

## 【論文】

## 乱数発生器を用いたデータ消去方法

井上 正人\*

## A Method of Erasing Data Using Random Number Generators

Masato Inoue

## Abstract

Erasing data is an indispensable step for disposal of computers or external storage media. Except physical destruction, erasing data means writing random information on entire disk drives or media. We propose a method which erases data safely using random number generators. These random number generators create true random numbers based on quantum processes.

**Key words:** Erasing Data, Random Number

## 1 はじめに

乱数は、暗号における鍵の生成や、使い捨てパッド(2進数で表現された平文とランダムな2進数列である使い捨てパッドとの排他的論理和をとり、元の平文を暗号化する方法、暗号文と使い捨てパッドの排他的論理和をとると元の平文が得られる)の生成などに用いられる。また、データ消去においては、元の情報にランダムなデータを上書きすることにより、元のデータの復元をできないようにするために用いられる。

乱数には、モンテカルロシミュレーション等に使用されるソフトウェア的に発生させる擬似乱数と、何らかの物理的現象を利用し発生させる物理乱数がある。

擬似乱数は、シード(seed)とよばれる出発値から、アルゴリズムに基づいて、順次ランダムに見える数値を発生させる。非常に長い周期の後に、元のシードの値に戻り、以後、同じ数値が繰り返される。シードが同じであれば、同じ値の乱数が発生するので再現性が必要なシミュレーションには都合がよいが、シードがわかってしまうと以後の値がすべてわかってしまい、使い捨てパッドなどではパターンがすべてわかってしまう危険性があり、ふさわしくない。

物理乱数には、放射性原子核の崩壊を用いるものや回路素子に流れる電流の熱によるゆらぎであるショット雑音を用いるもの、ハーフミラーや光の偏光など量子論的な確率現象を用いるものがある。

今回は、ハーフミラーを用いて、乱数を発生させる小型の乱数発生器を用いてハードディスクのデータ消去を行う方法を提案する。

## 2 データ消去

データ消去の重要性は言うまでもない。適切にデータを消去せず、使用済みコンピュータからデータ流出した例は枚挙にいとまがない。また、コンピュータの引き取りの際、データ消去の契約をしていたにも関わらず、そのまま再販売し、データ流出した例も見受けられる。

ハードディスク等のデータを消去するには、物理的に破壊しデータを消去する場合と、ランダムなデータをハードディスクのすべての領域に上書きして、記録してあったデータを復元できないようにする方法がある。物理的破壊には、高価な機械が必要であるが、ハードディスクに強磁場をかけて一瞬でデータを消去する方法がある。また、ディスクに穴をあけて読めなくする方法やハードディスクを分解し、情報の記録面をサンドペーパー等で傷付けて

\*海上保安大学校 inoue@jcga.ac.jp

読めなくする方法がある。ディスクに穴をあける方法は、記録面が一部残ってしまい、その部分だけ復元できる危険性がある。

これらの物理的破壊は、当然のことながらハードディスクの再利用はできなくなる。リース契約を結んでいて、リース終了後、リース会社に変換しなくてはならない場合には使用できない。また、買取りの場合であっても、まだ十分に使用できるコンピュータをデータ保護の観点のみで破壊するのは、特に機密性の高い情報を除き、資産の有効活用及び環境保護の観点から問題がある。そのような場合にランダムなデータをハードディスクに上書きする方法が用いられる。

ハードディスク等において、データは実際にデータが保存されるデータ領域とデータの保存場所を示す管理領域に分けられる。データが削除されると管理領域の情報のみが削除され、ファイル復元ソフト等を用いるとデータ領域の情報から管理領域が復元でき、データにアクセスできるようになる。このことからデータの削除は、データ領域も含めて情報を上書きしなければならない。また、ランダムな情報で上書きするのは、ハードディスク等においては磁気表面のミクロな分析により、情報が上書きされた状態でも、元の情報が復元できるからである<sup>1)</sup>。

本論文では、乱数の発生源として IDQ 社の乱数発生器 QUANTIS-USB を用い、データ消去ソフトとして GNU の Coreutils<sup>2)</sup> パッケージの中の shred を用いてデータ消去をする装置を作成する。

### 3 データ消去の装置について

データ消去を行うコンピュータの OS は Linux のディストリビューションの 1 つである Debian squeeze<sup>3)</sup> を使用した。アーキテクチャは 64 ビット PC(AMD64)を用いた。Debian は QUANTIS-USB の動作に必要な libusb-1.0 のモジュールを標準で備えている。Red Hat 系の OS 等ではこのパッケージを追加するか、ソースからインストールする必要がある。コンパイラなどの開発ツールとして gcc, make, libusb-1.0-0-dev などのパッケージを追加でインストールし利用した。コンピュータのスペックは以下の通りである。

CPU : Intel デュアルコア Xeon 3010(1.86GHz)  
 メインメモリ : 4GB  
 ハードディスク : 73GB SAS(15000rpm)

