# 一方向性関数を用いた任意の有効期間設定が可能な時限付き鍵管理技術

田中敏也[†] 栗林稔[††] 森井昌克[††]

ディジタルコンテンツを提供する情報関連サービスにおいて，コンテンツの利用を期間限定とする時限付き鍵管理システムが考案されている．短いビット系列で長い有効期間を実現させるためには，カオス写像を用いる方式では有効期間を離散値に，木構造を用いる方式では冗長な時間鍵を与えなければならなかった．本論文では木構造を用いる方式に改良を加え，一方向性関数の特性を利用する方式を提案する．その結果，短いビット系列で長い有効期間を実現でき，特定の条件が揃わない限りは冗長性のない任意の有効期間を設定することできる．

# Time-limited Key Management Scheme with Arbitrary Expiration Date Using One-way Functions

Toshiya Tanaka,[†] Minoru Kuribayashi[††] and Masakatsu Morii[††]

On a broadcast-type contents distribution, a time-limited key management scheme has been proposed, which enables a broadcaster to provide contents to each user with each period. In order to realize a long expiration date with a short bit-sequence, chaotic map based schemes must design the allowable period by discrete values and tree-structured schemes must allow a user for the access of extra period. In this paper, based on tree-structured scheme we propose a new time-limited key management scheme applying one-way function. In this scheme, we realize a long expiration date with a short bit-sequence, and make it possible to design arbitrary expiration date without redundancy unless a specific condition is satisfied.

## 1. はじめに

近年通信技術の進歩により，ディジタルコンテンツの電子配信サービスが普及しつつある．しかし，それにともない非合法な受信や複製が著作権侵害の脅威となっている．このような不正利用を防止し，正規ユーザ限定のコンテンツサービスを実現できる手法として，暗号関連技術を利用した暗号鍵管理方式が提案されている．放送型暗号（Broadcast Encryption）[1),2)]では，ユーザごとに異なる鍵を割り当てており，受信できるユーザを状況に応じて切り替えられるシステムを効率良く構築している．この方式の特徴は，ユーザごとに受信の許可を設定できる点であり，複数のユーザが結託したとしても，結託ユーザが持つ鍵を無効にすることが可能である．しかし，ユーザの受信状況に応じて送信する暗号文を作成する必要があり，ユーザの加入，脱退が激しいシステムにおいてはユーザ管理が非常に複雑になるという問題がある．有料のアクセスポイントへの一時的な接続サービスを提供するシステムにおいては，ユーザの結託に対する対策を施すよりも，システムを簡略化させた時限付き鍵管理方式[3)~9)]が有効である．この方式では，全ユーザ共通の鍵を使用して短時間で鍵更新を行い，各ユーザには個々に契約した一定期間の鍵を計算できる鍵情報を与える．そのため，サービス提供者はユーザの受信状況とは独立して送信データを作成することができ，また，ユーザの要望に対し柔軟な有効期間設定を行うことができる．このシステムでは，サービス提供者がユーザと通信を行うのは鍵情報を与えるときのみであり，有効期間終了時の手続きを必要としないため，ユーザ管理を容易に行える．

我々は，カオス写像を用いた時限付き鍵管理方式[3),4)]を提案しており，鍵選出法を時間ごとに変化させることによりそれらを効率化させた[5),6)]（以後，カオス方式）．カオス写像を用いた方式では鍵系列は 1 次元の数列であり，ユーザに配布する秘密情報である鍵サイ

† プロクター・アンド・ギャンブル・ジャパン株式会社
P & G JAPAN Corporation
†† 神戸大学大学院工学研究科
Graduate School of Engineering, Kobe University

ズは有効期間の長さに比例して大きくなる．秘密情報のサイズを小さく抑えるためには，方式の安全性を効慮すると有効期間終了時間を $\ell, 2\ell, 3\ell, \ldots$ といった離散値に設定する制約が必要であり，ユーザに対し柔軟な有効期間設定ができない．

文献 7) では，可換則を満たす 2 つの一方向性関数を用いて，鍵系列が 2 次元の格子構造をとる方式（格子方式）が提案されている．格子方式では，ユーザの有効期間の長さにかかわらず，配布する鍵サイズは一定である．しかし，安全な可換則を満たす 2 つの一方向性関数の組が見つかっていないという根本的な問題がある．

文献 8) では，鍵系列を葉とし，枝に一方向性関数を用いた木を用いる方式（木構造方式）が提案されている．木構造方式では，ユーザには鍵系列である葉ではなく，有効期間に合わせた節点を配布する．そのため．ユーザに配布する秘密情報のサイズを対数オーダで抑えることができる．しかし，配布する節点の数はユーザが指定する有効期間に依存し，秘密情報のサイズを一様に小さく抑えることはできない．そこで，木構造方式において，ユーザに有効期間外の冗長な鍵を与えることで，秘密情報のサイズをより小さく抑える方式（冗長木構造方式）が提案されている[9]．これによりユーザ端末において，記憶する鍵情報の削減，配布する秘密情報のサイズの縮小に成功している．しかし，有効期間開始前，終了後にそれぞれ最大でユーザが要求した有効期間と同じだけの冗長な鍵を与えることになる．つまり，最大で有効期間の 2 倍分の冗長な鍵を与えてしまう．ユーザに冗長な時間鍵を与えることは，ユーザが契約外の時間帯のコンテンツを視聴できることを意味し，大きな問題である．

ユーザに配布する秘密情報のサイズを小さくするために，カオス方式では，設定できる有効期間を離散値としなければならず，一方，冗長木構造方式では，秘密情報を小さくできるが冗長な鍵情報をユーザに与えなければならない．ゆえに秘密情報を小さくすることと，柔軟な有効期間設定の間にはトレードオフの関係があると考えられる．

本論文では，秘密情報のサイズを対数オーダで抑えることのできる木構造方式に着目し，ユーザに与える冗長な鍵を削減させた新しい木構造方式を 2 つ提案する．本方式の基本アイディアは，葉からさらに一方向性関数を施して得られる値を時間鍵とすることにより柔軟性を与えることである．提案方式 I では，葉から時間とは逆方向に一方向性関数を施すことで，有効期間終了後に冗長な鍵を与えない．有効期間開始前には冗長な鍵を与える必要があるが，冗長木構造が与える個数の半数に抑えることが可能である．この特性により提案方式 I は，空港の待合室で利用する無線 LAN サービスのような現時点から利用する時間限定サービスに用いるならば，冗長な時間鍵は皆無であり実用的である．提案方式 II では，隣接する 2 つの葉で組を作り，時間と逆方向と順方向にそれぞれ異なる一方向性関数を施し，次数の和が一定となるような組で排他的論理和をとり時間鍵とする．提案方式 II の特徴は，秘密情報のサイズは提案方式 I より若干増えるが，冗長な時間鍵は特定の条件の下ではいっさい与えない点である．冗長な時間鍵を与えてしまう場合においても，同じ条件下で冗長木構造方式，提案方式 I が与える時間鍵の量の半分以下である．この特性からネットワークを介した有料放送のような未来の時間から開始する時間限定サービスを行う場合に実用的であると考えられる．

