SecHack365

iGENC

Internal GPU encrypt. by 学習駆動コース 坂井ゼミ 竹田 大将

~ SoC内GPUを用いたユーザエクスペリエンスを低下させない暗号処理機構 ~

暗号は現在の情報社会では必要不可欠，でも暗号処理って**重い**ですよね？

大容量ファイル・ディスク全体の暗号化は 計算負荷が高い！ → **CPUリソースを占有し，全体のパフォーマンスが低下する**

CPUパフォーマンスを低下させずに高負荷な暗号処理を実現したい...

既存の手法は汎用的じゃないからラップトップPCなどへの適用は難しい.

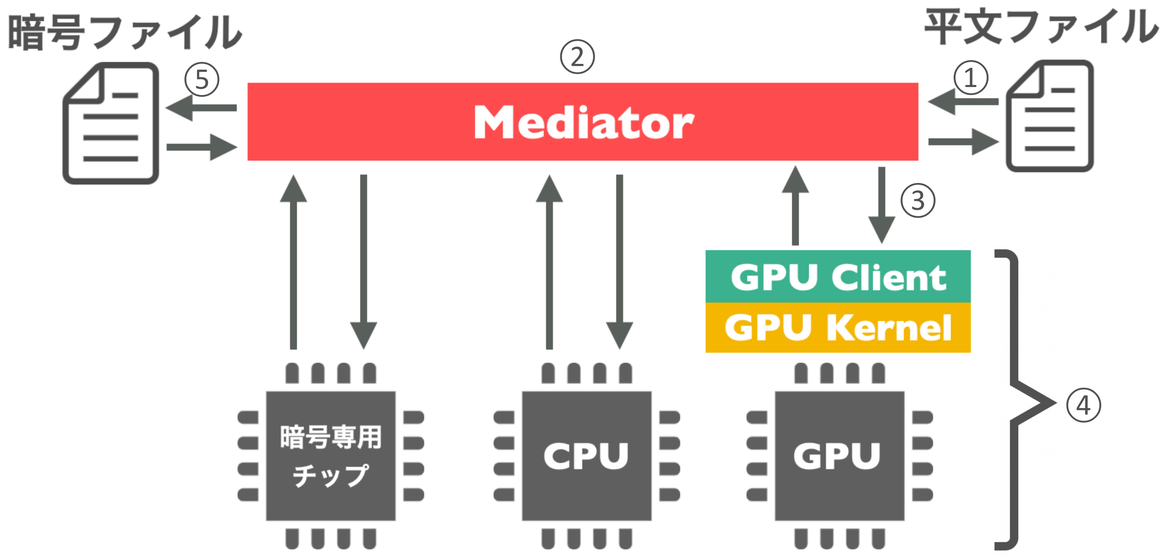
手法1: Intel AES-NIなどの暗号専用回路を使う手法
→ CPUの拡張機能であり，使用すればCPUは占有される

手法2: GPUに暗号処理をさせる手法
→ 多くがNVIDIA製などの外付けGPUでの実装前提である

現在普及しているラップトップPC，シングルボードコンピュータに適用できない(専用回路の実装やグラフィックボードの搭載が困難)

暗号処理を専用回路や追加ボードを使用せずに実現する手法が望まれる

SoC内GPUを始め，内蔵のリソースを無駄なく使う**iGencを提案！！**



① writeシステムコールをフックして，保存する平文ファイルを横取り
② Mediatorが状況に合わせてタスク割り当て先(CPU or GPU or ...)を決定
③ プロセス間通信により対象リソースへタスクを送る
④ GPU Clientがタスク受理 → GPU Kernelで暗号計算(今回はAES-128 CTR)
⑤ 暗号処理結果はMediatorを介してストレージに保存される

アイデア1: 内蔵GPUがあるじゃないか！
現在普及しているコンピュータ搭載SoCの多くはGPU内蔵.
しかも，外付けGPUがある環境なら空きリソースの場合が多い.
→ 内蔵GPUに暗号処理させることでCPUリソースの占有を防ぐ

アイデア2: MediatorがいるからUXを低下させない！
ゲームをプレイするから内蔵GPUの負荷を下げたいとき，
暗号専用回路を使って高速に暗号化したいときだってある.
→ 状況に合わせてMediatorがタスクを調停する

