L-010

GPGPUを用いた暗号攻撃

Attack against cipher algorithm using CUDA

西川 尚紀† 岩井 啓輔† 黒川 恭一†
Naoki Nishikawa Keisuke Iwai Takakazu Kurokawa

1 はじめに

GPU(Graphics Processing Unit) とは, グラフィクス処理に特化した半導体チップである. 2005 年ころから GPU の並列演算能力を汎用計算（GPGPU:Generally Purposed computation on GPU）に応用する研究が広がりを見せ, 成果を挙げた. 更に, 最近の世代の GPU からは整数・論理演算命令もサポートされたことにより, 暗号処理への応用も可能になった.

本稿は, GPU を用いた暗号処理の高速化の一例として, 暗号アルゴリズムへのパスワードクラックを実装し, その成果を元に同手法の可能性を検討したものである. 攻撃対象とする暗号部分はモジュール化し, 複数の暗号に対応できる. 今回は DES と AES を攻撃対象とした.

2 パスワードクラック

一般的に行われるパスワードクラックとして, Brute Force Attack や辞書攻撃等があるが, 本稿では, 最も一般的な方法である Brute Force Attack を図 1 に示すように GPU 実装した.

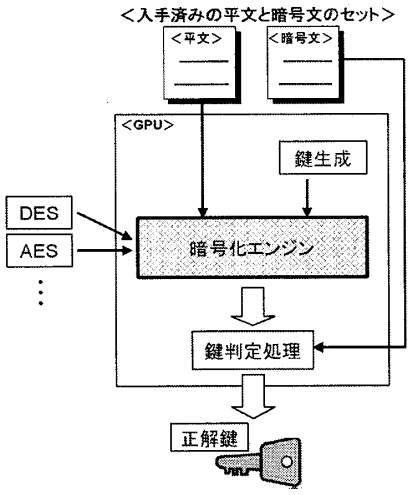

図 1: Concept of Key Crack using GPU

3 DES と AES

3.1 DES

DES(Data Encryption Standard) は 1976 年に決定された米国政府標準の共通鍵ブロック暗号である. 鍵長は 56 ビットで, ラウンドと呼ばれる暗号化ステップを繰り返す Feistel 構造をとっている. DES では, ラウンド処理を 16 回行う. 各ラウンド処理では 64 ビットの平文データが入力され, 入力は左半分 32 ビットと右半分 32 ビットに 2 等分され, 別々に処理される.

3.2 AES

AES(Advanced Encryption Standard) は, 安全性が低下した DES に代わる標準暗号規格として, 1997 年, 米国 NIST が世界中から公募し制定された米国政府標準の共通鍵ブロック暗号である. AES は, 16 バイトのデータを 1 単位として暗号化/復号を行うアルゴリズムであり, SubBytes,ShiftRows, MixColumns, AddRoundKey の 4 つの処理が繰り返し行われることにより暗号化される.

4 GPU 統合開発環境 CUDA

CUDA では GPU を複数のマルチプロセッサから成る並列計算アーキテクチャとして扱う. MP には 8 個のストリームプロセッサ (SP), 1 基の命令ユニット, 複数本のレジスタ, 1 基の共有メモリが内蔵されている. 命令ユニットは SP で共有される.

MP 内には, SP が使用するレジスタと, SP 間で共有する共有メモリがある. レジスタはスレッド毎に割り当てられ, 他のスレッドからは参照できない. 共有メモリはレジスタ並みの高速アクセスが可能だが, 容量が少ない. GM は大容量だが, メモリアクセスに遅延が生じる. 共有メモリはグローバルメモリからデータを読み出す際のキャッシュとして用いられることが多い. 今回用いた GPU である NVIDIA Geforce GTX 285 では MP30 個及びグローバルメモリ 1GB を搭載している.

CUDA では, CPU から GPU に発行されたジョブは, 1〜512 スレッドから成るスレッドブロックと呼ばれる単位で MP に割り当てられる. 更に MP 内では, 32 スレッドから成るワープと呼ばれる単位で処理が実行される. ワープ内で条件分岐が発生した場合, 速度が低下するので, ワープ内で条件分岐が発生しないよう, 工夫する必要がある.

5 実装

今回実装したパスワードクラックの基本的な実行フローは以下の通りとした.

1. 入手済である 1 組の平文と暗号文のセットとその他暗号化に必要なデータを GPU に転送
2. 暗号化に使用する鍵を GPU 内部で生成
3. 平文を暗号化
4. 鍵判定処理を行う

† 防衛大学校 情報工学科

5. 完成した暗号文と入手済みの平文が一致した場合に限り, 用いた鍵のみをリードバック

どのデータを 1 スレッドで処理させるかは, スレッドによる並列処理の効率を決定するために重要である. DES では 64 ビットの平文に対しビット単位で暗号化/復号を行うため, 本来は平文を 64 ビット領域に格納し, 1 ビット 1 スレッドで処理させたい. しかし, 1 ビット 1 スレッドだとメモリアクセスの際に並列処理できない. そのため, 各ビットは 1 バイト領域に格納し, 1 スレッドで 1 バイト領域に格納されたビットを処理させる. AES では 128 ビットの平文に対しバイト単位で暗号化/復号を行うため, 平文はそのまま 128 ビット領域に格納し, 1 バイト 1 スレッドで処理させた.

