論　文

認証鍵の生成・管理のための多次元インデックス法

董　　際国[†a)]　　森田　啓義[††b)]

A Multi-Dimensional Index Method for Key Generation and Key Management

Jiguo DONG[†a)] and Hiroyoshi MORITA[††b)]

あらまし　単一のマスターキーから大量の ID・パスワードの生成だけでなく，それらの安全な管理にも適した擬似乱数列の生成方法を提案する．我々は，CSB 生成器の内部関数の数を 2 から $m(m \geq 2)$ に拡大し，インデックス集合の大きさも可変にして，GGM 法を一般化する．また，我々は，キーの総数が M のとき，提案法における M 個のキーの生成に必要な平均時間を最小になるよう，最適な次元数を求める．そして，SHA1 を擬似乱数生成器として提案法に適用した場合の，生成されたキーの衝突とキーの生成の所要時間についての実験結果を報告する．

キーワード　多次元乱数生成，鍵の管理

1. まえがき

電子システムによる認証（自動識別）技術が多くの分野で使われている．一方，公共空間を介した認証は盗聴され易い．リモコンキーの信号が傍受され，車が盗まれ，車の中が荒らされることはその一例である．

認証コードの盗聴を防ぐには使い捨てが有効である．しかし，車のリモコンキーに使い捨て認証を使用する方法では，認証システムの故障やリモコンの紛失などに対処するアフターサービスのため，車にある全ての認証コードをその車が存在する限り，安全に保存・管理する必要がある．

これまで，認証，識別用 ID・パスワードや認証鍵（以下，キーと呼ぶ）の生成と管理についての議論は別々のカテゴリーでされている [1], [2].

高速に大量なキーを生成する一手段としては，非線形の擬似乱数生成器を使う方法が知られている [1]．例えば，N ビットのキーが必要ならば，擬似乱数生成器から生成したビット列 $r_0r_1...r_j...$（$r_j \in \{0,1\}, j = 0,1,\ldots$）を順次切り出せばよい．

キーの管理には木構造に基づく LKH 法 [3] とそれを改良した手法 [4]～[6] が提案されている．キー管理は，キーやその生成情報などを秘密情報として記憶媒体に保存し，暗号化やアクセス制御により保護するのが一般的である [1], [2]．保存形態については，一つの記憶媒体に集中管理するだけでなく，複数の媒体に分散したり，階層化する手法も提案されている [1], [2]．このようなキーの管理方法では多数のユーザを抱えるシステムではキーの使い捨て暗号や認証の実現には依然多大な保存・管理コストを要する．

また，階層関係をもつシステムにおいて，暗号化手法を用いたキーの管理方法が Akl ら [7] により提案されている．すなわち，セキュリティー許可レベルの低いキーはセキュリティー許可レベルの高いキーから暗号化関数 g を介して得られる：

$$K_{M-1} = g(K_M), \quad M = 2, 3, \ldots \tag{1}$$

Akl らの方法は高セキュリティーレベルのユーザ (K_M) が低セキュリティーレベルのユーザ (K_{M-1}) が保持する情報をアクセスできるが，逆方向のアクセスは許容されない機能をもち，各レベルのユーザは一つのキーだけを管理すればよいという特徴をもつ．彼らの方法を発展させ，ユーザの加入時などの拡張を容易にした方法 [8] やキーに有効期限を与える方法 [9] も報告されている．

† セリック株式会社，東京都
Selic Corporation, 705, 7–5, Sanbancho, Chiyoda-ku, Tokyo, 102–0075 Japan
†† 電気通信大学大学院情報システム学研究科，調布市
Graduate School of Information Systems, The University of Electro-Communications, 1–5–1 Chofugaoka, Chofu-shi, 182–8585 Japan
a) E-mail: jiguo@selicco.com
b) E-mail: morita@is.uec.ac.jp

しかし，Akl らのキー生成・管理法は，システムの規模 (M) が大きいときでも，管理者が一つのキーを管理（保存）するだけでよいが，最も低いセキュリティーレベルのキーを生成するために，$M-1$ 回の暗号化 (g) 計算が必要となる．また，Akl らのキー生成・管理法は上位のセキュリティーレベルキーから下位のセキュリティーレベルキーを生成するため，一般に使い捨て認証・暗号キーシステムなど，特権的な関係をもたないユーザからなるシステムには適用できない．

一方，Goldreich らは，ある条件を満たす擬似乱数生成器 $G : \{0,1\}^k \to \{0,1\}^{2k}$ と k ビットのシード x（$x \in \{0,1\}^k$）に対し，長さ k のインデックス $y = y_1 y_2 \cdots y_k$ から擬似乱数生成する方法 $f_x : \{0,1\}^k \to \{0,\ldots,2^k-1\}$ を提案している [10]．彼らの方法（GGM 法と呼ぶ）では，まず，G を用いて k ビットのシード x から $2k$ ビットの擬似乱数 $G(x)$ を生成する．次に，インデックス y の k ビット 2 進展開 $y_1 \cdots y_\ell \cdots y_k, y_\ell \in \{0,1\}$, $1 \le \ell \le k$ において，$\ell = 1$ とおいて，以下を繰り返す．すなわち，

