# 論　文　概　要

九州工業大学大学院情報工学府　情報創成工学専攻

| 学生番号 | 14675033 | 氏　名 | 福田　直人 |
|---|---|---|---|
| 論文題目 | ページキャッシュ暗号化を用いた Android 端末の盗難対策 | | |

## 1　はじめに

Android 端末の普及に伴い，端末の盗難による情報漏洩のリスクが高まっている．ディスクからの情報漏洩はディスク暗号化により防ぐことができるが，近年，コールドブート攻撃によるメモリからの情報漏洩が問題となってきている．コールドブート攻撃は端末を冷却した状態でリセットすることで，起動後にメモリ上に保持されたデータを盗む攻撃手法である．Android OS はディスク上のデータをメモリ上にページキャッシュとして保持しているため，コールドブート攻撃が行われるとページキャッシュ上のデータを盗み見られてしまう．その結果，暗号化されているディスク上のデータの一部が漏洩してしまう危険がある．

本研究では，メモリ上のページキャッシュを暗号化することで Android 端末の盗難時に情報漏洩を防ぐためのシステム Cache-Crypt を提案する．

## 2　Cache-Crypt

Cache-Crypt はメモリ上に保持されるページキャッシュの内容を暗号化する．これを暗号化ページキャッシュと呼ぶ．Android アプリがアクセスする時だけ暗号化ページキャッシュの必要な領域を復号し，アクセスが終わったら再び暗号化する．

アプリがファイルを読み込む際には，図 1 の破線のように，Cache-Crypt が暗号化ページキャッシュを復号し，そのデータをアプリのバッファにコピーする．読み込もうとしたファイルに対応するページキャッシュが存在しない時は，暗号化ディスクからページキャッシュ上に暗号化されたデータを読み込む．一方，アプリがファイルにデータを書き込む際には，図 1 の実線のように，Cache-Crypt がアプリのバッファ上のデータを暗号化し，暗号化ページキャッシュに書き込む．書き換えられたページキャッシュの内容は定期的に暗号化ディスクに書き戻す．

再暗号化のオーバヘッドを削減するために，一旦，復号したページキャッシュはしばらく復号したままにして，暗号化の遅延を行う．そして一定時間アクセスがなければページキャッシュを再び暗号化する．これにより，ファイルの読み書きの際には必要に応じてページキャッシュの復号のみを行えばよくなり，性能を向上させることができる．

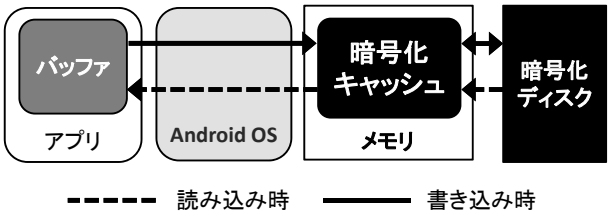


図 1: ファイルの読み書きの流れ

アプリがファイルをメモリマップした時には，ページキャッシュを復号したままにする．メモリマップにより，ページキャッシュがアプリのメモリにマップされ，アプリが OS を介さずに直接ページキャッシュにアクセスできるようになるためである．そして，ファイルのアンマップ時に再びページキャッシュを暗号化する．

ページキャッシュの暗号化を行うための暗号鍵は既存研究の ARMORED [1] を用いて保護する．ARMORED では，CPU レジスタに暗号鍵を保持することによって，コールドブート攻撃を受けても暗号鍵がメモリから漏洩するのを防ぐことができる．

## 3　実験

Cache-Crypt によるオーバヘッドを調べる実験を Nexus 7 (2013) を用いて行った．この実験では Benchmark というアプリを用いてベンチマークを実行し，Android の標準カーネルと Cache-Crypt を実装したカーネルとで比較を行った．Cache-Crypt を用いることによる読み込みのオーバヘッドは暗号化アルゴリズムに AES を用いた場合で 28%程度であった．

## 4　まとめ

本研究では，メモリ上のページキャッシュを暗号化することでコールドブート攻撃による情報漏洩を防ぐシステム Cache-Crypt を提案した．今後の課題はフルディスク暗号化との連携を行い，オーバヘッドをさらに削減することである．

## 参考文献

[1] J. Götzfried and T. Müller, ARMORED: CPU-bound Encryption for Android-driven ARM Devices, In *P*roc. ARES'13.