CRY-HELP: A Mobile Crime Reporting Application

Thabo Ndlovu Nina Otsweleng Tami Maiwashe

July 9, 2013

Table of Contents

1. Project Description

South Africa has one of the highest crime rates in the world. However, a recent article announced that in 2011 more than half the crimes of that year had not been reported to the police. Research has shown that people opt not to report crime not only because of associated costs – for instance, one's time and money, but also for fear of offenders, as well as of not sounding credible when reporting. These reasons illustrate the importance of a person either

feeling or being anonymous when they report a crime (a victim in the former case, and a third-party reporter in the latter). A study further supported this by showing that people are more likely to report a crime using a computer-based approach as opposed to the traditional telephonic approach. This sense of anonymity also envelops privacy: people usually don't want anyone but the police to know when they have reported a crime.

As such, the aim of this project is to create a mobile application, *Cry-Help*, which will allow users to report a crime electronically (moving away from the 'intimidating' telephonic approach) with an emphasis on anonymity and subtlety in an oppressive environment. Users will provide information relevant to the crime using covert means, such as gestures, on their mobile devices. This information will be sent via a secure protocol over telecommunications networks and delivered to a secure server. Finally, relevant officials will be able to promptly access only those reports relevant to them.

2. Problem statement

There currently does not exist a way for one to report crime covertly and securely, *and* have the relevant authorities access the information promptly while storing it securely.

2.1. Research Question

The question we would like to develop a solution for is this:

Can a mobile application that allows covert and secure crime reporting and further supports both safe storage of the crime report data and quick access to it be developed?

We explore this research question in terms of the three components of the proposed application: its user interface, the transmission of the crime reports, and its backend system.

2.1.1. User interface

The user interface will have two primary functions, namely, inputting data and reporting crimes. Mobile phones are multi-purpose devices that are not dedicated to emergency reporting – in fact, numerous other devices have been developed with the specific role of emergency reporting. Despite this, mobile phones being a common possession makes them a viable target device for this purpose – particularly modern mobile phones running the Android OS which are becoming cheaper and more prevalent.

The advantages of digitising the crime reporting process and the already existing security/crime-prevention and alerting capabilities inherent in a mobile device have been documented in numerous works (Lasley, J.R. & Palombo, B.J. 1995). The following questions however will be explored with regard to digital crime reporting on the mobile device:

i. **Is it effectively possible to allow users to report crime digitally on the mobile**

device?

- Is there a way to covertly gather relevant user and crime data in a hostile environment whilst considering time constraints in emergency situations?
- What means should be used to gather the data and how much data can be pre-collected?
- What kind of interface would best suit the data being collected given the context of the environment?
- Given crime reporting in both first and third person, what critical aspects differentiate the two and influence interface design?
- What data is most relevant?

ii. Is it possible to develop a system that allows effective covert message passing in an oppressive environment on mobile devices?

- Is there a way for a user to effectively conceal the reporting process from the individuals around them, who may even be looking over their shoulders?
- In what (crime) scenarios would concealing interaction with a mobile device be possible/feasible?
- Can the crime report attain a response, all the while leaving no traces that the perpetrator could find?

2.1.2. Secure transfer protocol

Mobile users are subject to a number of security threats. Attackers target mobile phones because they sometimes contain sensitive data and the user's identity information. In the context of mobile crime reporting an attacker could:

a. manipulate the phone to spam the authorities,
b. eavesdrop on conversations between the user and authorities,
c. impersonate the user and send false information to authorities.

Most mobile devices run on the GSM (2G) and 3G mobile networks. These are known to have weak encryption algorithms that can be broken easily (Gendrullis, Novotny, & Rupp, 2008), yet information that goes through the networks is unencrypted. Once the algorithms are broken, an attacker can access all unencrypted information that was made using the victim's phone. This becomes a bigger problem if the intercepted information is sensitive.

In order to tackle these problems, people have implemented some proxy-based security solutions, such as WTLS for WAP protocol. However, these solutions decrease application performance and do not protect against eavesdropping and data tampering (Mynttinen 2000). Kayayurt & Tuglular (2006) describe a way to implement end-to-end security, using TLS and SSL protocols, on mobile devices instead. Though these two protocols are known for their unsuitable high resource consumption, they claim to have a solution that uses TLS efficiently in mobile networks. This proposal doesn't appear to have been widely evaluated. We intend to

answer the following research question in the context of crime reporting for this part of the project:

i. Is the proposed mobile end-to-end TLS protocol implementation by Kayayurt & Tuglular (2006) safe and efficient for mobile crime reporting?

2.1.3. Database security and performance

To date, great efforts have been made in database security to create systems that enforce restrictions on users' access to data. These advances have yielded some effective solutions that are now common world-wide, such as role-based access control. However, the rise of inside jobs in organisations and some attackers' ability to sidestep authorisation schemes has led to the exploration of ways to further protect databases. Among these has been the encryption of databases.