ステップ 1. $y_\ell = 0$ なら，$2k$ ビットの前半を，$y_\ell = 1$ なら，$2k$ ビットの後半を z とおく．

ステップ 2. 次に，ℓ を 1 増した値が k を超えたら，z を $f_x(y)$ として出力して終了する．そうでなければステップ 3 に進む．

ステップ 3. $G(z)$ を求めて，ステップ 1 に戻る． □

論文 [10] が提案する擬似乱数生成関数は，各インデックス y に対応する k ビットの擬似乱数列を k の多項式時間で生成できることが示されている．すなわち，k 次元インデックス空間に置かれている大量の k ビットの擬似乱数列の中の任意の一つを k の多項式時間で生成できる．したがって，インデックス空間に置かれている擬似乱数列はシード x が与えられた下で，インデックス y によって $f_x(y)$ として管理できる．その結果，使い捨てとなる大量の乱数を保存しないで管理できる．すなわち，管理する秘密情報は一つのシード x だけのキー管理システムの構築が可能である．

GGM 法は，適用する擬似乱数生成器 G の満たすべき条件として，G は一方向性を有すること，CSB 生成器であることの二つを掲げている [10]．ここで，CSB 生成器（cryptographically strong pseudorandom bit generators）とは，①全てのネクストビットテスト（next-bit-tests）に合格し，②予測できないビット列を生成する，ビット生成器である [10], [11]．また，変換関数 f_x においての入出力の間に，必ずしも 1 対 1 ではないことを示唆している（[10], p.800）．これは，同じシード x，異なるインデックス y に対し，同じ出力が得られる可能性があることを意味する．

したがって，GGM 法を用いて，上述のキー管理システムを構築すると，適用するビット生成器 G によって，異なる多数なインデックスから同一のキーが生成される可能性がある．これは異なるユーザに異なるキーを与える必要がある認証システムには不向きである．

一方，応用上，キーの生成には，長いビット列が求められる（例えば 128 ビット以上）が，実際に使われるキーの数はキーの総数（2^{128}）に比べ遥かに少ない（例えば 2^{64} 以下）．

本研究では，GGM 法をキーの生成と管理を統一的に取り扱うキーの管理方法を提案する．提案法では，キーの長さとインデックスのビット数（生成・管理可能なキーの数）を独立させ，GGM 法を一般化する．以下では，この手法を多次元インデックス法と呼ぶ．

インデックスのビット数（生成・管理可能なキーの数）を可変にすることで，インデックスのビット数は管理する必要のキーの数や適用される擬似乱数生成器 G の周期性を考量して決め，異なるユーザに異なるキーを与える必要があるシステムの構築を可能にする．

本論文では，管理する N ビットのキーの総数 M，次元数 D，更に適用する擬似乱数生成器の 2 値系列生成速度 S が与えられたときにおけるキーの生成時間の最小，最大，平均値を導き，そして，適切な m の値と次元数 D の値について議論する．

本論文では，よく知られているハッシュ関数 SHA1 を擬似乱数生成器 G として多次元インデックス法に適用したとき，計算機でキーの実生成時間の最小，最大値とその計算値を示す．また，計算機数値計算により，鍵長 N ビットのときの管理できる無衝突の鍵の総数 M と次元数 D の間の近似的な関係を示す．

本論文の構成は以下のようになる．**2.** では本論文で提案する多次元インデックス法について述べ，**3.** ではハッシュ関数 SHA1 を多次元インデックス法に適用したとき，管理できる鍵の無衝突の空間サイズや生成速度などを評価する．**4.** では提案法を用いた応用例について考察し，最後に，**5.** において結論を述べる．

2. 多次元インデックス法

この節では，GGM 法を一般化し，管理するキーの総数 M，次元数 D と適用する擬似乱数生成器の 2 値

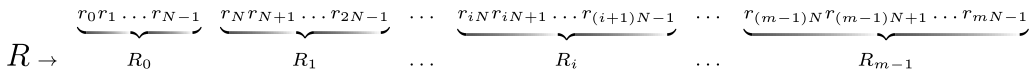

図 1　$G : \{0,1\}^N \to \{0,1\}^{mN}(m : M, r_i : \{0,1\})$
Fig. 1　$G : \{0,1\}^N \to \{0,1\}^{mN}(m : M, r_i : \{0,1\})$.

