**SCIS 2013** The 30th Symposium on
Cryptography and Information Security
Kyoto, Japan, Jan. 22 - 25, 2013
The Institute of Electronics,
Information and Communication Engineers

# 閾値型秘密分散法のシェアを用いたアクセス制御方式の拡張
# An Extended Access Control Method Using Threshold Secret Sharing

瀧本 克真 * 倉上 高史 * 今泉 祥子†
Katsuma TAKIMOTO Takashi KURAKAMI Shoko IMAIZUMI

藤吉 正明 * 貴家 仁志 *
Masaaki FUJIYOSHI Hitoshi KIYA

**あらまし** 本稿では，複数の制御対象を有するマルチメディアコンテンツに対して，閾値型秘密分散を用いた新たなアクセス制御方式を提案する．単純なハッシュ連鎖に加え，閾値型秘密分散のシェアを利用して暗号鍵を生成することでアクセス制御の自由度の向上を目指した．シェアを暗号鍵として利用して，一つのシェアを一つの暗号鍵とする場合と，複数のシェアから一つの暗号鍵を生成する方法を提案する．一般的なアクセス制御では結託を許可するか否かに応じて，権限とコンテンツの関係を変更する必要がある．しかし閾値型秘密分散のシェアを用いた鍵の生成法では，権限とコンテンツの関係を変えずに，設定された条件の下でのみ結託を許可する条件付き結託が実現できる．

**キーワード** ハッシュ　アクセス制御　秘密分散　シェア

## 1　はじめに

近年ではネットワークを介して音楽，動画像，文書を代表とした様々なメディアコンテンツが流通している．さらに，コンテンツの増加に伴い，それを求めるユーザも多様化してきている．ユーザ毎に求めるコンテンツの種類，品質等は異なる．コンテンツやユーザの多様化により，サービス事業者が展開するコンテンツ配信形態も様々である．そのため様々なコンテンツ形態に対して柔軟に対応可能なアクセス制御手法が求められている．

アクセス制御方式の一手法として，暗号鍵の生成にハッシュ連鎖を用いる手法が提案されている [2]〜[8]．この方式はサービス事業者が管理する鍵 (管理鍵と呼ぶ) およびユーザが受信する鍵 (配送鍵と呼ぶ) の削減に効果がある．文献 [3]〜[5] ではハッシュ連鎖による鍵生成方式を，コンテンツが階層構造を持つ場合に適用している．コンテンツが複数の品質尺度 (例えば，動画像ならば解像度やフレームレート) を持つ場合に対して，ハッシュ連鎖で生成した暗号鍵を階層を構成する単位データごとに割り当てることで，管理鍵および配送鍵の削減に成功している．一般的には，複数のユーザが結託することで本来は許諾されていないコンテンツを取得するのは好ましくない．そのため通常は結託を許さないコンテンツ形態を設定する．しかし従来法では，ある条件の下で結託を許可する場合が考慮されておらず，コンテンツ配信形態の変更が必要となる．例えば文献 [3]〜[5] では結託攻撃耐性のあるハッシュ連鎖構造をとっている．そのため結託を制御するにはハッシュ連鎖の構造を変えなければならない．

本稿では秘密分散のシェアを暗号鍵生成に用いることで，コンテンツの配信形態を変えずに結託を制御できる手法を提案する．ハッシュ連鎖を用いた鍵生成方式に加えて Shamir の閾値型秘密分散 [1] のシェアを暗号鍵生成に利用する．本稿では，単一もしくは複数のシェアを暗号鍵として利用し，シェアの割り当て方を変更するだけで結託耐性の有無が決定できる手法を提案する．提案法では閾値という条件を満たした場合のみ結託を許可する．

## 2　準備

本節では準備として，まずハッシュ関数を用いた暗号鍵生成手法について説明する．次に *(k, n)* 閾値型秘密分散法について要約する．

* 首都大学東京システムデザイン学部, 〒 191-0065 東京都日野市旭ヶ丘 6-6, Faculty of System Design, Tokyo Metropolitan University 6-6 Asahigaoka, Hino-shi, Tokyo 191-0065 Japan
† 千葉大学大学院融合科学研究科, 〒 263-8522 千葉県千葉市稲毛区弥生町 1-33, Graduate School of Advanced Integration Science, 1-33 Yayoicho, Inage-ku, Chiba-shi, Chiba 263-8522 Japan