For many years though, discussion of database encryption schemes has remained largely theoretical, with suggested approaches being too expensive to implement. But recently, a number of practical approaches have emerged, each a different trade-off between security and system performance. CryptDB is one such a practical solution that was recently proposed. It stands apart from most previous encryption schemes because its effect on some database systems' performance has been acclaimed as modest, only decreasing their speed by 26%, and it allows for the execution of some queries on encrypted data, thus preserving a high level of security where previous encryption schemes have left data exposed unnecessarily.

A crime reporting system requires that the incoming reports be made available to officials quickly, but also kept private as they may contain sensitive information. Consequently, in the third part of this project, we will investigate whether CryptDB is effective in the context of crime reporting. Specifically, we aim to answer these questions:

i. How well does the system perform under CryptDB?

ii. How can CryptDB's performance be improved for this context?

This part of the project does not investigate the level of security offered by CryptDB as this is well-avouched. Rather, it focuses on analysing the severity of CryptDB's performance overhead and limitations by looking at how the work of relevant officials in a crime reporting system using it in its backend is affected.

3. Procedures and methods

Design considerations and scope will be made with the target environment for the system being the University of Cape Town campus and its Campus Protection Services (CPS).

Furthermore, we will use the Scrum methodology to manage our project. Dr Anne Kayem will act as the Product Owner and the Scrum Master, Nina Otsweleng will be the team leader, and we will treat CPS as stakeholders; however, we will mostly rely on Dr Kayem to be their voice. Moreover, we will develop *Cry-Help* in two-week long sprints.

In light of Jakob Nielsen's analysis of the most useful number of testers for a system (2000), we will test *Cry-Help* as follows: we will have 15 volunteers from UCT's student body test different features of the application, and if time permits, we will have 5 volunteers different to the initial set test an improved version. The next three sections include elaboration on what the tests with the volunteers will evaluate in terms of each component of the application.

3.1. Interface

3.1.1. Design features

An interactive prototype will be developed that addresses the issue of data collection and crime reporting in a hostile environment. Data fields necessary for an effective crime report will be collected from authorities such as the CPS.

The interface will provide the following services:

- Users will be able to install the application and provide permission to for it to make use of pertinent personal data.
- Certain data will need to be stored about the user to reduce reporting time during an emergency. This data will have to be stored and encrypted.
- Users will be able to send crime reports using their mobiles to the CPS server using a subtle gesture. Reports will either be first-person reports or detailed third-person reports that describe the scene for authorities. Third-person reports need not be sent covertly.
- The application will provide some form of output in response to crime reports.
- The system will delete all traces of previous communications.

Effectively the system will allow covert message passing between the reporter and CPS that leaves no trace of communication.

3.1.2. Development strategy

The application will be developed for an Android target device with the Android SDK on Java. The design will be developed iteratively to create an interface suitable for the type and urgency of data.

Stored data will be password-protected and will only be available to the user until sent via a secure protocol to the CPS server in a report.

3.1.3. Test plan

The interface will be developed iteratively with volunteers with the design features in mind.
Testing will involve analysing:

- covert gesturing with a mobile device
- data traces

The final high-fidelity prototype will be implemented on campus grounds. Ten users will be selected to test the system along with CPS cooperation. Tests will cover UI usability and performance.

Evaluation will require acting out scenarios in which the application would likely be used in order to gauge the success of the Interface. Users will be expected to use the system without any help and give feedback on the interaction the system gave.

3.2. Secure protocol

3.2.1. Design features

The crime reporting application will need to have a secure channel for transporting a crime report from the user's phone to a backend database. To do this we will implement a prototype TLS security protocol for mobile devices as specified by Kayayurt & Tuglular (2006). This protocol is said to be based on TLS 1.0 specifications and adopted to work on J2ME mobile devices. This protocol is an application itself in which other applications such as *Cry-Help* can use its APIs to transfer data securely. The application will need to have a client/server architecture where the mobile phone acts as the client and the backend database acts as a server.

The proposed protocol has an architecture which covers both the TLS 1.0 specifications and necessary APIs architecture which involve cryptography model classes that abstract the real implementations of cryptography packages (Kayayurt & Tuglular, 2006). The TLS specification used in the paper is made up of many layers. These are described as

- *TLS Record Protocol* – responsible for messages, fragmenting into blocks, encrypting and transmitting them to higher clients
- *TLS Handshake protocol* – used for authentication between the client and server
- *TLS Alert Protocol* – send warning and calamitous errors that could take place in a TLS session
- *TLS Change Cipher Spec Protocol* – used to start new keys and encryption that the handshake protocol has established

3.2.2. Development strategy

Kayayurt and Tuglular (2006) have defined a class diagram for this security protocol which we

intend on using as a guide when developing the mobile protocol. This protocol will be implemented using Java, with the JBuilder IDE. For cryptographic algorithm implementation, we will use the Bouncy Castle Cryptography Package. Kayayurt & Tuglular (2006) spend very little time discussing key exchange algorithms. It is essential though for this project that the key exchange algorithm is taken into account. Due to the lack of discussion on this matter, we aim to experiment with the different key exchange algorithms such as Diffie-Hellman and RSA to see which are best suited for the mobile device. Since some information about the application's user will be captured and stored on the mobile device, some of it could be used as a parameter for the key exchange algorithm we end up using.