系列生成速度 S が与えられたときに，多次元インデックス法におけるキーの生成時間の最小，最大，平均値を導く．そして，適切な次元数 D の値について議論する．多次元インデックス法に適用する擬似乱数生成器 G は GGM 法と同様に，一方向性を有した，CSB 生成器とする．

2.1 GGM 法の一般化

2.1.1 GGM　法

自然数 $k \geq 1$ に対し，$I_k = \{0,1\}^k$ とおく．以下では k 桁の 2 進数と長さ k の 0 と 1 からなる 2 値系列を同一視する．任意の CSB 生成器 $G : I_k \to I_{2k}$ を擬似乱数生成器として用い，シード $x \in I_k$ から $2k$ ビットの 2 値系列 $G(x) = b_1^x \cdots b_j^x \cdots b_{2k}^x$ を生成する $(b_j^x \in \{0,1\}, 1 \leq j \leq 2k)$.

次に，二つの内部関数 $G_0, G_1 (G_i : I_k \to I_k, i = 0,1)$ を以下のように定める．まず，$G_0(x)$ を $G(x)$ の最初の k ビット（ここに，$G_0(x) = b_1^x \cdots b_k^x$ である），$G_1(x)$ を $G(x)$ の残りの k ビット（ここに，$G_1(x) = b_{k+1}^x \cdots b_{2k}^x$ である）とおく．そして，$y = y_1 y_2 \cdots y_k \in I_k$ に対して，

$$G_y(x) = G_{y_k}(\cdots(G_{y_2}(G_{y_1}(x)))\cdots)$$

と定める．

このとき，$x, y \in I_k$ に対し，関数 $f_x : I_k \to I_k$ は次のように定まる．

$$f_x(y) = G_y(x)$$

2.1.2 多次元インデックス法

自然数 $N \geq 1, m \geq 2$ に対し，任意の CSB 生成器 $G : I_N \to I_{mN}$ を擬似乱数生成器として用い，m 個の内部関数 $G_i : I_N \to I_N (i = 0, \ldots, m-1)$ を以下のように定める．(安全性証明が通るための N や m の条件については **2.4** で述べる．) まず，任意のシード $R \in I_N$ から mN ビットの 2 値系列 $G(R) = r_0 \cdots r_{mN-1}$ を生成する．

次に，$G_0(R)$ を $G(R)$ の先頭の N ビット（すなわち，$G_0(R) = r_0 \cdots r_{N-1}$），$G_1(R)$ を $G(R)$ の次の N ビット（すなわち，$G_1(R) = r_N \cdots r_{2N-1}$），$\cdots$，$G_{m-1}(R)$ を $G(R)$ の末尾の N ビット（すなわち，$G_{m-1}(R) = r_{(m-1)N} \cdots r_{mN-1}$），一般に，$G_j(R) = r_{jN} \cdots r_{(j+1)N-1}$，$0 \leq j \leq m-1$ とする．更に，インデックス

$i = (i_1, \ldots, i_d, \ldots, i_D)$ を D 次元座標とするとき $(i_d \in \mathcal{M}_d = \{0, \ldots, M_d - 1\},\ 1 \leq d \leq D),\ D \geq 2)$，

$$G_i(R) = G_{i_D}(\cdots(G_{i_d}(\cdots(G_{i_1}(R))\cdots))\cdots)$$

と定める．更に，$\mathcal{M} = \prod_{d=1}^{D} \mathcal{M}_d$ とおくと，

$R \in I_N, i \in \mathcal{M}$ に対し，関数 $f_R : \mathcal{M} \to I_N$ は次のように定まる．

$$f_R(i) = G_i(R)$$

2.1.3 多次元インデックス法と GGM 法

上述の二つの方法を比べてみると，

GGM 法においては

乱数生成器 $G : \{0,1\}^k \to \{0,1\}^{2k}$

シード（seed） $x \in \{0,1\}^k$

インデックス $y = y_1 \ldots y_j \ldots y_k \in I_k$,
$(y_j \in \{0,1\},\ 1 \leq j \leq k)$

に対し，**多次元インデックス法**では，2 以上の自然数，定数 $m, M_1, \ldots, M_D (D \geq 2)$ に対し，

乱数生成器 $G : \{0,1\}^N \to \{0,1\}^{mN}$（図 1 参照）

シード（seed） $R \in \{0,1\}^N$

インデックス $i = (i_1, \ldots, i_d, \ldots, i_D) \in \mathcal{M}$
$(i_d \in \mathcal{M}_d = \{0, \ldots, M_d - 1\}),\ 1 \leq d \leq D)$

と定めている．したがって，$m = 2, D = N, M_1 = \cdots = M_D = 2$ のとき，提案法は GGM 法と等価であることが分かる．

多次元インデックス法を用いて，任意の $i \in \mathcal{M}$ に対し，

インデックスの次元数：D

各次元のサイズ：$M_1, \ldots, M_D$

生成・管理するキーの総数：$M = M_1 \times \cdots \times M_D$

となるキー管理システム $R_i = f_R(i)$ を構築できる．

また，議論を単純にするため，本論文は $m = M_d$ にしているが，G において，$m \geq 2$ であれば，提案法法によるキー管理システムの構築が可能である．

