漏洩者の特定と配信停止が可能なマルチキャスト配信方式

Contents Multicast Method Enabling Leakers Tracing and Exclusion

豊島 鑑 [1] Kan Toyoshima
井上 武 [1] Takeru Inoue
上松 仁 [1] Hitoshi Uematsu
高橋 宏和 [1] Kazuhiro Takahashi
仁科 五月 [2] Satsuki Nishina
高木 剛 [2] Tsuyoshi Takagi
湊 真一 [3] Shinichi Minato

NTT 未来ねっと研究所 [1] NTT Network Innovation Labs.
公立はこだて未来大学 システム情報科学部 [2] Future University - Hakodate
北海道大学 大学院 情報科学研究科 [3] Hokkaido University

1 はじめに

ディジタルシネマ映像を始めとする高価なディジタルコンテンツのネットワーク配信は、その高価さ故に高いセキュリティを要求され、現在のところ、各受信者（映画館など）毎に個別に暗号化された後、各受信者に向けてユニキャストで配信されている [1]。しかしながら、今後、配信先が増え、また、マルチキャストによる配信が一般化していくにつれて、マルチキャスト配信への要望が次第に強くなると予想される。

今後、高価なディジタルコンテンツをマルチキャストでネットワーク配信しようとした場合、現在行われているマルチキャスト配信では IP Milticast プロトコルや Flexcast[2] などが使われているが、何れも一つのコンテンツをルータ等のネットワーク内装置でコピーしているため、どの受信者も同一コンテンツを受信している。このため、もしコンテンツ漏洩が起こった場合に、どの受信者が漏洩したかを特定することは困難である。

また、漏洩した受信者を特定できた場合に、その後の漏洩者への配信を停止するためには、全受信者の復号鍵が同じであるため、復号鍵を再度配信し直さなければならないという問題もある。

本稿では、これら二つの問題を解決する漏洩者特定と実質的な配信停止が可能なマルチキャスト配信方式を提案する。

2 漏洩者特定方法

マルチキャスト配信おいて、漏洩者を特定可能にする方式として、Parnes 等が考案したマルチキャスト配信方式が論文に発表されている [3]。本稿では、この方式をマルチキャスト透かし方式と呼ぶことにする。

この方式では、配信元は、コンテンツに異なる 2 種類の電子透かしを入れたものを作り、それぞれを m 個に分割する (例えば、m=100)。このようにしてできた $2m$ 個の分割コンテンツをそれぞれ個別の鍵で暗号化してネットワークに送信する。各受信者は、どちらかの電子透かしの入った分割コンテンツを順次受信するか事前に決められており、指定された受信パターンに従って受信された分割コンテンツのみが、配信元から渡された鍵で正しく復号される。

この方式では、各受信者の電子透かしのパターンは、受信者毎にユニークになるように、2^m 通りの中から予め配信元で決められており、コンテンツの要求条件に合わせて適切な m を選択すれば漏洩者を特定できる。2 種類の電子透かしを入れることによるネットワーク帯域幅の増加は高々 2 倍で抑えられる。

3 漏洩者への配信停止方法

マルチキャストでコンテンツを配信しつつ、コンテンツ漏洩者への配信を停止するには、ブロードキャスト暗号を使用する方法がある。

ブロードキャスト暗号は、1 人の送信者と複数の受信者間で使用され、送信者は同じ暗号文を受信者全体に送信するが、正しく復号できるのは、暗号化時に指定された受信者グループのみというものである。具体的には、復号を許可する全受信者の秘密鍵から公開鍵を作成し、それによって暗号化すると、許可された受信者のみがそれぞれの個人秘密鍵を用いて暗号文を復号できる。この画期的な暗号の概念は 1993 年、Fiat 等により提案された [4]。

この暗号を用いると、漏洩者が分かった時点で公開鍵を作り替えれば、漏洩者は例えコンテンツを受信できたとしてももはや復号できないので、漏洩者への配信を実質的に停止することができる。

4 提案方式

以上述べてきたように、本稿では、問題解決の方法として、マルチキャスト透かしとブロードキャスト暗号を組み合わせた方式を提案する。本方式により、マルチキャスト配信を行いながらも、コンテンツが漏洩した場合に、漏洩者を特定し、その後の漏洩者への実質的な配信停止を簡単に素早く行うことが可能である。