表 1: 複数のコンテンツと権限者例 (ハッシュ適用例)

| コンテンツ | $D_0$ | $D_1$ | $D_2$ |
|---|---|---|---|
| 暗号鍵 | $K_0$ | $K_1$ | $K_2$ |
| 権限 1 | ○ | × | × |
| 権限 2 | ○ | ○ | × |
| 権限 3 | ○ | ○ | ○ |

$$K_0 \leftarrow K_1 \leftarrow \cdots \leftarrow K_{N-2} \leftarrow K_{N-1}$$

図 1: ハッシュ関数による暗号鍵の生成：ただし矢印はハッシュ関数

### 2.1 ハッシュ連鎖による暗号鍵の生成

効率的なアクセス制御を目的として，ハッシュ関数による暗号鍵生成法を適用することで，コンテンツの提供者 (サーバ側) が管理する鍵 (管理鍵) の数と，ユーザがコンテンツの暗号を解除するために受け取る暗号鍵 (配送鍵) の個数の削減をする方法が検討されている [2]〜[8].

制御されるべきコンテンツ数 (制御対象数) を $N$ 個とする．表 1 は制御対象数 $N = 3$ の場合の一例である．コンテンツ $D_0, D_1, D_2$ に対応する暗号鍵をそれぞれ $K_0, K_1, K_2$ とおく．表 1 では 3 つの権限を想定し，上位の権限 (権限 1< 権限 2< 権限 3) ほど多くのコンテンツを取得可能な例を示す．ただし表 1 において各権限者は○のついた暗号鍵を取得可能とする．単純なアクセス制御では一般に $N$ 個の暗号鍵が必要となる．一方，ハッシュ連鎖による暗号鍵生成法は，管理鍵および配送鍵の削減に効果的である．コンテンツの暗号鍵は図 1 に示すように鍵 $K_{N-1}$ から繰り返しハッシュ値を求めることで従属的に生成する．ハッシュ関数を $H(\cdot)$ とすると

$$K_{N-2} = H(K_{N-1}) \quad (1)$$
$$K_{N-3} = H(K_{N-2}) \quad (2)$$
$$\vdots$$
$$K_1 = H(K_2) \quad (3)$$
$$K_0 = H(K_1) \quad (4)$$

のようにして暗号鍵が生成される．一般化すると以下のように表すことができる．

$$K_X = H^{(N-1)-X}(K_{N-1}), \quad (5)$$
$$(X = N-2, ..., 2, 1)$$

$H^X(\cdot)$ はハッシュの入力に対して $X$ 回ハッシュ値を求めることを意味している．

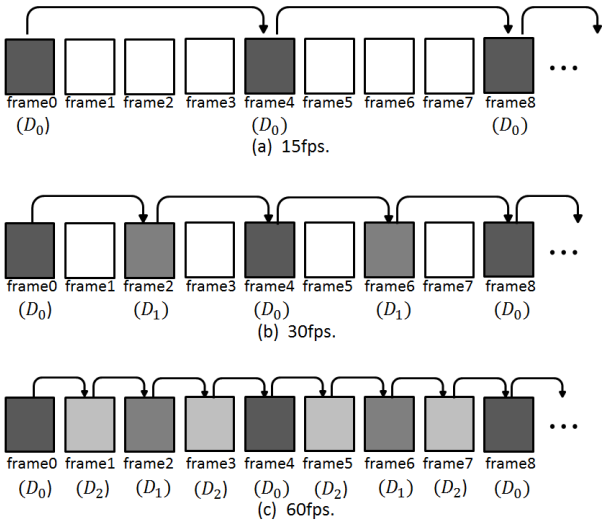



図 2: 各フレームレートにおける再生されるフレーム：影部フレームが再生される

表 1 の場合では管理鍵は $K_2$ ただ一つである．また配送鍵も権限によらず一つのみでよい．例えば，権限 3 を持つユーザには鍵 $K_2$ を配送し，権限 2 をもつユーザには鍵 $K_1$ を配送する．一方，各コンテンツにそれぞれ無関係な独立な鍵を割り当てた場合には複数の鍵を管理する必要があり，配送鍵の個数も上位の権限ほど多くなる．またハッシュ関数による暗号鍵生成では，ハッシュ関数の一方向性によって鍵 $K_X$ から鍵 $K_{X+1}$ を生成することは困難である．そのため下位の権限を持つ者が，上位の権限に属するコンテンツを取得することはできない．

動画像のフレームレートを制御対象とする例を図 2 に示す．フレームレートは 15fps(同図 (a)), 30fps(同図 (b)), 60fps(同図 (c)) の 3 種類 (制御対象数 $N = 3$) があるとする．3 種類のフレームレートは表 1 の権限 1〜3 に相当する．15fps で再生されるフレーム集合を $D_0$，$D_0$ に加え 30fps の時に追加で再生されるフレーム集合を $D_1$，そして $D_0$ および $D_1$ に加え 60fps の時に追加で再生されるフレーム集合を $D_2$ とする．暗号化はフレーム集合ごとに施される．フレーム集合 $D_0, D_1, D_2$ のそれぞれの暗号鍵を $K_0, K_1, K_2$ とすると，式 (5) より暗号鍵 $K_0$ および $K_1$ は $K_2$ から従属的に生成される．

