SCIS 2020 2020 Symposium on
Cryptography and Information Security
Kochi, Japan, Jan. 28 – 31, 2020
The Institute of Electronics,
Information and Communication Engineers

ILMAPM を MDIM に適用した場合のキーの衝突：IoT に有用なキーの生成と管理方法
Collision of Keys when ILMAPM is Applied to MDIM: Key Generation and Manegement Method for IoT

董 際国 * 森田 啓義 †
DONG Jiguo Hiroyoshi MORITA

あらまし IoT の普及にはセキュリティー大量のキーの管理方法である「多次元インデックス法」（MDIM）に「整数ロジスティック写像法」（ILMAPM）を適用したときに生成したキーの衝突実験の結果について報告する．ILMAPM に対しては，擬似乱数生成法の内部状態のビット数を n，入出力ビット長を N とすると，$n = N$ が成立つので IoT 機器への組込みに適している．実験では，代表的な擬似乱数生成法として広く利用されている SHA1（$n = 160, N = 44$）を比較対象とし，それぞれをた．MDIM に適用して生成したキーの系列に対する衝突実験を行った．その結果，ILMAPM が生成したキー系列は SHA1 よりよい統計的乱数性を有することが分かった．$n = N$ のとき，生成するキーの数が擬似乱数生成器が有する最短周期よりも少ない場合に，決定論的な無衝突を確保することができる．

キーワード IoT，ワンタイムパッド，鍵管理，擬似ランダム的衝突，決定論的衝突

1 前書き

IoT 環境においては、様々な計算能力を有する機器がネットに繋がり、情報交換をしている．その中、計算能力の低い RFID やネットノードが多く含まれており、そのセキュリティが難問である [1-3]．これまでの AES のような暗号では、RFID タグやセンサーネットワークなどの制約の厳しい環境には適さない [8]．そのため，軽量な認証プロトコール [1] や軽量暗号 [4-18] が提案されている．また、軽量暗号に対する攻撃テスト [19-24] や評価も行われている [25-31].

軽量な認証プロトコルは適切な頻度でキー再生成をすることで，なりすましの防止に有効ではあるが、ワンタイムパッドには既知のキー管理の問題がある [1].

軽量暗号は計算量を減らすため，S-Box やブロックのビット長，キーの長さを短くするなどして設計されていることが共通している．例えば，Piccolo[12] は，4 ビット S-Box，64 ビットブロック，80 ビットと 128 ビットの鍵，ラウンド数はそれぞれ 25 と 31 である．ANU[14] は，4 ビット S-Box，64 ビットブロック，80 ビットと 128 ビットの鍵，ラウンド数は 25 である．Loong[17] は，4 ビット S-Box，64 ビットブロック，64，80，128 ビットの鍵，それぞれ対応するラウンド数は 16、20、32 である．このように，より性能の低いデバイスに対し，より短い鍵とランド数で対応している．軽量暗号の性能とセキュリティの間にトレンドオフが存在している [27].

また，各種の軽量暗号はそれぞれ速度，占有メモリ，消費電力などの性能指標において，強みを持ち，ソフトウェア指向或はハードウェア指向も存在している．そのため，それぞれのデバイスに最も適切な軽量暗号が選択されたとき，異なるデバイスと通信を行う場合は複数の軽量暗号を持ち合わせる必要がある [27]．このことは明らかに，非効率的で，制約の厳しい環境に適さない．

従って，異なるデバイスに対し，異なるアルゴリズムで最適化できる同一の暗号或は擬似乱数生成法と，ワンタイムパッドに対応可能なキーの管理方法が求められる．

我々は「整数ロジスティック写像法」（ILMAPM）[32] と「多次元インデックス法」（MDIM）[33] を提案した．

ILMAPM は整数の加算，乗算，ビットの論理演算だけで，良い統計的な乱数性を持つ擬似乱数を高速に生成できる．

MDIM はキーの生成と管理を統一的に取り扱うキーの

* セリック株式会社，〒 102-0075 東京都千代田区三番町 7–5–705, selic corporation, 705, 7–5, Sanbancho, Chiyoda-ku, Tokyo 102–0075 Japan (jiguo@selicco.com)
† 電気通信大学大学院情報システム学研究科，調布市，東京都, Graduate School of Information Systems, The University of Electro-Communications, Chofu-ga-oka 1–5–1, Chofu-shi, Tokyo 182–8585 Japan (morita@is.uec.ac.jp)