## 2. 時限付き鍵管理方式

本章では，これまでに，提案された時限付き鍵管理方式として，木構造方式[8] および，効率的な時限付き管理方式であるカオス方式[5),6)]，冗長木構造方式[9] について述べ，それぞれの方式がユーザに与える冗長な鍵の個数について調べる．

### 2.1 準　　備

時限付き鍵管理システムにおいて，サービスを提供する全時間を $m \in \mathbf{Z}_+$（$m > 0$）とし，時間ごとに異なる鍵を割り当てる．ただし，$\mathbf{Z}_+$ は正整数を意味する．割り当てた鍵の 1 つ 1 つを “時間鍵” とし，時間 $i$（$i < m$）における時間鍵を $k_i$ と表記する．本論文を通して時間鍵の鍵長は $\alpha$ ビットとする．また，時間鍵を導出できる情報を “シード鍵” とし，時間鍵 $k_{i_1}, k_{i_2}, \ldots, k_{i_l}$ のシード鍵を $k_{\{i_1,i_l\}}$ と表記する．時間 $i$ から時間 $j$（$i < j < m$）の時間帯を $(i, j)$ と表記し，$n = j - i$ とする．あるユーザが時間帯 $(i, j)$ のコンテンツの受信権を購入した場合，センタは，時間鍵 $k_i, k_{i+1}, \ldots, k_j$ を導出できるシード鍵 $k_{\{i,j\}}$ をユーザに配布するものとする．

全方式とも，センタは時間 $t$（$1 \le t \le m$）に放送するコンテンツを時間鍵 $k_t$ で暗号化するものとする．

### 2.2 カオス方式

センタは一方向性のカオス写像 $f : \{0,1\}^\beta \to \{0,1\}^\beta$ （$\alpha \ll \beta$），効率化パラメータ $\ell$（$1 \le \ell \le \alpha$）を用意し，公開情報とする．ここで，鍵長 $\alpha$，時間 $t$ に対して，

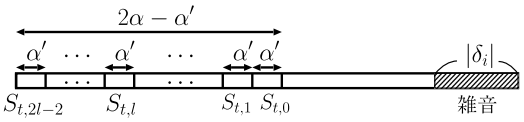


**図 1** シード鍵
Fig. 1 A seed-key of the chaotic scheme.

$$\alpha' := \frac{\alpha}{\ell}, \tag{1}$$

$$\overline{t} := t \pmod{\ell}, \tag{2}$$

と表記する．センタは，一様かつランダムに $K_0 \in \{0,1\}^{\beta}$ を生成し，系列 $K_t$（$1 \le t \le m$）を，

$$K_t = f^{\alpha' t}(K_0), \tag{3}$$

として計算する．得られた系列 $K_t$ において，MSB から $\alpha'$ ビットごとに $2\ell-1$ のパケットを設置し，MSB から順に $S_{t,2\ell-2}, S_{t,2\ell-1}, \ldots, S_{t,0}$ とする．ここで，次に示す 2 変数関数 $H$ を用いて，$K_t$ から連続する $\ell$ 個のパケットを選出し時間鍵 $k_t$ を作成する．

$$\begin{aligned} k_t &= H(K_t, \overline{t}), \\ &= S_{t,\overline{t}+\ell-1} \parallel S_{t,\overline{t}+\ell-2} \parallel \cdots \parallel S_{t,\overline{t}}, \\ \overline{t} &= t \pmod{\ell}. \end{aligned} \tag{4}$$

この関数 $H$ を公開情報とする．これより，時間鍵 $k_t$ は，

$$k_t = H\left(f^{\alpha' t}(K_0), \overline{t}\right), \tag{5}$$

により計算される．

時間帯 $(i,j)$, $n := j-i$ におけるシード鍵について考える．センタは一様かつランダムに選んだ雑音 $\delta_i$ を系列 $K_t$ に付加し，シード鍵とする．そのため，ユーザが契約した期間のみ有効となるように，雑音のサイズ $|\delta_i|$ を設定する必要がある．カオス写像の影響により，鍵更新のたびに雑音は平均 $\alpha'$ ビット拡大し，その分の情報が失われる．時間 $t > j$ において，鍵として抽出するパケットすべてに雑音の影響が及ぶためには，

$$n\alpha' + |\delta_i| = \beta - (2\ell-1)\alpha, \tag{6}$$

を満たすように $\delta_i$ を選ばなければならない．よって，

$$|\delta_i| = \beta - (2\ell-1)\alpha - n\alpha', \tag{7}$$

となる．以上より，ユーザに配布する秘密情報は，

$$k_{\{i,j\}} := K_i + \delta_i, \tag{8}$$

となる．このシード鍵を**図 1** に示す．このとき，時間鍵 $k_t$（$i \le t \le j$）は

$$k_t = H\left(f^{\alpha'(t-i)}(k_{\{i,j\}}), \overline{t}\right), \tag{9}$$

により計算できる．

この方式において，ユーザに与える冗長な時間鍵の個数を調べるため，まずユーザが有効期間外の時間鍵を得る可能性を考える．有効期間終了直後のユーザが次の時間鍵を得るには，時間 $\overline{j} = 0$ の場合，時間鍵の系列すべてを雑音が覆うため $2^{\alpha}$ 回の総当たり攻撃が必要である．一方，$\overline{j} \neq 0$ の場合，$\overline{j+h} = 0$（$h < \ell$, $h \in \mathbf{Z}_+$）となるまでの期間，攻撃なしで時間鍵が手に入る．手に入る時間鍵の個数は $h < \ell$ であり，たかだか $\ell$ 個である．そのため，ユーザに冗長な時間鍵を与えないためには，有効期間終了時間は $\overline{j} = 0$ を満たす離散値（$\ell$ の倍数）とならなければならない．ゆえに，カオス方式ではユーザに与える冗長な時間鍵は有効期間終了時間 $j$ に依存し，冗長を与えないためには $j$ はパラメータ $\ell$ に依存させる必要がある

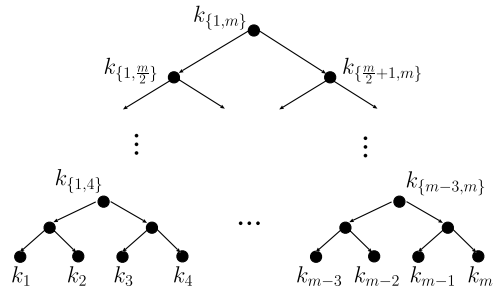


**図 2** 完全二分木 $T$
Fig. 2 Complete binary tree $T$.

### 2.3 木構造方式

センタは擬似乱数生成器 $G : \{0,1\}^{\alpha} \to \{0,1\}^{2\alpha}$ を用意し公開情報とする．$G_L(x)$ を $G(x)$ の上位 $\alpha$ ビット，$G_R(x)$ を $G(x)$ の下位 $\alpha$ ビットとする．一般性を失わずにある $\epsilon \in \mathbf{Z}_+$ が存在し，$m = 2^{\epsilon}$ であることとする．$m$ 個の葉を持つ完全二分木 $T = (T(V), T(E))$ を構成する．$T(V)$，$T(E)$ はそれぞれ完全二分木 $T$ の頂点集合，辺集合を意味している．ここで $v_p, v_l \in T(V)$ において $v_p$ が親，$v_l$ が子の親子関係であるならば $v_p \mapsto v_l$ とし，$v_p$ が先祖，$v_l$ が子孫の先祖関係であるならば $v_p \hookrightarrow v_l$ と表記する．センタは，以下のように再帰的に木の各点に鍵を割り当てる．

