4.2 コンテンツ暗号配信のための木構造鍵管理方式

4.2.1 はじめに

これからの図書館には、単に情報を蓄積する機能だけではなく、より積極的に情報を発信することが求められる。とくに電子図書館では、書籍や雑誌、論文、レポート等といった文字メディアに加えて、画像、動画、音声といったマルチメディアコンテンツの発信も重要な課題となる。大学に附置された図書館では、たとえば講義内容をビデオ配信する等の形で社会に大きく貢献することが可能になると期待される。商業ベースのサービスでは、たとえば映画や音楽等をネットワーク配信する業者も出現しており、情報を配信するためのインフラについては、かなり整いつつある。一方、配布すべきコンテンツを保護するための仕組みについては、まだ十分であるとはいえない。実際、多くのコンテンツ配信サービスにおいては、正当な対価を支払った信頼できるユーザ以外には、コンテンツ情報が漏洩しないような仕組みになっていることが要求される。そのため、コンテンツ自体は暗号化して配信し、正当なユーザのみが、あらかじめ配布された復号鍵を用いてコンテンツ情報を入手できるような方式が採用されることが多い。この際、料金を支払わないユーザ、あるいは不正行為に荷担したユーザに対して、無効化、すなわちコンテンツの復号をできなくするような仕組みが重要となる。

最も単純なユーザ無効化方式は、コンテンツの暗号化に用いる鍵を定期的に更新し、新しい鍵を正当なユーザにのみ配布する方式[8, 9]である。この方式は、一部の衛星放送事業者も採用しており、比較的迅速にユーザの無効化処理が行える反面、ユーザの保持する機器（デコーダやプレイヤ等）が、常にコンテンツ配信サーバと通信可能な状態であることが要求される。専用機を利用した放送型の配信サービスであれば、ユーザ機器が常にサーバからの情報を受け取れる状態にあると仮定することもそれほど不自然ではないが、ユーザがパソコン等の汎用機器を利用してサービスを享受する場合、端末が常にサーバの発信する情報を受け取るとの前提はなりたたない。より汎用で利便性の高いサービスを実現するためには、端末の常時接続性を前提としない鍵管理・ユーザ無効化方式が必要となる。

木構造鍵管理方式[7, 10, 11, 12]は、DVD プレイヤ等のオフライン型機器において鍵管理・ユーザ無効化を行うのに適した方式である。この方式では、

暗号化されたコンテンツを配信する際、正当なユーザのみが復号できるような形式で、コンテンツの暗号化に用いた鍵情報も同時に配信する。その際、システム全体で多数の異なる鍵を管理し、各ユーザに対して適切な組合せで鍵の事前配布を行う必要があるが、木構造を利用することで、ある程度効率的な鍵管理が可能となる。一般に、コンテンツ配信サービスのユーザ数は膨大なものになることが予想されるため、木構造鍵管理方式の効率を考えるにあたっては、

(1) コンテンツと同時に配信される鍵情報のサイズ

(2) ユーザ機器が記憶しておかなければならない秘密情報のサイズ

(3) ユーザ機器においてコンテンツを得るまでに必要となる計算量

などを評価する必要がある。

本研究では、主として上記 (2) および (3) のパラメータをできるだけ小さくするような方式について考察する。基本的なアイデアは、落とし戸付き一方向性関数を利用することで、サーバおよびユーザ機器が管理しなければならない秘密情報を、極力削減する点にある。一方向性関数の落とし戸はサーバのみが知っており、各ユーザ機器に配布する鍵を計算するのに利用される。各ユーザ機器は、一方向性関数を順方向に適用することで、コンテンツ復号に必要となる鍵を得る。具体的な鍵管理手法を 2 種類提案し、それぞれの安全性について議論する。最初の方式は、一方向性関数を 2 個用意し、木構造をたどる際に 2 種類の関数を切り替えて利用する方式である。安全性を確保するためには、2 個の関数が適当な性質を満たしている必要があるが、その性質について形式的に議論する。2 つ目の方式は、一個の一方向性関数のみを利用する方式であり、関数間の関係を考慮しなくて良い分、設計や運用等の負担を軽減できる。この方式についても、形式的手法により安全性の証明を行う。