C		32	64	96	128	160	192	224	256
S	PC(Gbps)	1.9	1.0	.71	.56	.51	.43	.36	.31
	MC(Kbps)	262	166	125	101	80	67	60	54

表 1: 計算精度（C ビット）と乱数生成速度（S）

管理方法として，多次元インデックスに基づくシードの繰り返し生成（更新）によるキーを生成する．したがって，インデックスを管理することにより，擬似ランダムのキーを生成・管理できる．

[33] において，SHA-1 を MDIM に適用して生成するキーの衝突実験を行ったが，MD5 や SHA-1 などの標準的な暗号化ハッシュ関数を実装することは、今日のタグの機能を超えている [1] ため，本論文は ILMAPM を擬似乱数生成器として MDIM に適用して生成するキーの衝突実験を行い，SHA-1 との比較を行う．ここに，生成された擬似乱数列の中に同じものがあったとき，それを衝突という．

本論文は以下のようになる．第 2 節は高速に良質な擬似乱数を生成できる ILMAPM の主な内容と MDIM の概要をまとめる．第 3 節は，MDIM を用いて生成したキーの衝突について，第 4 節は計算機によるキーの衝突実験とその結果を報告し，第 5 節はまとめである．

2 ILMAPM と MDIM

2.1 ILMAPM

2.1.1 整数ロジスティック写像（ILMAP）

ここに，N は計算精度のビット数，$\lfloor\ \rfloor$ は小数点以下を捨てる．

$$X_{t+1} = \lfloor 4X_t(2^N - X_t)/2^N \rfloor \quad (0 < X_t < 2^N,\ t = 0, 1, \dots) \tag{1}$$

2.1.2 擬似乱数の生成——撹拌

R_t の系列を生成した擬似乱数系列とすると、

$$R_t = \lfloor 4X_t(2^N - X_t)/2^N \rfloor XOR((4X_t(2^N - X_t)) \bmod 2^N) \quad (0 < X_t < 2^N,\ t = 0, 1, \dots) \tag{2}$$

2.1.3 擬似乱数の生成速度

計算速度は，プロセッサーの種類とプログラミング方法やコンパイラの種類によるが，ここにノート PC と 8 ビットマイコンでの計算例は表 1 にある．

2.1.4 ILMAP の周期性

計算機数値実験により、平均的な非周期状態長は

$$L_c \approx 5 \times 2^{N/2-3}$$

で与えられる [34].

2.2 MDIM

2.2.1 MDIM について

MDIM に適用する擬似乱数生成器 G は，一方向性を有した，CSB 生成器とする．

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 $G_i : I_N \to I_N (i = 0, \dots, m-1)$ を以下のように定める．

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．

さらに，インデックス $i = (i_1, \dots, i_d, \dots, i_D)$ を D 次元座標とするとき（$i_d \in \mathcal{M}_d = \{0, \dots, M_d - 1\},\ 1 \leq d \leq D),\ D \geq 2$），

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

と定める．さらに，$\mathcal{M} = \prod_{d=1}^{D} \mathcal{M}_d$ 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．

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

MDIM 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：$M_1, \dots, M_D$

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．

2.2.2 キーの生成時間

次元数 D の MDIM は位置座標 $i = (i_1, \dots, i_D)$ に基づき，N ビットのシード R からキー R_i を生成する．従って，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．従って，それぞれ以下のようになる．

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

2.2.3 適切な次元数 D

MDIM 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．

2.3 MDIM によるキー管理システムの一例

ILMAPM は，ハードウェア化した乗算器を利用することで，PC 上もマイコンの上も同じ擬似乱数を高速に生成される．FPGA 上では，大量の乗算器（$9\times9, 18\times18$）を備えているため，並列処理により，より高速にキーの生成ができる．ASIC に実装すれば更に高速化が可能である．異なるデバイスで，それぞれ利用可能な乗算器などに応じて，異なるアルゴリズムで同じ擬似乱数を生成可能である．

以下，リモコンキーシステムを一例に，階層化したキー管理システムについて説明する．

端末の使い捨て認証キー管理

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．

複数端末を含む製品のキー管理

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，ユーザーシステムのキー 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．

複数同製品のキー管理

同じ製品のユーザーシステムのキー R_u を管理するキーを R_p とし，ユーザーを特定できる番号を u とすると，R_p と u によるキー R_u を管理するシステムである．

