UNIVERSITI PUTRA MALAYSIA

AN EMBEDDED DATABASE DESIGN AND IMPLEMENTATION OF A PARALLEL IEEE XTS STORAGE ENCRYPTION FOR MOBILE DEVICES

MOHAMMAD AHMED MOHAMMAD ALOMARI

FK 2018 8

AN EMBEDDED DATABASE DESIGN AND IMPLEMENTATION OF A PARALLEL IEEE XTS STORAGE ENCRYPTION FOR MOBILE DEVICES

By

MOHAMMAD AHMED MOHAMMAD ALOMARI

Thesis Submitted to the School of Graduate Studies, Universiti Putra Malaysia, in Fulfilment of the Requirement for the Degree of Doctor of Philosophy

November 2017

DEDICATION

I dedicate this PhD thesis

To

My beloved parents,

My great wife,

and

My respected teachers,

Whose love, affection, encouragement, patience, and prayers make me able to achieve such success.

Abstract of thesis presented to the Senate of Universiti Putra Malaysia in fulfilment of the requirement for the degree of Doctor of Philosophy

AN EMBEDDED DATABASE DESIGN AND IMPLEMENTATION OF A PARALLEL IEEE XTS STORAGE ENCRYPTION FOR MOBILE DEVICES

By

MOHAMMAD AHMED MOHAMMAD ALOMARI

November 2017

Chairman : Khairulmizam Samsudin, PhD
Faculty : Engineering

The ubiquity and huge proliferation of mobile and handheld devices, such as smartphones and tablets, are globally undeniable where Google's Android operating system dominates the largest share of mobile platforms in the market. The vast spread and increased capabilities of these devices have come with major challenges to mobile security and data confidentiality. Every year different threats against sensitive data resting inside the storage of these mobile devices continue to rise sharply. Encryption might be the most efficient technique to ensure storage confidentiality; however it comes with great impact on these small gadgets which suffer from lack of resources such as processing power and battery. Performance is also a major concern for implementing security solutions, such as full storage encryption, inside mobile devices. A security solution might not be welcomed by consumers if it causes tangible performance degradation. With the wide spread of multi-core processors in current smart gadget devices, parallelization is no more luxury and can be used to enhance encryption performance in mobile gadgets significantly.

This study focuses on evaluating and enhancing the performance of data storage encryption inside mobile devices. In this thesis, a parallel encryption system for the protection of sensitive data stored inside Android-based mobile devices is developed and successfully implemented. To ensure higher security level, the developed system is implemented using the NIST-certified XTS-AES block encryption algorithm. Other storage encryption algorithms, i.e. XTS-Twofish and XTS-RC6, have also been implemented in both serial and parallel designs and then evaluated. Overheads occurring due to parallel implementations have been identified and successfully mitigated to achieve proper performance speedup. Since

the most user sensitive data are residing inside persisting databases, an SQLite implementation of the parallel XTS-AES system is proposed. This developed parallel SQLite-XTS system encrypts data stored in databases transparently on-the-fly without the need for any user intervention. To design the parallel computation side of the proposed system and improve the overall system performance, a specific version of OpenMP API is integrated inside the architecture of targeted Android platform. This allows the developed encryption system to exploit the multi-core commodity processors, equipped with current mobile devices, in order to enhance performance. Different serial and parallel experiments have been conducted on an Android testbed device, where performance analysis and comparisons of different SQLite implementations have been carried out.

During the file-based experiments, the parallel XTS-AES has shown a performance speedup of 1.71 with 86% efficiency faster than its serial counterpart; with higher encryption throughput achieved in the testbed device up to 8290 KB/s and 11380 KB/s when using XTS-AES and XTS-RC6 ciphers respectively. Additionally, the developed parallel SQLite-XTS system have been successfully implemented and integrated into the mobile testbed device. To assess the performance and feasibility of this system, it has been compared with three other SQLite implementations, i.e. Plain SQLite, Serial XTS SQLite, and SQLCipher-CBC. Results show that the developed parallel SQLite system has reduced the overhead of database encryption from 30.8%, with serial implementation, up to 17.8% when parallel SQLite is used. That provides the developed system with an efficiency of 73% compared to serial counterpart. These results clarify that the developed SQLite system introduces significant performance improvement compared to other implementations.

Abstrak tesis yang dikemukakan kepada Senat Universiti Putra Malaysia sebagai memenuhi keperluan untuk ijazah Doktor Falsafah

REKA BENTUK DAN PELAKSANAAN PANGKALAN DATA TERBENAM BAGI PENYULITAN STORAN IEEE XTS SELARI UNTUK PERANTI MUDAH ALIH

Oleh

MOHAMMAD AHMED MOHAMMAD ALOMARI

November 2017

Pengerusi : Khairulmizam Samsudin, PhD
Fakulti : Kejuruteraan

Kewujudan dan penggunaan yang meluas bagi peranti tangan dan mudah alih, seperti telefon pintar dan tablet, tidak dapat dinafikan lagi di mana sistem operasi Google Android mendominasi sebahagian besar pasaran platform mudah alih. Perkembangan dan peningkatan keupayaan peranti-peranti ini telah datang dengan beberapa cabaranutama kepada keselamatan dan kerahsiaan data. Saban tahun, pelbagai ancaman terhadap data peribadi yang tersimpan di dalam peranti mudah alih terus mencatatkan peningkatan yang ketara. Penyulitan data mungkin kaedah yang paling berkesan bagi memastikan kerahsiaan storan; namun ia memberi impak yang besar terhadap gajet-gajet kecil yang mempunyai sumber kuasa pemprosesan dan bateri yang terhad. Prestasi juga merupakan satu kebimbangandalam pelaksanaan penyelesaian keselamatan, seperti penyulitan storan penuh, di dalam peranti mudah alih. Sesuatu penyelesaian keselamatan mungkin tidak dapat diterima oleh pengguna jika ia menyebabkan kemerosotan prestasi yang ketara. Dengan penggunaan pemproses multi-teras yang meluas pada peranti gajet pintar semasa, keselarian (parallelization) bukanlah sesuatu yang asing dan ia boleh digunakan untuk meningkatkan prestasi gajet mudah alih dengan ketara.

Kajian ini bertumpu kepada penilaian dan peningkatan prestasi penyulitan storan data peranti mudah alih. Dalam tesis ini, sistem penyulitan selari bagi melindungi data sensitif didalam peranti mudah alih berasaskan Android telah dibangunkan dan berjaya dilaksanakan. Bagi memastikan tahap keselamatan yang lebih tinggi, pelaksanaan sistem yang dibangunkan itu dilakukan menggunakan algoritmapenyulitan blok XTS-AES yang disahkan oleh NIST. Lain-lain algoritma bagi penyulitan storan,seperti XTS-Twofish and XTC-RC6, juga telah dilaksanakan

dalam kedua-dua rekabentuk siri dan selari yang kemudiannya dinilai. Overhed yang berlaku disebabkan oleh pelaksanaan selari telah dikenalpasti dan berjaya dikurangkan untuk mencapai kelajuan prestasi yang sepatutnya. Oleh kerana kebanyakan data sensitif pengguna tersimpan di dalam pangkalan data kekal, pelaksaan SQLite bagi system XTS-AES selari telah dicadangkan. Sistem ini menyulitkan data yang tersimpan di dalam pangkalan data secara telus dan terus tanpa memerlukan sebarang campur tangan pengguna. Untuk mereka bentuk bahagian pengkomputeran selari bagi sistem yang dicadangkan dan meningkatkan pretasi sistem secara keseluruhan, versi khusus OpenMP telah disepadukan dalam senibina platform Android yang digunakan. Ini membolehkan sistem ini mengeksploitasi pemproseskomoditi multi-teras yang dilengkapi dengan peranti mudah alih semasa, bagi meningkatkan prestasi. Eksperimen siri dan selari yang berbeza telah dijalankan pada peranti tapak uji Android, yang mana analisis prestasi dan perbandingan antara pelaksanaan SQLite yang berbeza telah dijalankan.