4.2.2 木構造鍵管理方式

木構造鍵管理方式では、各ユーザ機器（以下では端末と呼ぶ）に事前配布する鍵を、鍵管理センター（配信サーバ、以下では単にセンターと呼ぶ）が定める。以下では端末の全体集合を U とし、簡単のため、U は 2^h 個の端末を含むと仮定する。

センターは、高さが h であるような完全 2 分木 T を構成し、葉節点と端末とを一対一対応させる。以下の説明では、混乱のない限り葉と端末とを同一視して記述を行う。次に、(葉節点も含めて) 全ての節点に対し、対称鍵暗号系の鍵

を割り当てる。この際、同じ鍵が複数の異なる節点に割り当てられることがないようにする。以下では、節点 n の親節点を p(n)とし、n に対応する鍵を k(n)と表記する。また、節点 n_1 が 節点 n_2 の先祖であるとき、$n_1 \leq n_2$ と書く。さらに、n が木 T の節点であるとき、$n \in T$ と書くことにする。

各端末には、その端末のアドレス（２分木の中のどの場所に位置するかの情報）と、その端末の先祖に割り当てられた鍵を全て与える。すなわち、葉節点 l に対応する端末には、鍵集合 $\{k(n) \mid n \in T,\ n \leq l\}$ が与えられる。各端末は、与えられた鍵を他者に知られないように管理することが義務づけられる。

サーバが、ユーザの部分集合 $U' \subseteq U$ に対してコンテンツ c を配信したい場合を考える。ここで、R=U–U' は無効化端末の集合である。この場合、サーバは対称鍵暗号系の鍵 r をランダムに生成し、r を鍵とする c の暗号文 E(r, c) を配信する。また、U'に属する端末のみが r を入手できるよう、以下のように r を暗号化して配信する。

(1) $T(R) = \{n \mid n \in T,\ \exists l \in R,\ n \leq l\}$ を計算する。T(R) は R に属する葉の先祖全体からなる集合である。

(2) $N(R) = \{n \mid n \in T,\ \neg(n \in T(R)),\ p(n) \in T(R)\}$ を計算する。すなわち、N(R)は、それ自身は T(R) に属さないが、その親節点が T(R)に属するような節点全体からなる集合である。

(3) $K(R) = \{\langle$n のアドレス，E(k(n)，r)$\rangle \mid n \in N(R)\}$を計算し、配信する。

端末への鍵配布方法より明らかに、任意の $n \in T(R)$ に対し、鍵 k(n)の値を知るような無効化端末が少なくともは一台存在する。一方、任意の $n \in N(R)$ に対し、どの無効化端末も k(n) を知ることはない。また、U' に属する任意の端末 l について、N(R) には l の先祖がちょうど一個だけ含まれる。したがって、無効化されていない任意の端末は、K(R)の中に自分の先祖に対応する組を発見し、自分が保有している鍵を用いて r を入手することができる。

例えば図１において、$n_i\ (1 \leq i \leq 7)$, $l_j\ (1 \leq j \leq 8)$ を節点とすると、端末 l_7 は事前に$\{k(n_1), k(n_3), k(n_7), k(l_7)\}$ を与えられる。また、無効化端末を $\{l_3, l_5, l_6\}$（図中の黒丸）とすると無効化端末の先祖全体からなる集合は $T(R) = \{n_1, n_2, n_3, n_5, n_6, l_3, l_5, l_6\}$（黒丸、斜線に相当）なので、$N(R) = \{n_4, n_7, l_4\}$となり，K(R) の計算に用いる鍵は $\{k(n_4), k(n_7), k(l_4)\}$となる。このとき、端末 l_7 は $k(n_7)$を利用して復号を行う。