異なる製品のキー管理

製品のキー R_p を管理するメーカ側の管理キーを R_m とし製品を特定できる番号を p とすると，R_m と p によるキー R_p を管理するシステムである．

複数メーカーのキー管理

メーカのキー R_m を管理するキーを R としメーカを特定できる番号を m とすると，R と m によるキー R_m 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，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，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，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_m と製品情報 p とユーザー番号 u 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_m だけである．

また，第三者機間によるキーの管理とサポートサービスを提供することで，ばらばらのメーカごとのキー管理よりもコストの面でもセキュリティー面でも有利である．特に中小企業の場合である．メーカの存続に依存しないキーのサポートも可能となる．そのとき，上記メーカが管理する情報に，メーカ情報 m を追加して管理するだけでよい．秘密に管理する必要な情報は R だけである．

MDIM は規則あるインディックスを用いて，大量のキーの非保存管理を可能にした．また，規則あるインディックスを利用することで，使用済み或いは飛ばされたインディックスによる認証が求められたときに，不正アクセス（攻撃）の可能性があり、それを早期察知することに役立つ．

また，使い捨て暗号キーの管理も同様にできる．

そして，認証や暗号だけでなく，生産物に認証キーの付与・管理も同じように行え，一物一キーによりモノの認証即ち偽生産物の検出（防止）にも役立つ．

3 MDIM を用いて生成したキーの衝突

3.1 決定論的系列と擬似ランダム系列

3.1.1 決定論的系列

本論文は，計算機を用いて関数を計算して，生成した系列において，計算機の内部状態（計算精度）のビット長 n と，出力（観測）値のビット長 N の間に，$N = n$ が成立つとき，その系列を決定論的系列という．明らかに，そのとき，系列の中の任意の N ビットの値の次の N ビットも前の N ビットも決まっている．

j	$p_j = q_{j-1} \times (j-1)/2^N$	$q_j = 1 - p_j, q_0 = 1$
1	0	1
2	$1/2^N$	$1 - 1/2^N$
3	$2/2^N - 2/2^{2N}$	$1 - (2/2^N - 2/2^{2N})$
4	$3/2^N - 2 \times 3/2^{2N} + 2 \times 3/2^{3N}$	$1 - 3/2^N + 2 \times 3/2^{2N} - 2 \times 3/2^{3N}$
5	$4/2^N - 3 \times 4/2^{2N} + 2 \times 3 \times 4/2^{3N} - 2 \times 3 \times 4/2^{4N}$	$1 - 4/2^N + 3 \times 4/2^{2N} - 2 \times 3 \times 4/2^{3N} + 2 \times 3 \times 4/2^{4N}$
⋮	⋮	⋮

表 2: 衝突発生確率の遷移

3.1.2 擬似ランダム系列

上記決定論的系列において，内部状態のビット数 n と出力（観測）のビット数 N が，$n > N$ である場合を見てみよう．

$n = N + 1$ の場合，すなわち，内部状態が n ビットで生成した系列に対し，$N = n - 1$ で観測すると，ある N ビットの値の後に 2 通り値を観測される可能性があり，その値の前にも 2 通りがある．$n = N + 2$ の場合，ある N ビットの値の後に 4 通りの値を観測される可能性があり，その値の前にも 4 通りがある．

したがって，$n \geq 2N$ 場合，その系列において，ある N ビットの数値の次の N ビットの数値は 2^N 通りの可能性がある．実際に 2^N 通りの値がすべて含まれるかどうかは，用いられる擬似乱数生成法（関数）によるが，$n >> N$ の場合，その可能性はより高い．著名な擬似乱数生成法 MT[35] が内部状態 19997 ビット，入出力 32 で，その典型的な例である．

本論文は，$n >> N$ の場合のキー系列を擬似ランダムなキー系列という．

3.2 衝突確率

N ビットのキーがランダムに生成されるとき，j 番目のキーで最初の衝突が起きる確率 p_j と一度も衝突が発生していない確率 $q_j = 1 - p_j$ の推移は表 2 となる（$q_0 = 1$）．