- 頂点の鍵 $k_{\{1,m\}} \in \alpha$ を一様かつランダムに選ぶ．
- ある節点 $v$ の鍵が $k_v$ のとき，その左の子の節点には $G_L(k_v)$ を割り当て，その右の子の節点には $G_R(k_v)$ を割り当てる．

上の操作をすべての節点に鍵が割り当てられるまで行う．この操作で生成した完全二分木を**図 2** に示す．$m$ 個の葉の中で $(i,j)$ と対応する葉の集合を $V(R_{i,j})$ として，さらに

$$\begin{aligned} T(R_{i,j}) := \{v \mid v \in T(V), \forall v_l \hookrightarrow v, \\ v_l \in V(R_{i,j})\}, \end{aligned} \tag{10}$$

とする．このとき，あるユーザに配布する秘密情報は，

$$k_{\{i,j\}} := \{\langle v, k_v\rangle \mid v \in T(R_{i,j}), \exists v_p \in T(V)\backslash T(R_{i,j}), v_p \mapsto v\}, \quad (11)$$

となる．この木構造方式では，ユーザに冗長な時間鍵は与えない．

### 2.4 冗長木構造方式

時間帯 $(i,j)$, $n := j - i$ におけるシード鍵に対して，$2^{\eta-1} < n \le 2^{\eta}$ を満たす $\eta \in \mathbf{Z}_+$ を考える．$n$ 個の葉は，葉の数が $2^{\eta}$ のたかだか 2 つの連続した完全二分木に連なっている．これら 2 つの完全部分木を左から順に $T_L$，$T_R$ と呼ぶ．さらに $T_L$ の完全部分木で，葉の数が $2^{\eta}/2$ の 2 つの連続した（1 つの葉も共有していない）完全部分木を左から順に $T_1$，$T_2$ そして $T_R$ の場合も同様に $T_3$，$T_4$ とする．

$$V(R'_{i,j}) := \{T_y(V) \mid V(R_{i,j}) \cap T_y(V) \neq \phi, y = 1,2,3,4\}, \quad (12)$$

に関して，センタは，

$$k_{\{i,j\}} := \{\langle v, k_v\rangle \mid v \in T(R_{i,j}), \exists v_p \in T(V)\backslash T(R'_{i,j}), v_p \mapsto v\}, \quad (13)$$

をユーザに配布する．

この方式において，ユーザに与える冗長な時間鍵の個数を調べる．現実的な状況を考え，木の深さが $\log(m/q)$（$1 \le q \le m$，$q \in \mathbf{Z}_+$）までの階層を木構造方式として，深さが $\log(m/q+1)$ から $\log m$ までの階層を冗長木構造方式として扱う．この様子を**図 3** に示す．ここで，時間 $t$ に対し $\widetilde{t} = t \pmod q$ と表記する．2 つの部分木を利用する場合，$2q - n$ 個の余分な鍵を与えることになり，$\widetilde{i} = 0$ かつ $\widetilde{j} = 1$ のとき最大 $2q-2$ 個の冗長な時間鍵を与えることになる．3 つ以上の部分木を用いる場合も同様である．よって，ユーザに配布する冗長な時間鍵は最大 $2q-2$ 個である．

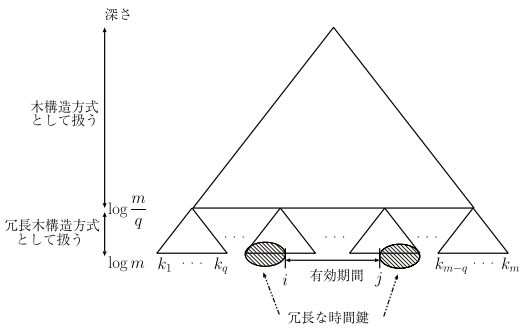


**図 3** 冗長な時間鍵
Fig. 3 Redundant time-key of the redundant tree-structure scheme.

## 3. 提案木構造方式

本章では，木構造方式を用いて，冗長な時間鍵の削減を可能とする方式を 2 つ提案する．提案方式 I では，有効期間終了後にユーザに冗長な時間鍵は与えない．有効期間開始前の時間鍵は与える必要があるが，冗長木構造方式の半分の個数に抑えることが可能である．提案方式 II では，有効期間開始前，終了後ともにユーザに冗長な時間鍵をまったく与えない．ここでは，2.3 節，2.4 節で用いた書式を直接用いる．

### 3.1 提案方式 I

センタは擬似乱数生成器 $G : \{0,1\}^{\alpha} \to \{0,1\}^{2\alpha}$ と一方向性関数 $g : \{0,1\}^{\alpha} \to \{0,1\}^{\alpha}$ を用意し，それぞれ公開情報とする．木構造方式と同様に $G_L(x)$ を $G(x)$ の上位 $\alpha$ ビット，$G_R(x)$ を $G(x)$ の下位 $\alpha$ ビットとする．一般性を失わずにある $\epsilon \in \mathbf{Z}_+$ が存在し，$m = 2^{\epsilon}$ であることとする．木構造方式と同様の手法で，$m/q$ 個の葉を持つ完全二分木 $\tau = (\tau(V), \tau(E))$ を構成し，その葉を左から順に $S_1, S_2, \ldots, S_{m/q}$ とする．それぞれの葉に一方向性関数 $g$ を時間軸と逆向きに用いて次のように $m$ 個の時間鍵を生成する．

$$k_{b+(a-1)q} = g^{q-b}(S_a) \quad 1 \le a \le m/q, \ 1 \le b \le q \quad (14)$$

$m = 16$，$q = 4$ における時間鍵を**図 4** に示す．

ここで，時間帯 $(i,j)$, $n := j - i$ におけるシード鍵について考える．$m$ 個の時間鍵の中で $(i,j)$ と対応する集合を $V(R_{i,j})$ として，$S(R_{i,j}) := \{S_k \mid S_k \rightharpoonup k_t, k_t \in V(R_{i,j})\}$ とする．ただし，$S_a$ から生成される $q$ 個の時間鍵の集合を $V(S_a)$ とし，$\forall k_t \in V(S_a)$ に対して $S_a$ と $k_t$ の関係を $S_a \rightharpoonup k_t$ と表記する．さらに $S(R_{i,j})$ の先祖の集合を，

$$\tau(R_{i,j}) := \{v \mid v \in \tau(V), \forall v_l \hookleftarrow v, v_l \in S(R_{i,j})\}, \quad (15)$$