ここで，フレームレートは図 3 に示すような階層構造を持つことに注意されたい．図 3 における集合 $A$，集合 $B$，集合 $U$ の関係は以下の式で与えられる．

$$A \subset B \subset U \quad (6)$$

集合 $A$ および $B$ は全体集合 $U$ の部分集合であり，かつ集合 $A$ は集合 $B$ の部分集合である．また集合 $A$ の集合 $B$ に関する補集合を集合 $\bar{A}$，集合 $B$ の集合 $U$ に関する補集合を集合 $\bar{B}$ とすると，これらの関係は以下の式で

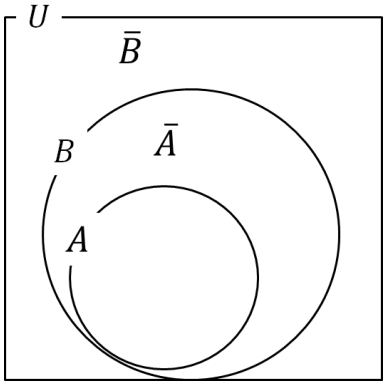



図 3: 制御対象の階層関係

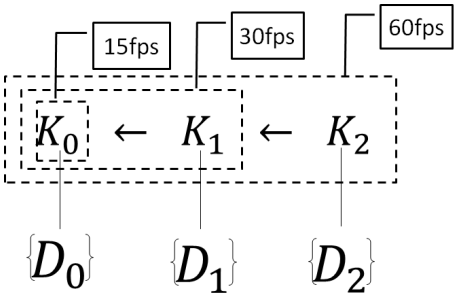



図 4: 暗号鍵の生成順序と各フレームレートに必要な暗号鍵：ただし矢印はハッシュ関数

表すことができる．

$$\bar{A} = B - A \tag{7}$$

$$\bar{B} = U - B \tag{8}$$

集合 $A$ は図 2 のフレーム集合 $D_0$ に，集合 $\bar{A}$ はフレーム集合 $D_1$ に，集合 $\bar{B}$ はフレーム集合 $D_2$ に，それぞれ相当する．暗号鍵の生成順序と各フレームレートに必要な暗号鍵の関係を図 4 に示す．各フレームレートの動画をそのまま暗号化するよりも，このようにフレーム集合ごとに暗号化した方が暗号化するデータ量も少なくて済み，かつ管理鍵や配送鍵の個数は同じである．

### 2.2 Shamir の$(k, n)$閾値型秘密分散法

この節では Shamir の $(k, n)$ 閾値型秘密分散法 [1] について簡単に要約する．Shamir の閾値型秘密分散法は，秘密にしたい情報 $S$ を $n$ 個の分散情報 (以下「シェア」と呼ぶ) に分散し，その $n$ 個のうち $k(\leq n)$ 個のシェアが集まれば秘密情報 $S$ を復元することができる，というものである．しかし集まったシェア数が $k$ 個に満たない場合は秘密情報 S に関する情報は一切得られず，復元できない．次に秘密の分散，および復元の処理について述べる．秘密の分散処理ではまず，定数項 $a_0$ が $S$ となるような $k-1$ 次の 1 変数多項式 $q(x)$ を生成する．

$$q(x) \equiv a_0 + a_1 x + \cdots + a_{k-1} x^{k-1} \mod p \tag{9}$$

ただし式 (9) において，$p$ は素数，$0 \leq a_i < p (i = 1, 2, \cdots, k-1)$，である．素数 $p$ はランダムに設定し，それを上限として多項式の係数 $a_i$ を正整数の中からランダムに決定する．また $x = 0$ のときの $q(x)$ の値，すなわち $a_0$ が秘密情報 $S$ である．そして分散情報であるシェア $w_t (t = 1, 2, \cdots, n)$ は

$$w_1 = q(1), w_2 = q(2), \cdots, w_t = q(t), \cdots, w_n = q(n)$$

として生成される．

復元処理では Lagrange 補間を用いる．シェアを $k$ 個集めたら $k$ 個の点 $(t, w_t)$ から $q(x)$ を以下の式で求める．

$$q(x) = \sum_{j=1}^{k} q(x_j) l_j(x) \tag{10}$$

$$l_j(x) = \prod_{1 \leq i \leq k, i \neq j} \frac{x - x_i}{x_j - x_i}$$

秘密情報 $S$ は $a_0 = q(0)$ から得られる．すなわち

$$S \equiv \sum_{j=1}^{k} q(x_j) \prod_{1 \leq i \leq k, i \neq j} \frac{-x_i}{x_j - x_i} \mod p \tag{11}$$

