ストリーム暗号を用いて特定符号を回避する暗号化方式に関する提案

池田　裕樹 †　　岩村　惠市 †

† 東京理科大学
102-0073　東京都千代田区九段北 1-14-6

ikeda@sec.ee.kagu.tus.ac.jp

あらまし　ファイルフォーマットが定まった画像データを部分的に暗号化する事を考える．しかし, このような符号化画像にはそのフォーマット特有の意味を定めた系列を特定符号として設定している場合が多い．よって, 暗号化によりデータ部に特定符号が発生した場合, データ構造が変化するため正常に再生されず, 誤動作すると考えられる. この問題に対して特定符号を回避しながら部分暗号化を行う手法がいくつか提案されているが, 従来手法 [1,2] は非暗号化部分が比較的多く存在し安全性に問題がある．本論文ではストリーム暗号を用いて従来手法よりも非暗号化部分を減少させ安全性を向上させた暗号化方式を提案する.

An encryption scheme without output of the predefined specific code with stream cipher

Hiroki Ikeda†　　Keiichi Iwamura†

†Tokyo University of science
1-14-6 Kudankita,Chiyoda-Ku Tokyo,102-0073,Japan

ikeda@sec.ee.kagu.tus.ac.jp

Abstract　In general, coded images such as JPEG2000 have marker codes that define peculiar meanings of the file format as specific codes in header. We consider partial encryption of the coded image. If encrypted codes include a code which is the same as a specific code, some editors of the coded image occur errors. For this problem, some methods are proposed. However these methods are not secure since they have many non-encryption parts. In this paper, we propose a method that decreased non-encryption parts using stream cipher in one byte.

1　はじめに

近年, ネットワーク環境の普及に伴ってデジタルコンテンツの流通が盛んになっており, コンテンツが容易に取得できるようになった反面, 不正コピー等の著作権侵害が問題になっている．この問題の対策として, コンテンツの暗号化, 電子透かし, デジタル署名等の研究がされている．なかでも暗号化は、一般にデータ全体を暗号化する方法と, 部分的に暗号化する方法がある．データ全体を暗号化する方法は, データの一部を確認するときに全てのデータを復号する必要があり制約がある．部分的に暗号化する方法は, 機密を守りたい部分のみ暗号化することで検索等が容易である．本論文では, この部分暗号化を考える．ここで, 暗号化対象は従来法 [1,2] と同様に JPEG2000 とし, 圧縮前や圧縮処理中のデータの暗号化は, 圧縮効率の低下等の問題があるため考えない．よって, 本論文では圧縮された符号

化画像の部分暗号化・復号, 及び再生を考える. ただし, このような暗号化方式においては以下のような問題が発生する.

一般に符号化画像にはそのフォーマット特有の意味を定めた系列をマーカコード（以後, 特定符号）としてヘッダ部に設定している場合が多い. よって, 部分暗号化された画像を再生する場合, 部分暗号化によって特定符号が発生すれば, 再生器が誤動作することが予想される.

今回対象とするJPEG2000の特定符号は$FF90_h$〜$FFFF_h$の値であるため, 部分暗号化において回避したいのは,$FF90_h$〜$FFFF_h$の値である.

しかし, ブロック暗号を用いる文献 [1] の手法は1バイトの前半の半バイトは常に暗号化されないので, 暗号化部分が暗号文から容易に解析される可能性があり, 安全性に問題がある. ストリーム暗号を用いる文献 [2] は安全性について大きく改善されているが, 非暗号化部分が一部存在するという問題点は残されている.

本論文では, ストリーム暗号を用いて1バイト単位で処理を行う事により, 下記の要件を満たす暗号化方式を提案する.

(a). 部分暗号化したデータに特定符号を含まないことが保証される.

(b). 暗号化する部分を選択でき, 画質の制御が可能である.

(c). 元々のファイルフォーマットが持つスケーラビリティを保持した形で部分暗号化することができる.

(d). 少なくとも暗号文攻撃と既知／選択平文攻撃対して安全性が評価されている.

(e). 非暗号化部分が従来法 [1,2] に比べて少ない.

以降,2 章に従来方式の概要を示す.3 章において, 提案法を説明し,4 章で考察を行う. また, 提案法はストリーム暗号を用いるため, 同じストリーム暗号を用いている文献 [2] と比較を行うことにし, 文献 [1] の概要は頁数の都合で省略する.