とする．有効期間終了時間 $j$ の時間鍵 $k_j$ を生成するシードを $S_y$ とおき，$S(R_{i,j})$ のうち $S_y$ を除外した

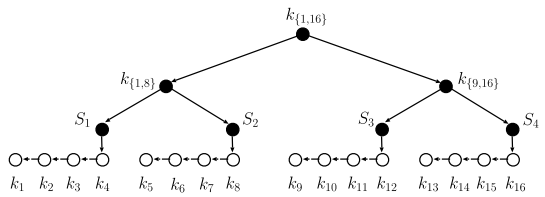


**図 4** 提案方式 I の時間鍵例
Fig. 4 An example of time-keys of the proposal scheme I.

集合の先祖の集合を，

$$\tau(R_{i,j} \setminus S_y) := \{v \mid v \in \tau(V), \forall v_r \hookleftarrow v, \\ S_y \rightharpoonup k_j, v_r \in S(R_{i,j}) \setminus S_y\}, \quad (16)$$

とする．このとき，あるユーザに配布する秘密情報は，

$$k_{\{i,j\}} := \begin{cases} \{\langle v, k_v\rangle \mid v \in \tau(R_{i,j}), \\ \exists v_p \in \tau(V)\setminus\tau(R_{i,j}), v_p \mapsto v\} & (\widetilde{j}=0) \\ \{\langle v, k_v\rangle, k_j \mid v \in \tau(R_{i,j} \setminus S_y), \\ \exists v_p \in \tau(V) \setminus \tau(R_{i,j} \setminus S_y), \\ v_p \mapsto v\} & (\widetilde{j} \neq 0) \end{cases}$$

$$\widetilde{j} = j \pmod q \quad (17)$$

となる．

**定理 1** 提案方式 I が余分に与える時間鍵はたかだか $q-1$ 個である．

**証明.** 有効期間終了時間 $j$ 以降に与える時間鍵について考える．$\widetilde{j}=0$ のとき，$S_y = k_{\{j-q+1,j\}}$ となりユーザに余分な時間鍵は与えない．一方，$\widetilde{j} \neq 0$ のとき，$0 \leq h \leq q-1$ を満たす $h \in \mathbf{Z}_+$ を考えると，$S_y = k_{\{j-q+1+h,j+h\}}$ となる．式 (17) で示すように，$k_j$ を生成するシード鍵として，ユーザには秘密情報は $S_y$ ではなく，$k_j$ が与えられる．これらの情報から求められる時間鍵は，一方向性関数 $g$ の性質より，$k_{j-q+1+\delta}, \ldots, k_j$ であり，有効期間終了後の時間鍵 $k_{j+1}, \ldots, k_{j+\delta}$ を得ることはできない．よって，ユーザに $j$ 以降の余分な時間鍵は与えない．

次に，有効期間開始時間 $i$ 以前に与える時間鍵について考える．$S_v \rightharpoonup k_i$ を満たす $S_v$ を考えると，$S_v = k_{\{i-q+1+\delta,i+\delta\}}$ である．$0 \leq \delta \leq q-1$ より，$i-q+1+\delta \leq i-q$ となり，$i$ よりたかだか $q-1$ 個前までの時間鍵を与えることになる．

以上より，余分な時間鍵数は $q-1$ であることが証明された． □

### 3.2 提案方式 II

センタは，提案方式 I の場合と同様，擬似乱数生成器 $G$，一方向性関数 $g_1, g_2 : \{0,1\}^\alpha \to \{0,1\}^\alpha$ を用意し，それぞれ公開情報とする．また，提案方式 I と同様の手順で完全二分木 $\tau$ の 2 倍の葉数 $2m/q$ を持つ完全二分木 $\tau = (\tau(V), \tau(E))$ を構成する．この完全二分木 $\tau$ の葉を左から順に $S_1, S_2, \ldots, S_{2m/q}$ とする．この完全二分木の奇数の葉 $S_{2a-1}$ ($1 \leq a \leq m/q$, $a \in \mathbf{Z}_+$) に一方向性関数 $g_1$ を時間軸と順方向に $q$ 回作用させ左部分時間鍵 $\psi_{1+(a-1)q}, \psi_{2+(a-1)q}, \ldots, \psi_{aq}$ を生成する．また，偶数の葉 $S_{2a}$ に一方向性関数 $g_2$ を時間軸と逆方向に $q$ 回作用させ右部分時間鍵 $\varphi_{1+(a-1)q}, \varphi_{2+(a-1)q}, \ldots, \varphi_{aq}$ を生成する．これら部分時間鍵の次数の等しい組で排他的論理和をとり，$m$ 個の時間鍵を生成する．つまり，

$$\psi_{b+(a-1)q} = g_1^{b-1}(S_{2a-1}),$$
$$\varphi_{b+(a-1)q} = g_2^{q-b}(S_{2a}),$$
$$k_{b+(a-1)q} = \psi_{b+(a-1)q} \oplus \varphi_{b+(a-1)q},$$
$$1 \leq a \leq m/q,$$
$$1 \leq b \leq q. \quad (18)$$

とする．ただし，$\oplus$ はビットごとの排他的論理和とする．$m=8$，$q=4$ における時間鍵を**図 5** に示す．

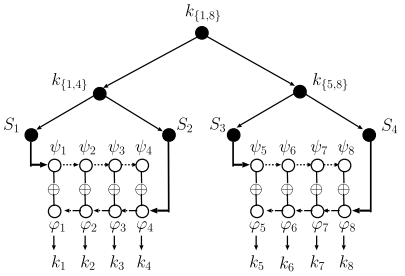


**図 5** 提案方式 II の時間鍵例
Fig. 5 An example of time-keys of the proposal scheme II.

ここで，時間帯 $(i,j), n := j-i$ におけるシード鍵について考える．$S_{2a-1}$，$S_{2a}$ から生成される $q$ 個の時間鍵の集合を $V(S_{2a-1}, S_{2a})$ とし，$\forall k_t \in V(S_{2a-1}, S_{2a})$ に対して $S_{2a-1}$，$S_{2a}$ と $k_t$ の関係を $(S_{2a-1}, S_{2a}) \rightharpoonup k_t$ と表記する．さらに，$S(R_{i,j})$ の先祖の集合を，

$$\tau(R_{i,j}) := \{v \mid v \in \tau(V), \forall v_l \hookleftarrow v, v_l \in S(R_{i,j})\}, \quad (19)$$

とする．有効期間開始時間 $i$ の時間鍵 $k_i$ を生成するシードを $(S_{2x-1}, S_{2x})$ とおき，$S(R_{i,j})$ のうち $S_{2x-1}$ を除外した集合の先祖の集合を，

$$\tau(R_{i,j} \setminus S_{2x-1}) := \{v \mid v \in \tau(V), \forall v_r \hookleftarrow v, \\ (S_{2x-1}, S_{2x}) \rightharpoonup k_i, v_r \in S(R_{i,j}) \setminus S_{2x-1}\}. \quad (20)$$

