M-94　MPEGストリーミングのためのブロック暗号処理方式

A Block Cipher Mode for MPEG Streaming

佐藤　雅史†　山本　秀樹†
Masashi Sato　Hideki Yamamoto
†沖電気工業株式会社 ブロードバンドメディアカンパニー
Broadband Media Company, Oki Electric Industry Co.,Ltd.

1．はじめに

ブロードバンドの普及に伴い、より高画質の映像を使ったサービスが可能となってきている。また、符号化技術の進歩により、MPEG-4 ASP(Advanced Simple Profile)のような、従来の MPEG より低ビットレートで画質のよい映像が配信可能になっている。一方でコンテンツを供給する側からは、コンテンツ保護のためのストリーム暗号化が要求されている。サービスプロバイダ側からすると、コンテンツ配信におけるストリームの暗号化はサーバや端末での負荷をできるだけ増加させない高速なものが必要となる。

本研究では上記の要求を満たす速度を持つブロック暗号化方式として OFB モードを改良した方式を開発する。本方式は一度生成した擬似乱数ビット列をある程度スライドさせながら重複して使うことで高速化をはかる。本稿では以下 SlidingOFB と呼ぶ。

2．従来技術の問題点

2.1.OFB モード

MPEG ストリームの暗号化には OFB モードが多く用いられている[1]。OFB を使う方式として、各パケットに鍵と初期ベクトルを持たせる方式と、一つのコンテンツを流すすべてのパケットに同一の鍵と初期ベクトルを使用する方式が考えられる。

2.2.処理速度の問題

2.1.で述べた方式のうち前者は、例えばディジタル放送で MPEG-2 の暗号化に使用されている[2]。しかし実際のブロードバンド配信で主流となっている MPEG-4 では、Audio パケットの大きさはおおよそ 200 バイトであり（Audio 単体で 64kbps とした場合）、このパケットに対してリアルタイムに合計 32 バイトの鍵と初期ベクトルを書き込んでいくのは、サーバの処理能力・ネットワークのトラフィック双方にとって大きな負担となる。

2.3.パケットロスの問題

一方、後者は前者より処理は軽くできるがパケットロスに弱いという欠点がある。OFB モードでは暗号化対象との XOR 演算に用いた擬似乱数ビット列の終点から、また暗号化関数を用いて擬似乱数ビット列の延長を行う。このためパケットロスが発生すると、サーバが暗号化に使った擬似乱数ビット列と、クライアントが復号化に使う擬似乱数ビット列が違うものになってしまう。そこで著者らは、後者を改良して、高速化とパケットロス耐性を実現させた。

3．SlidingOFB の仕組み

3.1.概要

SlidingOFB では各パケットに番号を割当て、擬似乱数ビット列の先頭から何バイト目を使うかを決定できるようにする。これによりクライアント側でパケットロスが発生しても、パケット番号を参照することで復号化が滞りなく行えるようになる（図1）。更に生成した擬似乱数ビット列をある程度重複して再利用できるようにして高速化を行う（図2）。

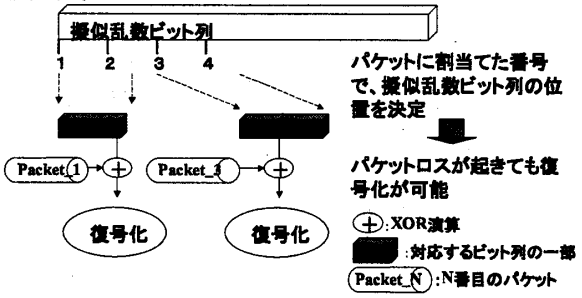


図1　パケットロスが発生した時の対処

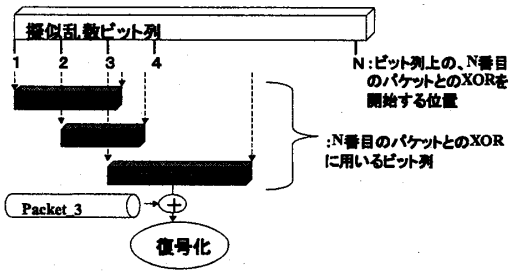


図2　擬似乱数ビット列の再利用

3.2.詳細

SlidingOFB ではあらかじめ擬似乱数ビット列を生成しておく。ある程度重複して再利用するためには各パケットの長さの最大値（以下 maxplen）よりも長くなくてはならないので maxplen×4 バイトの擬似乱数ビット列を生成しておく。擬似乱数ビット列の何バイト目を使うかを決定するためのパケットへの番号割当てには RTP のシーケンス番号（以下 seq）を用いる。これであればすべてのパケットに自動的に振り当てられ、単調増加であるため計算も単純になる。具体的なパケットと XOR を行う擬似乱数ビット列の位置として、擬似乱数ビット列の先頭から seq×slide バイト目を用いる。seq に掛ける定数 slide を調整することで重複の割合を変更することが可能である。これにより OFB と同じく、復号化に使用する擬似乱数ビット列をまったく重複させないことが可能であり、必要に応じて安全性と速度のバランスを取ることができる。ただしメモリ上の擬似乱数ビット列の長さは有限であるので、

$$position = (seq \times slide) \bmod maxplen \times 4$$

をパケットと XOR を開始する擬似乱数ビット列上の位置としている。更に擬似乱数ビット列を maxplen バイト毎に seq をトリガーとして更新を行っていく。