Bagi eksperimen berasaskan fail, XTS-AES selari telah menunjukkan prestasi peningkatan kelajuan sebanyak 1.71 dengan 86% keberkesanan, lebih laju daripada sistem siri; dengan data penyulitan yang lebih tinggi dicapai dengan peranti tapak uji sehingga 8290 KB/s and 11380 KB/s apabila menggunakan pengekodXTS-AES dan XTS-RC6. Di samping itu, sistem SQLite-XTS selari telah berjaya dilaksanakan dan disepadukan dalam peranti tapak uji mudah alih. Bagi menilai prestasi dan kebolehlaksanaan sistem ini, ia telah dibandingkan dengan tiga SQLite yang lain, iaitu Plain SQLite, Serial XTS-SQLite dan SQLCCipher-CBC. Keputusan kajian menunjukkan bahawa sistem SQLite selari yang dibangunkan telah mengurangkan overhed penyulitan pangkalan data dari 30.8%, dengan pelaksanaan siri, sehingga 17.8% apabila SQLite selari digunakan. Ini menunjukkan bahawa sistem ini dilengkapi dengan kecekapan 73% berbanding dengan sistem siri. Keputusan kajian ini menjelaskan bahawa sistem SQLite yang dibangunkan telah meningkatkan prestasi denganketara berbanding dengan pelaksanaan sistem lain.

ACKNOWLEDGEMENTS

First and foremost I would like to thank almighty Allah for all the strength, support, patience, and courage he has given to me to reach such point of completion.

I would like also to express my appreciation and deep gratitude to my supervisor Dr. Khairulmizam Samsudin for his insightful comments, motivation, patience, and immense knowledge. I appreciate all his contributions of time, ideas to make this PhD thesis productive and fruitful. Without his tremendous mentoring efforts, encouragement, and guidance, the completion of this work would not have been possible. The enthusiasm and willingness he has for his research was contagious and motivational to me, even during the tough times of this PhD pursuit.

My sincerest thanks go also to members of the supervisory committee Dr. Abdul Rahman Ramli, and Dr. Shaiful Jahari Hashim, who provided me with precious comments and support to complete this thesis. The committee members have been a good source of friendship and advice.

Additionally, I would like to thank the lab staff of Computer and Communication Systems Department in Engineering Faculty, for their continuous help and cooperation to achieve lab experimental works.

Finally, I would like to thank my family for their continuous love and encouragements. For my parents, the words can not express how grateful I am for all their sacrifices and prayers.

And most of all for my encouraging, supportive, loving, and patient wife whose faithful support during the tough times of this PhD is so appreciated.

I also express my regards to all of those who helped me in any respect for the completion of this work.

I certify that a Thesis Examination Committee has met on 14 November 2017 to conduct the final examination of Mohammad Ahmed Mohammad Alomari on his thesis entitled "An Embedded Database Design and Implementation of a Parallel IEEE XTS Storage Encryption for Mobile Devices" in accordance with the Universities and University Colleges Act 1971 and the Constitution of the Universiti Putra Malaysia [P.U.(A) 106] 15 March 1998. The Committee recommends that the student be awarded the Doctor of Philosophy.

Members of the Thesis Examination Committee were as follows:

Aduwati binti Sali, PhD
Associate Professor
Faculty of Engineering
Universiti Putra Malaysia
(Chairman)

M. Iqbal bin Saripan, PhD
Professor
Faculty of Engineering
Universiti Putra Malaysia
(Internal Examiner)

Mohammad Hamiruce Marhaban, PhD
Professor
Faculty of Engineering
Universiti Putra Malaysia
(Internal Examiner)

Nathan Luke Clarke, PhD
Professor
Plymouth University
United Kingdom
(External Examiner)

NOR AINI AB. SHUKOR, PhD
Professor and Deputy Dean
School of Graduate Studies
Universiti Putra Malaysia

Date: 28 December 2017

This thesis was submitted to the Senate of Universiti Putra Malaysia and has been accepted as fulfilment of the requirement for the degree of Doctor of Philosophy. The members of the Supervisory Committee were as follows:

Khairulmizam Samsudin, PhD
Senior Lecturer
Faculty of Engineering
Universiti Putra Malaysia
(Chairman)

Abdul Rahman Ramli, PhD
Associate Professor
Faculty of Engineering
Universiti Putra Malaysia
(Member)

Shaiful Jahari Hashim, PhD
Associate Professor
Faculty of Engineering
Universiti Putra Malaysia
(Member)

ROBIAH BINTI YUNUS, PhD
Professor and Dean
School of Graduate Studies
Universiti Putra Malaysia

Date:

Declaration by graduate student

I hereby confirm that:

- this thesis is my original work;
- quotations, illustrations and citations have been duly referenced;
- this thesis has not been submitted previously or concurrently for any other degree at any other institutions;
- intellectual property from the thesis and copyright of thesis are fully-owned by Universiti Putra Malaysia, as according to the Universiti Putra Malaysia (Research) Rules 2012;
- written permission must be obtained from supervisor and the office of Deputy Vice-Chancellor (Research and Innovation) before thesis is published (in the form of written, printed or in electronic form) including books, journals, modules, proceedings, popular writings, seminar papers, manuscripts, posters, reports, lecture notes, learning modules or any other materials as stated in the Universiti Putra Malaysia (Research) Rules 2012;
- there is no plagiarism or data falsification/fabrication in the thesis, and scholarly integrity is upheld as according to the Universiti Putra Malaysia (Graduate Studies) Rules 2003 (Revision 2012-2013) and the Universiti Putra Malaysia (Research) Rules 2012. The thesis has undergone plagiarism detection software.

Signature: _____ Date: _____

Name and Matric No.: Mohammad Ahmed Mohammad Alomari (GS26874)

Declaration by Members of Supervisory Committee

This is to confirm that:

- the research conducted and the writing of this thesis was under our supervision;
- supervision responsibilities as stated in the Universiti Putra Malaysia (Graduate Studies) Rules 2003 (Revision 2012-2013) are adhered to.

Signature: _____

Name of Chairman of Supervisory Committee: Dr. Khairulmizam Samsudin

Signature: _____

Name of Member of Supervisory Committee: Associate Professor Dr. Abdul Rahman Ramli

Signature: _____

Name of Member of Supervisory Committee: Associate Professor Dr. Shaiful Jahari Hashim

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

FDE	Full Disk Encryption
API	Application Programming Interface
XTS	XEX encryption mode with Tweak and ciphertext Stealing
SQLite-XTS	SQLite with XTS encryption
SQLCipher-CBC	SQLCipher encryption with CBC mode
ICS	Ice Cream Sandwich
CPU	Central Processing Unit
GPU	Graphics Processing Unit
IDS	Intrusion Detection System
DoS	Denial of Service
TPM	Trusted Platform Module
ASIC	Application-Specific Integrated Circuit
SISWG	Security in Storage Working Group
XTS	XEX encryption mode with Tweak and ciphertext Stealing
XEX	Xor-Encrypt-Xor
NIST	National Institute of Science and Technology
CBC	Cipher Block Chaining
AES	Advanced Encryption Standard
RDBMS	Relational Database Management System
ACID	Atomic, Consistent, Isolated, and Durable
SQL	Structured Query Language
VFS	Virtual File System
VDBE	Virtual Database Engine

SEE	SQLite Encryption Extension
PII	Personally Identifiable Information
eMMC	Embedded Multi-Media Card
SDK	Android Software Development Kit
SMP	Symmetric Multiprocessing
MPI	Message Passing Interface
GCC	GNU Compiler Collection
ADB	Android Debug Bridge
ADT	Android Developer Tools
DDMS	Dalvik Debug Monitoring Service

CHAPTER 1

INTRODUCTION

1.1 Background

During the recent years, mobile and handheld devices, such as smartphones and tablets, have spread out in the market pervasively. This huge proliferation of high-end mobile devices has introduced new threats to sensitive data that are stored inside these devices. Due to the small size, mobile devices are prone to be temporarily forgotten, completely lost or even stolen which can risk the huge amount of private data residing inside them. Besides having personal sensitive information, current mobile devices can also hold employee organizational data which can cause huge losses in case of intentionally stolen or even misplaced [1]. A survey carried out at the Infosecurity Europe Show claims that 41% of IT professionals are using mobile data without encryption or any other forms of protection [2]. An IEEE 2015 report shows that although 6 billion out of 7 billion people use mobile devices or tablets for shopping, banking, and posting on social media, protecting the transferred data is still rarely addressed [3].