表 2 にあるように，j 個のキーが生成されたとき，衝突が発生した確率 P_j は以下のようになる

$$\begin{aligned} P_j &= p_1 + p_2 + \cdots + p_j \\ &= 0 + 1/2^N + (2/2^N - 2/2^{2N}) + (3/2^N \\ &\quad -2 \times 3/2^{2N} + 2 \times 3/2^{3N}) + \cdots \\ &= (1/2^N + \cdots + (j-1)/2^N) - (1 \times 2/2^{2N} + \cdots \\ &\quad +(j-2)(j-1)/2^{2N}) + (1 \times 2 \times 3/2^{3N} + \cdots \\ &\quad +(j-3)(j-2)(j-1)/2^{3N}) - \cdots \end{aligned}$$

上の式を整理すると，式 (3) となる．式 (3) において，j が 2^N より十分に小さいとき（$j << 2^N$），P_j の近似値 $P_j^\star$ は式 (4) で表せる．

$$\begin{aligned} P_j = &\sum_{k=1}^{j-1} \frac{k}{2^N} - \sum_{k=1}^{j-2} \frac{k(k+1)}{2^{2N}} \\ &+ \sum_{k=1}^{j-3} \frac{k(k+1)(k+2)}{2^{3N}} - \cdots \end{aligned} \tag{3}$$

$$P_j^\star = \frac{1}{2} \frac{(j-1)j}{2^N} \tag{4}$$

3.3 キーの長さと衝突の検出

キーがランダムなキー系列から得られるとき，キーの衝突の発生は確率的に生じる．一方，キーが決定論的なキー系列から得られるとき，キーの衝突の発生は，系列の周期性に決定される．すなわち，生成したキーの数 (j) が系列の非周期状態長を超えた場合，確実に衝突が検出される．逆に，j が系列の非周期状態長を超えない場合，衝突が検出されない．

3.3.1 キーの長さ N の決定

式 (3) から，ある j に対し，P_j は N の増大に指数的に小さくなることがみてとれる．一方，j の増大は衝突検出に要する時間が指数的に増大させる．衝突の検出結果を統計的有意にするには，ある j と N で得られる P_j に対し，適切な実験回数 T も必要である．したがって，計算機を用いて，生成したキーの衝突の有無の検出は，検出可能のキーの長さ N が計算機の性能に制限される．我々は，経験的に，$N = 44$ ビットが妥当と分かった．

3.3.2 $n >> N$ の場合

$n >> N$ の場合，即ち，キーが擬似ランダムなキー系列から得られるとき，キー衝突の発生は式 (3) に従う．非線形擬似ランダム系列生成法 SHA1 はよく知られている．内部状態長 $n = 160$ ビットである SHA1 を MDIM に適用し，生成したキーの衝突実験を次節で行う．

3.3.3 $n = N$ の場合

$n = N$ の場合，すなわち，キーが決定論的なキー系列から得られるとき，キーの衝突が検出されるかどうかは，衝突の検出に使われるキーの数と系列の非周期状態長に

よる．次節は内部状態長を任意に設定できる ILMAPM を MDIM に適用して，生成したキーの衝突実験を行う．

4 衝突実験

本節は SHA1 と ILMAPM を MDIM に適用して，生成したキーに対する衝突実験を行う．

4.1 実験目的

SHA1 から擬似ランダム的な衝突，ILMAPM から決定論的な衝突を計算機実験で確認する．

4.2 実験方法

ここに，MDIM 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．

検索はキーの長さ N を 44 ビットにし，対応する多次元空間サイズを $M = 2^{22}, 2^{20}, \ldots, 2^{10}$ とした．それぞれの次元数を $D = 11, 10, \ldots, 5$ 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．

ランダムなキー系列において，式 (4) から，ある N において，生成したキーの数により，衝突の発生する確率が異なることが分かる．実験結果を有意にするため，各 M に対し，理論的な衝突の回数を 50 となるように，実験回数 T を設定した．

各ケースにシード R を無作為に選び，取り得るすべての多次元座標 i に対応する R_i を生成し，R_i 間に衝突の有無を検索する．なお検索は R_i 間に衝突が検出された時点で終了する．

表 3 にあるように，$D = 5$ のとき，$T = 100 \times 2^{24}$ となっている．ILMAPM を MDIM に適用した場合，T の値は取り得るシード R の総数の万分の一以下になっている．ランダムにシードを生成する場合，同じものをとる確率が万分の一以下ということは，同じシードを取らない要求にとって，十分に小さい値とはいえない．したがって，本実験において，D をさらに小さくすることは有意ではないと考えられる．