図 2 に $D = 2$ を例にとり，提案法によるキーの生成 $R_i = f_R(i)$ を示す．この場合インデックス i は二

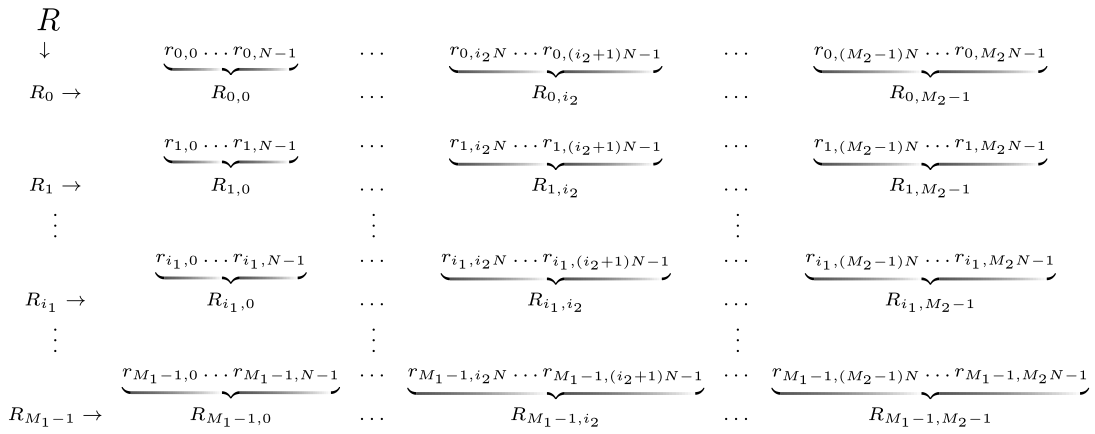

図 2　キーの二次元配置
Fig. 2　Two dimensional arrangement of keyes.

つの部分インデックスの対 (i_1, i_2) として表される．

まず，シード R から生成される M_1 個のシード $R_0, \ldots, R_{M_1-1}$ の中から，R_{i_1} を選ぶ．次に，シード R_{i_1} から生成される M_2 個の擬似乱数 $R_{i_1,0}, \ldots, R_{i_1,M_2-1}$ の中から，R_{i_1,i_2} を選んで出力する．この例では，次元数 D のとき，シードの更新が $D-1$ 回繰り返されることに注意する．

なお，$m = 1$ の場合に，多次元インデックス法によるキー管理システムの構築ができないことは多次元インデックス法の構成から容易に分かる．

2.2　キーの生成時間

次元数 D の多次元インデックス法は位置座標 $i = (i_1, \ldots, i_D)$ に基づき，N ビットのシード R からキー R_i を生成する．したがって，擬似乱数生成器 G の 2 値系列の生成速度を S *bps*（ビット/秒）とすると，任意の N ビットの R_i の生成に必要な時間 T_{R_i} は

$$T_{R_i} = (D + i_1 + \cdots + i_D) \times N/S$$

で計算される．ここに，次元数 D，総数 $M = M_1 \times \cdots \times M_D$ の N ビットのキーの生成に必要の時間の最小値，最大値，平均値をそれぞれ $T^D_{min}, T^D_{max}, T^D_{ave}$ とするときの値を示す．明らかに，最小値は $i_1, \ldots, i_D$ が全て最小値（$0, \ldots, 0$）となるとき，最大値は $i_1, \ldots, i_D$ が全て最大値（$M_1 - 1, \ldots, M_D - 1$）となるとき，平均値は全ての R_i の生成時間の和にその総数 M で割って計算される．したがって，それぞれ以下のようになる．

$$T^D_{min} = ND/S$$

$$\begin{aligned} T^D_{max} &= N(M_1 + \cdots + M_D)/S \\ T^D_{ave} &= \frac{N}{\prod_{j=1}^{D} M_j} \sum_{i_1=0}^{M_1-1} \cdots \sum_{i_D=0}^{M_D-1} (D + i_1 + \cdots \\ &\quad + i_D)/S \\ &= N(D + M_1 + \cdots + M_D)/(2S) \end{aligned}$$