となる．以上が Lagrange 補間による $k$ 個のシェアから秘密情報 $S$ の復元である．

本稿ではハッシュ関数に加え，この $(k, n)$ 閾値型秘密分散法を用いたアクセス制御方式を提案する．

## 3 提案法

ここでは，$(k, n)$ 閾値型秘密分散をアクセス制御に用いる．あるコンテンツの暗号鍵を秘密分散し，秘密分散によって生成されたシェアを別のコンテンツの暗号鍵として利用する．これによってシェアは，閾値を満たせば鍵を復元するシェア本来の機能と，コンテンツの暗号鍵としての機能の二つを有することになる．

### 3.1 閾値型秘密分散を用いたアクセス制御

閾値型秘密分散をアクセス制御に用いる手法について，$(2, 3)$ 閾値型秘密分散を例に図 5 を用いて説明する．まずコンテンツを配信する側は，あるコンテンツ $D$ を暗号化するための暗号鍵 $W$ を決定する．続いて，その暗号鍵 $W$ を秘密情報として $(2, 3)$ 閾値型秘密分散法を適用し，3 つのシェア $w_1, w_2, w_3$ を生成する．生成されたシェアをさらに別のコンテンツ $D_1, D_2, D_3$ を暗号化する鍵としてそれぞれ割り当てる．ここまでが暗号鍵の生成となる．コンテンツを取得するユーザ側は，取得した

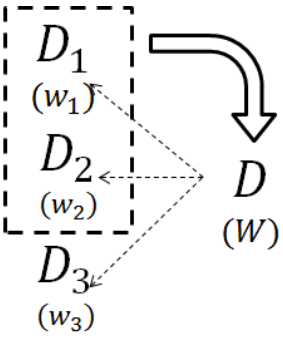



図 5: 秘密分散による暗号鍵生成：ただし，$W$ およびシェア $w_1, w_2, w_3$ はそれぞれコンテンツの暗号鍵として扱う．また点線矢印はシェアの生成，白矢印はシェアによる秘密情報の復元を意味する．

表 2: コンテンツと権限の関係の一例：表中の○はその権限で許諾されるコンテンツであることを示す

| コンテンツ | $D_1$ | $D_2$ | $D_3$ | $D$ |
|---|---|---|---|---|
| 権限 1 | ○ | × | × | × |
| 権限 2 | × | ○ | × | × |
| 権限 3 | × | × | ○ | × |
| 権限 4 | × | × | × | ○ |

いコンテンツに応じた暗号鍵を受け取る．ここで，暗号鍵 $w_1, w_2, w_3$ のうち 2 つを手に入れた者は，入手した暗号鍵をシェアとして用い，秘密分散の復元処理によってシェアから暗号鍵 $W$ を生成し，$W$ で暗号化されたコンテンツ $D$ の取得が可能になる．表 2 のようにコンテンツ $D$ および $D_1, D_2, D_3$ がそれぞれ権限 1〜4 に割り振られていると考える．ユーザは権限に応じてコンテンツを取得するが，権限 1〜3 のうち二つ以上の権限を持つ場合，本来権限 4 でのみ与えられるコンテンツ $D$ を入手することができる．

ここでは一つのコンテンツに一つのシェアを暗号鍵として割り当てたが、3.3 節ではコンテンツに応じて暗号鍵として割り当てるシェアの数を変えることを考える．一つのコンテンツに複数のシェアを割り当てた場合は，割り当てられたシェアを結合し暗号鍵とする．今回の場合はシェアの排他的論理和を取ることで，コンテンツの暗号鍵を生成するものとする．排他的論理和ならば，割り当てられるシェアが決まれば，暗号鍵を一意に決定でき，桁上がりによる鍵長の変化も起こらないためである．

### 3.2 提案法が想定する問題

本節では提案法が対象とする問題設定について説明する．例えば表 3 のような権限とコンテンツの関係を仮定する．表 3 の例では，それぞれのコンテンツに無関係な

表 3: コンテンツと権限の関係 (結託あり)：表中の○および●はその権限で許諾されるコンテンツであることを示す．ただし●はハッシュ連鎖で生成される鍵を示す．

| コンテンツ | $D_0$ | $D_1$ | $D_2$ | $D_3$ |
|---|---|---|---|---|
| 暗号鍵 | $K_0$ | $K_1$ | $K_2$ | $K_3$ |
| 権限 1 | × | × | × | × |
| 権限 2 | ● | × | × | × |
| 権限 3 | ● | × | × | ○ |
| 権限 4 | ● | ● | ● | × |
| 権限 5 | ● | ● | ● | ○ |