2 従来方式

文献 [2] の暗号化アルゴリズムは,2 バイト単位で暗号化を行い, 特定符号になった場合は暗号化しないことで, 特定符号の発生を回避する方式である. 2 バイト単位で処理を行うのは, 特定符号が 2 バイトであるためである. JPEG2000 ストリームの最小単位が 1 バイトであり, 前後のバイトとの組み合わせが特定符号になる可能性があるため, 暗号化した結果とその前後のバイトとの組み合わせが特定符号でないかの検査も行うとしている.

以上を踏まえて, 文献 [2] の暗号化アルゴリズムの概要は以下のようになる.

(1) 2 バイト単位で暗号化を行い, 所定の検査により, その 2 バイトの暗号化結果が特定符号でないことを検査する.

(2) 検査により特定符号でないとされたその 2 バイトは. その暗号化結果と置き換えられる. そうでない場合は, 暗号化されずそのまま用いられる.

(3) 次の 2 バイトに進み,(1) の処理から繰り返す.

(1) における所定の検査とは, 暗号化された 2 バイトと, 次の 2 バイトの暗号文と平文に対して, 現在の 2 バイトとの全てのパターンが特定符号にならないことを検査することを指す. また, 復号は暗号化の逆処理である.

復号と暗号化は同じ処理であるので, 暗号化時に特定符号となる場合は, 復号の際も特定符号となる. この手法は, 全てのデータが暗号化対象であり, 非暗号化部分は正当な鍵を持つ利用者しかわからない. したがって, 選択平文攻撃は暗号文攻撃と同様に困難である. この手法により暗号化されたデータは, およそ 128 符号に 1 符号非暗号化部分が存在するが, 安全となっている.

位置が特定できない非暗号化部分を利用した有効な攻撃があるかは今のところ不明だが, 提案法は, より安全性を増すために, 非暗号化部分の発生を少しでも抑えることが目的である.

3　特定符号を回避する暗号化方式

文献 [2] は, 全ての組み合わせを検査し 1 組でも特定符号となった場合は暗号化しない. そのため, 結果として特定符号になりえない部分も暗号化していないことになる.

提案法は 1 バイト単位で処理を行う事により, 非暗号化部分を減少させる. 1 バイト毎に暗号化する場合, 暗号化した結果が FF_h でなければ, 次に来るバイトとの組み合わせは特定符号になり得ないので検査する必要がない. 暗号化して FF_h になったとき, 次のバイトを暗号化して特定符号になっているかどうか検査し, 暗号化を行うかを決める. ただし, 暗号化されなかった 2 バイト目が FF_h のとき 3 バイト目を暗号化して特定符号になる可能性がある. よって,2 バイト目の平文が FF_h のときも検査を行う必要がある.

復号は暗号化と同様の検査をし, 復号を行うかを決めるが, 暗号化の際に注目するバイトの平文が FF_h で, 次のバイトを暗号化させたものが FF_h になりかつ暗号化されなかった場合に, 暗号化結果は注目するバイトは暗号化されており, 次のバイトは暗号化されていないデータになる. この 2 バイトの復号結果は $FFFF_h$ となるため, 注目しているバイトは暗号化されているにもかかわらず, 復号されないという問題が発生する. そこで, 平文が FF_h であったときも検査を行う.

よって,1 バイト単位で処理する場合は平文もしくは暗号文が FF_h になった時点で検査を行うため, 前のバイトとの組み合わせを検査する必要がない. さらに, 後ろに続くバイトとの組み合わせは, 注目するバイトの暗号文と次のバイトの暗号文, 注目するバイトの平文と次のバイトの暗号文, 次のバイトの平文と 2 個先のバイトの暗号文の 3 組でよく, 最後の組み合わせは検査する必要がない場合もある. したがって従来法より非暗号化部分が少なくなる.

以上を踏まえて提案する暗号化アルゴリズムを以下に示す. 下記において, 第 i 番目のバイトを第 i バイトと呼び, それを暗号化したものを第 i 暗号化バイトと呼ぶ.

(1) $i = 1$ とする. ただし, 暗号化対象は特定符号が存在しない圧縮データである.

(2) 暗号化対象となる第 i バイトを暗号化し, 第 i 暗号化バイトを出力する.

(3) 第 i バイトが最終バイトなら, 第 i 暗号化バイトを第 i バイトの暗号化結果として確定し終了する.

(4) 第 i 暗号化バイトと第 i バイトが FF_h であるか調べる.