2.3　適切な次元数 D

適切な次元数 D の値は，キーの生成に必要な時間の平均値 T^D_{ave} の最小化により求められる．ここで，最小化問題を容易にするため，各次元の大きさ M_d を全て等しい値にする．したがって，

$$T^D_{ave} = ND(1 + M^{1/D})/(2S)$$

となる．N, S, M が定数で，T^D_{ave} が最小値となるとき，その導関数

$$(ND(1 + M^{1/D})/(2S))' = 0$$

従って，

$$1 + M^{1/D} - M^{1/D} \ln M^{1/D} = 0$$

が得られ，

$$3.59 < M^{1/D} < 3.60$$

が得られる．

各次元の大きさ $M_d = M^{1/D}$ が整数なので，T^D_{ave} は最小値にならないが，$M^{1/D} = 4$ がより最小の T^D_{ave} に近い．以上の考察から，キー生成の高速化という観点からは，GGM 法における $M_1 = \cdots = M_D = 2$ と

いう設定では，最も速い生成速度が得られていないことを示している．

したがって，多次元インデックス法において，管理するキーの総数 $M = M_1 \times \cdots \times M_D$，$M_1 = \cdots = M_D$ のときの適切な次元数 D は，$M^{1/D} = 4$ より

$$D = \log_2 M^{1/2}$$

で計算される．

2.4 安全性について

2.4.1 GGM 法の安全性

文献 [10] では，GGM 法の安全性について，次の二つの内容の主な定理が示されている．

（定理 3）**2.1.1** に示されている GGM 法に対し，$F_k = \{f_x\}_{x \in I_k}$，$F = \{F_k\}$ として，擬似乱数生成器 G が CSB 生成器であれば，F は全ての多項式時間の統計的検査法に合格する．

（定理 4）その上，F は全ての多項式時間の統計的検査法に合格するとき，それに限るとき，F は推測できない．

定理 3 に対し，文献 [10] は，**2.1.1** に示されている $G : \{0,1\}^k \to \{0,1\}^{2k}$ を用いて，シード $x \in \{0,1\}^k$ で，インデェックス $y = y_1 \ldots y_j \ldots y_k$，に従って（$y$ に基づき G の $2k$ ビットの出力から k ビットを選択する）得られた F に対し，F が統計的検査法に合格しない場合に，G も統計的検査法に合格しないことを示した．この結果は G が CSB 生成器である前提と矛盾することにより，定理 3 の成立を証明した．

ここに，系列が統計的テストに合格するとは，k が十分に大きいとき，系列に対するテストの結果（統計量）が乱数で得られる結果（理論値）と十分に近いときとしている．

また，定理 4 に対して，x に対し，ランダムに選んだ $y \in I_k$ において，$f(x)$ が正しく選ばれる（2 択 1 の選び）確率はちょうど 2 分の 1 になることから，逆に正しく選ばれる確率が 2 分の 1 より大きいとき，すなわち，F の推測テストにおいて，F を推測できるとき，F は統計的テストを通過できないことを示したことにより，F は全ての多項式時間の統計的検査法に合格する前提と矛盾することを示し，定理 4 の成立を証明した．

すなわち，GGM 法の安全性は，用いる擬似乱数生成器 G が CSB 生成器で一方向性を有するという条件を満たせば安全である．

多次元インデックス法は GGM 法においての 2 値であるインデックスを多値にした．これはインデックスに基づき，生成器 G から生成した系列の中に乱数列を選ぶ（抽出する）ときの範囲が広がっただけであり，選び出した乱数列の性質が変わらないことがわかる．GGM 法における定理 3，定理 4 は多次元インデックス法において，それぞれ以下の定理 1 と定理 2 となる．

2.4.2 多次元インデックス法の安全性

定理 1

2.1.2 で定まった $f_R(i) = G_i(R)$ において，$F_D = \{f_x\}_{x \in I_N}$，$F = \{F_D\}$ として，G は CSB 生成器であれば，F は全ての多項式統計的テストに合格する．

証明.

GGM 法の定理 3 の証明においての $G : \{0,1\}^k \to \{0,1\}^{2k}$ の k を N に，インデックス $y = y_1 y_2 \cdots y_k$ を $i = (i_1, \ldots, i_d, \ldots, i_D)$ に，インデックスの次元数 k を D に置き換え，CSB 生成器の内部で用いる擬似乱数生成器の内部関数の個数 $m = 2$ を，各 d $(1 \le d \le D, D \ge 2)$ に対し，$m = M_d$ とすると，GGM 法の定理 3 の証明と同様の手順で定理 1 の証明が行える． □

同様に，系列が統計的テストに合格するとは，定数 N が十分に大きいとき，系列に対するテストの結果（統計量）が乱数で得られる結果（理論値）と十分に近いときとする．

定理 2

2.1.2 で定まった $f_R(i) = G_i(R)$ において，$F_D = \{f_x\}_{x \in I_N}$，$F = \{F_D\}$ として，$f_R(i)$ を計算する多項式アルゴリズムが存在するとすると，F が全ての多項式時間の統計的テストに合格するとき，そして，それに限るとき，F は推測できない．

証明.