表 4: コンテンツと権限の関係 (結託なし)：表中の○および●はその権限で許諾されるコンテンツであることを示す．ただし●はハッシュ連鎖で生成される鍵を示す．

| コンテンツ | $D_0$ | $D_1$ | $D_2$ | $D_3$ | $D_4$ |
|---|---|---|---|---|---|
| 暗号鍵 | $K_0$ | $K_1$ | $K_2$ | $K_3$ | $K_4$ |
| 権限 1 | × | × | × | × | × |
| 権限 2 | ● | × | × | × | × |
| 権限 3 | ● | × | × | ○ | × |
| 権限 4 | ● | ● | ● | × | × |
| 権限 5 | ● | ● | ● | ○ | ○ |

独立な鍵を割り当てた場合には管理鍵は 4 個，配送鍵も最大 4 個が必要になる．単純なハッシュ連鎖を用いて暗号鍵を生成する場合は，式 (5) を用いて暗号鍵 $K_2$ から $K_1, K_0$ を従属的に生成してもよい．しかし権限 3 の制約を保護するため，$K_3$ は別に生成しなければならない．また表 3 の関係では権限 3 を持つ者と権限 4 を持つ者が結託した場合，権限 5 に届いてしまう．そこで結託しても上位の権限に届かないようにするためには，コンテンツと権限の関係を変更する必要がある．変更した関係を表 4 に示す．

表 4 では権限 3 を持つ者と権限 4 を持つ者が結託しても権限 5 には届かない．独立な鍵を割り当てる場合は管理鍵は 5 個，配送鍵も最大 5 個である．単純なハッシュ連鎖で鍵を生成する場合は，$K_2$ から $K_1, K_0$ を生成し，それとは別のハッシュ連鎖で $K_4$ から $K_3$ を生成する．その場合は管理鍵は 2 つ配送鍵も最大 2 つである．

このように従来法では結託を許すか否かを決定する際，権限とその権限で許諾されるコンテンツの関係を変える必要がある．しかし秘密分散を用いると，鍵の生成手法で結託の有無を設計することができる．提案法では権限とコンテンツの関係を変えることなく，設定された条件の下でのみ結託を許可する「条件付き結託」が可能である．提案法では閾値型秘密分散を用いることで条件付き

表 5: 各プライバシー保護映像生成に必要な情報：表中の○および●は映像の生成にその情報が必要であることを示す．ただし●はハッシュ連鎖で生成される鍵を示す

| 観察者の閲覧権限 ＼ 分解された被写体情報 | 領域情報 $R$ | 色差成分 $C$ | 低解像度成分 $Y_S$ | 高解像度成分 $Y_W$ |
|---|---|---|---|---|
| 透明化映像 | × | × | × | × |
| ボックス表示映像 | ● | × | × | × |
| エッジ処理映像 | ● | × | × | ○ |
| モザイク処理映像 | ● | ● | ● | × |
| 実写映像 | ● | ● | ● | ○ |

表 6: 結託を許さない場合の，各プライバシー保護映像生成に必要な情報：表中の○および●は映像の生成にその情報が必要であることを示す．ただし●はハッシュ連鎖で生成される鍵を示す．

| 観察者の閲覧権限 ＼ 分解された被写体情報 | 領域情報 $R$ | 色差成分 $C$ | 低解像度成分 $Y_S$ | 高解像度成分 (レベル 1) $Y_{W1}$ | 高解像度成分 (レベル 2 以上) $Y_{W2}$ |
|---|---|---|---|---|---|
| 透明化映像 | × | × | × | × | × |
| ボックス表示映像 | ● | × | × | × | × |
| エッジ処理映像 | ● | × | × | ○ | × |
| モザイク処理映像 | ● | ● | ● | × | × |
| 実写映像 | ● | ● | ● | ○ | ○ |