Current mobile devices depend mainly for their security on user passwords which can be breached easily, especially with the newly emerging high-tech attacks. To ensure better confidentiality for the data stored in these devices, encrypting the complete storage area needs to be introduced and heavily investigated. While full storage encryption provides more security to handheld devices, it also reduces the users' burden when dealing with file encryption in which specific files can be encrypted selectively. On the other hand, the complete storage area is transparently encrypted, using full storage encryption, without any need for user intervention [4]. Although full encryption of stored data is a vital necessity to warrant confidentiality in desktop and mobile devices, it is involved with different challenges. Encrypting large amount of data can impose great processing load on the CPUs of these devices which already suffer, mainly mobile devices, from lack of resources such as processing power, memory and battery. Using general purpose single core CPUs for encrypting full disks of data can greatly impact the overall performance of any system [5]. The use of dedicated chips (ASIC) can introduce an alternative solution but it comes with more cost and less flexibility to reprogram or update these chips. With the wide spread of multicore processors in current mobile devices, speeding up storage encryption using parallelization is possible. Parallelization is no more luxury and can enhance the performance significantly. The increase of performance comes with a saving in cost due to the current availability of multicore technologies [6].

Introducing multicore processors to mobile gadget devices has brought a new era to improve device performance and capabilities without the need to increase CPU clock rates which causes more heat dissipation. In desktops, an AMD chip can

consume 60% more power with every 400MHz rise in clock rate which may cause unacceptable levels of heat inside the chip when increasing the clock rate to high limits [7]. Multicore technology allows higher performance with less energy consumption which is an important requirement for mobile devices that suffer from small size limitations and fast battery drainage. Multicore also enhances user experience and control in multitasking environments where different heavyweight applications can run simultaneously such as encryption, virus protection, and compression. The performance gained by the parallel processing using multicore depends mainly on how much is the size of the parallelizable part in the implemented application. This means the parallelization will be very efficient only if the executed application can be divided into different modules which can be processed separately and concurrently by different processor cores.

As a storage encryption algorithm, different algorithms have been introduced to meet the requirements. In this work, XTS-AES [8] has been chosen to be implemented in order to enforce the confidentiality of data stored in mobile devices while preventing from known attacks against previous cipher modes such as CBC. The P1619 XTS-AES is an IEEE standard for data protection on narrow-block storage devices. It integrates XTS mode of operation with the well known AES encryption algorithm to provide a solid standard for storage protection. XTS-AES is developed by the IEEE Security in Storage Working Group (SISWG) in 2007, and approved by the National Institute of Standard and Technology (NIST) in 2010 in an attempt to protect block-oriented storage devices. XTS-AES addresses different types of attacks such as copy-and-paste attacks that may lead to data leakage. One main advantage of XTS-AES algorithm is that it is fully parallelizable which can greatly help to speedup data encryption process [8]. This is not true for some other encryption modes of operation like CBC, OFB and CFB when compared to the parallelizable XTS mode.

The operating systems for mobile devices have evolved dramatically in recent years. This improvement ranges from attractive GUI to processing capabilities and PC-like services. This was mainly driven by the huge advances in mobile devices hardware, such as touch screens and high processing speeds, as well as great consumer demands for smarter devices. To cope with mobile devices needs, such as limited processing and memory as well as small display size, different operating systems have been specifically tailored for that purpose. The most common mobile platforms are Nokia Symbian, Apple's iOS, BlackBerry OS, Windows mobile, and Google's Android [9, 10]. Some of these mobile platforms do not have an efficient way to protect stored data, such as full storage encryption, which should be made as an integral part of the operating system to guarantee high level of protection while maintaining proper device performance. Alternatively, they rely on complementary applications to provide the required level of security for data inside the device. These applications are usually not efficient enough to protect against modern attacks which are able to retrieve user data. Attacks through viruses and malware are spreading to mobile devices from different sources such as Wi-Fi, Internet, cellular networks, Bluetooth, and others [11].

Google's Android is a relatively new mobile operating system. It is generally a complete framework that is developed by Open Handset Alliance. Android is an open source, programmable OS where its source code is available for community developers to edit and enhance [12]. The latest version of Android platform, specifically Android 6.0, mandates the use of full disk encryption (FDE) feature in its devices. However this comes with different complaints that full disk encryption heavily affect device performance. The security framework and content protections of this platform are based on type of permissions given to an application rather than tying access controls to the data handled by these applications. This may allow a malicious application with suitable permission to breach the most sensitive data on device. Moreover, the feature of Android as a multi-environment open source operating system can be used against it where Android code is available for attackers to study and manipulate. These different security and performance issues in mobile devices introduce great need to address data storage confidentiality from inside the platform itself which can significantly enhance the confidentiality of such devices while maintaining better performance. Exploiting the spread of commodity multi-core mobile devices can be an interesting area to address performance issues that occur due to introducing security mechanisms such as encryption [9, 13].

To manage the structured data of applications in its mobile devices, Android as well as many other mobile platforms have adopted SQLite [14] database system. Android uses SQLite database as the main medium to store structured data so that it can be easily accessed, queried, and modified. SQLite is an open source relational database software library that is known to be serverless, self-contained, zero-configuration, and embeddable engine used in many small scale systems. The wide deployment of SQLite is due to its many features such as the small memory footprint, high storage efficiency, and fast query operation. Another advantage of SQLite is that its complete conformance with the well known transactional Structured Query Language (SQL) standard. On the other hand, the heavy use of SQLite databases can be slow due to the continuous need for I/O access to non-volatile memory (i.e. disk storage). SQLite is also used with operating systems of other mobile devices, such as Apple's iOS, Nokia's Symbian, and recently Blackberry 10 OS, in order to manage their application data [15, 16].

The data of each Android application is stored in one or more SQLite database files that can be accessed only by that application. Additionally, data stored in SQLite files do not have any content protection mechanism except that the access control permissions of a database file are locked to its specific application. That makes the confidentiality of SQLite data easy to access and can be breached if an intruder gets access to the database files by one way or another. Other measures of data protection, such as encryption, for SQLite files are not provided with the original SQLite library. However encryption support is provided to SQLite through external extensions such as SQLite Encryption extension (SEE) [17], SQLCipher [18], and wxSQLite [19]. Many of these extensions suffer from performance issues inside mobile devices due to the heavy impact of encryption which in some cases may render the device useless [20, 21].

1.2 Research Motivation and Problem Statement

In recent years, the pervasive use of embedded and handheld devices globally has given rise to serious issues and challenges related to protecting these data-rich devices. One of the most important challenges that threaten these devices is how to maintain the confidentiality and privacy of user's sensitive data residing inside these devices. The storage size of newer devices can hold large amount of data that can be personal, corporate, or governmental data. Due to their small size, these portable devices are prone to be lost or stolen easily causing their data to be breached or tampered with through an adversary who might has physical access to device contents. This can cause severe consequences and huge organizational loss since large number of consumers uses their handheld devices for both personal and corporate purposes. Among many smartphone platforms, Android is the most common mobile platform globally. The vast proliferation and openness of source code make Android platform more vulnerable to different attacks including data security breaches.

The security of different current mobile devices, including Android platform, depends mainly on simple user passwords which have been proven to be weak enough especially with the latest attack models that benefit from advanced technology [4, 22]. As an alternative, encryption of the storage area in smart gadget devices can provide a far trusted solution to ensure confidentiality of stored data. In recent years, many mobile platforms, such as Google's Android, have introduced the feature of full disk encryption (FDE) to protect data stored in their devices. FDE is an encryption mechanism that encrypts the whole contents of the mobile device which may include encrypting the operating system itself [23, 24]. However, since mobile and embedded devices suffer from lack of resources such as processing power and memory, it is more appropriate that their encryption mechanisms do not involve heavy-weight computational operations which can greatly impact their performance and drain battery [25]. Another drawback of FDE is that these heavy encryption-decryption operations need to take place every time you need to access a non-confidential file or switch on and off your device. These limitations validate that encrypting the most important and sensitive data residing in databases is more practical than encrypting the whole storage contents.