SP のパイプラインレイテンシを隠蔽するには最低 6 ワープ起動するのが良いとされているため [3], 共有メモリの容量を考慮して, スレッドブロック 1 つ当たりのスレッド数は 512 に決定した. 同条件だとスレッドブロックが 2 つ起動し, MP の最大アクティブスレッド数 1024 が保証され, MP を最大効率で使用できる.

GPU で鍵を内部生成した理由は, CUDA の仕様では 1 度のジョブ発行で 2^{35} 個の鍵を生成して同鍵空間をチェックできるのに対し, メモリには 2^{23} 個程度しか蓄積できないためであり, また CPU〜GPU 間のデータ通信を極力低減させたいためでもある. 鍵生成はスレッド並列で高速処理するよう工夫し, 全体の計算時間に対し鍵生成が占める割合は DES で実験したところ 4%〜35%から 0.1%〜1.2%程度となり, GPU 内部での鍵生成によるオーバーヘッドが極力回避できていることがわかる.

GPU 内部での鍵判定処理も鍵生成と同じ理由から GPU 内部で処理した. 鍵判定部分の処理時間は全体の実行時間と比較して DES では 0.1%〜2.5%程度と, GPU 内部での判定処理が負担にならないことがわかった.

CUDA ではワープ (=32 スレッド) を 1 単位として処理を行う. しかし, DES や AES では, 56 ビットの転置等, ワープ内での条件分岐を発生させる処理が存在する. そのため, 56 ビットの転置表であれば 64 ビットの共有メモリ領域に多少冗長に格納し, 条件分岐を避けるよう工夫した.

6 性能評価

CUDA による検索速度を評価するために CPU による実装と比較する. 用いたマシンの仕様を表 1 に, それぞれ実装したプログラムの実行時間のグラフを図 2 に示す. グラフの横軸は鍵空間, 縦軸は計算時間を対数で表している. ここでは DES のみ示す. 2^{35} の鍵空間に対し, GPU では約 7.45 時間, CPU では約 67.8 時間と, 約 9.1 倍の速度向上を示した. また, 我々は, 2009 年 3 月に行われたマルチプログラミングコンテスト GPU Challenge 2009 で AES を実装し Tesla-C1060 上で 2^{14} 個の平文ブロックに対し 2.933ms の処理時間を達成している. 従って, 2^{35} 個の平文を AES 処理するのにかかる時間はおよそ 6150s であることが予想できる. 一方, 2^{35} 個の鍵空間の場合の暗号処理を除いた時間は 1073s であった. 従って, 暗号化モジュールを AES に載せ換えた場合には, 2^{35} 個の平文に対し, およそ 1.9 時間でキークラックできると見積もられる.

表 1: Spec of computer

CPU	Corei7 Quadi7-920(2.66GHz)
Memory	6GB
OS	CentOS5.2 (カーネル ver2.6.18)
Compilier	gcc ver4.1.2(option -O3)
GPU Accelerator	NVIDIA Geforce GTX 285
GPU Memory	1GB
CUDA Complier	nvcc ver2.1

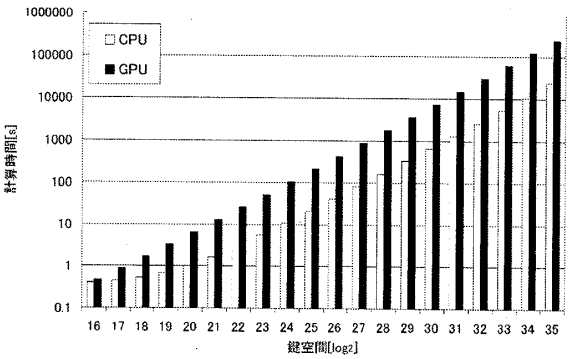

図 2: Comparison of Calculation Time

7 結論

本稿では, GPGPU によるパスワードクラックの可能性の検討を行った. 2^{35} 乗個の鍵空間に対し, DES では Geforce GTX 285 を用いて約 7.45 時間, AES では Tesla-C1060 を用いて約 1.9 時間で探索可能であることを示した. また, DES の場合, Intel Core i7-920 2.66GHz に対し, 約 9 倍の速度向上を示した.

本研究の発展としては, 複数の GPU を使用してパフォーマンスを向上させることが考えられる. 渡辺らは, データ転送に対して計算時間の割合が高い場合には, 複数 GPU を用いて性能を発揮しやすいという結論を示している [4]. 本稿による実装はこの条件に合致しているため, 複数 GPU で期待通りの性能が引き出せるアプリケーションであると考えられる.

参考文献

[1] Svetlin A.Manavski, "CUDA COMPATIBLE GPU AS AN EFFICIENT HARDWARE ACCELERATOR FOR AES CRYPTOGRAPHY", ICSPC (2007).

[2] Sandeep Kumar, Christof Paar, Jan Pelzl, Gerd Pfeiffer, and Manfred Schimmler. "Breaking Ciphers with COPACOBANA - A Cost-Optimized Parallel Code Breaker", CHES 2006, pp.101-118 (2006).

[3] Vasily Volkov, James W.Demmel, "Benchmarking GPUs to Tune Dense Linear Algebra", SC'08 (2008).

[4] 渡辺 裕也, 遠藤 敏夫, 松岡 聡, "複数 GPU におけるセルフスケジューリングによる並列数値演算", 情報処理学会 ARC 研究報告, Vol.2008 No.75, pp85-90 (2008).