GGM 法の定理 4 の証明においての $G : \{0,1\}^k \to \{0,1\}^{2k}$ の k を N に，インデックス y を i に置き換え，定数 $m = M_d (1 \le d \le D,\ D \ge 2)$ にすると，x に対し，ランダムに選んだ i において，$f(x)$ が正しく選ばれる（M_d 択 1 の選び）確率はちょうど M_d 分の 1 になり，GGM 法の定理 4 の証明と同様の手順で定理 2 の証明が行える． □

したがって，多次元インデックス法の安全性は，GGM 法と同様で，用いる擬似乱数生成器 G が CSB 生成器で一方向性を有するという条件を満たせば，安全である．

3. 評　　価

この節では，一例として，容易に $G : I_N \to I_{mN}$ を構成できるハッシュ関数 SHA1 を多次元インデックス法に適用するときに，計算機実験による衝突実験の結果と得られるキーの生成速度とを示す．

3.1 SHA1 による衝突実験

ここに，多次元インデックス法が管理する総数 M の N ビットのキー R_i の中に，同じものが存在するとき，衝突という．

多次元インデックス法は SHA1 を用いたとき，N ビットのキーの生成可能なサイズ M の多次元空間にあるキーに対する全件検索を行い，衝突があったかどうかを確認する．

SHA1 の入力は 8 ビット単位で指定できるため，検索はキーの長さ N を $24, 32, 40$ ビットの 3 通りにし，対応する多次元空間サイズを $M = 2^{N/2-2}$ と $M = 2^{N/2}$ とした．それぞれの次元数を $D = 5, 7, 9$ と $D = 6, 8, 10$ で与え，各次元のサイズ M_d を同じ値（$M_1 = \ldots = M_D = 4$）にした．各ケースに 100 回シード R を無作為に選び，100 回検索して衝突があった回数を表 1 に示す．

表 1 に，キー長 $24, 32, 40$ ビット，それぞれの次元数 $D = 5, 7, 9$ のとき，100 回検索の中それぞれ $5, 3, 3$ 回において，同じキー（衝突）が検出されていること，それぞれの次元数を 1 増やした $D = 6, 8, 10$ のとき，それぞれ $32, 34, 38$ 回において，同じキーが検出されている結果が示されている．なお，それぞれの次元数を 1 減らした $D = 4, 6, 8$ のとき，全て同じキーが検出されていないことを確認した．

検索結果に基づき，我々は N ビットときの R_i の総数（空間サイズ）$M = 2^{N/2-4}$ は高い確率で衝突が発生しないことを推測できる．

また，GGM 法においての擬似乱数生成器 G に SHA1 を用いるときの N と衝突についても，同様に計算機による実験を行った．その結果を表 2 に示す．

表 1　多次元インデックス法における N と衝突
Table 1　N and collision in multi-dimensional index method.

$N(bit)$	24		32		40	
多次元の空間サイズ M	2^{10}	2^{12}	2^{14}	2^{16}	2^{18}	2^{20}
次元数 D	5	6	7	8	9	10
各次元サイズ M_d	4	4	4	4	4	4
衝突検出回数	5	32	3	34	3	38

表 1，表 2 の実験結果から，擬似乱数生成器 G に SHA1 を用いるとき，全体的に多次元インデックス法と GGM 法の N と衝突については，ほぼ同等で大きな差がないことが分かる．

3.2 生 成 速 度

多次元インデックス法が用いる擬似乱数生成器の 2 値系列 $r_0 r_1 \ldots$ の生成速度を S bps（ビット/秒），初期値を R，次元数 D，生成・管理する N ビットのキーの総数を M とし，2 進数列を生成する以外の作業時間（例えば，ビット列を切り出す時間など）が十分に小さく，計算しないことにする場合，初期値 R を用いたインデックス $i = (i_1, \ldots, i_D)$ に対応する N ビットのキー R_i の生成に要する時間と実際に計算機を使ったキーの生成に要する時間を示す．

このような近似的に計算された生成時間を推測時間 (t') とし，実際に汎用 PC を用いて計算にかかった時間を実測時間 (t) として，$D = 30$，$M_1 = \cdots = M_d = \cdots = M_D = 4$，2 値系列の生成速度が 136 Mbps のときの最小値 Min，最大値 Max と平均値 Ave の結果を表 3 に示す．また，GGM 法，すなわち $N = 128$ ビット，$D = 60$，$M_1 = \cdots = M_d = \cdots = M_D = 2$ の場合，と $D = 15$，$M_1 = \cdots = M_d = \cdots = M_D = 16$ の場合の結果も表 3 に示す．なお，それぞれの平均値の実測値はそれを測るために膨大な時間が必要で実現困難なため，推測値だけを示している．

表 3 から，キーの長さ $N = 128$ ビット，2 値系列の生成速度 136 Mbps，次元数 $D = 30$，各次元のサイズ $M_1 = \cdots = M_d = \cdots = M_D = 4$ のとき，キーの総数 $M = 2^{60}$ の中の一つの生成に，最も速いのは

表 2　GGM 法における N と衝突
Table 2　N and collision in GGM's method.