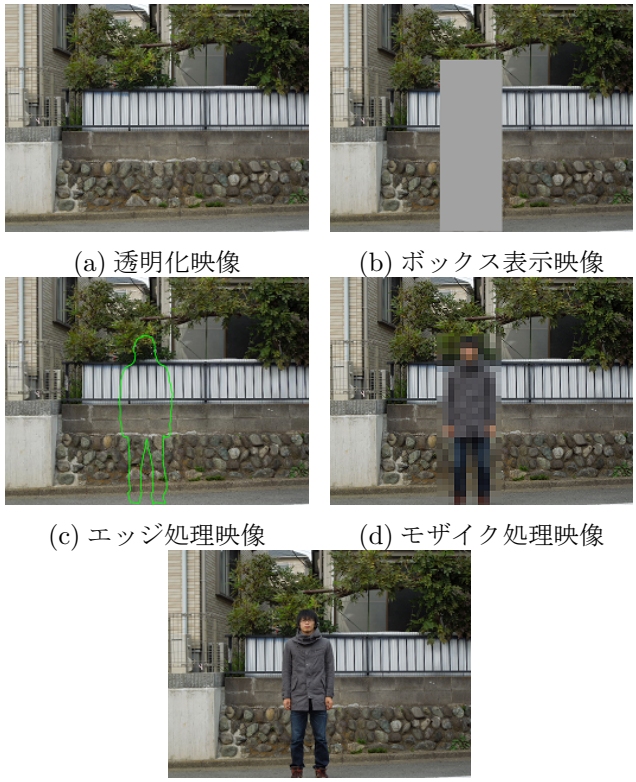
結託を実現する．

### 3.3 条件付き結託が可能なアクセス制御

文献 [9],[10] では結託耐性を有する観察者の閲覧権限に応じたプライバシー保護映像の配信手法が提案されている．もともとの手法では各プライバシー保護映像の再生に必要な情報は表 5 のような関係であった．これに結託耐性を与えるため表 6 に示す関係に変更した手法が提案されている．表 5 の関係は表 3 と等しく，表 6 の関係は表 4 と等しい．分解された被写体情報の暗号鍵を以下のようにおく．領域情報を $R$，色差成分を $C$，低解像度成分を $Y_S$，高解像度成分を $Y_{W_1}$，$Y_{W2}$ とする．図 6 に各プライバシー保護映像の例を示す．

表 6 に示した関係において，秘密分散を用いてアクセス制御を施す．制御モデルを図 7 に示す．まずレベル 2 以上の高解像度成分の暗合鍵 $Y_{W2}$ を (3,4) 秘密分散し，シェア $w_1, \ldots, w_4$ を生成する．

図 7(a) のモデル 1 では低解像度成分の暗号鍵 $Y_S$ としてシェア $w_1$ を，レベル 1 の高解像度成分の暗号鍵 $Y_{W1}$ としてシェア $w_2$ を割り当てている．低解像度成分 $Y_S$ からは 2.1 節の単純なハッシュ連鎖により，色差成分 $C$ と領域情報 $R$ の暗号鍵が式 (14),(15) のように順次生成さ



(a) 透明化映像　(b) ボックス表示映像

(c) エッジ処理映像　(d) モザイク処理映像

(e) 実写映像

図 6: 各プライバシー保護映像

れる．モデル 1 ではシェア $w_3, w_4$ には特に何も割り当てない．

$$Y_S = w_1 \tag{12}$$

$$Y_{W1} = w_2 \tag{13}$$

$$C = H(w_1) \tag{14}$$

$$R = H(C) = H^2(w_1) \tag{15}$$

同図 (b) のモデル 2 では，低解像度成分 $Y_S$ にシェア $w_1$ と $w_2$ の 2 つを割り当て，シェア $w_1$ と $w_2$ の排他的論理和を暗号鍵とする．サーバ側はコンテンツに割り当てられたシェアを配送し，ユーザ側は受け取ったシェアから暗号鍵を生成する．またレベル 1 の高解像度成分の暗号鍵 $Y_{W1}$ としてシェア $w_3$ を割り当てている．モデル 1 と同様ハッシュ連鎖により，低解像度成分 $Y_S$ から色差成分 $C$ と領域情報 $R$ の暗号鍵が式 (18),(19) のように順次生成される．モデル 2 ではシェア $w_4$ には特に何も割り当てない．

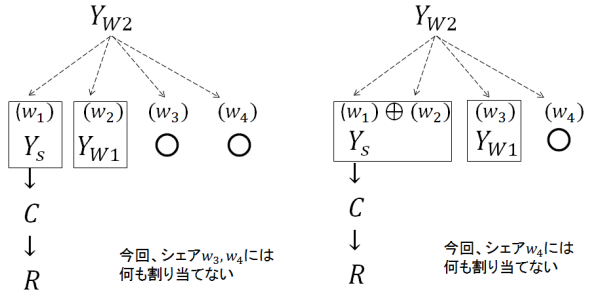
$$Y_S = w_1 \oplus w_2 \tag{16}$$

$$Y_{W1} = w_3 \tag{17}$$

$$C = H(w_1 \oplus w_2) \tag{18}$$

$$R = H(C) = H^2(w_1 \oplus w_2) \tag{19}$$

モデル 1 とモデル 2 では低解像度成分 $Y_S$ に割り当てる




(a) モデル 1（結託なし）　(b) モデル 2（結託あり）

図 7: シェアを暗号鍵として利用した方式：ただし実線矢印はハッシュ関数，点線矢印は秘密分散，() 内はシェア，四角の枠はシェアが各成分の暗号鍵として割り当てられていることを示す．