### 4 乱数発生器について

コンピュータにおいてはランダムなデータを発

生させるのが難しい。通常のコンピュータにおいて、出力は内部状態と入力から完全に決定され、予測不可能で再現不可能な現象の発生源を備えていないからである。Linux 系の OS では、乱数の発生源として /dev/random や /dev/urandom が使用される。どちらもキーボードやマウスなどを押したり開放したりするときの割込み信号の時間差を 2 進数に変換して一時的に蓄えておき、必要なときに取り出して乱数を発生させるもので特殊なファイルとして実装されている。/dev/random は十分なビット数の 2 進数が蓄積された後に出力を行う。それまで出力はブロックされる。一方、/dev/urandom は乱数の質が低下しても強制的に出力を行う。データ消去など、たくさんの乱数を発生させなければならない場合には /dev/urandom が使用されることが多い<sup>4)</sup>。

QUANTIS-USB は量子力学的な現象であるハーフミラーを用いて乱数を発生させる。図 1 において、光源から光子が 1 個放出された場合、透過か反射のどちらかが起こり、同時に起こることはない。また、時間差で発生させた光子が透過するか反射されるかは全く独立で、互いに影響することはない。さらにこの現象は内在的で、この系の外部の影響を受けない。ハーフミラーで光子が反射される確率と、透過する確率は等しくしておく。検出器で光子を検出した場合を 1、検出されない場合を 0 とすると光源から光子を連続的に発生させた場合、1 または 0 になる確率は等しく、0 と 1 は全くランダムな値が得られる。

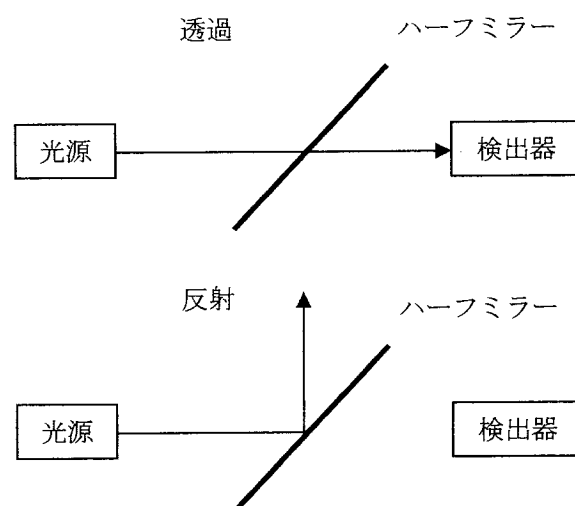


図1 ハーフミラーによる透過と反射

IDQ 社の QUANTIS の製品には PCI バスと UBS バスの 2 種類のモジュールがあり、これらのモジュールはアメリカ国立標準技術研究所

NIST(National Institute of Standards and Technology)の NIST SP800-22 の乱数テスト<sup>5)</sup>や DIHARD テスト(Diehard Battery of Tests of Randomness<sup>6)</sup>)など高度な乱数テストに適合している。今回は USB バスの QUANTIS-USB を用いる。この装置の乱数の生成能力は 4Mbit/s ± 10%(1Mbit は 10<sup>6</sup>bit)である。

## 5 データ消去のプログラムについて

Linux で使用できるデータ消去ソフトとして GNU の Coreutils パッケージの中に含まれる shred がある。shred はファイル消去やハードディスクのデータ消去に用いられる。Linux ディストリビューション Debian squeeze にも標準で GNU の Coreutils がインストールされているが、QUANTIS-USB を使用するためにプログラムに変更を加える必要があるため、最新のパッケージ coreutils-8.14 のソースをダウンロードして展開し、必要な変更を加えた後にコンパイルを行う。

標準で shred は /dev/urandom から乱数を読み込むようになっているが、これを QUANTIS-USB から乱数を読み込むよう変更してコンパイルする。このときコンパイルに必要なライブラリやヘッダファイル等は参照できるように設定しておく必要がある。

## 6 発生する乱数について

データ消去を行う前に QUANTIS-USB で発生させた乱数の簡単なテストを行った。QUANTIS のモジュールは高度な乱数テストに適合しているが、プログラムや設定ミス等により乱数が適切に発生していない危険性を排除するためである。

0 以上で正の整数  $r$  より小さな整数の乱数を  $N$  個発生させたとする。このとき  $i$  番目の乱数の発生数を  $f_i$  として、

$$\chi^2 = \frac{r}{N} \sum_{i=0}^{r-1} \left( f_i - \frac{N}{r} \right)^2$$

とおく。 $N$  の値が十分大きいと、この値を利用して  $\chi^2$  検定ができる<sup>7)</sup>。

乱数は 1 バイトずつ発生させて、unsigned char 変数に値を入れる。2 進数 1 バイトは 10 進数で 0 から 255 までの数値で、 $\chi^2$  検定における自由度が 255 となり通常の  $\chi^2$  分布表には載っておらず、検定を行うには自由度が大きすぎるので、上位 4 ビットと下位 4 ビットを独立な数値として 0 から 15 まで

