2J-01

FPGA ベース並列マシン RASH の タイムメモリトレードオフ解読法への適用

高橋 勝己, 飯田 全広 †, 中島 克人

三菱電機 (株), † 三菱電機エンジニアリング (株)

1 はじめに

近年の FPGA(Field Programmable Gate Array) の大容量化と高速化により, ハードウェアのプロトタイプ設計手段や他のプロセッサの補助としての利用以外に, それ自体を処理要素として扱うことが可能になった. これに伴い, 我々は FPGA を用いた処理プラットフォームとして, これまでに複数の用途のために FPGA を搭載したスケーラブルシステム RASH(Reconfigurable Architecture based on Scalable Hardware) を作成した [1][2]. FPGA による実装は, ビット操作を伴う演算が多い大規模演算において特に威力を発揮できる. そのような演算の 1 つに暗号解読がある.

DES(Data Encryption Standard) は最も有名な暗号である. DES は, 64 ビットのブロックを 1 つの単位として, 56bit の鍵によって暗号化と復号を行なう暗号であるため, 2^{56} の探索空間 (鍵空間) を持つ. RSA 社はこの DES に対する既知平文攻撃 (平文と暗号文から鍵を探索) を行なう DES チャレンジを開催し, ここ数年の様々なチームがそのチャレンジに成功している [3].

我々は, この DES の鍵探索時間大幅な短縮 (1 ～ 2 時間のオーダ) を目指す. この時, より少ない規模で実現を目指すため, 事前準備として数カ月分の処理を必要としても, 探索によって鍵が見つかる確率が 100% でなくても良いとした. この条件の下, 我々は, タイムメモリトレードオフ解読法 (Time-Memory Trade-Off Cryptanalysis: 以下 "TMTO 法" と略す) [4] を選択し, その選択のために必要な RASH の拡張と, 拡張後の性能を見積もった. 構成の 1 例を示すと, 50 箱からなる RASH では, 事前に 1 カ月かけて特殊な表を作成すれば, 80% の確率で約 1 時間以内に鍵を見つけることができる.

2 RASH の構成

RASH は, Compact-PCI(Peripheral Component Interconnect) の筐体複数と PC で構成され, これらは LAN によって接続される. 各筐体には, 筐体内の制御を行なう CPU ボードと, ハードディスク, 電源, ファンなどが予め用意されており, これにボードを 7 枚まで挿入できるようになっている. RASH では, ここに FPGA を複数搭載した EXE ボードを挿入する. PC は, ユーザとのインタフェースと RASH 全体の管理のために用いられる. 図 1にその基本構成を示す.

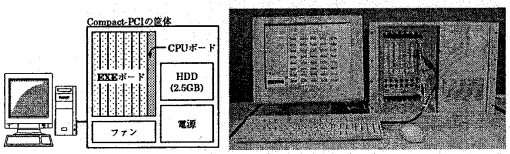

図 1: RASH 基本構成 (構成図と装置写真)

RASH における演算の基本構成要素は 1 石の FPGA である. FPGA には, 1 石 10 万ゲート規模相当の ALTERA 社 FLEX10K100A シリーズを選択した. EXE ボードには, FPGA 8 石, Compact-PCI のインターフェース回路, ローカルメモリ, ボードの制御回路を搭載する. 図 2に EXE ボードの構成を示す.

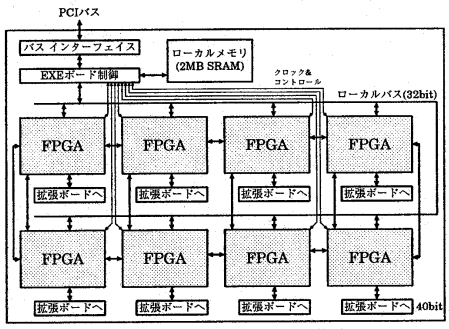

図 2: EXE ボードの構成

3 DES における TMTO 法

TMTO 法は, 平文は既知だが暗号文が未知である期間に特殊な表を作成し, 暗号文が既知となった際, その表を用いることで, 暗号文作成に用いられた鍵の探索時間を短縮する方法である.

この表は, DES の暗号化 "$C = E(P, K)$"(C は暗号文 (64bit), P は平文 (64bit), K は鍵 (56bit)) と調節関数 "$K = R(C)$" を組み合わせた変形暗号化関数 "$C' = E'(C) = E(P, R(C))$" を用いて作成する. M 種の初期値と L 種の調節関数 R を用意し, 個々の初期値に対して (同一の調節関数を用いた) 変形暗号化を T 回繰り返し, これを 1 枚の表とする. これを別の調節関数で再度繰り返し, 最終的に L 枚の表を作成する [5]. ここまでが事前準備に相当する.

鍵の探索は, 暗号文と 1 枚の表の各要素とを比較し, 一致すれば検証処理 (詳細 [5]) を実施し, 一致しなければ, 暗号文を表の作成に用いた変形暗号化関数で変換し, 再度表の要素との比較を行なう (これを最大 T 回繰り返

Time-Memory Trade-Off Cryptanalysis on FPGA-based Parallel Machine RASH

K. Takahashi, M. Iida, K. Nakajima

Mitsubishi Electric Corporation, †Mitsubishi Electric Engineering Co.,LTD.

す). 以上の作業を全ての表との間で実施する. これにより, 表の作成中に調節関数の出力として現れた値の中に鍵と同じ値が含まれていれば, この探索によって鍵を発見することができ, その確率は L, M, T の値によって変化する [6].