シェア数が異なる．これによって，モデル 2 ではエッジ処理映像の閲覧権限を持つユーザとモザイク処理映像の閲覧権限を持つユーザが結託した場合に，シェア数が秘密分散の閾値を満たす．そのため高解像度成分 $Y_{W2}$ の暗号鍵を取得可能で，実写映像の再生に必要な全ての被写体情報を得ることができる．モデル 1 では実写映像以外の映像の閲覧権限を持つ者たちが結託しても，実写映像の閲覧権限を得ることはできない．このように秘密分散のシェアを暗号鍵として割り当てることで，シェアの割り当て方によって結託攻撃の許否を変化させることができる．この方法では，表 6 に示される元のコンテンツ形態を維持したまま，閾値という条件を満たした時のみ結託を認可させることができる．

表 6 の関係では，高解像度成分の暗号鍵 $Y_{W2}$ を低解像度成分 $Y_S$ と高解像度成分 $Y_{W1}$ から以下の式のように生成すれば，ハッシュによる鍵生成でも結託を許す構造を作ることができる．

$$Y_{W2} = H(Y_S \oplus Y_{W1}) \tag{20}$$

表 7 のように，高解像度成分 $Y_{W3}$ とエッジ処理の閲覧権限が増えた場合を考える．暗号鍵の生成順序を図 8 に示す．$Y_{W3}$ を $(4, 6)$ 秘密分散し図 8 のようにシェアを割り当てると，エッジ処理映像 1，エッジ処理映像 2，モザイク処理映像のうちからどの 2 つ権限でも結託を許可できる．また，シェアの割り当て方を変えれば，結託攻撃耐性を持たせたり，限られた権限のみで結託を許可することも可能である．このような，結託の許可はハッシュ連鎖を用いた鍵生成方式では作ることが困難である．

## 4　評価

表 6 の関係に対して以下の二つの手法を用いた場合を考える．まず最もわかりやすい方法として，分解された

表 7: 提案法が有効な例：表中の○および●は映像の生成にその情報が必要であることを示す．ただし●はハッシュ連鎖で生成される鍵を示す．

| 観察者の閲覧権限 ＼ 分解された被写体情報 | $R$ | $C$ | $Y_S$ | $Y_{W1}$ | $Y_{W2}$ | $Y_{W3}$ |
|---|---|---|---|---|---|---|
| 透明化映像 | × | × | × | × | × | × |
| ボックス表示映像 | ● | × | × | × | × | × |
| エッジ処理映像 1 | ● | × | × | ○ | × | × |
| エッジ処理映像 2 | ● | × | × | × | ○ | × |
| モザイク処理映像 | ● | ● | ● | × | × | × |
| 実写映像 | ● | ● | ● | ○ | ○ | ○ |

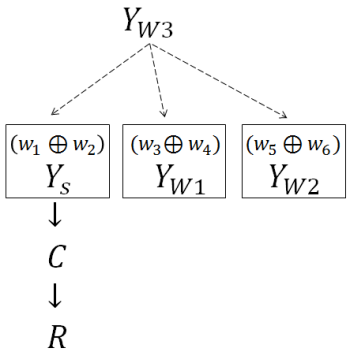



図 8: 表 7 における鍵生成順序：ただし実線矢印はハッシュ関数，点線矢印は秘密分散，() 内はシェア，四角の枠はシェアが各成分の暗号鍵として割り当てられていることを示す．

被写体情報の 5 つにそれぞれ個別の独立な鍵を割り当てる場合，次に，単純なハッシュ連鎖で暗号鍵を生成する場合について考える．この 2 つの手法を検証し，秘密分散を利用した場合と比較する．各手法における管理鍵の個数および配送鍵の最大数，条件付き結託の可不可を表 8 に示す．

分解された 5 つ被写体情報のそれぞれに個別の独立な鍵を割り当てる場合は，手法としては単純だがそれぞれの暗号鍵が関連を一切持たない．そのため条件付き結託の設定は難しく，また管理鍵および配送鍵の数も多い．