電子透かしによる漏洩者特定能力を向上させるためにコンテンツの分割数 m を増やすと、各受信者が復号に使用する鍵の数 ($=m$) も増加するが、提案方式では、分割数 m を増やしても各受信者の復号鍵は一つで済むと

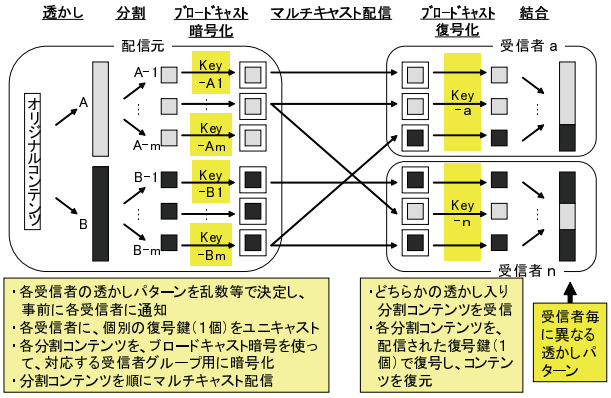

図 1 提案方式の概略図

いう大きなメリットがある。

また、マルチキャスト透かし方式に従来の秘密鍵の暗号を組み合わせた場合には、各受信者の分割コンテンツの透かしパターンに合わせた m 個の秘密鍵を、個別にユニキャストしなければならない。提案方式では、各分割コンテンツの透かしパターン毎に、対応する受信者の秘密鍵からペアリング暗号を使って公開鍵を作成してコンテンツを暗号化するため、各受信者に 1 個の秘密鍵を事前にユニキャストするだけでよいという長所もある。

5 ブロードキャスト暗号の詳細

全受信者数を n、復号化を許可する受信者数を s、復号化を許可しない受信者数を r とする (即ち、$s+r=n$)。最も単純な方法、すなわち、受信者各個人の公開鍵により暗号化を行うことでブロードキャスト暗号を実現した場合、受信者全体の公開鍵のサイズ (公開鍵サイズ) は $O(n)$、受信者個人の秘密鍵のサイズは $O(1)$、暗号文のサイズ (暗号文サイズ) は $O(s)$ である。そのため、公開鍵サイズと暗号文サイズを小さくすることが、効率的なブロードキャスト暗号を実現する上で重要である。

2005 年、Boneh, Gentry, Waters によって、暗号文サイズが $O(1)$ であるブロードキャスト暗号の方式が提案された [5]。ただし、公開鍵サイズは $O(n)$ である。本稿では、この方式を BGW 方式と呼ぶことにする。また、2007 年に Delerablée 等は受信者の動的な追加を可能にした Dynamic Broadcast Encryption (同 DBE 方式) を提案した [6]。DBE 方式では、公開鍵サイズが $O(n)$、暗号文サイズが $O(r)$ となる。さらに、2007 年に Sakai 等と Delerablée 等は独立に、公開鍵を更新することなく動的に受信者を追加することができる Identity-Based Broadcast Encryption (同 IBBE 方式) を提案した [7, 8]。ただし IBBE 方式では、復号化を許可できる受信者の数に上限がある。この上限を k とすると、公開鍵サイズは $O(k)$、暗号文サイズは $O(1)$ となる。

受信者の数が固定されている場合、BGW 方式が効率的である。そこで、本稿では BGW 方式を C 言語により実装し、計算速度の評価を行った。

6 BGW 方式

BGW 方式は、鍵生成、暗号化、復号化の 3 つのアルゴリズムにより構成されている。各受信者に対して ID として 1 から n までの自然数を割り振る。つまり受信者全体は $\{1、2、\cdots、n\}$ である。各受信者は個人秘密鍵として d_1、d_2、$\cdots$、d_n を所有し、公開鍵を PK とし、送信者が復号化を許可する受信者の集合を $S \subset \{1、2、\cdots、n\}$ とする。