とする．また，有効期間終了時間 $j$ の時間鍵 $k_j$ を生成するシードを $(S_{2y}, S_{2y-1})$ とおき，$S(R_{i,j})$ のうち $S_{2y}$ を除外した集合の先祖の集合を，

$$\tau(R_{i,j} \setminus S_{2y}) := \{v \mid v \in \tau(V), \forall v_r \hookleftarrow v, \\ (S_{2y-1}, S_{2y}) \rightharpoonup k_j, v_r \in S(R_{i,j}) \setminus S_{2y}\}. \quad (21)$$

$S(R_{i,j})$ のうち $S_{2x-1}$，$S_{2y}$ を除外した集合の先祖の集合を，

$$\tau(R_{i,j} \setminus S_{2x-1,2y}) := \{v \mid v \in \tau(V), \\ \forall v_r \hookleftarrow v, v_r \in S(R_{i,j}) \setminus S_{2x-1} \cap S_{2y}\}. \quad (22)$$

とする．以上のパラメータを用いるならばユーザに配布する秘密情報は，

$$k_{\{i,j\}} := \begin{cases} \{\langle v, k_v\rangle \mid v \in \tau(R_{i,j}), \\ \exists v_p \in \tau(V) \setminus \tau(R_{i,j}), v_p \mapsto v\}, \\ (\widetilde{i}, \widetilde{j} = 0) \\ \\ \{\langle v, k_v\rangle, \varphi_j \mid v \in \tau(R_{i,j} \setminus S_{2y}), \\ \exists v_p \in \tau(V) \setminus \tau(R_{i,j} \setminus S_{2y}), v_p \mapsto v\}, \\ (\widetilde{i} = 0, \widetilde{j} \neq 0) \\ \\ \{\langle v, k_v\rangle, \psi_i \mid v \in \tau(R_{i,j} \setminus S_{2x-1}), \\ \exists v_p \in \tau(V) \setminus \tau(R_{i,j} \setminus S_{2x-1}), v_p \mapsto v\}, \\ (\widetilde{i} \neq 0, \widetilde{j} = 0) \\ \\ \{\langle v, k_v\rangle, \psi_i, \varphi_j, \mid v \in \tau(R_{i,j} \setminus S_{2x-1,2y}), \\ \exists v_p \in \tau(V) \setminus \tau(R_{i,j} \setminus S_{2x-1,2y}), v_p \mapsto v\} \\ (\widetilde{i}, \widetilde{j} \neq 0), \end{cases}$$

$$\widetilde{j} = j \pmod{q}, \tag{23}$$

となる．

**定理 2** 提案方式 II は，単一期間契約したユーザに対して余分な時間鍵は与えない．

**証明.** 有効期間開始時間 $i$ 以前に与える時間鍵について考える．$\widetilde{i} = 0$ のとき，$S_{2x-1}, S_{2x} = k_{\{i,i+q-1\}}$，$0 \leq h \leq q-1$ となり，ユーザに余分な時間鍵は与えない．$\widetilde{i} \neq 0$ のとき，$S_{2x-1}, S_{2x} = k_{\{i-q+1+h,i+h\}}$ である．$k_i$ を生成するシード鍵として，ユーザに与えられる秘密情報は式 (23) より，$\psi_i$，$S_{2x}$ である．これらの情報から求めれるものは，一方向性関数 $g_2$ の性質より，左部分時間鍵 $\psi_i, \ldots, \psi_{i+h}$ および右部分時間鍵 $\varphi_{i-q+1+h}, \ldots, \varphi_{i+h}$ であり，得られる時間鍵は $k_i, \ldots, k_{i+h}$ である．よって，有効期間以前の時間鍵を計算することはできない．

次に，有効期間終了時間 $j$ 以降に与える時間鍵について考える．$\widetilde{j} = 0$ のとき，$S_{2y-1,2y} = k_{\{j-q+1,j\}}$ となりユーザに余分な時間鍵は与えない．$\widetilde{j} \neq 0$ のとき，$S_{2y-1}, S_{2y} = k_{\{j-q+1+h,j+h\}}$，$0 \leq h \leq q-1$ となる．$k_j$ を生成するシード鍵として，ユーザに与えられる秘密情報は $S_{2y-1}$，$\varphi_j$ である．これらの情報から求められるものは，一方向性関数 $g_1$ の性質より，左部分時間鍵 $\psi_{j-q+1+h}, \ldots, \psi_{j+h}$ および右部分時間鍵 $\varphi_{j-q+1+h}, \ldots, \varphi_j$ であり，得られる時間鍵は $k_{j-q+1+h}, \ldots, k_j$ である．よって，有効期間終了後の時間鍵を計算することはできない．

以上より，ユーザに余分な時間鍵を与えないことが証明された． □

## 4. 考　　察

本章では提案木構造方式を，2 章で述べた他の時限付き鍵管理方式と，冗長な時間鍵の個数，端末秘密量，端末計算量，安全性において比較，検討を行う．

### 4.1 冗長な時間鍵

ユーザに与える冗長な時間鍵について比較，検討する．カオス方式では，ユーザに設定できる有効期間終了時間 $j$ が $\overline{j} = 0$ を満たす離散値をとる必要があり，この値は効率化パラメータ $\ell$ に依存する．そのため，方式の効率を上げるほどユーザに設定できる有効期間の幅が大きくなってしまう．ユーザに冗長な時間鍵を与えることを許す場合，有効期間終了後に平均 $\ell/2$ 個与える必要がある．

図 3 の現実的な状況での冗長木構造方式では有効期間開始前，終了後それぞれに平均 $(q-1)/2$ 個，合わせて平均 $q-1$ 個の冗長な時間鍵を与える必要がある．

提案方式 I では，有効期間終了後には与えず，有効期間開始時には平均 $(q-1)/2$ 個の冗長な時間鍵を与える必要がある．ただし，$q$ はカオス方式の場合と異なり定数であり，冗長木構造方式の $q-1$ に比べ半分の個数である．また，提案方式 I でユーザに与える冗長な時間鍵はすべて有効期間開始前である．実際のシステム運用を考えた場合，ユーザの要求が現在から時間 $j$ までの鍵であれば，過去の鍵を与えても問題にならないが，未来の鍵を与えることはシステム運用者側が不利益を被る．よって，有効期間終了後に冗長な時間鍵を与えない提案方式の特性は実用的であると考えられる．提案方式 II では，有効期間開始前，終了後ともに冗長な時間鍵は与えない．そこで，未来の時間から開始する時間限定サービスでの運用において実用的であると考えられる．

### 4.2 端末秘密量，端末計算量

端末秘密量，端末計算量とは，文献 9) で提案されている時限付き鍵管理方式の効率に関するパラメータである．

**定義 1** 本論文では，端末秘密量，端末計算量を次のように定義する．

- 端末秘密量は，端末が秘密に保管する必要のある鍵数を意味し，シード鍵に含まれる鍵すなわち節点の個数である．有効期間 $n$ とパラメータ $q$ を用いて評価する．
- 端末計算量は，有効期間内のすべての時間鍵をシード鍵の集合から計算するために必要なステップ数であり，端末秘密量同様，$n$，$q$ を用いて評