の数値とし、自由度を 15 として検定を行う。5000 個の 1 バイトの乱数を発生させ、上位ビットと下位ビットを分けた 4 ビットでの数  $N$  を 10000 とし、 $r$  を 16 とする乱数を発生させる。有意水準 0.05 で自由度 15 の棄却域は 25.0 であるから、これより大きな領域で、 $\chi^2$  がこの領域にあると 0 から 15 まで同確率で乱数を発生している仮説が棄却される。これらの検定を 100 回繰り返して 95 回は棄却されず、5 回は仮説が棄却された。

## 7 ファイルの消去

shred は -u オプションを付けない場合、ファイルを消去せずファイルの内容をランダムな値で上書きするため、実際にファイルが書き換えられているかがチェックできる。1GB のファイルを作成し、これらのファイルを消去した。これらは -u オプションを付けて行った。ファイルの消去は 10 回行い消去に要する時間の平均を求めた(表1)。参考のため、/dev/urandom から乱数を発生させた場合の消去に要する時間を掲載している。

表1 ファイル消去に要した時間(秒)

	QUANTIS-USB	/dev/urandom
1	2450	12
2	2450	11
3	2448	11
4	2447	11
5	2448	11
6	2447	11
7	2447	12
8	2448	12
9	2448	11
10	2450	12
平均	2448.30	11.40

QUANTIS-USB の乱数の生成能力 4Mbit/s から計算して、1GB の乱数を発生するのに必要な時間は、1 バイトが 8 ビットであるから、

$$\frac{1024 \times 1024 \times 1024 \times 8}{4 \times 10^6} = 2147.48$$

秒であり、大体、本来の乱数発生性能を発揮している。

次にハードディスク全体のデータを消去する例としてマイクロドライブ(容量340MB, 4500rpm)のデータを消去する。マイクロドライブはカードリ

ーダに装着し、USB 接続でコンピュータとつないだ状態で消去する。実際にデータが消去されているかどうかは dd コマンドでマイクロドライブのデータをファイルに書き出してチェックすればできる。今回もデータの消去は10回行い消去に要する時間の平均を求めた(表2)。マイクロドライブはハードディスクのアクセスに時間がかかるため、上書きするデータの容量がファイルの消去の場合より少なくても、ファイルの消去の時には短時間であった。/dev/urandom から乱数を発生させたときの消去にかなり時間がかかっている。

表2 ハードディスク消去に要した時間(秒)

	QUANTIS-USB	/dev/urandom
1	934	191
2	822	188
3	935	184
4	824	184
5	832	183
6	839	182
7	824	184
8	836	182
9	822	183
10	826	183
平均	849.40	184.40

通常のコンピュータのハードディスクはちょっと古いタイプでも容量が数十 GB 以上ある。これを、乱数発生器を用いてデータを消去しようとする2日はかかる計算になる。実際、Linux をインストールしたハードディスクと同じ、容量 78GB の SAS のハードディスクを消去してみたが 47 時間ほど必要であった。

## 8 おわりに

今回、乱数発生器と Linux マシンで、ハードディスクのデータを消去する方法を提案した。IDE や SATA の内蔵ハードディスクを外部ハードディスクとして認識できるようなキットも出ており、Linux のカーネルが認識するハードディスク等であればデータを消去できる。上書きされるデータは真乱数であり、ミクロレベルでも、データが復元される危

険性は減少する。ただし、復元可能性の分析は顕微鏡レベルの詳細な分析が必要であり、今回は行わなかった。

今回の乱数発生器においては、数十 GB 以上の容量のハードディスクを消去するには2日以上かかり、安全にかつ現実的にデータを消去するにはもっと高速に乱数を発生させる装置を用いなければならない。

また、RAID 環境や SSD など、データの書き込みに関して再配置を行う装置などにおいては、うまくデータが上書きされない危険性があるのでデータ消去において注意が必要である<sup>8)</sup>。最近では SSD などに Secure Erase(全領域のデータ消去)機能が付属しているものが出ていますので安全にデータ消去を行うには、このような製品を選ぶ必要がある。

## 参考文献

- 1) Gutmann, P., *Secure Deletion of Data from Magnetic and Solid State Memory*, SYM'96: Proceedings of the 6th conference on USENIX Security Symposium, Focusing on Applications of Cryptography, USENIX Association, Berkeley, CA, (1996) 77-90
- 2) <http://www.gnu.org/s/coreutils/>
- 3) <http://www.debian.org/>
- 4) Gutterman, Z., Pinkas B., Reinman, T., *Analysis of the Linux Random Number Generator*, SP '06 Proceedings of the 2006 IEEE Symposium on Security and Privacy, IEEE Computer Society Washington, DC, (2006) 385-399
- 5) <http://csrc.nist.gov/groups/ST/toolkit/rng/index.html>
- 6) <http://i.cs.hku.hk/~diehard/cdrom/>
- 7) Sedgewick, R., *Algorithms in C*, Addison-Wesley, Publishing Company, Boston, MA, (1990), 509-519
- 8) Wei, M., Grupp, L. M., Spada, F. E., Swanson, S., *Reliably Erasing Data from Flash-Based Solid State Drives*, Proceedings of FAST '11: 9th USENIX Conference on File and Storage Technologies, USENIX Association, San Jose, CA, (2011) 105-117