表 8: 表 6 に対する各手法の比較

| | 個別に独立な鍵 | 単純なハッシュ連鎖 | 提案法 |
|---|---|---|---|
| 条件付き結託 | × | △ | ○ |
| 管理鍵の個数 | 5 | 2 | 1 |
| 配送鍵の最大の個数 | 5 | 2 | 3 |

$$R \leftarrow C \leftarrow Y_S$$

$$Y_{W1} \leftarrow Y_{W2}$$

図 9: プライバシー保護映像配信問題における単純なハッシュ連鎖による暗号鍵の生成 : ただし矢印はハッシュ関数

単純なハッシュ連鎖で鍵を生成する場合は，管理鍵および配送鍵の削減が期待できる．ハッシュ連鎖の構造の一例を図 9 に示し，生成順序を以下に示す.

$$C = H(Y_S) \tag{21}$$

$$R = H(C) = H^2(Y_S) \tag{22}$$

$$Y_{W1} = H(Y_{W2}) \tag{23}$$

ここで，低解像度成分 $Y_S$ とレベル 1 の高解像度成分 $Y_{W1}$ の暗号鍵が同じハッシュ連鎖内で生成されてはいけない．表 6 を見てわかるように同じハッシュ連鎖内にあると，エッジ処理映像の閲覧権限を有していることは，モザイク処理映像の閲覧権限を有していること同じになる．そのため，低解像度成分 $Y_S$ とレベル 1 の高解像度成分 $Y_{W1}$ の暗号鍵は別々のハッシュ連鎖で生成されなければならない．ハッシュ連鎖の構造を変えることで，条件付き結託が可能な場合もある．管理鍵と配送鍵の個数は，個別の独立な鍵を割り当てた場合に比べ少ない.

秘密分散を用いた場合では，シェアを生成する際に使用する式 (9) の多項式の係数 $a_i(i = 1, 2, \cdots, k-1)$ を保存しておけば，秘密分散とハッシュ連鎖によってレベル 2 以上の高解像度成分 $Y_{W2}$ からすべての暗号鍵を生成可能である．よって管理鍵は $Y_{W2}$ 一つでよい．配送鍵はハッシュにより削減できてはいるが，コンテンツに割り当てるシェア数によって変化するため注意が必要である．秘密分散を利用した場合では鍵の生成にシェアを用いているため，閾値によって条件付き結託の設定が可能である.

## 5 まとめ

本稿ではアクセス制御に秘密分散を利用し，シェアをコンテンツの暗号鍵として割り当てた．秘密分散を利用した場合ではシェアから暗号鍵を生成するため，権限とコンテンツの関係を変えずに結託を許すか否かの設定が可能である．シェアの割り当ては自由度があるため，より柔軟なアクセス制御が可能である．しかし一つのコンテンツに多数のシェアを割り当てすぎると，その分配送鍵が増加するため注意が必要である．今回はプライバシー保護映像配信手法 [9],[10] を例に提案法を適用したが，今後は提案法のより一般的なモデルへの適用を検討する.

## 参考文献


[1] A.Shamir“How to share a secret,” Communications of the ACM, vol.22,no.11,pp.612-613(1979)

[2] 今泉祥子，渡邊修，藤吉正明，貴家仁志，“ディジタル動画配信サービスにおけるアクセス制御方式,”SCIS2010,3F4-1-b

[3] 今泉祥子, 藤吉正明, 貴家仁志 “再帰型ハッシュ連鎖を用いた暗号鍵生成方式と多次元階層的アクセス制御への適応” 情報メディア学会誌 vol.65, No.2, pp.193〜202 (2011)

[4] 今泉祥子, 青木直和, 小林裕幸, 貴家仁志, “メディアアクセス制御のための再帰型ハッシュ連鎖型多次元暗号鍵派生方式,” SCIS2012, 4F1-2

[5] S. Imaizumi, M. Fujiyoshi, and H. Kiya, “An efficient access control method for composite multimedia content,” IEICE Electronics Express,, vol.7, no.20, pp.1534-1538, 2010.

[6] 須賀祐治，岩村恵市，“一方向性ハッシュ関数を用いた階層構造を持つアクセス制御方式,” CSS2003, pp.295-300 (Oct. 2003)

[7] 須賀祐治, 岩村恵市, “有効グラフで表現されたアクセス構造における一方向性ハッシュ関数を用いた鍵生成方式,” 信学 SCIS, 1C5-1, 2004.

[8] 須賀祐治, 岩村恵市, “DAG における鍵派生方式の枝切り改良方式,” 信学 SCIS, 2C2-4, 2005.

[9] 福岡直也，伊藤義道，馬場口登，“結託耐性を有する観察者の権限に応じたプライバシー保護映像の配信手法,”電子情報通信学会 Technical Report,pp.25-30,2012

[10] Naoya Fukuoka, Yoshimichi Ito, and Noboru Babaguchi, “Delivery method for viewer-specific privacy protected video using discrete wavelet transform,” ICIP, 2012, pp.2285-2288