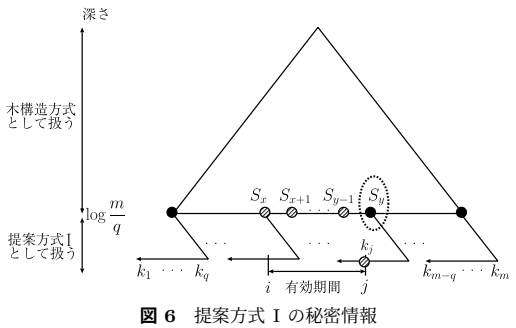


**図 6** 提案方式 I の秘密情報
Fig. 6 The secret information of the proposal scheme I.

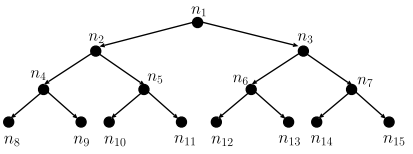


**図 7** 端末秘密量の変化
Fig. 7 The change of the amount of the secret information in each terminal.

価する．

図 3 の現実的な状況での冗長木構造方式と提案木構造方式とを比較，検討する．すべての方式において，深さ 0 から $\log(m/q)$ までの範囲は木構造方式として扱うため，その葉の下に続く階層について議論する．

冗長木構造方式では，ユーザに与える秘密情報は深さ $\log(m/q)$ の節点でたかだか $\lfloor n/q \rfloor + 2$ 個となる．よって，冗長木構造方式として扱う範囲での端末秘密量は，$\lfloor n/q \rfloor + 2$ となる．次に，端末計算量について考える．冗長木構造方式として扱う範囲の部分木の深さは $\log q$ であり，深さ $\log q$ の部分木が生成する時間鍵をすべて計算するためには，すべての枝に相当する個数分だけステップが必要である．その計算量は，$2 + 4 + 8 +, \ldots, +q = 2q - 2$ である．部分木はたかだか $\lfloor n/q \rfloor + 2$ 個あるため，有効期間内のすべての時間鍵を計算するには，

$$\left(\left\lfloor \frac{n}{q} \right\rfloor + 2\right)(2q-2) = 2n + 4q - 2\left\lfloor \frac{n}{q} \right\rfloor - 4 \tag{24}$$

ステップの計算が必要となる．よって，冗長木構造方式として扱う範囲の端末計算量は，$2n - 2\lfloor n/q \rfloor + 4q - 4$ となる．

提案方式 I では，ユーザに与える秘密情報は深さ $\log(m/q)$ の節点をたかだか $\lfloor n/q \rfloor + 1$ 個と時間鍵 $k_j$ である．よって，提案方式 I での端末秘密量は冗長木構造方式の場合と同様 $\lfloor n/q \rfloor + 2$ である．ただし，ここで注意が必要である．ユーザに与える深さ $\log(m/q)$ の節点の数に注目すると，提案方式 I では冗長木構造方式と比べ**図 6** で示す一番右の節点 $S_y$ が 1 つ少ない．ユーザに配布するシード鍵は，深さ $\log(m/q)$ の節点の先祖集合であるため，提案方式 I と冗長木構造方式では端末秘密量が変化する．そこで，提案方式 I が冗長木構造方式と比べてどの程度端末秘密量が変化するか見積もる．提案方式 I，冗長木構造方式では，深さ 0 から $\log(m/q)$ までの範囲を木構造方式として扱う．よってユーザに与えるシード鍵は，深さ $\log(m/q)$ のユーザに与える節点集合を $T(R_{tree})$ とすると，式 (11) より，

$$k_{\{i,j\}} := \{\langle v, k_v \rangle \mid v \in T(R_{tree}), \exists v_p \in T(V) \backslash T(R_{tree}), v_p \mapsto v\}, \tag{25}$$

と表せる．端末秘密量はこの $k_{\{i,j\}}$ に含まれる節点の個数である．節点集合 $T(R_{tree})$ の位数を $n_{tree}$ とすると，冗長木構造方式ではたかだか $n_{tree} = \lfloor n/q \rfloor + 2$ となるのに対し，提案方式 I ではたかだか $n_{tree} = \lfloor n/q \rfloor + 1$ となり一番右の節点 $S_y$ が少ない．この $S_y$ の有無で変化する端末秘密量について調べる．

**定理 3** 提案方式 I における端末秘密量は，冗長木構造方式より平均 $1 - 1/n_{tree}$ 増加する．

**証明.** $S_y$ がある節点の左の子である場合を考える．部分木の左の子である確率は $n_{tree}/2$ であり，このとき節点 $S_y$ を除去すると端末秘密量が 1 小さくなる．たとえば，**図 7** で $S_y = n_{14}$ とし $n_{14}$ を除去すると，ユーザに与える節点は $n_2$，$n_6$，$n_{14}$ から $n_2$，$n_6$ に変わり配布する節点は 1 つ少なくなる．$S_y$ がある節点の右の子である確率は $n_{tree}/2$ である．このとき節点 $S_y$ を除去すると，ユーザに与える端末秘密量は $S_y$ が属する部分木の深さに比例して増加する．部分木の深さが少なくとも 1 以上である確率は $n_{tree}/2$ であり，深さが 1 の場合 $S_y$ を除去しても端末秘密量は変わらない．たとえば，図 7 で $S_y = n_9$ とし $n_9$ を除去すると，ユーザに与える節点は $n_4$ から $n_8$ に変わるが，配布する節点の個数は変わらない．次に，部分木の深さが少なくとも 2 以上ある確率は $n_{tree}/4$ であり，深さが 2 の場合 $S_y$ を除去すると端末秘密量は 1 増加する．たとえば，図 7 で $S_y = n_{11}$ とし $n_{11}$ を除去すると，ユーザに与える節点は $n_2$ から $n_4$，$n_{10}$ に変わり配布する節点は 1 つ増える．部分木の深さが少なくとも 3 以上ある確率は $n_{tree}/8$ であり，深さが 3 の場合 $S_y$ を除去すると端末秘密量は 2 増加する．たとえば，図 7 で $S_y = n_{15}$ とし $n_{15}$ を除去すると，ユーザに与える節点は $n_1$ から $n_2$，$n_6$，$n_{14}$ に変わり配布する節点は 2 つ増える．同様に，部分木の深さが少

なくとも $k$ 以上ある確率は $n_{tree}/2^k$ であり，深さが $k$ の場合 $S_y$ を除去すると端末秘密量は $k-1$ 増加する．部分木の深さ $\log n_{tree}$ のとき最大となる．ここで，深さ $k$ の部分木の右の子は，深さが $k+1$ 以上となる部分木の右の子をすべて包含することに注意する．それゆえ，部分木の右の子において変化する端末秘密量の総量は，

$$\begin{aligned}&\frac{n_{tree}}{4}+\frac{n_{tree}}{8}+\frac{n_{tree}}{16}+\ldots+\frac{n_{tree}}{2^{\log n_{tree}}}\\&=\frac{n_{tree}}{4}\sum_{k=1}^{\log n_{tree}-1}\left(\frac{1}{2}\right)^{k-1}\\&=\frac{n_{tree}}{2}\left(1-\left(\frac{1}{2}\right)^{\log n_{tree}-1}\right)\\&=\frac{n_{tree}}{2}\left(1-\frac{2}{n_{tree}}\right)\end{aligned}\tag{26}$$