鍵生成： 受信者の総数 n を入力とし、公開鍵 PK と個人秘密鍵 d_i $(i=1、2、\cdots、n)$ を出力とする。

暗号化： メッセージ M、公開鍵 PK、復号化を許可する受信者の集合 S を入力とし、暗号文 C とヘッダ Hdr を出力する。

復号化： 暗号文 C、ヘッダ Hdr、受信者 ID i、個人秘密鍵 d_i、公開鍵 PK、復号化を許可された受信者の集合 S を入力とし、メッセージ M を出力する。

7 性能評価

双線形写像として、高速に動作することが知られている η_T ペアリング [9] を用いた。また、標数 3、拡大次数 97 の有限体 $\mathbb{F}_{3^{97}}$ を使用した。測定に使用した PC のスペックは CPU : Athlon X2 3800+、メモリ : 2GB、OS : Windows Vista (32bit) である。コンパイラは gcc 3.4.4 を使用し、最適化オプションは -O2 -fomit-frame-pointer とした。また、多倍長ライブラリとして GMP 4.2.2 を使用した。

BGW 方式はハイブリッド暗号であるが、共通鍵暗号による処理時間は従来と同等であるため、今回の測定ではその演算を省略した。表 1 に、分割コンテンツ 1 個に対して、共通鍵暗号の処理部分を除いた BGW 方式に必要となる演算時間の測定結果を示す。人数は、鍵生成については全受信者数 n を、暗号化と復号化については復号化を許可した受信者数 s を表している。鍵生成の計算時間は受信者の総数 n に、暗号化と復号化の演算時間は復号化を許可した受信者の数 s に比例していることがわかる。

表 1　BGW 方式の計算時間 (秒)

人数	鍵生成	暗号化	復号化
100	0.479	0.010	0.008
500	2.377	0.025	0.023
1000	4.783	0.044	0.042
1500	7.144	0.063	0.061
2000	9.522	0.082	0.080

この結果から、鍵生成、暗号化および復号化に必要な時間は十分短く、また、漏洩者への配信を停止するための公開鍵の作り替えに必要な時間も十分短いことが分かった。

8 おわりに

本稿では、マルチキャスト配信においても、コンテンツの漏洩者を特定して配信を停止できる方式を提案した。また、Boneh 等によって提案された BGW 方式のブロードキャスト暗号を実装してその性能評価を行い、実用可能な見通しを得た。

参考文献

[1] 阪本 等, "ディジタルシネマ共同トライアル「4K Pure Cinema」", NTT 技術ジャーナル, Vol. 18, No. 4, pp. 47-50, 2006.

[2] 井上 等, "Flexcast による段階的導入に優れたマルチキャストシステムの設計と実装", 電子情報通信学会論文誌 D-1, Vol. J88-D-1, No. 2, pp. 272-291, 2005.

[3] R. Parnes and R. Parviainen, "Large scale distributed watermarking of multicast media through encryption, Proc. Of IFIP Communications and Multimedia Security Issues of the N ew Century, pp. 17-26, 2001.

[4] A. Fiat and M. Naor, "Broadcast Encryption", CRYPTO 1993, LNCS 773, pp. 480-491, 1993.

[5] D. Boneh, C. Gentry and B. Waters, "Collusion Resistant Broadcast Encryption With Short Ciphertext and Private Keys", CRYPTO 2005, LNCS 3621, pp. 258-275, 2005.

[6] C. Delerablée, P. Paillier and D. Pointcheval, "Fully Collusion Secure Dynamic Broadcast Encryption with Constant-Size Ciphertexts or Decryption Keys", Pairing 2007, LNCS 4575, pp. 39-59, 2007.

[7] R. Sakai and Jun Furukawa, "Identity-Based Broadcast Encryption", Cryptology ePrint Archive Report, 2007/217, 2007.

[8] C. Delerablée, "Identity-Based Broadcast Encryption with Constant Size Ciphertexts and Private Keys", ASIACRYPT 2007, LNCS 4833, pp. 200-215, 2007.

[9] M. Shirase and Y. Kawahara, T. Takagi, E. Okamoto, "Universal η_T Pairing Algorithm over Arbitrary Extension Degree", WISA 2007, LNCS 4867, pp.1-15, 2007.