4.2.1 SHA1 を MDIM に適用した場合

MDIM に SHA1 を用いたとき，$N = 44$ 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．しかし，$N = 44$ ビットとした場合，8 の整数倍ではない。本実験では，SHA1 の入力を 48 ビットとし，上位 4 ビットを 0 にする。

4.2.2 ILMAPM を MDIM に適用した場合

MDIM に ILMAP を用いたとき，$N = 44$ 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，衝突の有無を確認する．

4.3 実験結果

MDIM に SHA1 及び ILMAP を用いたとき，$N = 44$ ビットでの各 M に対応した実験回数 T で，衝突が検出された回数を表 3 に示す．表 3 にある T 欄の値に対し 100 を乗じた値が実験回数になる．

M		2^{22}	2^{20}	2^{18}	2^{16}	2^{14}	2^{12}	2^{10}
D		11	10	9	8	7	6	5
M_d		4	4	4	4	4	4	4
$T(\times 100)$		1	2^4	2^8	2^{12}	2^{16}	2^{20}	2^{24}
C	SHA1	46	55	51	48	74	0	0
	ILMAP	94	357	384	459	442	447	538

表 3: MDIM における衝突

4.3.1 SHA1 の実験結果

表 3 から，$D = 11, \ldots, 8$ において，衝突実験で衝突が検出された回数は，50 近辺であることをみてとれる．このことは，SHA1 で生成された系列が確率論的な振る舞いが観察されているとのことである．

一方 $D = 7$ では、期待値 50 から大きく離れている．又、$D = 6, 5$ はともに衝突が検出されなかった．これはランダムでない振る舞いを意味する [36].

4.3.2 ILMAPM の実験結果

表 3 から，$D = 11$ の実験において，100 回のうち 94 回衝突が検出された．$D = 11$ のとき，実験に使われるキーの数 M は ILMAP の平均的な非周期状態長（節 2.2.4）を超えていることが分かる．MDIM の周期性は ILMAP の周期性と同一のものではないが，ILMAP の周期に支配されていることが察される．

$D = 10, \ldots, 5$ において，期待値 50 に対し，衝突が検出された回数は 357,384, 459,442,447,538 であった．衝突が検出される回数が緩やかな増加傾向も見えるが，実験回数の 16 倍ごとの増加からみれば，ほぼ安定しているといえる．すなわち，生成するキーの総数 M が 1/4 に減少すると，短い周期に入る回数が約 1/16 に減少した．

4.3.3 周期性と無衝突

SHA1 の場合、衝突が検出されたときに，系列が周期に入ったかどうかは本論文の目的ではないので、確認をしなかった．一方，ILMAP の場合，$n = N$ なので，系列が周期に入ったことは明らかである．

生成したキーの数が系列が存在する最短周期長よりも少ない場合，周期は検出されないが，本実験では計算機の計算能力の制限で，キーのビット数を 44 ビットとしたため，それを実験で示すことができなかった．

本実験では $D = 5$ のとき，実験回数（使用した初期値の数）が 100×2^{24} で 2^{31} に近い，関数の対称性も考慮

すると，使った初期値が全初期値の約 2^{12} 分の一になっている．この状況では，無作為で選ばれる初期値が同一のものになる確率が低いとはいえない．このことを踏まえ，$D < 5$ の場合の衝突実験を行わなかった．

4.4 追加衝突実験

SHA1 を MDIM に適用したときの衝突実験結果から SHA1 を擬似乱数生成器として生成した系列のランダム性を確認するため，追加実験を行う．

4.4.1 SHA1 に関する追加衝突実験

本追加実験は MDIM に適用せずに，SHA1 の入力を任意の 160 ビットとし，SHA1 を用いて生成した 160 ビットの系列を繋ぎ，そこから，M 本の 44 ビットのキーを切り出し，その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．

表 4 から，$M = 2^{22}$ と $M = 2^{10}$ 以外において，衝突実験で衝突が検出された回数は，50 近辺であることをみてとれる．このことは，SHA1 で生成された系列が確率論的な振る舞いが観察されているとのことである．

一方 $M = 2^{22}$ と $M = 2^{10}$ では、期待値 50 から大きく離れている．この結果は偶然であるかどうかを確認するため，初期値を変えてそれぞれ 4 回の追加実験を行った結果，$M = 2^{22}$ の場合，それぞれの衝突回数は $38, 34, 43, 47$ であった．この結果から，$M = 2^{22}$ の場合の衝突回数は期待値より少ない傾向があることを確認できる．また，$M = 2^{10}$ の場合，それぞれの衝突回数は $55, 53, 57, 59$ であった．この結果から，$M = 2^{10}$ の場合の衝突回数は期待値より多い傾向があることを確認できる．この結果は，SHA1 で生成された系列がランダムでない振る舞い [36] が観察されているとのことである．

