FPGA ベース並列マシン RASH における部分 TMTO 法暗号解析の実装

2W−05

高橋 勝己 †, 飯田全広 ‡, 浅見廣愛 †
† 三菱電機 (株), ‡ 三菱電機エンジニアリング (株)

1　はじめに

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)[1] は、事前にテーブルルックアップ法よりも容量の少ない特殊な表を作成することで、暗号文を入手した際、全数探索よりも少ない演算量で鍵を求める解読手法である。一方、我々は FPGA(Field Programmable Gate Array) を多数用いた可変構造型計算機として FPGA ベース並列マシン RASH (Reconfigurable Architecture based on Scalable Hardware) を試作し、DES (Data Encryption Standard) を始めとする秘密鍵暗号において、鍵の全数探索が高速に行えることを示した [2]。更に、TMTO 法を RASH で実装することでより高速な鍵探索が行えることも示した [3]。

ここで、DES を対象として、その鍵が、パスワードの入力のように印字可能な文字の入力によって設定されているとする。この場合、鍵の取り得る値には、コードによる限定が生じることになる。本稿では、この限定が加わった鍵の部分集合に対して、TMTO 法による暗号解析を行う方法について報告する。

2　TMTO 法と鍵の限定

2.1　TMTO 法の原理

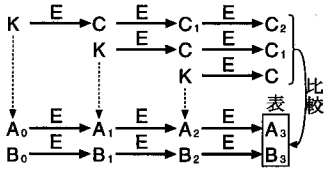

図 1: TMTO の原理

TMTO 法は、暗号化関数を E、入手した暗号文を C、暗号文に使用した鍵を K とした時、K を確率的な探索で求める方法であり、関数の写像が 1 対 1 なら、次のような手順で鍵を探索できる (図 1 参照)。

表作成: 初期値 (A_0, B_0) に対して暗号化 E を繰り返し、表 (A_3, B_3) を作成する。これを探索表と呼ぶ。

鍵探索: 入手暗号文 C と表 (A_3, B_3) を比較し、一致するものがあれば、対応する A_2, B_2 が鍵 K となるため、初期値 (A_0, B_0) からその値を生成する。一致するものがなければ C_1 を生成し、同様に表 (A_3, B_3) と比較し、一致するものがあれば、同様の手順でを鍵を生成することができる。これを C_3 の生成と比較まで繰り返す。

鍵探索が終了しても一致するものがなければ、作成した探索表から鍵を見つけることはできない。

2.2　印字可能文字よる鍵の限定

ASCII コードは、英数字、記号、制御文字を 7bit で表現したものである。パスワードの入力のように、8 文字分のキー入力があれば、これを 56bit の DES の鍵とすることができる。このようなキー入力を考えた場合、各 1 文字において設定される範囲は、(制御文字を除く) 印字可能な 96 文字種類のみであり[1]、この限定を利用すれば、鍵の探索範囲を狭めることができる。

2.3　TMTO 法と限定

TMTO 法は、事前に多数の要素を持つ探索表を複数枚用意することで、暗号文入手後は鍵探索だけで鍵を求めることができるようになる。TMTO 法では、探索表毎の多様性の確保と暗号文と鍵のビット長の調整のため[2]、暗号化関数そのものではなく、それに調節関数を加えた変形暗号化関数を用いて表作成や鍵探索を行う。

TMTO 法で探索できるのは、表作成で行う暗号化において鍵として使用された値である。従って、印字可能文字よる鍵の限定が加わっている場合には、この値を限定の範囲内に留めるようにすればよい。

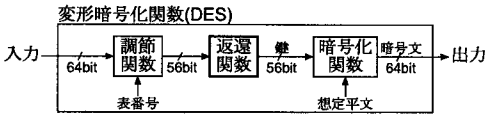

図 2: 変形暗号化

図 2 は、DES を対象とした場合の変形暗号化関数の例である。調節関数と暗号化関数の間に、限定の範囲

Implementation Subset Time-Memory Trade-Off Cryptanalysis on FPGA-based Parallel Machine "RASH"
K.Takahashi, M.Iida, H.Asami
Mitsubishi Electric Corporation & Mitsubishi Electric Engineering Co.,LTD.