(5) 第 i 暗号化バイトと第 i バイトが共に FF_h でなかった場合, 第 i 暗号化バイトが第 i バイトの暗号化結果として確定される.$i = i+1$ とし,(2) の処理から継続する. 第 i 暗号化バイトと第 i バイトのどちらか一方が FF_h であった場合は, 第 $i+1$ 暗号化バイトを出力する.

(6) 第 i バイトが FF_h であった場合, 第 $i+1$ 暗号化バイトが FF_h であるか調べる.

(7) 第 $i+1$ 暗号化バイトが FF_h でなかった場合, 第 i 暗号化バイトが第 i バイトの暗号化結果として確定される. さらに、$i = i+1$ とし (2) の処理から継続する. 第 $i+1$ 暗号化バイトが FF_h であった場合, 第 i バイトと第 $i+1$ バイトの暗号化は行われない. さらに, 第 $i+1$ バイトが最終バイトなら終了する. そうでなければ,$i = i+2$ とし (2) の処理から継続する.

(8) 第 i 暗号化バイトが FF_h であった場合, 第 $i+1$ 暗号化バイトが 90_h ～FF_h であるか調べる.

(9) 第 $i+1$ 暗号化バイトが 90_h ～FF_h でなかった場合, 第 i 暗号化バイト, 第 $i+1$ 暗号化バイトがそれぞれ第 i バイト, 第 $i+1$ バイトの暗号化結果として確定される. さらに, 第 $i+1$ バイトが最終バイトなら終了する. そうでなければ,$i = i+2$ とし (2) の処理から継続する. 第 $i+1$ バイトが 90_h ～FF_h であった場合, 第 i イトと第 $i+1$ バイトの暗号化は行われない. 第 $i+1$ バイトが最終バイトなら終了する.

(10) 第 $i+1$ バイトが FF_h であるか調べる.

(11) 第 $i+1$ バイトが FF_h でなかった場合,$i=i+2$ とし (2) の処理から継続する. 第 $i+1$ バイトが FF_h であった場合, 第 $i+2$ バイトを暗号化し, 第 $i+2$ 暗号化バイトを出力する.

(12) 第 $i+2$ 暗号化バイトが 90_h ～FF_h であるか調べる.

(13) 第 $i+2$ 暗号化バイトが 90_h ～FF_h でなかった場合, 第 $i+2$ 暗号化バイトが第 $i+2$ バイトの暗号化結果として確定される. 第 $i+2$ 暗号化バイトが 90_h ～FF_h であった場合, 第 $i+2$ バイトの暗号化は行われない. 第 $i+2$ バイトが最終バイトなら終了する. そうでなければ $i=i+3$ とし (2) の処理から繰り返す.

復号アルゴリズムは以下のようになる.

(1) $i=1$ とする. 復号対象は予め上に示した暗号化アルゴリズムにより暗号化されたデータである.

(2) 復号対象となる第 i バイトを復号し, 第 i 復号バイトを出力する.

(3) 第 i バイトが最終バイトなら, 第 i 復号バイトを第 i バイトの復号結果として確定し終了する.

(4) 第 i 復号バイトと第 i バイトが FF_h であるか調べる.

(5) 第 i 復号バイトと第 i バイトが共に FF_h でなかった場合, 第 i 復号バイトが第 i バイトの復号結果として確定される.$i=i+1$ とし, (2) の処理から継続する. 第 i バイトと第 i バイトのどちらか一方が FF_h であった場合は第 $i+1$ バイトを復号し, 第 $i+1$ 復号バイトを出力する.

(6) 第 i バイトが FF_h であった場合, 第 $i+1$ 復号バイトが FF_h であるか調べる.

(7) 第 $i+1$ 復号バイトが FF_h でなかった場合, 第 i 復号バイトは第 i バイトの復号結果として確定される. さらに $i=i+1$ とし (2) の処理から繰り返す. 第 $i+1$ 復号バイトが FF_h であった場合, 第 i バイトと第 $i+1$ バイトの復号は行われない. さらに, 第 $i+1$ バイトが最終バイトなら終了する. そうでなければ,$i=i+2$ とし (2) の処理から継続する.

(8) 第 $i+1$ 復号バイトが 90_h ～FF_h であるか調べる.