SQLite database management system has been adopted by most mobile platform to store and maintain their application and system databases. This system manages databases without providing any protection mechanisms for data stored. If an adversary can have access to a database file, he can easily retrieve database contents using any simple text editor or database viewing tool. Providing a security mechanism, such as encryption, for SQLite database can greatly enhance data confidentiality of mobile devices as long as that does not affect the overall device performance. XTS-AES encryption algorithm is mainly developed by IEEE for the protection of storage data. Incorporating multicore processors with new smartphones and handheld devices brought a new era to enhance data confidentiality while maintaining suitable system performance. Taking advantage of

multicore technology equipped with current mobile devices can help improve device performance, if a proper parallel system is built for that purpose.

The goal of this research is to develop a parallel encryption system for the protection of sensitive data stored inside embedded and handheld devices. The developed system will implement the well known IEEE XTS-AES tweakable block encryption algorithm, which is developed for storage encryption, to protect data inside mobile devices. XTS-AES is known to be a secure cipher against different attacks such as copy-and-paste attack while allowing parallelization in its implementations [8]. A parallel design of XTS-AES algorithm is to be developed and implemented in this work. In recent years, Android became the most dominant platform for smartphone devices. Based on Canalys [26], Android has the highest growth rate of 615% from 2009(Q4) to 2010(Q4). The developed parallel XTS-AES system will be designed for Android-based mobile devices. Since the most important user data are in databases, the developed system is to be tailored more specifically to target the data stored in SQLite databases which resides inside the device persistent memory. The system will encrypt and decrypt data in SQLite transparently on-the-fly without the need for any user intervention. It is well known that Android SQLite is a light database management system that does not provide any security mechanism, such as encryption, for its stored data.

As encryption is a resource-intensive operation that can affect the performance of a mobile device, the developed system will implement a parallel XTS-AES design to make full use of commodity multicore processors equipped with current and future handheld devices. To achieve this target, a specifically tailored version of OpenMP will be integrated to Android architecture where it will used to implement the parallel design required for this system. Since multi-core programming in embedded and mobile systems is relatively new, many of these systems still does not have full support for parallel programming architectures (in both CPUs and GPUs). OpenMP is a multi-platform application programming interface (API) which is designed to be used for shared-memory parallel programming. Parallel encryption utilizing different cores in mobile device can maintain data confidentiality while enhancing system performance significantly. Introducing multicore processors, such as CPUs and GPUs, to mobile devices will continue to spread prevalently in the future, which makes it important to develop encryption systems that can exploit this feature in order to reduce system overhead and improve overall performance. During implementation part of this work, different testbed experiments will be conducted using a multicore testbed device to evaluate parallel gain in performance of the proposed system as compared to serial implementations.

1.3 Aims and Objectives

In general, the aim of this thesis is to design and implement an XTS parallel encryption system that can be used for the protection of data stored in Google's Android-based mobile devices. The developed system offers significant improve in

performance when compared to the current available storage encryption systems. Performance evaluations will be performed using an experimental testbed Android device.

In specific, the objectives of this study can be summarized as follows:

1. To design a parallel XTS data storage encryption system and integrate it into the architecture of Android-based mobile devices.
2. To implement a transparent, on-the-fly XTS-AES parallel encryption system tailored specifically for SQLite RDBMS of Android devices.
3. To evaluate, through testbed device experiments, the performance of the developed XTS encryption system and then compare it against different implementations.

1.4 Scope of Work

Security of data storage in mobile devices is a wide field with a vast range of research topics. In this study, it is first highlighted how the developed system generally works. Then this will be narrowed down to the main methodology, which focus on how to design and implement a file-based parallel XTS encryption system for the protection of persisting data resting inside Android mobile devices. The work is then narrowed further to give focus to the protection of data residing in databases (structured data) which are maintained by SQLite RDBMS. Other related topics such as Android architecture, OpenMP parallel API, SQLite architecture are clarified. On the other hand, security of the developed system is not covered since it is based on the proven security of XTS algorithm which is based on the security of the well known Rogaway's XEX (Xor-Encrypt-Xor) encryption algorithm [8, 27].

The work in this thesis focuses on the development and implementation of an XTS parallel encryption system that is designed to protect data-at-rest stored in Android-based mobile devices. Since encrypting the whole contents of a mobile device can incur performance issues and device resources overhead, the developed system is tailored to encrypt data inside SQLite databases. To improve the overall system performance, OpenMP parallel API (Application Programming Interface) is used to design the parallel side of the proposed system which allows the system to exploit the multicore processors equipped with current samrtphone and handheld devices. Therefore, the implementation stage includes testbed experiments on a multicore mobile device to test the developed system and measure gain in performance. Additionally, the main concern of this study is the security of data stored inside mobile devices (smartphones, tablets, etc) that is data resting in these devices. Hence it does not cover any aspect of data in communication between mobile devices. Finally, although the implementations in this work may be applicable to other mobile platforms, such as iOS and Nokia, with minor modification, they are developed specifically to target Google's Android platform.

1.5 Research Contribution

Since current and future handheld and embedded devices will be increasingly equipped with multicore processor technology (such as CPUs and GPUs), there is a vital necessity to direct more research to this area especially with the lack of resources in these small-size devices. This lack of resources comes with an urgent need to secure data inside these mobile devices without affecting their performance. The need for more research in parallel processing is obvious to mitigate the impact of heavy-weight operations, such as encryption which became necessary for the protection of mobile device from daily emerging threats.

The main contribution of this thesis is to design and implement a parallel encryption system, using the well known IEEE XTS-AES encryption algorithm, to ensure confidentiality of persisting data (data at rest) residing in mobile devices while maintaining suitable device performance. The system is incorporated in Android architecture, specifically to SQLite library, in such a way that it will transparently encrypt structured data stored inside SQLite databases. This is achieved in a user-friendly manner. To overcome any performance bottlenecks due to encryption operations, the system is designed to encrypt data in parallel fashion exploiting the newly introduced multicore feature with current mobile devices. This parallel design can significantly enhance performance of the developed system.

The following points clarify more thesis contributions:

- The integration and implementation of multi-core parallel design of XTS-AES in mobile devices can contribute to the protection of these limited-resource devices while enhancing their performance at same time.
- The developed SQLite-XTS encryption system process data on-the-fly where it encrypts database pages as they transfer to or from storage. This ensures no performance burden may occur on mobile device when encrypting the whole database in one time.
- The proposed system provides more user friendly way to ensure mobile security since the storage encryption of SQLite databases will be performed transparently without the need for user intervention.
- Testing the proposed system on a real testbed device (not only simulations), gives the system the opportunity to be commercialized or added to official distribution of Android in the future.
- Since the use of encryption for transparently protecting the storage data inside mobile devices is considered to be a new field, a great deal of research is required from both performance and security perspectives. This work might have achieved one step in that necessary research.
- As far as the author knowledge, there no parallel XTS-AES study to encrypt data stored in SQLite library of embedded devices have been reported in literature.
- While this work mainly targets mobile devices that use Android platform, it could be implemented in other SQLite-based mobile platforms, such as iOS and Nokia, with subtle modifications.

1.6 Outline of Thesis

This PhD thesis is organized into five chapters as follows. Chapter 1 introduces research background, motivation and problem statement, research objectives, and contributions. In Chapter 2, detailed literature of the whole work is provided. The threats and challenges facing data security in mobile devices are discussed and related research to that is explored. Different platform techniques to protect data through encryption are detailed. Since this work mainly target Android platform, the architecture of this platform as well as various issues related to data security are explored in this part of thesis. Mechanism used for data storage in Android are also discussed and compared to provide the pros and cons of each mechanism. The storage encryption algorithms and techniques are explained giving focus to the XTS-AES which remains the centre encryption algorithm of this study. Next is detailed literature about SQLite architecture and more specifically its use in Android. The security and data protection in SQLite is explored where different encryption mechanisms and extensions are explained and compared.