計算精度 $N(bit)$	24		32		40	
多次元空間サイズ M	2^{10}	2^{12}	2^{14}	2^{16}	2^{18}	2^{20}
次元数 D	10	12	14	16	18	20
各次元サイズ M_d	2	2	2	2	2	2
衝突検出回数	1	34	6	40	1	33

表 3　キーの生成時間の比較
Table 3　Comparison of generation time of key where i is a hexadecimal number. (Unit: μsec)

	GGM 法		多次元インデックス法			
	(D, M_d)=(60, 2)		(D, M_d)=(30, 4)		(D, M_d)=(15, 16)	
i	t'	t	t'	t	t'	t
Min	56.6	65.6	28.3	31.2	14.1	21.8
Max	113	123	113	125	226	234
Ave	84.8		70.6		120	

Intel Pentium M® processor 1.60 GHz

31.2 μsec，最も遅いのは 125 μsec，推測値と近い結果となっていることを確認できる．

また，生成時間の平均値の推測値を見ると，$D = 60, 30, 15$ のときの生成時間の平均値の比は 12 : 10 : 17 で，提案法は最も小さい値となっている．

以上の結果から提案法は生成速度において，GGM 法よりも速いことが分かる．

4. 提案法を用いたキー管理法の例

ロックを生産するあるメーカは，一部の安全性の高い機械式鍵に対し，鍵の製造データを厳密に管理している．当該メーカ以外では同じ鍵を作ることが極めて困難であるので，安全である．

4.1 従 来 法

上述の機械式鍵の管理方法から，メーカ側が安全性を確保するため，キー情報の一元管理が必要と認識していることが分かる．しかし，同じメーカのリモコン電子キーの場合，認証コードを一元化管理していない．全ての認証コードをユーザシステムに書き込んで，つまり，ユーザ自身の責任で管理されている．

すなわち，システムの操作方法を知って，システムに近づけられる者であれば，同じ性能をもつリモコンで，システムから認証コードをリモコンに書き込むことにより，リモコンキーを作ることができる．機械式鍵に比べ安全性に問題が残る．

4.2 リモコンロック認証コードの管理の特殊性

一部のリモコンキーシステムでは，安全性を確保するため，リモコンキーの認証コードは使い捨てとなっている．異なるリモコンキーの認証コードは異なる乱数系列を使う．また，リモコンをなくしたとき，異なる認証コードのリモコンを発行する必要がある．

一方，住宅や車などは引越しや転売により，使用者が変わるという状況が発生する．

そのため，ロックシステムに対し，運用する期間，必要とされるリモコンの数などは一定ではない．

メーカによる一元化管理であれば，システムが廃棄されたという情報がメーカに伝わらない限り，出荷した全てのシステムの認証コードを管理する必要がある．**1.** に述べた従来の記憶媒体に保存して管理する方法では，安全性を確保しながら，全ての可能性に対応できるためには，多大なメモリと管理コストが必要となる．

4.3 提 案 法

多次元インデックス法において，キー R_i を生成するための次元座標情報 i は管理対象を特定できる管理情報から抽出（変換）できる．次元座標情報 i は意味をもたない規則ある数値（例えば，$0, 1, 2, \ldots$）だけでもよい．以下にその一例を述べる．

①ユーザキー管理システム

全てのユーザシステムのキー R_u を管理するメーカ側の管理キーを R とし，ユーザを特定できる番号（例えば $MW002014010100001$）を u とすると，R と u によるキー R_u を管理するシステムである．

②リモコンキー管理システム

ロックシステムに，それぞれのリモコンキーに異なる番号 j（例えば $0, 1, \ldots$）を与え，ユーザシステムのキー R_u と j によるリモコンキー R_j を管理するシステムである．

③使い捨て認証コード管理システム

リモコンキーの使い捨て認証コードに異なる番号 i（例えば $0, 1, \ldots$）を与え，リモコンキー R_j と i による使い捨て認証コード R_i を管理するシステムである．

認証

リモコンキーは j, R_j, i を管理し，未使用の i と R_j を使って，③で R_i を生成して，j, R_i, i をロックシステムに送り，使い捨て認証を行う．

ロックシステムは R_u と各リモコンの使い済み i を管理し，リモコンから送られた j, R_i, i に対し，リモコン j の i が未使用であることを確認し，②で R_u と j を使って R_j を生成し，そして i と R_j を使って，③で R_i を生成し，リモコンから送られた認証コードと比較して使い捨て認証を行う．

キーの管理

メーカは R とユーザ番号及び該当ロックシステムに発行済みのリモコン情報 j を管理すれば，ロックシステムの復元や新しいリモコンキーの発行ができる．秘密に管理する必要な情報は R だけである．