となることが分かる．左の子においては $(-1)\cdot n_{tree}/2$ であるため，端末秘密量の平均変化量は，

$$\left(-\frac{n_{tree}}{2}+\frac{n_{tree}}{2}\left(1-\frac{2}{n_{tree}}\right)\right)\cdot\frac{1}{n_{tree}}=-\frac{1}{n_{tree}}\tag{27}$$

となる．ただし提案方式 I では，深さ $\log(m/q)$ の節点以外に時間鍵 $k_j$ をユーザに与える必要があるため，端末秘密量の平均変化量はさらに 1 大きくなり，

$$1-\frac{1}{n_{tree}}\tag{28}$$

となる． □

端末計算量は，一方向性関数 $g$ を作用させる回数が $q$ であり，作用させる節点の数がたかだか $\lfloor n/q\rfloor+1$ であるので，

$$q\cdot\left(\left\lfloor\frac{n}{q}\right\rfloor+1\right)=n+q\tag{29}$$

となる．冗長木構造方式の式 (24) と比較すると約半分である．

提案方式 II では，ユーザに与える秘密情報は深さ $\log(m/q)$ の節点がたかだか $\lfloor n/q\rfloor$ 個，深さ $\log(m/q)+1$ の節点がたかだか 2 個，部分時間鍵がたかだか 2 個であり，端末秘密量は $\lfloor n/q\rfloor+4$ である．ユーザに与える深さ $\log(m/q)$ の節点の数は $n_{tree}=\lfloor n/q\rfloor$ であり，冗長木構造方式の場合と比べると，**図 8** で示す一番左の節点 $S_x$ と一番右の節点 $S_y$ が少ない．定理 3 の式 (27) から，$S_y$ が少ないことで端末秘密量は平均 $-1/n_{tree}$ 変化する．また，同様に $S_x$ が少ないことにより，端末秘密量は平均 $-1/n_{tree}$ 変化する．提案方式 II では深さ $\log(m/q)$ の節点以外に $\log(m/q)+1$ の節点をたかだか 2 個，部分時間鍵をたかだか 2 個与える必要があるため，端末秘密量は冗長木構造方式に比べて平均 $4-2/n_{tree}$ 増加する．

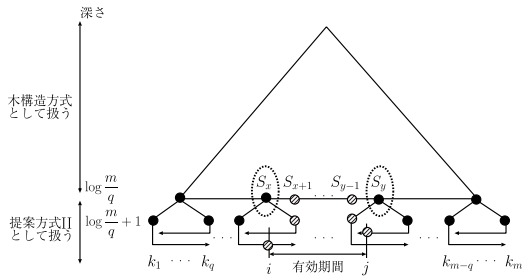


**図 8** 提案方式 II の秘密情報
Fig. 8 The secret information of the proposal scheme II.

端末計算量は，一方向性関数 $g_1$，$g_2$ を作用させる回数が $q$ 回で作用させる節点がたかだか $2\lfloor n/q\rfloor+4$ 個であり，さらに排他的論理和を $q$ 回施すため，

$$q\cdot(2\lfloor n/q\rfloor+4)+q=2n+5q\tag{30}$$

となる．冗長木構造方式の式 (24) と比較すると，ほぼ同等であることが分かる．

提案木構造方式の端末秘密量は，冗長木構造方式と比べると提案方式 I で平均 $1-1/n_{tree}$ 増加し，提案方式 II で平均 $4-2/n_{tree}$ 増加する．端末計算量は，提案方式 I で冗長木構造方式の約半分の量で，提案方式 II でほぼ同等である．

### 4.3 安 全 性

時限付き鍵管理方式の安全性は，ユーザが与えられたシード鍵から有効期間外の時間鍵を入手できる可能性で決まる．そこで本節では，木構造の構成要素である擬似乱数生成器に明確な定義を与え，時間鍵漏洩の原因となる周期性が現れる確率を求めることで提案木構造方式の安全性を調べる．また，定理 1，定理 2 で単一期間契約したユーザに与える冗長な時間鍵について調べたが，さらに複数期間契約したユーザの結託攻撃で漏洩する冗長な時間鍵について調べる．

冗長木構造方式，提案木構造方式共通となる木構造を実現させるために，ある親となる節点の値を擬似乱数生成器に入力し，その出力を左右の子とする操作を葉に至るまで繰り返している．この木構造において安全性を考えると，親から子を求めることは容易であるが，子から親を求めることが困難となる性質が不可欠である[10]．この擬似乱数生成器 $G$ の定義を以下に与える．

**定義 2** 擬似乱数生成器 $G$ は，$\alpha$ ビットの系列を $2\alpha$ ビットに写像する決定的多項式時間アルゴリズムであり，その出力系列は真性乱数と多項式計算量で区別できない．

さらに木構造方式において，もし任意の節点の組もし

くは葉の組が同じ値となった場合，周期性が現れることとなり問題となる．そこで，提案木構造方式において周期性が現れる確率を評価する．

$N$ 個の集合から重複を許して無作為に $Q$ 個を選出した際に，同じものが含まれている可能性は，

$$P = 1 - \exp\left(\frac{-Q(Q-1)}{2N}\right) \tag{31}$$

となることが Birthday Paradox[11] より知られている．

木構造を採用する冗長木構造方式および提案木構造方式では，完全二分木を作成するために用いる擬似乱数生成器は $\alpha$ ビットの入力に対して，出力は $2\alpha$ ビットであり，上位 $\alpha$ ビットと下位 $\alpha$ ビットに分けて 2 分木としている．その完全二分木の節点および葉の総数は，葉の総数を $n$ とすると $2n-1$ 個である．それゆえ，任意の節点もしくは葉の組が同じ値となる確率は，$1-exp(-(2n-1)(n-1)/2^{\alpha})$ となる．木構造の葉の個数 $n$ をたかだか $2^{32}$ と仮定し，$\alpha = 128$ とすれば，この確率は無視できるほど低い．それゆえ，木構造を実現させる際に，定義を満たす擬似乱数生成器を用いれば，衝突が生じる可能性はきわめて低いことが分かる．また，提案木構造方式では木構造の葉に一方向性関数を適用させて時間鍵を生成しており，それらの個数もたかだか $q$ 個であるため，任意の葉が重複する確率も同様にきわめて低いことが明らかである．

時限付き鍵管理方式では，結託したユーザを追跡する機能を有しておらず，ユーザに与えた期間分の時間鍵は結託ユーザに漏洩してしまう．そこで，提案木構造方式についてユーザに与えた期間以外の時間鍵に関して漏洩する可能性があるか考察する．