M	2^{22}	2^{20}	2^{18}	2^{16}	2^{14}	2^{12}	2^{10}
$T(\times 100)$	1	2^4	2^8	2^{12}	2^{16}	2^{20}	2^{24}
C	36	48	49	48	45	51	60

表 4: SHA1 における衝突

4.4.2 ILMAPM に関する追加衝突実験

この実験は，SHA1 と比較するため，計算精度（内部状態）を 160 ビットにして，ILMAPM への初期入力を任意の 160 ビットとし，ILMAPM を用いて生成した 160 ビットの系列を繋ぎ，そこから，M 本の 44 ビットのキーを切り出し，その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．

表 5 から，$M = 2^{18}$ だけ、33 で期待値 50 から大きく離れている．この結果は偶然であるかどうかを確認するため，ILMAPM への初期入力を変えて 4 回の追加実験を行った．その結果，それぞれの衝突回数は $57, 53, 47, 44$ であった．この追加実験の結果から，$M = 2^{18}$ の場合の最初の衝突回数は期待値より少ないことは偶然であることを確認できる．

計算精度を 160 ビットの ILMAPM への実験から，SHA1 で生成された系列から見られたランダム的でない振る舞いは観察されなかった．このことは ILMAPM が SHA1 よりも良質な擬似乱数系列を生成していることを意味する．

M	2^{22}	2^{20}	2^{18}	2^{16}	2^{14}	2^{12}	2^{10}
$T(\times 100)$	1	2^4	2^8	2^{12}	2^{16}	2^{20}	2^{24}
C	44	52	33	50	48	49	48

表 5: ILMAPM における衝突

5 終わりに

我々は，本論文において，擬似乱数生成法の内部状態のビット長を n，入出力ビット長を N としたとき，生成された系列に対し，$n >> N$ のときの系列を擬似ランダム的な系列とし，$n = N$ のときの系列を決定論的な系列とした．

その上，異なる性能を有する計算機の上で擬似乱数生成できる ILMAPM を大量のキーの一元化管理を可能にする MDIM に適用して生成したキーに対し衝突実験を行った．その結果，

ILMAPM（$n = N = 44$）の場合，$M = 2^{22}$ で 100 回実験の内，94 回において衝突が検出された．これは実験に使われるキーの数 M は ILMAP の平均的な非周期状態長を超えていること意味する．MDIM の周期性は ILMAP の周期性と同一のものではないが，ILMAP の周期に支配されていることが察される．

$D = 10, \ldots, 5$ において，衝突が検出される回数が緩やかな増加傾向であるが，実験回数が 16 倍ごとの増加からは，ほぼ安定しているといえる．すなわち，生成するキーの総数 M が 1/4 に減少すると，短い周期に入る回数が約 1/16 に減少することが分かった．

SHA1（$n = 160, N = 44$）の場合その結果，SHA1 で生成された系列は，$D = 11, \ldots, 8$ において，ランダム的な振る舞いが観察された．又，$D = 7$ では、期待値から大きく離れて，$D = 6, 5$ はともに衝突が検出されなかった．これは非ランダム（作為）的な振る舞いを意味する．

上記 SHA1 の結果を踏まえ，追加実験をした結果，SHA1 を擬似乱数生成器として生成した系列がランダム的でない振る舞いを観察された．ILMAPM に対し $n = N = 160$ とし，SHA1 と同様な実験を行った結果，ILMAPM が SHA1 よりも良質な擬似乱数系列を生成していることを確認できた．

ILMAPM はマイコンのような計算性能の低いものでも，乗算器が装備されていれば，高速に擬似乱数生成が

可能で，また，FPGA のように多数の小さい乗算器が装備されているとき，より高速に擬似乱数（キー）を生成できる．

ILMAPM を MDIM に適用したとき，計算精度を高くして，生成するキーの数がその最短周期よりも少ない場合に，決定論的な無衝突を確保することができる．

我々は提案した ILMAPM と MDIM が IoT 環境の秩序の樹立に役立つと寄与する．

参考文献

[1] Vajda I, Buttyan L. Lightweight Authentication Protocols for Low-Cost RFID Tags, 2nd Workshop on Security in Ubiquitous Computing, in conjunction with Ubicomp 2003.