5. む す び

GGM 法をキーの生成と管理を統一的に取り扱うキーの管理方法として，キーの長さとインデックスのビット数（生成・管理可能なキーの数）を独立させ，一般化し，シードの繰り返し生成（更新）によるキーを生成する多次元インデックス法を提案した．

管理するキーの総数 M，次元数 D と適用する乱数生成器の 2 値系列生成速度 S が与えられたときにおけるキーの生成時間の最小，最大，平均値を導き，そして，適切な次元数 D の値を $D = \log_2 M^{1/2}$ と与えた．

また，多次元インデックス法に SHA1（2 値系列の

生成速度が 136 Mbps）を適用したとき，生成・管理する $N = 128$ ビットのキーの総数が 2^{60}，次元数 30 のときの一本のキーの生成時間は，最長 125 μsec，最短 31.2 μsec 程の結果であった．そして，キーの平均生成速度は GGM 法より速い．

そして，キーの長さ N を $24, 32, 40$ ビット，次元座標の空間サイズ $M = 2^{N/2}$ と $M = 2^{N/2-2}$ などにして生成したキーの間の衝突について検索した結果，$M = 2^{N/2-2}$ のとき，約 0.95 の割合で衝突が検索されなかった．更に多次元空間サイズを一次元減らしたとき（$M = 2^{N/2-4}$），同じキー（衝突）は検出されなかった．GGM 法の場合と同等な結果であったことを実験で確認した．

したがって，セキュリティ分野では，これまで難しいとされてきた，盗聴の防止に有効な使い捨て認証システム，解読が困難な鍵の使い捨て暗号システム，不正にコピーした複製品の検出に有効な推測できない識別番号を製品ごとに付与・認証するシステムの構築など幅広い応用が期待できる．

謝辞　本論文は，これまでの数回の投稿で，査読委員から頂いた貴重な指摘と助言によって完成したと言っても過言ではない．ここに，匿名の査読委員の方々に感謝の意を心から申し上げたい．

文　献

[1] S. Burnett and S. Paine, RSA セキュリティオフィシャルガイド　暗号化，翔泳社，東京，2002.
[2] B. Schneier，暗号技術大全，ソフトバンクパブリッシング株式会社，東京，2004.
[3] C.K. Wong, M. Gouda, and S.S. Lam, “Secure group communications using key graphs,” IEEE/ACM Trans. Netw., vol.8, no.1, pp.16–30, Feb. 2000.
[4] D. Naor, M. Naor, and J. Lotspiech, “Revocation and tracing schemes for stateless receivers, in advances in cryptology,” Crypto 2001, LNCS 2139, pp.41–62, 2001.
[5] T. Asano, “A revocation scheme with minimal storage at receivers,” Proc. ASIA CRYRT 2002, LNCS 2501, pp.433–450, 2002.
[6] 菊池浩明，中村雄一，“セキュアなコンテンツ配信における SD 法に基づいた効率のよい鍵管理方式,” 情処学論，vol.47, no.2, pp.426–433, Feb. 2006.
[7] S.G. Akl and P.D. Taylor, “Cryptographic solution to a problem of access control in a hierarchy,” ACM Trans. Comput. Syst., vol.1, no.3, pp.239–248, 1983.
[8] G.C. Chick and S.E. Tavares, “Flexible access control with master keys,” Crypto1989, LNCS 435, pp.316–322, 1990.
[9] W-G Tzeng, “A time-bound cryptographic key assignment scheme for access control in a hierarchy,” IEEE Trans. Knowl. Data Eng., vol.14, no.1, pp.182–188, Jan./Feb. 2002.
[10] O. Goldreich, S. Goldwasser, and S. Micali, “How to construct random functions,” J. ACM, vol.33, no.4, pp.792–807, 1986
[11] M. Blum and S. Micali, “How to generate cryptographically strong sequences of pseudo-random bits,” SIAM J. Comput., vol.13, pp.850–864, Nov. 1984.

（平成 26 年 2 月 26 日受付，6 月 20 日再受付）

董　際国　（正員）

1984 上海科学技術大学精密機械工学卒．1997 山形大学人文学部経済学科卒．1999 同大学院社会文化システム研究科修士課程修了．2012 電気通信大学情報システム研究科博士後期修了．カオスの応用に関する研究に従事．工博．電子情報通信学会会員．

森田　啓義　（正員：シニア会員）

1983 阪大大学院博士後期課程了．同年豊橋技術科学大学助手．1990 電気通信大学電気通信学部講師を経て，現在同大大学院情報システム学研究科教授．3D 画像処理，MPEG 応用，データ圧縮，誤り訂正など，情報理論とその応用に関する研究に従事．工博．情報処理学会，情報理論とその応用学会，IEEE, ACM 各会員．