アイデア3: システムコールをフックすることで楽々導入！
既存システム/アプリの改造や，専用プログラムを書くのは面倒...
→ システムコールをフックして，勝手にファイルを暗号化する
ユーザが面倒な作業をする必要は無し

Mediator	大量に発行される暗号タスクの調停を行う
GPU Client	GPUへのタスクを待ち受ける常駐プログラム
GPU Kernel	暗号処理を行うGPUプログラム (この部分を差し替えることで他の暗号も使える)

iGencのここがすごい！！

CPUの負荷を軽減

CPUに高負荷をかけた状態で実行
暗号専用回路とGPUでスループット(Gbps)を比較

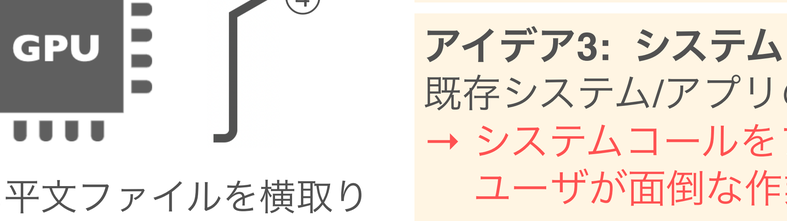
- Intel Core i3-8109U (2C/4T, 3.00 GHz)
- AES-128(CTR)：入力ファイルサイズ16MB
- CPU動作周波数固定

GPU実装はほとんど性能低下なし
→CPUを占有していない(依存してない)

	CPU負荷なし (Gbps)	CPU負荷あり (Gbps)	性能低下率 (%)
OpenSSL CTR + AES-NI	39.625	28.875	約27%
GPU実装	2.969	2.967	**約 0.0007%**

ゲーミングモード搭載

全ての暗号処理をGPUにスケジューリング
→ グラフィック性能が**約30% 低下**



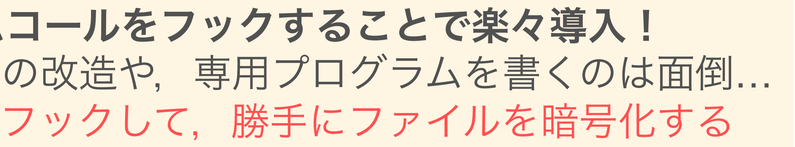
グラフィックを優先するゲーミングモード搭載

ゲーミングモードを実行すると...
→ GPU以外に暗号化タスクをスケジューリング
→ グラフィック性能が低下しない
→ ゲームをするときもUXが低下しない！

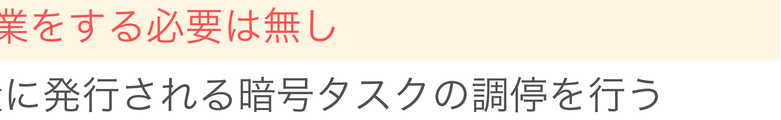
楽々暗号化

iGencを起動した状態で,
試しにVimでテキストを保存してみよう！

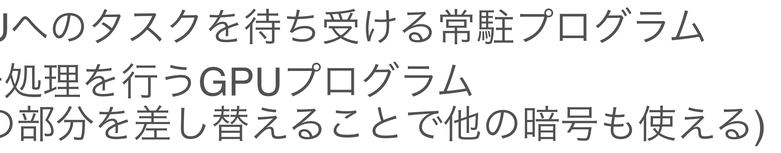
・保存するテキスト



・実際に保存されたファイル，読めない！



・AES複号プログラムで復号すると見れる！



普段使いのアプリでも,
ファイルを簡単にセキュアにできた！

今後の展望

サイズによるスケジューリング，ゲーミングモードの他にも，状況に合わせたスケジューリングの追加.
Intel GPU以外にも，より多くのSoC内GPUに対応.
AES以外の暗号やEdDSAなどの署名にも対応.
今後の最新情報は，GitHubやブログをチェック！ →

国立研究開発法人 情報通信研究機構

IWATE Low Layer Lovers' association.

http://t-takeda.work
GitHub, Blog, Twitter