[2] Y Xiao, X Shen, B Sun, L Cai, Security and privacy in RFID and applications in telemedicine. IEEE Commun. Mag. 44, 64-72 (2006)

[3] Y Xiao, S Yu, K Wu, Q Ni, C Janecek, J Nordstad, Radio frequency identification: technologies, applications, and research issues. Wireless Commun. Mobile Comput. 7, 457-472 (2007)

[4] D Hong, J Sung, S Hong, J Lim, S Lee, B Koo, H Kim, in Proceedings of the 8th International Workshop on Cryptographic Hardware and Embedded Systems. HIGHT: a new block cipher suitable for low-resource device (Springer, Yokohama, 2006), pp. 46-59

[5] B Sun, CC Li, K Wu, Y Xiao, A lightweight secure protocol for wireless sensor networks. Comput. Commun. 29, 2556-2568 (2006)

[6] A Poschmann, G Leander, K Schramm, C Paar, in Proceedings of the IEEE International Symposiumon Circuits and Systems. New light-weight crypto algorithms for RFID (IEEE, New Orleans, 2007), pp. 1843-1846

[7] G Leander, C Paar, A Poschmann, K Schramm, in Proceedings of the 14th International Workshop on Fast Software Encryption. New lightweight DES variants (Springer, Luxembourg, 2007), pp. 196-210

[8] A Bogdanov, LR Knudsen, G Leander, C Paar, A Poschmann, MJB Robshaw, Y Seurin, Vikkelsoe C, in Proceedings of the 9th International Workshop on Cryptographic Hardware and Embedded Systems. PRESENT: an ultra-lightweight block cipher (Springer, Vienna, 2007), pp. 450-466

[9] C Canniere De, O Dunkelman, M Knezevic, in Proceedings of the 11th International Workshop on Cryptographic Hardware and Embedded Systems. KATAN and KTANTAN-a family of small and efficient hardware-oriented block ciphers (Springer, Lausanne, 2009), pp. 272-288

[10] W Wu, L Zhang, in Proceedings of the 9th International Conference on Applied Cryptography and Network Security (ACNS). LBlock: a lightweight block cipher (Springer, SPAIN, 2011), pp. 327-344

[11] A Olteanu, Y Xiao, F Hu, B Sun, H Deng, A lightweight block cipher based on a multiple recursive generator for wireless sensor networks and RFID. Wireless Commun. Mobile Comput. 11, 254-266 (2011)

[12] K Shibutani, T Isobe, H Hiwatari, A Mitsuda, T Akishita, Shirai T, in Proceedings of the 13th International Workshop on Cryptographic Hardware and Embedded Systems. Piccolo: an ultra-lightweight blockcipher (Springer, Nara, 2011), pp. 342-357

[13] Karakoc, F., Demirci, H., Harmanci, A. E., AKF: A key alternating Feistel scheme for lightweight cipher designs. INFORMATION PROCESSING LETTERS; FEB 2015, 115 2, p359-p367, 9p.

[14] Gaurav Bansod, Abhijit Patil, Swapnil Sutar, Narayan Pisharoty, ANU: an ultra lightweight cipher design for security in IoT. SECURITY AND COMMUNICATION NETWORKS; DEC 2016, 9 18, p5238-p5251, 14p.

[15] Evangelina Lara, Leocundo Aguilar, Jesus A. Garcia, Mauricio A. Sanchez, A Lightweight Cipher Based on Salsa20 for Resource-Constrained IoT Devices. Sensors, Vol 18, Iss 10, p 3326 (2018)

[16] Lang Lia, Botao Liua, Yimeng Zhou, Yi Zoua, SFN: A new lightweight block cipher. Microprocessors and Microsystems Volume 60, July 2018, Pages 138-150

[17] Botao LIU, Lang LI, Ruixue WU, Mingming XIE, and Qiuping LI, Loong: A family of Involutional Lightweight Block Cipher Based on SPN Structure. DOI10.1109 / ACCESS.2019.2940330, IEEE Access

[18] Shifa, A.; Asghar, M.N.; Noor, S.; Gohar, N.; Fleury, M. Lightweight Cipher for H.264 Videos in

the Internet of Multimedia Things with Encryption Space Ratio Diagnostics. Sensors 2019, 19, 1228

