいのではないでしょうか。

6. 暗号解析の立場からの評価

本章では、提案アルゴリズムを(アルゴリズムの逆演算を行う立場から)解析する場合の弱点などについてコメントします。

提案方式の安全性は、PANAMA が疑似乱数生成器として振舞うことを期待しており、実際に自己評価書には PANAMA の安全性に関する議論が記述されています。

ただ、統計的性質に関する評価において、PANAMA 単体での結果より悪くなる場合があります。評価者は、「暗号文と平文の排他的論理和」を疑似乱数とみなして評価を行ったわけですが、(自己評価書に記載されている)PANAMA 単体の統計的性質や、評価者が行った PANAMA の疑似乱数性評価の値よりも結果が悪くなると言うことは、「平文と PANAMA の生成する乱数との排他的論理和」による単純な排他的論理和型暗号方式よりも安全性が劣っている可能性は否定できません。ただ、<u>この事実が即安全でないという結論にはなりません</u>。

7. ドキュメントについて

「仕様書 MULTI-S01 暗号」の下記の記述内容が不明です。

#	場所	内容
1	P.11 下から 3 行目	$a_i \oplus a_{i+1} \oplus a_{i+}, 0 \le i < 17$ とありますが、最後の a のインデックス内容が不明です。

8. 性能について

ソフトウェアとして実装した際の性能評価を行いました。測定条件は下記の通りです。

#	項目	条件	
1	プログラム	提案方式の開発者により提供	
2	記述言語	C 言語	
3	測定装置の性能	Pentium III 600MHz, RAM128MB	
4	OS	Windows98	RedHat Linux 7.0J
5	コンパイラ	Visual C++ 6.0 Win32 Release 用オプション(/O2 等)	GNU C コンパイラ gcc 2.96 オプション –O2

実験では、10000000 バイトの平文から 10000016 バイトの暗号文を生成するのに要する時間を測定しました。暗号文の生成に要した時間を_ftime(Visual C++)/gettimeofday(gcc)を用いて 5 回測定しました。

測定結果を以下に示します。

#	測定結果	
	Windows98(Visual C++)	RedHat(gcc)
1 回目(秒)	1.610	2.159
2 回目(秒)	1.530	.2.515
3 回目(秒)	1.590	2.609
4 回目(秒)	1.590	2.139
5 回目(秒)	1.590	2.929
平均(秒)	**1.582**	**2.459**
レート(Mbps)	**48.226**	**31.026**

「自己評価書 MULTI-S01」の p.21 に記載されている値(Celeron 350MHz で 55Mbps)より悪い値が出ました。本テストでは、10MB という巨大なメモリを扱うため、キャッシュに入りきらないメモリへのアクセスが多発し、性能劣化したのではないかと思われます。