Chapter 3 reports on the design and implementation of the proposed XTS-AES parallel encryption system inside mobile devices, mainly Android-based devices. The stages to implement the proposed XTS-AES in Android in both serial and parallel design are introduced here. The parallel integration of XTS as well as other ciphers to Android architecture is discussed including the tools used to achieve that such as OpenMP API. Performance overheads that affect system speedup are elaborated and their elimination is discussed. The mechanisms to measure the performance are also provided. Experimental setup for both software and hardware sides are also provided in details. The design and implementation details of the proposed parallel XTS-AES system in SQLite are explained thoroughly.

In Chapter 4, the testbed experimental results obtained are presented. Findings in this work are classified into two sections: Device File-based evaluations of XTS storage encryption algorithms, and SQLite transparent XTS encryption system (SQLite-XTS). In File-based evaluations, performance comparisons (serial and Parallel) of different storage encryption algorithms, inside the mobile device, are presented and analyzed with respect to XTS. In SQLite-XTS section, the testbed experimental results obtained from four different SQLite implementations will be discussed and compared. The enhancements gained from the proposed parallel system will be compared to serial one; and then these results will be benchmarked and compared with current implementation of SQLite encryption. Finally, conclusions from the current study as well as recommendations for future research are presented and thoroughly discussed in Chapter 5 of this thesis.

REFERENCES

[1] B. Halpert, "Mobile Device Security," in *Proceedings of the 1st annual conference on Information security curriculum development* Kennesaw, Georgia: ACM, 2004.

[2] Infosecurity, "41% of IT professionals using mobile data without protection", http://www.infosecurity-magazine.com/view/17906/41-of-it-professionals-using-mobile-data-without-protection/, [Accessed Dec 2012].

[3] The Institute, "Mobile Devices Remain Vulnerable to Attacks, Special Report on Cybersecurity", IEEE, http://theinstitute.ieee.org/static/special-report-cybersecurity [Accessed April 2015].

[4] A. Pfitzmann, B. Pfitzmann, M. Schunter, and M. Waidner, "Trusting Mobile User Devices and Security Modules," *IEEE Computer journal*, vol. 30, pp. 61-68, 1997.

[5] M. A. Alomari, K. Samsudin, and A. R. Ramli, "A Study on Encryption Algorithms and Modes for Disk Encryption," in *International Conference on Computer Design and Applications (ICCDA 2009)*, Singapore, 2009, pp. 793-797.

[6] M. A. El-Fotouh and K. Diepold, "A New Narrow Block Mode of Operations for Disk Encryption," in *The Fourth International Conference on Information Assurance and Security*, 2008, pp. 126 - 131

[7] W. Knight, "Two heads are better than one [dual-core processors]," *IEE reviews*, vol. 51, pp. 32-35, 2005.

[8] IEEE, "IEEE Standard for Cryptographic Protection of Data on Block-Oriented Storage Devices," pp. c1-32, 2008.

[9] M. Ongtang, K. Butler, and P. McDaniel, "Porscha: Policy Oriented Secure Content Handling in Android," in *Proceedings of the 26th Annual Computer Security Applications Conference (ACSAC '10). ACM*, 2010, pp. 221-230.

[10] T. Müller, M. Spreitzenbarth, and F. C. Freiling, "Frost Forensic Recovery of Scrambled Telephones," in *11th International Conference on Applied Cryptography and Network Security.* , Canada, 2013, pp. 373-388.

[11] A. Shabtai, Y. Fledel, and Y. Elovici, "Securing Android-Powered Mobile Devices Using SELinux," *IEEE Security & privacy magazine*, vol. 8, pp. 36-44, 2010.

[12] Android website, http://www.android.com/ , [Accessed 15 May 2015].

[13] A. Shabatai, Y. Fledel, U. Kanonov, Y. Elovici, S. Dolev, and C. Glezer, "Google Android: A Comprehensive Security Assessment," *IEEE Security & Privacy magazine*, vol. 8, pp. 35-44 2010.

[14] SQLite, https://www.sqlite.org/, [Accessed 18 November 2015].

[15] Z. Wang, R. Murmuria, and A. Stavrou, "Implementing and Optimizing an Encryption Filesystem on Android," *2012 IEEE 13th International Conference on Mobile Data Management*, pp. 52-62, July 2012

[16] H. Kim, N. Agrawal, and C. Ungureanu, "Examining Storage Performance on Mobile Devices," in *3rd ACM SOSP Workshop on Networking, Systems, and Applications on Mobile Handhelds, MobiHeld '11*: ACM, 2011.

[17] SQLite, "SQLite Encryption Extension Documentation," http://www.sqlite.org/see/doc/trunk/www/index.wiki, [Accessed 17 December 2014].

[18] Zetetic, "SQLCipher: Full Database Encryption for SQLite," https://www.zetetic.net/sqlcipher/, [Accessed 19 January 2015].

[19] Wxcode, wxSQLite3, http://wxcode.sourceforge.net/components/wxsqlite3/, [Accessed 05 February 2015].

[20] C. E. Loftis, T. X. Chen, and J. M. Cirella, "Attribute-Level Encryption of Data in Public Android Databases, RTI Press," September 2013.

[21] L. Haiyan and G. Yaowan, "Analysis and Design on Security of SQLite," *International Conference on Computer, Networks and Communication Engineering (ICCNCE 2013)*, 2013.

[22] A. Skillen and M. Mannan, "On Implementing Deniable Storage Encryption for Mobile Devices," in *20th Annual Network and Distributed System Security Symposium (NDSS '13),* San Diego, CA United States, Feb. 2013.

[23] A. Skillen and M. Mannan, "Mobiflage: Deniable Storage Encryption for Mobile Devices," *IEEE Transactions on Dependable and Secure Computing*, vol. 11, pp. 224-237, May 2014.

[24] B. Bosen, "FDE Performance Comparison Hardware Versus Software Full Drive Encryption," *Trusted Strategies LLC*, 2010.

[25] P. Gasti and Y. Chen, "Breaking and Fixing the Self Encryption Scheme for Data Security in Mobile Devices," in *18th Euromicro Conference on Parallel, Distributed and Network-based Processing*, 2010, pp. 624 - 630.

[26] Marketing Charts, "Android Conquers World - Canalys statistics," http://www.marketingcharts.com/direct/android-conquers-world-15908/, 2011

[27] P. Rogaway, "Efficient Instantiations of Tweakable Block ciphers and Refinements to Modes OCB and PMAC," *Advances in Cryptology – ASIACRYPT*, vol. 3329 of Lecture Notes in Computer Science, pp. 16–31, 2004.

[28] ITU, "Measuring the Information Society, http://www.itu.int/en/ITU-D/Statistics/Documents/publications/mis2013/MIS2013_without_Annex_4.pdf," 2013.

[29] Gartner, "Gartner Says Annual Smartphone Sales Surpassed Sales of Feature Phones for the First Time in 2013, http://www.gartner.com/newsroom/ id/2665715," 2014.

[30] Portio Research, "17 Incredible Facts about Mobile Messaging that you should know", http://www.portioresearch.com/en/blog/2013/17-incredible-facts-about-mobile-messaging-that-you-should-know.aspx, 2013.

[31] comScore Reports, U.S. Smartphone Subscriber Market Share, https://www.comscore.com/Insights/Market-Rankings/comScore-Reports-July-2014-US-Smartphone-Subscriber-Market-Share, [Accessed 24 July 2014].

[32] On Device Research, "Mobile Malaysia: ahead of the pack", https://ondeviceresearch.com/blog/mobile-malaysia-internet-mobile-ecommerce-trends, 2014.

[33] Satista, "Global smartphone shipments forecast from 2010 to 2018", http://www.statista.com/statistics/263441/global-smartphone-shipments-forecast/, [Accessed 06 June 2014].