今回はこの各値として, "$L = 2^{22}$, $M = 2^{14}$, $T = 2^{21}$" を用いる. この時, 探索に成功する確率は 80% である.

3.1 拡張

RASH は, DES の全数探索であれば基本構成のまま拡張なしに実現できる. その時の性能は, 約 30M 鍵 /s になる (4 クロックで 1 回の暗号化を行なう回路を 3 回路実装, 動作周波数は 39.5MHz)[7]. これに対し, TMTO 法を用いるためには, 変形暗号化, 作成した表の格納, 及び, 表要素との比較の 3 つの機能を RASH 上に実現する必要がある.

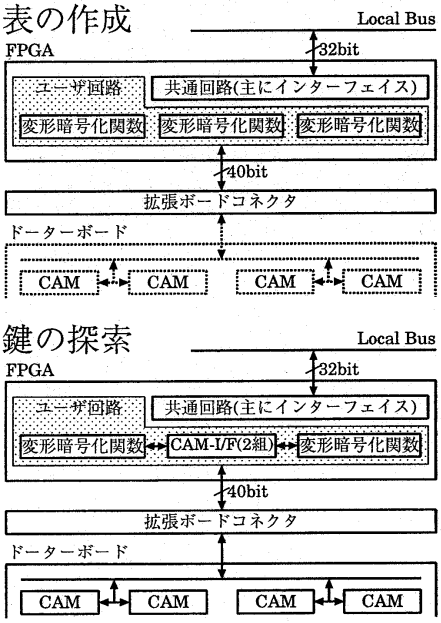

図 3: FPGA 内の回路と拡張

変形暗号化関数は, DES の暗号化に調節関数を加え, 再帰呼び出し可能な関数に変形させているのみである. この調節関数として, 上位ビットの削除と任意の値との排他的論理和を行なうものを採用した. これにより, ハードウェアの拡張が不要なだけでなく, 回路自体も先の DES の回路に大きな修正を加えずに実現することができる. また, 表の格納はハードディスクの容量を増やすことで容易に対応できる. 一方, 表要素との比較は, 鍵探索時の速度を大きく左右し, 1 枚の表の大きさも FPGA 上に保持するには大き過ぎるため, 比較的大きな拡張を必要とする. RASH では, FPGA からの制御の容易さと検索の高速性から, 単なるメモリを使用した 2 分探索やハッシュによる実現ではなく, CAM(Contents Addressable Memory: 連想記憶メモリ, 8K エントリ 2 個) を用いることでこの機能を実現する. この CAM はドーターボード上に配置し, 拡張ボードコネクタによって EXE ボードと接続する.

変形暗号化のみを繰り返し実行する表作成時には, DES の全数探索時同様に FPGA あたり変形暗号化を 3 回路実装し, 表要素との比較も行なう鍵探索時には, CAM の制御用回路のため, 変形暗号化を 2 回路実装して処理を行なう. 図 3に FPGA 内部の回路構成を示す.

3.2 実行速度

TMTO 法の演算時間は, 表作成で $O(LMT)$, 鍵探索で $O(LT)$ となり, 必要記憶容量は, 表の格納用として $O(LM)$ となる. 表作成の演算時間の基本となる, 変形暗号化 1 回路の実行速度は 7.9M 鍵 /s と見積もれる. 鍵探索の表要素との比較と変形暗号化は, 並行実行可能だがより遅い速度によって規定され, 3M 鍵 /s となる. その結果, 1 時間以内に 80% の確率で鍵を探索することができるシステムが, 表 1にあるような構成で実現できることになる.

表 1: システムの構成と性能

ユニット数	51	26
記憶容量 / ユニット	10.0GB	19.7GB
EXE ボード枚数 / ユニット	6	5
ドーター搭載 EXE ボード枚数	1	2
事前準備期間	28.7 日間	67.7 日間
鍵の散策時間	1 時間	1 時間

4 おわりに

本稿では, FPGA ベース並列マシン RASH を TMTO 法による DES の鍵探索に用いた場合に必要となる拡張と, その結果得られる性能について述べた. 本装置は, 異なる暗号への対応はもちろん, TMTO 法のように処理が複数のフェーズ (表作成と鍵探索) に分かれ, 必要となる機能が異なる場合にも, 回路情報の書き換えによって対応することができる. また, 回路の改良により, 全数探索 /TMTO 法とも, システムとして性能向上が期待できる. 今後も, これら性能向上対策や適用範囲の拡大を行なってゆく予定である.

参考文献

[1] 中島他, "FPGA ベースの並列マシン RASH の概要," 情報処理学会第 58 回全国大会 1H-08, 1999

[2] 浅見他, "FPGA ベースの並列マシン RASH のシステム機能と構成," 情報処理学会第 58 回全国大会 1H-09, 1999

[3] "RSA - DES Cracked!," DES Challenge home page, RSA Data Security, Inc.
Available at http://www.rsa.com/des/

[4] M.E.Hellman, "A Cryptanalytic time-memory trade-off," IEEE Transactionson on Infomation Theory, Vol.IT-26, No. 4, pp.401-406, 1980.

[5] 高橋他, "タイムメモリトレードオフ解読法に基づく暗号強度評価装置の実現 性について," 情報処理学会論文誌第 40 巻第 8 号, 1999

[6] 楠田他, "タイム・メモリ・トレードオフ解読法の最適化とブロック暗号への適用," 1995 年暗号と情報セキュリティシンポジウム講演集, SCIS95-A3.2, 1995.

[7] 浅見他, "FPGA ベース並列マシン RASH での DES 暗号回路の改良," システム LSI 設計技術研究会, 信学技報, 2000