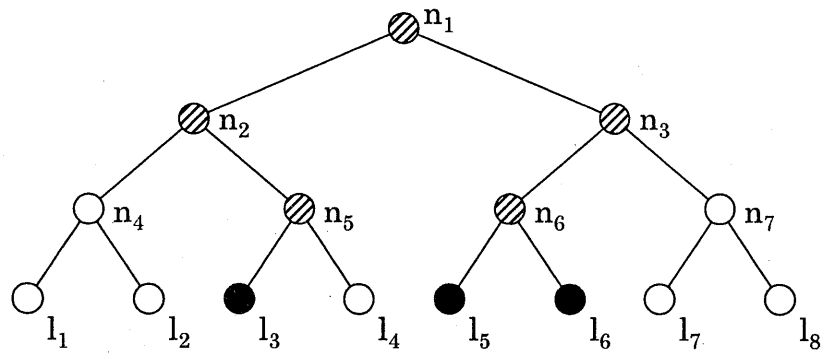

図 1：鍵の割当例

以上で述べた方式により，無効化端末を除外してコンテンツを配信することが可能となるが，この方式については，以下のような問題点が指摘されている。

- メッセージオーバヘッドの問題：コンテンツ c の暗号文のほか，r の暗号文を全部で N(R) 個送信する必要があるため，N(R) が大きくなる場合は，鍵 r の配送にかなり大きなオーバヘッドがかかってしまうことになる。
- 端末の秘密サイズの問題：各端末は h+1 個の鍵を秘密情報として保持する必要がある。一般に，情報を秘密に保管するためのコストは小さくはないので，h が比較的大きい場合，各端末のコストが増加する恐れがある。

本論文では，端末の秘密サイズを削減する手法として，木構造鍵管理方式において，木の各節点の鍵を割り当てる際に落とし戸付き一方向性関数を利用する方式について検討する。鍵割り当てに一方向性関数を利用する方式については [11]などでも検討されているが，[11]の方法では，端末が木の内部節点に対応する鍵を入手する際，場合によってはオンライン接続で鍵情報を参照する必要があった。本稿では，完全なオフライン端末でも利用可能な方式を考える。

4.2.3 提案方式 1

4.2.3.1 鍵の生成と配布

サーバは 2 個の落とし戸付き一方向性関数 h_L と h_R を選択し，公開する。ここで，h_L の定義域および値域，h_R の定義域および値域は全て等しいとする。

また，h_L と h_R とはできるだけ相関のない関数を選択し，たとえば $h_L(h_R(x))=h_R(h_L(x))$ といった性質が成り立たないものとする。両関数とも，落とし戸情報はサーバが秘密に保管しておき，木の各節点に鍵を割り当てる際に利用する。木の各節点に対し，以下に示す方法で鍵を決定する。

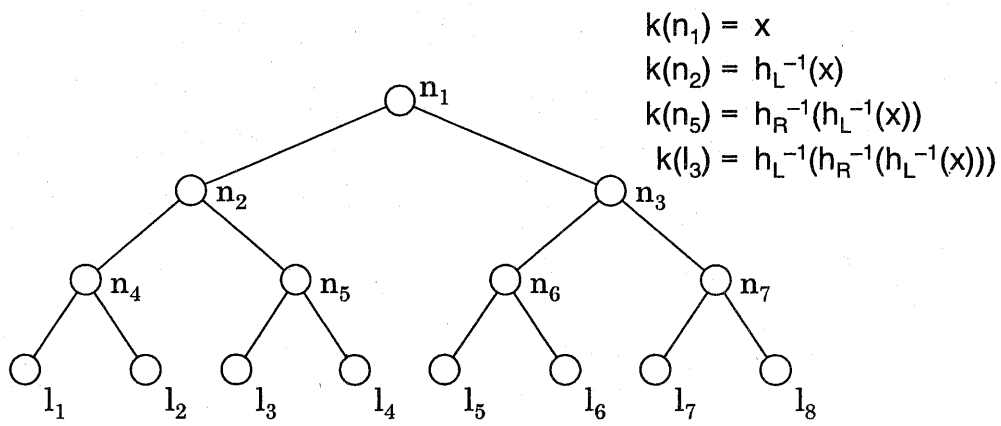

図 2：方式 1 における鍵の計算

- 根節点の鍵はランダムに決定する。
- 節点 n の鍵が k である場合，
  - n の左の子節点の鍵は $h_L^{-1}(k)$ とする。
  - n の右の子節点の鍵は $h_R^{-1}(k)$ とする。