[34] Y. Chen and W.-S. Ku, "Self-Encryption Scheme for Data Security in Mobile Devices," in *Proceedings of the 6th IEEE Conference on Consumer Communications and Networking Conference* Las Vegas, NV, USA: IEEE Press, 2009.

[35] J. AL-MUHTADI, D. MICKUNAS, and R. CAMPBELL, "A lightweight reconfigurable security mechanism for 3G4G mobile devices," *IEEE Wireless communications*, vol. 9, pp. 60- 65 2002.

[36] G. Agosta, A. Barenghi, F. D. Santis, and A. D. Biagio, "Fast Disk Encryption Through GPGPU Acceleration," in *International Conference on Parallel and Distributed Computing, 2009*: IEEE Computer Society, 2009.

[37] "Lookout Mobile Threat Report," Lookout Mobile Security 2011, https://www.lookout.com/_downloads/lookout-mobile-threat-report-2011.pdf.

[38] D. Dagon, T. Martin, and T. Starner, "Mobile phones as computing devices: the viruses are coming," *IEEE Pervasive Computing*, vol. 3, pp. 11-15, 2004.

[39] D. Emm, "Mobile malware - new avenues," *Network Security, Elsevier Science Publishers B. V.*, vol. 2006, pp. 4-6, November 2006.

[40] "The National Cyber-Security Advisory Council (CNCCS), Smartphone Malware Report," June 2011, http://press.pandasecurity.com/usa/wp-content/uploads/2011/06/CNCCS-Smartphone-Malware-Full-Report-Translated-06-7-11-FINAL.pdf.

[41] J. Viega and B. Michael, "Mobile Device Security - Guest Editors' Introduction," *IEEE Security & privacy journal*, vol. 8, pp. 11-12, 2010.

[42] C. Eric, "Motivations of Recent Android Malware", 2011, www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/motivations_of_recent_android_malware.pdf

[43] M. Ongtang, S. McLaughlin, W. Enck, and P. McDaniel, "Semantically rich application-centric security in Android," *Security and Communication Networks*, vol. 5, pp. 658–673, 2012.

[44] W. Enck, M. Ongtang, and P. McDaniel, "On Lightweight Mobile Phone Application Certification," in *Proceedings of the 16th ACM conference and communications security*, New York, USA, 2009, pp. 235-245.

[45] R. Racic, D. Ma, and H. Chen, "Exploiting MMS Vulnerabilities to Stealthily Exhaust Mobile Phone's Battery," *SecureComm*, vol. 6, pp. 1-10, 2006.

[46] H. Kim, J. Smith, and K. G. Shin, "Detecting energy-greedy anomalies and mobile malware variants," in *Proceedings of the 6th international conference on Mobile systems, applications, and services*, 2008, pp. 239-252.

[47] L. Liu, G. Yan, X. Zhang, and S. Chen, "Virusmeter: Preventing your cellphone from spies," in *Recent Advances in Intrusion Detection*, 2009, pp. 244-264.

[48] D. C. Nash, T. L. Martin, D. S. Ha, and M. S. Hsiao, "Towards an intrusion detection system for battery exhaustion attacks on mobile computing devices," in *Third IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom 2005 Workshops)*, 2005, pp. 141-145.

[49] B. R. Moyers, J. P. Dunning, R. C. Marchany, and J. G. Tront, "Effects of wi-fi and bluetooth battery exhaustion attacks on mobile devices," in *Proceedings of the 43rd Hawaii International Conference on System Sciences (HICSS), 2010*, USA, 2010, pp. 1-9.

[50] M. Chandramohan and H. B. K. Tan, "Detection of mobile malware in the wild," *IEEE Computer*, vol. 45, pp. 65-71, 2012.

[51] H. Bojinov, D. Boneh, and R. Cannings, "Address space randomization for mobile devices," in *Proceedings of the fourth ACM conference on Wireless network security*, 2011, pp. 127-138.

[52] D. Muthukumaran, A. Sawani, J. Schiffman, B. M. Jung, and T. Jaeger, "Measuring integrity on mobile phone systems," in *Proceedings of the 13th ACM symposium on access control models and technologies*, USA, 2008, pp. 155-164.

[53] W. Enck, P. Gilbert, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth, "TaintDroid: an information flow tracking system for real-time privacy monitoring on smartphones," *Communications of the ACM*, vol. 57, pp. 99-106, 2014.

[54] D. Y. Zhu, J. Jung, D. Song, T. Kohno, and D. Wetherall, "TaintEraser: Protecting sensitive data leaks using application-level taint tracking," *ACM SIGOPS Operating Systems Review*, vol. 45, pp. 142-154, 2011.

[55] J. Bickford, R. O'Hare, A. Baliga, V. Ganapathy, and L. Iftode, "Rootkits on smart phones: attacks, implications and opportunities," in *Proceedings of the eleventh workshop on mobile computing systems & applications*, 2010, pp. 49-54.

[56] F. M. David, E. M. Chan, J. C. Carlyle, and R. H. Campbell, "Cloaker: Hardware supported rootkit concealment," *2008 IEEE Symposium on Security and Privacy*, pp. 296-310, 2008.

[57] Microsoft Corporation, "BitLocker Drive Encryption Overview", http://technet.microsoft.com/en-us/library/cc732774.aspx, [Accessed 18 October 2014].

[58] N. Ferguson, "AES-CBC + Elephant diffuser: A Disk Encryption Algorithm for Windows Vista," 2006.

[59] S. M. Diesburg and A.-I. A. Wang, "A survey of confidential data storage and deletion methods," *ACM Computing Surveys (CSUR),* vol. 43, 2010.

[60] S. Türpe, A. Poller, J. Steffan, J.-P. Stotz, and J. Trukenmüller, "Attacking the bitlocker boot process," *Trusted Computing, Springer Berlin Heidelberg,* pp. 183-196, 2009.

[61] Apple, "Best Practices for Deploying FileVault 2, http://training.apple.com/pdf/WP_FileVault2.pdf," 2012.

[62] O. Choudary, F. Gröbert, and J. Metz, "Infiltrate the Vault: Security Analysis and Decryption of Lion Full Disk Encryption," *IACR Cryptology ePrint Archive 2012,* vol. 374, 2012.

[63] J. A. Halderman, S. D. Schoen, N. Heninger, W. Clarkson, W. Paul, J. A. Calandrino, A. J. Feldman, J. Appelbaum, and E. W. Felten, "Lest We Remember: Cold Boot Attacks on Encryption Keys," *In Proceedings of 17th Usenix Security Symposium,* 2008.

[64] M. A. Halcrow, "eCryptfs: An enterprise-class encrypted filesystem for linux," in *Proceedings of the 2005 Linux Symposium,* 2005, pp. 201-218.

[65] C. Laird, "Taking a Hard-Line Approach to Encryption," *IEEE Computer Society,* vol. 40, pp. 13-15, 2007.

[66] L. Hars, "Discryption: Internal Hard-Disk Encryption for Secure Storage," *IEEE Computer Society,* vol. 40, pp. 103-105, 2007.

[67] Android Source, "Notes on the implementation of encryption in Android 3.0", https://source.android.com/devices/tech/encryption/android_crypto_implementation.html [Accessed 13 September 2014]

[68] WhisperCore, http://downloadsquad.switched.com/2011/03/16/whispercore-beta-for-android-encrypts-your-data-for-free-only-w/, [Accessed 18 June 2014]

[69] Whisper Systems, "WhisperCore: Device and Data Protection for Android," Beta Version (0.5.5)", http://whispersys.com/whispercore.html, [Accessed 12 March 2012]

[70] Apple, "iOS Security, http://www.apple.com/ipad/business/docs/iOS_Security_Feb14.pdf," 2014.

[71] R. Bathurst, R. Rogers, and A. Ghassemlouei, *The Hacker's Guide to OS X: Exploiting OS X from the Root Up*: Elsevier, 2012.

[72] RIM, " Security Technical Overview, BlackBerry Enterprise Solution, Version: 5.0, http://docs.blackberry.com/en/admin/deliverables/16650/BlackBerry_Enterprise_Server-Security_Technical_Overview--1153051-0615043613-001-5.0.2-US.pdf," 2011.