提案方式 I において，2 つの期間 $A:\{i_a, j_a\}$，期間 $B:\{i_b, j_b\}$（$i_a < j_a < i_b < j_b$）を契約した場合について考える．定理 1 より，時間 $i_a$，$i_b$ 以前にはそれぞれ最大 $q-1$ 個，平均 $(q-1)/2$ 個時間鍵を与え，時間 $j_a$，$j_b$ 以降には冗長な時間鍵はいっさい与えない．期間 $j_a < t < i_b$ のの時間鍵を計算するためには，その期間のシード鍵 $k_{\{j_a,i_b\}}$ が必要である．しかし，シード鍵 $k_{\{j_a,i_b\}}$ を入手するためには，一方向性関数 $g$ を逆向きに解き，かつ擬似乱数生成器 $G$ を解析する必要があり困難である．よって，2 つの契約期間の場合，最大で $2q-2$ 個，平均 $q-1$ 個の時間鍵を与える必要がある．期間数を $c$ ($c \in \mathbf{Z}_+$) とすると，ユーザに与える冗長な時間鍵は契約期間の数に比例し平均 $c(q-1)/2$ 個である．

提案方式 II において，ユーザが 2 つの期間 $A:\{i_a, j_a\}$，期間 $B:\{i_b, j_b\}$（$i_a < j_a < i_b < j_b$）を契約した場合について考える．期間が $i_a$ 未満および $j_b$ より先の時間鍵に関しては期間 $A$，$B$ が互いに干渉を引き起こさないため，定理 2 よりユーザに冗長な時間鍵が漏洩することはない．ここで，期間 $j_a < t < i_b$ に着目する．$j_a$ と $i_b$ が同じシードに属する場合，左部分鍵 $\psi_a$ と右部分鍵 $\varphi_b$ を用いれば，この期間内の時間鍵を求めることができる．この際，最大で $q-2$ 個，平均 $(q-2)/2$ の冗長な時間鍵を与えることになる．ただし，$j_a$ と $i_b$ が異なるシードに属する場合，定理 2 より冗長な時間鍵を与えることはない．期間数を $c$ とし，すべての期間において上記の条件を満たす場合，ユーザに与える冗長な時間鍵は平均 $(c-1)(q-2)/2$ 個である．提案方式 II では，$j_a$ と $i_b$ が同じシードに属する場合のみ冗長な時間鍵を与えるが，提案方式 I の場合と比較すれば少なくなっている．実装したアプリケーションにおいてユーザが複数期間契約した際，$j_a < t < i_b$ となる期間は契約しない制約を設けることで，この冗長な時間鍵を与えることは防止することができる．

## 5. ま と め

本論文では，冗長木構造方からユーザに与える秘密情報のサイズを増大させることなく，冗長な時間鍵の個数を削減できる方式を提案した．提案木構造方式では葉からさらに一方向性関数を施すことにより，冗長な時間鍵を与えることなく任意の有効期間設定を行える時限付き鍵管理方式を実現した．提案方式 I では，有効期間開始前に冗長な時間鍵を与える必要があるが，その個数は冗長木構造方式の半分に抑えることができる．この特性により，サービス開始時が現時点からのサービスにおいては実用的である．提案方式 II では，提案方式 I に比べユーザに与える秘密情報のサイズは若干増加するが，単一期間契約のユーザにはいっさい冗長な時間鍵を与えない．よって，未来の時間から開始するサービスにおいて運用が可能である．この方式では，ある条件の整った結託に関しては部分的に冗長な時間鍵を与えてしまうが，その個数は提案方式 I よりも少ない．また，実際のアプリケーションで制約を設けることで，冗長な時間鍵を与えることを防止することができる．

## 参 考 文 献

1) Fiat, A. and Naor, M.: Broadcast encryption, *Proc. Crypto1993*, LNCS 773, pp.480–491 (1994).
2) Naor, D., Naor, M. and Lotspiech, J.: Revocation and tracing schemes for stateless re-

civers, *Proc. Crypto2001*, LNCS 2139, pp.41–62 (2001).
3) Kuribayashi, M. and Tanaka, H.: A new key generation method for broadcasting system with expiration date, *Proc. SITA2004*, pp.323–326 (2004).
4) Kuribayashi, M. and Tanaka, H.: Key generation scheme for broadcast encryption exploiting chaotic sequences, *Proc. SCIS2005*, pp.1135–1140 (2005).
5) 田中敏也，栗林　稔，森井昌克，田中初一：カオス写像を利用した期間限定サービス用鍵更新方式，2005 年情報理論とその応用学会，pp.275–278 (2005).
6) 田中敏也，栗林　稔，森井昌克，時間限定サービスのための鍵管理技術，信学技報 OIS，Vol.105, No.529, pp.5–9 (2006).
7) 楫　勇一，野島　良，時間限定サービスを実現するための時系列鍵管理方式，2005 年暗号と情報セキュリティシンポジウム，pp. 289–294 (2005).
8) Yoshida, M., Kaji, Y. and Fujiwara, T.: A time-limited key management based on a one-way permutaion tree, *IEICE and SITA Joint Conference on Information Theory*, Vol.105, No.85, pp.165–170 (2005).
9) 野島　良，古原和邦，今井秀樹：現実的な時限付鍵管理方式の考察，2005 年情報理論とその応用学会，pp.591–594 (2005).
10) 岡本栄司：暗号理論入門第 2 版，共立出版，(2002).
11) Stinson, D.R.: *Cryptography theory and practice, 3rd edition*, Chapman and Hall/CRC (2006).

(平成 18 年 11 月 24 日受付)
(平成 19 年 6 月 5 日採録)

**田中　敏也**

2005 年神戸大学工学部電気電子工学科卒業．2007 年神戸大学大学院自然科学研究科電気電子工学専攻修了．2007 年 P & G ジャパン株式会社入社，現在に至る．在学時，暗号と情報セキュリティ等の研究に従事．電子情報通信学会会員．

**栗林　　稔**

1999 年神戸大学工学部電気電子工学科卒業．2001 年神戸大学大学院自然科学研究科電気電子工学専攻修了．2004 年同大学博士（工学）．2002 年同大学工学部電気電子工学科助手，2007 年同大学大学院工学研究科助教，現在に至る．電子透かし，暗号と情報セキュリティ，符号理論等の研究に従事．IEEE，電子情報通信学会，情報理論とその応用学会各会員．

**森井　昌克**（正会員）

1983 年佐賀大学理工学部卒業．1989 年大阪大学大学院工学研究科通信工学専攻博士課程修了．工学博士．同年京都工芸繊維大学工芸学部電子情報工学科助手．1990 年愛媛大学工学部情報工学科講師，1992 年同助教授，1995 年徳島大学工学部知能情報工学科教授を経て，2005 年神戸大学工学部電気電子工学科教授．情報セキュリティ，代数的符号理論，離散数学，ディジタル信号処理アルゴリズム，コンピュータネットワーク等の研究，教育に従事．IEEE，電子情報通信学会，情報理論とその応用学会各会員．