[19] AA Priyanka, SK Pal, A survey of cryptanalytic attacks on lightweight block ciphers. Int. J. Comput. Sci. Inf. Technol. Secur. (IJCSITS). 2, 472-481 (2012)

[20] Lin Ding, Chenhui Jin, Jie Guan, Qiuyan Wang, Cryptanalysis of Lightweight WG-8 Stream Cipher. IEEE Transactions on Information Forensics and Security IEEE Trans.Inform.Forensic Secur. Information Forensics and Security, IEEE Transactions on. 9(4):645-652 Apr, 2014

[21] Ding Lin, Jin Chenhui, Guan Jie, Slide attack on standard stream cipher Enocoro-80 in the related-key chosen IV setting. In Special Issue on Secure Ubiquitous Computing, Pervasive and Mobile Computing December 2015 24:224-230

[22] Bay A., Nakahara J., Vaudenay S. Cryptanalysis of Reduced-Round MIBS Block Cipher. In: Heng SH., Wright R.N., Goi BM. (eds) Cryptology and Network Security. CANS 2010. Lecture Notes in Computer Science, vol 6467, pp 1-19 .

[23] Coban M., Karakoc F., Boztas O. (2012) Biclique Cryptanalysis of TWINE. In: Pieprzyk J., Sadeghi AR., Manulis M. (eds) Cryptology and Network Security. CANS 2012. Lecture Notes in Computer Science, vol 7712, pp 43-55 .

[24] T A Darumaya and B H Susanti, Forgery Attack on LightMAC Hash Function Scheme using SIMECK 32/64 Lightweight Block Cipher. Materials Science and Engineering, Volume 453, conference 1, International Conference on Design, Engineering and Computer Sciences 2018

[25] M Cazorla, K Marquet, Minier M., Survey and benchmark of lightweight block ciphers for wireless sensor networks, in Proceedings of the 2013 International Conference on Security and Cryptography (SECRYPT).(IEEE, Reykjavik, 2013), pp. 1-6

[26] Dalmasso, L., Bruguier, F., Benoit, P., Torres, L., Evaluation of SPN-Based Lightweight Crypto-Ciphers. IEEE Access Access, IEEE. 7:10559-10567 2019

[27] Chao Pei, Yang Xiao, Wei Liang and Xiaojia Han. Trade-off of security and performance of lightweight block ciphers in Industrial Wireless Sensor Networks. EURASIP Journal on Wireless Communications and Networking (2018) 2018:117.

[28] Wei LI, Vincent RIJMEN, Zhi TAO, Qingju WANG, Hua CHEN, Yunwen LIU, Chaoyun LI and Ya LIU, Impossible meet-in-the-middle fault analysis on the LED lightweight cipher in VANETs Sci China Inf Sci March 2018 Vol. 61 032110:3

[29] Malik Qasaimeh, Raad S. Al-Qassas and Sara Tedmori, Software randomness analysis and evaluation of lightweight ciphers: the prospective for IoT security. Multimed Tools Appl (2018) 77:18415-18449

[30] Annelie Heuser, Stjepan Picek, Sylvain Guilley, and Nele Mentens, Lightweight Ciphers and their Side-channel Resilience. DOI 10.1109/TC.2017.2757921, IEEE Transactions on Computers

[31] Yuechuan Wei, Peng Xu, Yisheng Rong, Related?key impossible differential cryptanalysis on lightweight cipher TWINE. Journal of Ambient Intelligence and Humanized Computing (2019) 10:509-517

[32] 董際国，森田啓義，“整数ロジスティック写像と撹拌演算による乱数生成”，信学論誌，Vol.J94-A, No.12, pp.923-931, Dec. 2011.

[33] 董際国，森田啓義，“認証鍵の生成・管理のための「多次元インデックス法」”，信学論誌，vol.J98-D, no.1, pp.206-213, Jan. 2015.

[34] 董際国，森田啓義，“整数ロジスティック写像の諸性質：発散，収束，周期性”，信学論誌，vol.J96-A, no.2, pp.90-99, Feb. 2013.

[35] Matsumoto, M., AND Nishimura,T., “Mersenne Twister: A 623-dimensionally equidistributed uniform pseud random number generator,” ACM Trans. on Modeling and Computer Simulations 8, pp. 3–30, 1998.

[36] D.E. Knuth, The Art of Computer Programming vol.2 Seminumerical Algorithms, Third Ed., Adisson-Wesley, 1997.