[73] A. Belenko and D. Sklyarov, ""Secure Password Managers" and "Military-Grade Encryption" on Smartphones: Oh, Really?," *Blackhat Europe*, 2012.

[74] V. Gough, "EncFS: Encrypted Filesystem, http://arg0.net/wiki/encfs," vol. 1, p. 22, 2003.

[75] R. Canetti, C. Dwork, M. Naor, and R. Ostrovsky, "Deniable encryption," *Advances in Cryptology—CRYPTO'97, Springer Berlin Heidelberg*, pp. 90-104, 1997.

[76] Strategy Analytics, "Android Captured Record 85 Percent Share of Global Smartphone Shipments in Q2 2014", http://blogs.strategyanalytics.com/WSS/post/2014/07/30/Android-Captured-Record-85-Percent-Share-of-Global-Smartphone-Shipments-in-Q2-2014.aspx, July 2014

[77] Android, "Application Framework Architecture", http://developer.android.com/about/versions/index.html, [Accessed 23 October 2013]

[78] R. J. G. Vargas, E. A. Anaya, R. G. Huerta, and A. F. M. Hernandez, "Security controls for Android," in *Computational Aspects of Social Networks (CASoN)*, 2012, pp. 212-216.

[79] R. Meier, *Professional Android 2 Application Development*, 1st ed.: Wrox Press, Wiley Publishing, Inc., 2010.

[80] D. Compton, "A Technical Survey of Mobile Device Security," Auburn University, 2012.

[81] Kaspersky Lab, "Kaspersky Security Bulletin 2012", http://securelist.com/analysis/kaspersky-security-bulletin/36703/kaspersky-security-bulletin-2012-the-overall-statistics-for-2012/, [Accessed 05 November 2012].

[82] Z. Wang, R. Johnson, R. Murmuria, and A. Stavrou, "Exposing security risks for commercial mobile devices," in *Computer Network Security, LNCS*: Springer Berlin Heidelberg, 2012, pp. 3-21.

[83] Android Security, "Full Disk Encryption", https://source.android.com/security/encryption/, [Accessed 13 March 2015].

[84] W. Enck, M. Ongtang, and P. McDaniel, "Understanding Android Security," *IEEE Security & privacy journal,* vol. 7, pp. 50-57, 2009.

[85] M. Conti, V. T. N. Nguyen, and B. Crispo, "CRePE: Context-Related Policy Enforcement for Android," *Information Security, LNCS* vol. 6531, pp. 331-345, 2011.

[86] A. Shabtai, Y. Fledel, U. Kanonov, Y. Elovici, S. Dolev, and C. Glezer, "Google Android: A Comprehensive Security Assessment," *IEEE Security & Privacy magazine,* vol. 8, pp. 35-44 2010.

[87] T. Cannon and S. Bradford, "Into the Droid: Gaining Access to Android User Data," in *DefCon Hacking Conference (DefCon '12),* Las Vegas, Nevada, July 2012.

[88] L. Davi, A. Dmitrienko, A.-R. Sadeghi, and M. Winandy, "Privilege escalation attacks on android," *Information Security, LNCS,* vol. 6531, pp. 346-360, 2011.

[89] S. Bugiel, L. Davi, A. Dmitrienko, T. Fischer, and A.-R. Sadeghi, "Xmandroid: A new android evolution to mitigate privilege escalation attacks," Technische Universitat Darmstadt, Technical Report TR-2011-04, 2011.

[90] S. Bugiel, L. Davi, A. Dmitrienko, T. Fischer, A.-R. Sadeghi, and B. Shastry, "Towards Taming Privilege-Escalation Attacks on Android," *NDSS Symposium 2012,* 2012.

[91] J. Gotzfried and T. Muller, "ARMORED: CPU-bound encryption for Android-driven ARM devices," in *2013 Eighth International Conference on Availability, Reliability and Security (ARES)*, 2013, pp. 161-168.

[92] Android Developers, "Storage Options", http://developer.android.com/guide/topics/data/data-storage.html#pref, [Accessed 12 August 2014].

[93] T. Blasing, L. Batyuk, A.-D. Schmidt, S. A. Camtepe, and S. Albayrak, "An Android Application Sandbox System for Suspicious Software Detection," in *2010 5th International Conference on Malicious and Unwanted Software (MALWARE)*, 2010, pp. 55-62.

[94] T. Šimunić, L. Benini, and G. D. Micheli, "Cycle-Accurate Simulation of Energy Consumption in Embedded Systems," in *Proceedings of the 36th annual ACM/IEEE Design Automation Conference*, 1999, pp. 867-872.

[95] C. Fruhwirth, "New Methods in Hard Disk Encryption," Institute for Computer Languages Theory and Logic Group, Vienna University of Technology, 2005.

[96] K. Scarfone, "Guide to Storage Encryption Technologies for End User Devices," *NIST*, vol. Special Publication 800-111, Nov 2007.

[97] TureCrypt, on-the-fly disk encryption software, http://www.truecrypt.org/, [Accessed 17 Feb 2014].

[98] FreeOTFE, Free disk encryption software, http://www.freeotfe.org/, [Accessed 18 Feb 2014].

[99] E. Riedel, M. Kallahalla, and R. Swaminathan, "A Framework for Evaluating Storage System Security," in *Proceedings of the 1st USENIX Conference on File and Storage Technologies (FAST'02)*. vol. 2: USENIX Association, 2002, pp. 15-30.

[100] C. P. Wright, J. Dave, and E. Zadok, "Cryptographic file systems performance: What you don't know can hurt you," in *Proceedings of the Second IEEE International Security in Storage Workshop, SISW'03*, 2003, pp. 47-47.

[101] A. Rajgarhia and A. Gehani, "Performance and extension of user space file systems," in *Proceedings of the 2010 ACM Symposium on Applied Computing*: ACM, 2010, pp. 206-213.

[102] C. Hohmann, "CryptoFS - An Encryption Filesystem, http://reboot.animeirc. de/cryptofs/," 2008.

[103] W. A. Bhat and S. M. K. Quadri, "After-deletion data recovery: myths and solutions," *Computer Fraud & Security 2012*, vol. 4, pp. 17-20, 2012.

[104] E. Zadok, I. Badulescu, and A. Shender, "Cryptfs: A stackable vnode level encryption file system. Vol. 184. Technical Report CUCS-021-98," Computer Science Department, Columbia University 1998.

[105] M. Halcrow, "eCryptfs: A Stacked Cryptographic Filesystem," *Linux Journal 2007*, vol. 156 p. 2, 2007.

[106] Dm-crypt: Linux kernel device mapper crypto target, https://code.google.com/p/cryptsetup/wiki/DMCrypt, [Accessed 16 October 2014]

[107] A. Menezes, P. V. Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*: CRC Press, 1996.

[108] W. Stallings, *Cryptography and Network Security Principles and Practices*, Fourth ed.: Prentice Hall, 2005.

[109] M. V. Ball, C. Guyot, J. P. Hughes, L. Martin, and L. C. Noll, "The XTS-AES disk encryption algorithm and the security of ciphertext stealing," *Cryptologia 36:1* pp. 70-79, 2012.

[110] M. V. Ball, "Follow-up to NIST's Consideration of XTS-AES as standardized by IEEE Std 1619-2007 (Draft 2)," IEEE SISWG, p. 17, Mar 2009.

[111] S. Ahmed, K. Samsudin, A. R. Ramli, and F. Z. Rokhani, "Advanced Encryption Standard-XTS implementation in field programmable gate array hardware," *Security and Communication Networks*, vol. 8, pp. 516-522, 2015.

[112] A. P. Kakarountas, E. Hatzidimitriou, and A. Milidonis, "A survey on throughput-efficient architectures for IEEE P1619 for shared storage media," *IEEE Symposium on Computers and Communications (ISCC)*, pp. 758-763, 2011.