[1]NUL コードは文字列の終了として印字可能文字種に含める。
[2]ビット数の調整が行われる場合は写像が 1 対 1 にならないため、もう 1 段判定処理が加わる。

外に出た鍵を引き戻す返還関数を挿入した形になっている。

返還関数としては、印字可能文字は96種であるため、全体の96^8の剰余をとり、これを96毎に区切ったものをASCIIのコードに戻すという形も考えられる。しかし、FPGAに搭載する回路としては、複数の独立したブロックに分かれている方が都合が良いため、各1文字分7bitを独立に処理する次のようなものを用意する。

固定方式: ASCIIコード0〜$7F_h$のうち、NULとDELを入れ換えれば、20_h〜$7F_h$が印字可能文字の範囲となる。そこで、0〜$1F_h$の範囲だった場合には$20_h, 40_h, 60_h$のいずれかを加え、その後、NULとDELを入れ換える。

縮退利用方式: DESを対象とした場合、縮退関数において8bitの情報が破棄される。この破棄情報を1文字毎に1bit使用する。そして、0〜$1F_h$の範囲だった場合に加える値を$20_h, 40_h, 60_h$から2つ選択しておき、そのどちらの値を加算するかをこの1bitを用いて選択する。

循環方式: 表の番号に従って、各文字毎に、0〜$1F_h$の範囲だった場合に加える値を$20_h, 40_h, 60_h$の3種から選択する。このバリエーションは$6561(=3^8)$種となるため、表は6561枚ごとに同じ組合せが使用されることになる。

前者の2つは、探索領域を外れた値が固定的に引き戻されるため、暗号化関数と縮退関数において偏りがないとしても、返還関数によって探索領域内に偏りが生じる。この偏りが大きいほど、探索表から探索できる鍵の範囲を狭められる。この偏りを積極的に利用すれば、英小文字が多い場合に発見しやすくなる探索表が出来るだけでなく、英大文字や数字記号を発見し易い表も同様に用意できるようになる。

一方、循環方式は表の内部では偏りがあるが、この偏りを表毎にずらすことで全体として偏りが無いのと等価とする方式である。

3 性能評価

3.1 FPGA内の回路構成

図3に表作成と鍵探索それぞれにおけるFPGA内の回路構成を示す。表作成用は、スループット性能を重視し、調節関数、返還関数、8段パイプラインの構成のDES関数からなる変形暗号化回路と制御及びバスI/F(インターフェイス)回路で構成し、鍵探索用は、探索表の検索速度に合わせて、DES関数を1段パイプラインの構成に変えた変形暗号化回路2個と制御及びバスI/F回路に、探索表を格納し比較するために用意するメモリとのメモリI/F回路を加えた構成とした。

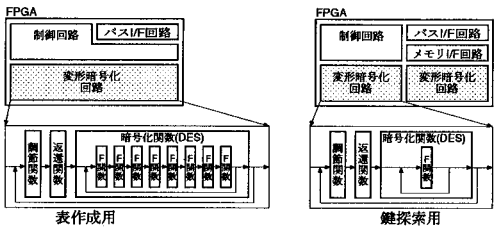

図3: FPGAの回路構成

3.2 探索時間と装置規模

印字可能文字という限定によって、鍵の探索領域は約$\frac{1}{10}$に縮小される。これに対し、返還関数の挿入による速度低下は表作成で約1割、鍵探索で約3割となり、回路増加による影響はFPGA内の構成を変えない程度に収まる。

この結果、30筐体構成のRASH[3]は、「事前に約52日間かけて表作成を行い、2時間程度で8割の確率で鍵を発見する暗号解析」を実施できる[3]のと同様に、「事前に約1週間かけて表作成を行い、20分程度で8割の確率でキー入力による鍵を発見する暗号解析」も実施できるようになる。

4 おわりに

本稿では、FPGAベース並列マシンRASHをパスワード入力のような限定を加えた鍵を対象とした部分集合に対するTMTO法による暗号解析に適用した結果について述べた。鍵の限定に対応するための返還関数は、いずれも簡単な回路で実現でき、実行速度や回路規模にほとんど影響を与えずに実現することができる。その結果、鍵の探索効率を大幅に向上させることができることから、鍵が部分集合となるケースにおいてもTMTO法に基づく暗号解析を活用することができる。

参考文献

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

[2] 浅見 他, "FPGAベース並列マシンRASHでのDES暗号解析処理の改良," 情報処理学会論文誌:ハイパフォーマンスコンピューティングシステム Vol.41, No.SIG 5(HPS 1), pp.50-57, 2000-8

[3] 飯田 他, "FPGAベース並列マシンRASHにおけるTMTO法暗号解析の実装(2)〜性能評価〜," 情報処理学会第62回全国大会 2S-08, 2001

[3] 1つの筐体には、FPGA8個搭載のボード6枚(うち1枚はメモリボードをドータとして搭載する)が挿入される。