(9) 第 $i+1$ 復号バイトが 90_h ～FF_h でなかった場合, 第 i 復号バイト, 第 $i+1$ 復号バイトがそれぞれ第 i バイト, 第 $i+1$ バイトの復号結果として確定される. さらに, 第 $i+1$ バイトが最終バイトなら終了する. そうでなければ,$i=i+2$ とし (2) の処理から継続する. 第 $i+1$ バイトが 90_h ～FF_h であった場合, 第 i バイトと第 $i+1$ バイトの復号は行われない. 第 $i+1$ バイトが最終バイトなら終了する.

(10) 第 $i+1$ バイトが FF_h であるか調べる.

(11) 第 $i+1$ バイトが FF_h でなかった場合,$i=i+2$ とし (2) の処理から継続する. 第 $i+1$ バイトが FF_h であった場合, 第 $i+2$ バイトを復号し, 第 $i+2$ 復号バイトを出力する.

(12) 第 $i+2$ 復号バイトが 90_h ～FF_h であるか調べる. 第 $i+2$ 復号バイトが 90_h ～FF_h でなかった場合, 第 $i+2$ 復号バイトが第 $i+2$ バイトの復号結果として確定される. 第 $i+2$ 復号バイトが 90_h ～FF_h であった場合, 第 $i+2$ バイトの復号は行われない. 第 $i+2$ バイトが最終バイトなら終了する. そうでなければ $i=i+3$ とし (2) の処理から継続する.

4 考察

以下では提案法が次の命題を満たす事を証明する.

命題 1 暗号化データは特定符号を含まない.

命題 2 暗号化データは正しく元のデータに復号される.

ここで、以下の条件を定義する.

条件 A 第 i 暗号化バイトが FF_h でない.

条件 B 第 i 暗号化バイトと第 $i+1$ 暗号化バイトが特定符号でない.

条件 C 第 i 暗号化バイトと第 $i+1$ 暗号化バイトが特定符号で, 第 $i+1$ バイトが FF_h でない.

条件 D 第 i 暗号化バイトと第 $i+1$ 暗号化バイトが特定符号で, 第 $i+1$ バイトが FF_h であり, 第 $i+2$ 暗号化バイトが 90_h ～FF_h でない.

条件 E 第 i 暗号化バイトと第 $i+1$ 暗号化バイトが特定符号で, 第 $i+1$ バイトが FF_h であり, 第 $i+2$ 暗号化バイトが 90_h ～FF_h である.

命題 1 の証明

(i) 条件 A の場合 : 暗号化アルゴリズムにより, 第 i 暗号化バイトが選択されるが, 条件 A により, これらが特定符号となることはない.

(ii) 条件 B の場合 : 暗号化アルゴリズムにより, 第 i 暗号化バイト, 第 $i+1$ 暗号化バイトが選択されるが, 条件 B より, これらが特定符号となることはない.

(iii) 条件 C の場合 : 暗号化アルゴリズムにより, 第 i バイト, 第 $i+1$ バイトが選択されるが, これらが特定符号でないことは明らかである. また, 条件 C よ, 第 $i+2$ 暗号化バイトがどのような値でも, 第 $i+1$ バイトと第 $i+2$ 暗号化バイトは特定符号になりえない.

(iv) 条件 D の場合 : 暗号化アルゴリズムにより, 第 i バイト, 第 $i+1$ バイト, 第 $i+2$ 暗号化バイトが選択される. 条件 D より第 $i+1$ バイトと第 $i+2$ 暗号化バイトをつなげたものは特定符号ではないので, 第 i バイトから第 $i+2$ バイトまでに特定符号は含まれない.

(v) 条件 E の場合 : 暗号化アルゴリズムにより, 第 i バイト, 第 $i+1$ バイト, 第 $i+2$ バイトが選択されるが, これらに特定符号が含まれないことは明らかである.

命題 2 の証明

(i) 暗号化時に条件 A かつ第 i バイトが FF_h でない場合 : 条件により, 暗号化された第 i 暗号化バイトが選択されているが, 復号アルゴリズムにより復号が行われ正しい平文に戻すことが出来る.

(ii) 暗号化時に条件 A かつ第 i バイトが FF_h であった場合 : 条件により, 暗号化された第 i 暗号化バイトが選択されている. 復号アルゴリズムにより, 第 i バイトの復号は第 $i+1$ バイトの復号結果で決まるが, 今回の条件では, 第 $i+1$ バイトの復号結果は 00_h ～$8F_h$ であるので復号が行われ, 正しい平文に戻すことができる.

(iii) 暗号化時に条件 B の場合 : 条件により, 暗号化された第 i 暗号化バイト, 第 $i+1$ 暗号化バイトが選択されている. 第 i 復号バイトと第 $i+1$ 復号バイトは特定符号とならないので, 復号が行われ, 正しい平文に戻すことができる.