4．SlidingOFBの評価

通常のOFBと比較してのSlidingOFBの評価を3つの観点から行う。

4.1.暗号化単体での評価

第1に擬似乱数ビット列上での、隣接パケットとのXOR位置のスライド量を変化させた場合の処理速度を計測した（表1）。10kbyteのパケットを用いたので、表1の10240byteスライドしたものは擬似乱数ビット列の再利用を行わないという意味で従来のOFBモードと同じ安全性を持つ。そのデータと16byteスライドさせたものとを比較すると、処理速度に71.8倍の差が出た。

表1　暗号化方式の処理速度

隣接パケットとのXOR位置のスライド量(byte)	処理速度(Mbit/sec)
16	8963.8
1024	996.8
5120	214.7
10240	124.8

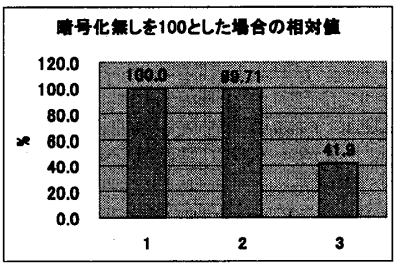
4.2.サーバ上での評価

第2に2.で紹介したパケットに鍵と初期ベクトルを持たせて公開鍵で暗号化した上で持たせる方式と、SlidingOFBがサーバにかける負荷を比較した。なお実際のMPEGの再生方式を考えるとイントラフレーム分、つまり30回に一度すべてのパケットに対して書き込めばよいのでここでもそうしている。方法としてはサーバに同時多数アクセスを行い、暗号化を行いながら実際に何本のストリームを配信できるかを調べた。表2は暗号化無しを100とした場合の各方式の相対値である。横軸の数字はそれぞれ

1：暗号化無し
2：SlidingOFBで1kバイトスライドしたもの
3：通常のOFB

に対応している。SlidingOFBが通常のOFBに比べて、ストリームの配信本数の減少率が約200分の1であることが分かる。

表2　各方式で配信できたストリーム本数の相対値



4.3.パケットロス耐性に対する評価

第3にパケットロスの発生する実際のインターネット環境で、OFBとSlidingOFBとを比較したところ、従来のOFBではパケットロスが起きるとそれ以上再生できなかったが、SlidingOFBでは上手く再生できた。更に画像も比較したところ、暗号化されていないコンテンツと、SlidingOFBで暗号化されたものとで視覚的に認識できる差は見られなかった。

5．安全性についての考察

ここでは従来のOFBと比較しての安全性について述べる。OFBの問題点としてよく知られているのは、擬似乱数ビット列とのXORを行っているため、復元される平文に対して予想できる形で変更を加えられることである[1]。このリスクはSlidingOFBでも同じである。ただしSlidingOFBの場合では擬似乱数ビット列をある程度重複して使用している。そのためpacket_1に対応する擬似乱数ビット列の中身を盗聴されると、packet_2以降の擬似乱数ビット列の、データの一部も知られてしまうことになる。しかしここで扱っているのはMPEGデータである。テキストデータと比べて、イントラフレームとその差分で構成されているMPEGデータは、平文の一部から全体を推測するのが難しい。また、ネットワーク上で配信する時に一つのフレームをパケット分割していることが多いので、もし平文の一部から全体を推測できるならそのリスクは通常のOFBでも同様に発生する。つまり、MPEGデータの暗号化に限れば、SlidingOFBは通常のOFBと同程度に安全といえる。

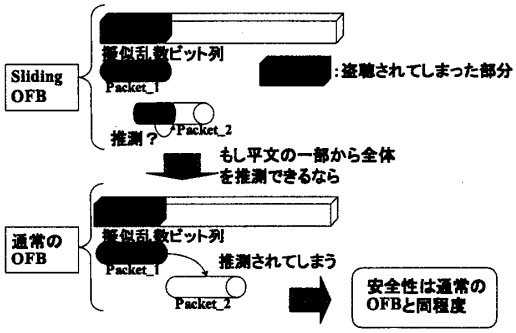


図3　通常のOFBとSliding OFBの安全性の比較

6．まとめと今後の課題

本稿では従来のOFBよりも高速で、パケットロスにも強いブロック暗号処理方式SlidingOFBについて報告した。処理速度を従来のOFBと比較すると、実験結果は5.1.によると単体で71.8倍、5.2.によるとサーバ上で動作した場合は約200倍の差が出る。またパケットロスが起きると従来のOFBでは復号化できない場合でも、SlidingOFBでは上手く復号化を継続できた。これにより、現実のブロードバンド環境上で、従来の技術よりも有用性が高いと推測できる。

今後の課題としては、コンテンツのダウンロード時の応用、マルチキャストでの応用、スライド量の指定方法の検討が挙げられる。

本研究は、通信・放送機構の委託研究として行われた。

[参考文献]

[1]ウィリアム・スターリング著,石橋啓一郎,三上荘子,福田剛士訳：暗号とネットワークセキュリティ,pp.93-100,ピアソン・エデュケーション,東京,2001.

[2]社団法人電波産業界：BSデジタル放送限定受信方式標準規格 ARIB STANDARD ARIB STD-B25 1.0版,pp.13,東京,Oct.1999.

以上