[113] E. Hatzidimitriou and A. P. Kakarountas, "Implementation of a P1619 Crypto-Core for Shared Storage Media," in *15th IEEE Mediterranean Electrotechnical Conference (MELECON)*, Malta, 2010, pp. 597-601.

[114] F. Gurkaynak, R. Schilling, M. Muehlberghuber, F. Conti, S. Mangard, and L. Benini, "Multi-core data analytics Soc with a flexible 1.76 Gbit/s AES-XTS cryptographic accelerator in 65 nm CMOS," in *Proceeding of the Fourth Workshop on Cryptography and Security in Computing Systems*, 2017, pp. 19-24.

[115] A. Elbirt, W. Yip, B. Chetwynd, and C. Paar, "An FPGA Implementation and Performance Evaluation of the AES Block Cipher Candidate Algorithm Finalists " *In The Third Advanced Encryption Standard Candidate Conference*, pp. 13-27, 1999.

[116] E. J. Swankoski, R. R. Brooks, V. Narayanan, M. Kandemir, and M. J. Irwin, "A Parallel Architecture for Secure FPGA Symmetric Encryption," in *Proceedings of 18th International Parallel and Distributed Processing Symposium.*, 2004, pp. 132-139.

[117] NIST, "FIPS-197, Announcing the Advanced Encryption Standard (AES)," http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf, Nov 2001.

[118] S. Ahmed, K. Samsudin, A. R. Ramli, and F. Z. Rokhani, "An Effective Storage Encryption Solution," *Indian Journal of Science and Technology* vol. 6, pp. 4384-4389, 2013.

[119] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Ferguson, "Twofish: A 128-Bit Block Cipher," in *NIST AES Proposal*, 1998.

[120] Khronos OpenCL, http://www.khronos.org/opencl/, [Accessed January 2015].

[121] Nvidia, "Nvidia CUDA Programming Guide, Version 3.0," 2010.

[122] C. Wang, S. Chandrasekaran, and B. Chapman, "An OpenMP 3.1 Validation Testsuite," in *OpenMP in a Heterogeneous World*, 2012, pp. 237-249.

[123] Google Patents, "RC6 patent," http://www.google.com/patents?vid=5835600, 1998, [Accessed August 2012].

[124] J. Kreibich, *Using SQLite*, First ed.: O'Reilly Media, Inc, 2010.

[125] M. Owens, "Embedding an SQL database with SQLite," *Linux Journal*, vol. 2003, p. 2, 2003.

[126] S. Haldar, *Inside sqlite*: O'Reilly Media, Inc, 2007.

[127] M. Owens and G. Allen, *The definitive guide to SQLite*, Second ed.: Apress, 2010.

[128] SQLite, "The SQLite Amalgamation," http://www.sqlite.org/amalgamation.html, [Accessed 11 December 2014].

[129] R. A. Popa, C. Redfield, N. Zeldovich, and H. Balakrishnan, "Cryptdb: Protecting Confidentiality with Encrypted Query Processing," in *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*, 2011, pp. 85-100.

[130] SQLiteCrypt, "SQLiteCrypt: Transparent SQLite database encryption," http://sqlite-crypt.com/documentation.htm, [Accessed 12 February 2015].

[131] Botansqlite, "Botansqlite3: SQLite Encryption," https://github.com/OlivierJG/botansqlite3, [Accessed 19 February 2015].

[132] M. S. Ferdous and R. Poet, "Portable Personal Identity Provider in Mobile Phones," *12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 736-745, July 2013.

[133] The Guardian Project, "SQLCipher For Android: Encrypted Database," https://guardianproject.info/code/sqlcipher/, 2014.

[134] S. Fahl, M. Harbach, M. Oltrogge, T. Muders, and M. Smith, "Hey, You, Get Off of My Clipboard," *In Financial Cryptography and Data Security, Springer Berlin Heidelberg*, pp. 144-161, 2013.

[135] T. Graf, S. Zickau, and A. Kupper, "Enabling Location-Based Services on Stationary Devices Using Smartphone Capabilities," *Mobile Web Information Systems (MobiWIS 2013), LNCS 8093*, pp. 49-63, 2013.

[136] M. Yamamoto, "A Speed-up Method of Light RDBMS SQLite for Stream Processing Utilizing Multi-core CPU Configurations," *Electronics and Communications in Japan*, vol. 96, pp. 19-31, 2013.

[137] Strategy Analytics, " Multi-core chip penetration increased to 66% in 2013", https://www.strategyanalytics.com/strategy-analytics/blogs/components/handset-components/2013/09/09, [Accessed 15 September 2017].

[138] G. R. Andrews, *Foundations of multithreaded, parallel, and distributed programming*: Addison Wesley, 2000.

[139] R. Chandra, L. Dagum, D. Kohr, D. Maydan, R. Menom, and J. McDonald, *parallel programming in openmp*: Morgan Kaufmann, 2001.

[140] J. D. Owens, D. Luebke, N. Govindaraju, M. Harris, J. Krüger, A. E. Lefohn, and T. J. Purcell, "A Survey of General-Purpose Computation on Graphics Hardware," *In Eurographics, Computer Graphics Forum*, vol. 26 pp. 80–113, 2007.

[141] M. A. Alomari and K. Samsudin, "A framework for GPU-accelerated AES-XTS encryption in mobile devices," in *TENCON 2011-2011 IEEE Region 10 Conference*: 144-148, 2011.

[142] B. Schauer, "Multicore Processors – A Necessity," *ProQuest discovery guides (2008)*, pp. 1-14, 2008.

[143] B. Chapman, G. Jost, and R. V. D. Pas, *Using OpenMP: portable shared memory parallel programming*: MIT press, 2007.

[144] J. Whipple, W. Arensman, and M. S. Boler, "A Public Safety Application of GPS-Enabled Smartphones and the Android Operating System," in *IEEE International Conference on Systems, Man and Cybernetics*, USA, 2009, pp. 2059-2061.

[145] J. Lerner and J. Tirole, "The Simple Economics of Open Source," *NBER Working Paper Series*, 2000.

[146] Eclipse Foundation, http://www.eclipse.org, [Accessed 27 March 2015].

[147] K. Furlinger and M. Gerndt, "ompP: A Profiling Tool for OpenMP," *Lecture Notes in Computer Science*, vol. 4315, pp. 15-23, 2008.

[148] B. Mohr, "OPARI-OpenMP Pragma And Region Instrumentor", 2006.

[149] B. Mohr, A. D. Malony, S. Shende, and F. Wolf, "Towards a performance tool interface for OpenMP: An approach based on directive rewriting," *Proceedings of the Third Workshop on OpenMP (EWOMP'01),* 2001.

[150] S. L. Graham, P. B. Kessler, and M. K. M. Kusick, "Gprof: A call graph execution profiler," *ACM Sigplan Notices* vol. 17, pp. 120-126, 1982.

[151] K. Furlinger and M. Gerndt, "Analyzing Overheads and Scalability Characteristics of OpenMP Applications," *Lecture Notes in Computer Science*, vol. 4395, pp. 39-51, 2007.

[152] K. Furlinger and S. Moore, "Continuous Runtime Profiling of OpenMP Applications," *Parallel Computing: Architectures, Algorithms and Applications, NIC series*, vol. 38, pp. 677-684, 2007.

[153] K. Furlinger, "OpenMP application profiling - state of the art and directions for the future," *Procedia Computer Science* vol. 1, pp. 2107-2114, 2010.

[154] ClockworkMod, "Clockworkmod: ROM Manager and Recovery Image," https://www.clockworkmod.com/rommanager, [Accessed 13 April 2013].

[155] D. B. Stewart, "Measuring execution time and real-time performance," in *Embedded systems conference* Boston, Sep 2006.

[156] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, and C. Hall, "Performance Comparison of the AES Submissions," in *Proceedings of the Second AES Candidate Conference*, 1999, pp. 15-34.

[157] T. H. Romer, W. H. Ohlrich, A. R. Karlin, and B. N. Bershad, "Reducing TLB and memory overhead using online superpage promotion," *ACM SIGARCH Computer Architecture News*, vol. 23, pp. 176-187, 1995.