(iv) 暗号化時に条件 C の場合 : 条件により, 暗号化されていない第 i バイト, 第 $i+1$ バイトが選択されている. 復号した第 i 復号バイトと第 $i+1$ 復号バイトは条件より, 特定符号となるので復号が行われず正しい平文に戻すことができる.

(v) 暗号化時に条件 D の場合 : 条件により, 暗号化されていない第 i バイト, 第 $i+1$ バイトと暗号化されている第 $i+2$ 暗号化バイトが選択されている. 復号した第 i 復号バイトと第 $i+1$ 復号バイトは条件より, 特定符号となり. 第 $i+2$ 復号バイトは 00_h

〜$8F_h$ となる. したがって, 第iバイトと第$i+1$バイトは復号が行われずに, 第$i+2$バイトは復号が行われ, 正しい平文に戻すことができる.

(vi) 暗号化時に条件 E の場合 : 条件により, 暗号化されていない第iバイト, 第$i+1$バイト, 第$i+2$バイトが選択されている. 復号した第i復号バイトと第$i+1$復号バイトは条件より, 特定符号となり, 第$i+2$復号バイトは90_h 〜FF_h となる. したがって第iバイトから第$i+2$バイトまで復号が行われず, 正しい平文に戻すことができる.

以上より, 提案法は命題 1, 命題 2 を満足する.

安全性　提案法は, 文献 [2] と同様に, 非暗号化部分を特定出来るのは正当な鍵を持つ利用者のみである. したがって, 暗号文攻撃, 既知/選択攻撃は困難である. さらに提案法において暗号化されない確率は以下の場合である. ただし, 暗号化および圧縮処理を完全なランダム化とみなす.

(1) 第i暗号化バイトと第$i+1$暗号化バイトが特定符号の場合

(2) 第iバイトと第$i+1$暗号化バイトが共にFF_h の場合

(3) (1) かつ第$i+1$バイトと第$i+2$暗号化バイトが特定符号の場合

(1) から (3) の確率を P_1 から P_3 とすると, それぞれ以下のように表せる.

$$P_1 = \frac{1}{2^8} \times \frac{1}{2^8}$$

$$P_2 = \frac{1}{2^8} \times \frac{7 \times 16}{2^8}$$

$$P_3 = P_1 \times \frac{1}{2^8} \times \frac{7 \times 16}{2^8}$$

文献 [2] と比較しやすくするために,2 バイト単位で確率を計算する. 提案法は,(1) と (2) の場合で 2 バイトが暗号化されない. また,(3) の場合では 3 バイト暗号化されないが, 初めの 2 バイトが暗号化されない確率は P_1 で既に考慮しているので, P_3 の確率は 1 番後ろの 1 バイトだけが暗号化されない確率とみなす. したがって, 提案法によって暗号化されない確率は以下の通りである.

$$P_1 + P_2 + \frac{P_3}{2} \approx \frac{1}{579}$$

文献 [2] の暗号化方式に比べ, 非暗号化部分が $\frac{1}{5}$ ほどになっている. 非暗号化部分が多いほど解読される危険性が増すので, 非暗号化部分が少ないほど安全である.

5　まとめ

提案法は，命題 1 で証明したように, 要件 (a) を満たしている. また, ストリーム暗号を用いているため, 暗号化対象となるデータに特定符号が含まれていなければ, そのデータ長は任意に設定できる. したがって, レイヤ単位や画像の部分単位で暗号化することが可能であり, 暗号化後のデータ長は変化しないため, ファイルフォーマットのスケータビリティが保持されている. よって, 要件 (b), 要件 (c) を満たしている. さらに, 安全性の項で述べたように要件 (d), 要件 (e) を満たしている.

以上より本論文では，初めに示した全ての要件を満たす暗号化方式を提案した.

参考文献

[1] 貴家仁志，今泉祥子，渡邊修 :「マーカコードの発生を考慮した JPEG2000 符号化画像の情報開示法」，電子情報通信学会論文誌，Vol. J86-D-II，No. 11，pp.1628–1636，2003 年 11 月.

[2] 岩村惠市，林淳一 :「JPEG2000 符号化画像のマーカコード発生を回避できる暗号化方式」，電子情報通信学会論文誌，Vol. J90-A，No. 11，pp.839–850，2007 年 11 月.