上記の規則を再帰的に利用することにより，2 分木 T の任意の節点について，その節点の鍵が一意に定まる。葉 l に対応する端末には，その端末のアドレスと，k(l) 一個だけを配布する。よって，葉が秘密に管理すべき情報は k(l) 一個だけとなる。各端末は，自分自身のアドレス情報，鍵情報 k(l)，公開されている一方向性関数 h_L，h_R を利用して，自分の先祖に割り当てられた鍵の集合 $\{k(n) \mid n \leq l\}$ を導出することが可能である。一方，自分の先祖でない節点の鍵情報を導出するには，一方向性関数 h_L または h_R の逆関数を計算する必要がある。したがって，十分な強度を有する一方向性関数を利用すれば，ある端末の鍵情報から他の端末の保有する鍵情報を導出することは困難である。

図 2 に鍵生成の一例を示す。葉 l_3 に相当する端末は，$h_L^{-1}(h_R^{-1}(h_L^{-1}(x)))$ を事前に配布される。

4.2.3.2 安全性について

方式 1 において利用する一方向性置換は，できるだけ相関のないものが望

ましい。置換対の相関性については，クローフリー置換対に関する一連の議論が知られているが，方式１にて要求される一方向性置換の特徴は，次の２点でクローフリー性とは異なる。

(1) クローフリー置換対を提案方式 1 に適用した場合を考える。提案方式 1 の攻撃者は固定された z（攻撃者の先祖鍵の一つに相当）から，$h_L(x) = h_R(z)$ を満たす x（コンテンツの暗号化に用いられる鍵に相当）を計算しなければならない。直感的には，クローフリー置換の攻撃者は $h_L(x)=h_R(z)$ を満たす任意の (x, z) を計算すればよいのに対し，提案方式１では z が固定されているため，攻撃がより難しくなっていると考えられる。

(2) ユーザの端末の結託で逆置換に関する何らかの情報が漏れる可能性がある。 注意すべきは，提案方式１では，ルート鍵に逆置換を適用した値を各ユーザが保持している事である。逆像の集合は，攻撃者に一方向性置換の落とし戸情報に関する糸口を与えてしまう可能性がある。

本研究では，クローフリー性に比較的類似した概念として「強準置換対族」の概念を定式化し，以下の定理を証明した。

定理１：(h_L, h_R) が強準置換対族から選択された置換対ならば，提案方式１は安全である。

4.2.4 提案方式2

4.2.4.1 鍵の生成と配布

方式１では，２つの強準置換対族から一方向性置換を選ぶことにより安全性を保証することが出来るが，そのような置換対族の具体的な構成法は明らかではない。本節では，一方向性置換を一つだけ利用するような鍵管理方式を提案する。

センターは，一方向性落とし戸置換 f および一方向性ハッシュ関数 r を準備する。各節点 n に対して r(n) を割り当て，これを節点 n のシードと呼ぶ。センターは，一方向性落とし戸置換 f と一方向性ハッシュ関数 r を公開して，一方向性落とし戸置換の逆置換 f^{-1} を秘密にする。各節点への鍵の割り当ては以下に示す方法で再帰的に決定する。

- 根節点の鍵はランダムに決定する。

◆ 親節点 n_p の子節点 n_c の鍵を，$k(n_c) = f^{-1}(k(n_p) + r(n_c))$とする。

この規則を適用した結果，2 分木 T の任意の節点について，その節点の鍵が一意に定まる。コンテンツを配信する際には，上の定義にしたがって各節点の鍵を生成し，前節で述べたのと同様の手続きでコンテンツの暗号化および配信を行う。葉 l に対応する端末には，自分自身の葉がどの位置にあるかを示すアドレス情報 l と葉 l に割り当てた k(l)を事前に配布する。各端末は，自分自身のアドレス情報 l，鍵情報 k(l)，f，r を利用して，自分の先祖に割り当てられた鍵

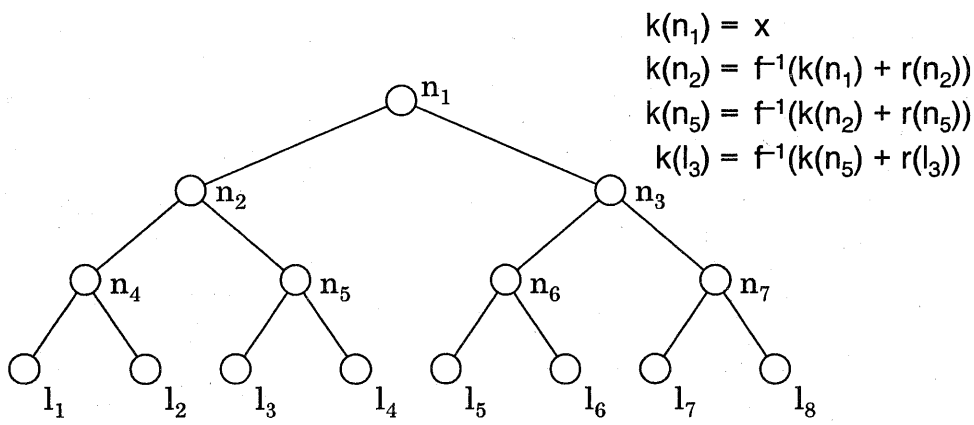

図 3：方式 2 における鍵の計算

の集合 $\{k(n) \mid n \leq l\}$ を導出することが可能である。すなわち，子節点の鍵 $k(n_c)$ が既知のとき，親節点の鍵 $k(n_p)$ は $k(n_p) = f(k(n_c)) + r(n_c)$として計算可能である。

図 3 に鍵生成の一例を示す。葉 l_3 に相当する端末には，事前に秘密情報として $f^{-1}(f^{-1}(f^{-1}(x)+r(n_2))+r(n_5))+r(l_3))$ を配布する。ここで x は，ランダムに選ばれたルート鍵である。

4.2.4.2 安全性について

提案方式 2 の安全性は，f の逆像計算が困難であることに依存している。逆像計算の困難性により，端末には自分自身の先祖以外の鍵を得ることが難しくなっている。詳細な議論は省略するが，提案法の安全性は以下の定理により得られる。

定理 2：f が一方向性落とし戸置換族から選ばれた置換ならば，提案方式 2 は

安全である。

4.2.5 まとめ

木構造鍵管理法において，鍵生成に一方向性落とし戸置換を使用する方式を二つ検討した。提案方式１では置換を２つ利用し，木構造上のパスに応じて置換を使い分ける。提案方式２は一個の置換だけで構成可能であり，より実用的であると考えられる。

参考文献

[1] T. Asano: "A Revocation Scheme with Minimal Storage at Receivers," ASIACRYPT 02, LNCS 2501, pp.433 - 450, 2002.

[2] A. Fiat, M. Naor: "Broadcast Encryption," CRYPTO 93, LNCS 773, pp.480 - 491, 1993.

[3] Y. Dodis, L. Reyzin: "On the Power of Claw-Free Permutations," Conference on Security in Communication Networks (SCN), pp.55 - 73, 2002.

[4] S. Goldwasser, S. Micali, R. Rivest: "A Digital Signature Scheme Secure against Chosen Message Attacks," SIAM Journal on Computing, 17, 2, pp.281 - 308, 1988.

[5] D. Halevy, A. Shamir: "The LSD Broadcast Encryption Scheme," CRYPTO 02, LNCS 2442, pp.47 - 60, 2002.

[6] Y .H. Hwang, C. H. Kim, P. J. Lee: "An Efficient Revocation Scheme with Minimal Message Length for Stateless Receivers," Information Security and Privacy 2003, LNCS 2727, pp.377 - 386, 2003.

[7] D. Naor, M. Naor, J. Lospiech: "Revocation and Tracing Schemes for Stateless Receivers," CRYPTO 01, LNCS 2139, pp.41 - 62, 2001.

[8] D. Wallner, E. Harder, R. Agee: "Key Management for Multicast: Issues and Architecture," RFC 2627, 1999.

[9] C. Wong, M. Gouda, S. Lam: "Secure Group Communications Using Key Graphs," IEEE/ACM Trans. Networking, vol. 8, pp.16 - 30, 2000.

[10] 中野 稔司，大森 基司， 神林 誠，"デジタルコンテンツ保護用鍵管理方

式”， 2001年暗号と情報セキュリティシンポジウム講演論文集，2001.
[11] 中野 稔司，大森 基司， 松崎 なつめ，“デジタルコンテンツ保護用鍵管理方式”, 2002年暗号と情報セキュリティシンポジウム講演論文集, 2002.
[12] 中野 稔司，大森 基司， 松本 勉，“複数システムに対応した木構造鍵管理方式”, 2002年暗号と情報セキュリティシンポジウム講演論文集, 2002.