3.2.3. Test plan

The first thing to test in this protocol is if it actually works. This will involve testing the key exchange algorithm, the handshake protocol and also the transmission and reception of data. From then on, we would like to test if the protocol is indeed secure by looking at a few attacks that might be used on it. If time allows, performance of the protocol – especially the key exchange, decrypting and encrypting – will be tested. We will need to test performance because the application is essentially meant to help a victim get a timely response and works on a mobile phone.

3.3. Database

3.3.1. Design features

At the backend of the crime reporting system, we will have a database management system (DBMS) whose architecture will be as CryptDB requires. For comparison, we will also have an unencrypted database protected by role-based access control. A simple application to be used by the CPS officers will be developed and adapted for the different DBMSes being tested. Its main functions will be to automatically receive notifications of a relevant crime report (for example, officers should only be notified of crimes in the region that they patrol), as well as to query crime reports.

3.3.2. Development strategy

Java will be used to programme the application for the CPS officers. Also, CPS representatives will need to be interviewed and/or observed in order to understand what functions such an application should support, over and above instant notifications of crime and queries on reports. Such functions will be added to the application, and this will allow our evaluation of both DBMSes' performance in the context of crime reporting to be more accurate.

The two DBMSes will be developed in MySQL.

3.3.3. Test plan

The backend will be tested for the following:

- Time taken per required function (e.g. notifications, queries)
- Accuracy of data retrieved

Since we will be developing the different parts of this project concurrently, for initial testing, we will need to manually enter crime reports and run tests on those. When the different parts have been integrated, we will then run tests on crime reports sent from the actual *Cry-Help* application.

4. Expected challenges

- Acquiring cooperation from CPS
- The security protocol implementation proposed by Kayayurt & Tuglular (2006) is developed in Java so that it is compliant with J2ME, however, our goal is to implement Cry-Help for Android mobile devices. It might be difficult to make the protocol suitable for, or work on the Android platform.
- GSM and 3G are insecure yet we need to have a secure way of exchanging keys between the client (mobile device) and server (database).

5. Ethical, Professional and Legal issues

5.1. User Testing

Ethical clearance will be needed in order to perform any user testing. Participants will be students/affiliates of the University selected via a voluntary process in which they sign their retractable consent. Anonymity will be ensured, and no data outside of the study will be collected (such as names). Participants will be informed on the project details and will not be in any danger during participation.

5.2. CPS

Clearance and participation from the CPS will be necessary to carry out full scenario testing. Anonymity of individual officers will be ensured, as well as keeping any data from their actual storage or databases private. Most of all this project, has nothing to do with CPS' performance or effectiveness and they will be informed of this.

6. Related work

Numerous applications have been developed that address similar problems (Lasley, J.R. & Palombo, B.J. 1995, Blythe, M.A., Wright, P.C. & Monk, A.F. 2004). These explore the digitisation and automation of crime reporting, while others address the issue of needing a mobile device usable in emergencies. The needs of the users are weighted to attempt to create an effective interface (Satchell, C. & Foth, M. 2011).

Furthermore, many studies on mobile and communication security have taken place since the growth of mobile device use in the world. Jøsang & Sanderud (2003) describe communication security in terms of confidentiality, integrity, authentication and non-repudiation of transmitted data. Misra & Wickamasinghe (2004) speak of two categories of security challenges a device must account for, namely, content security and channel security. Content security deals with protecting data on the device, and channel security involves protecting data from unauthorised access. Security is usually implemented using cryptographic algorithms which should only be used when the data transmitted is extremely sensitive because they affect the way mobile applications run (Jøsang, 2003).

Mobile communication security is not only dependent on cryptographic algorithms, but also on the network architectures, such as GSM and 3G. A number researchers have come up with ways to increase mobile security. These include Arapinis et al. (2012), who proposes an improvement to 3G that helps avoid linkability, and Kayayurt & Tuglular (2006), who have come up with an end-to-end mobile security protocol using TLS and SSL.

Finally, there has been a growing interest in securing data-at-rest (RSA Security, 2013). The aim is to keep the database in an unreadable form so that even when it is compromised, the attacker will be unable to retrieve the sensitive data stored (Elovici et al., 2004). However, as much as advances have been made in finding ways to encrypt databases inexpensively (for example, in the work of Elovici et al. (2004)), another issue to further consider is that of performing database queries on encrypted data.

Gentry's (2009) fully homomorphic encryption scheme allows for any kind of query to be executed on the encrypted database. However, its performance is very poor, even for actions as simple as a search. Research like that of Sesay et al. (2004) and of Elovici et al. (2004) did not focus on query execution on encrypted databases, but instead looked more into improving the level of security provided by database encryption.

As such, CryptDB (Popa et al., 2011) appears to be one of the first database encryption solutions to emerge with an empirical evaluation of various database systems' performance that includes the overhead encryption imposes, while maintaining the same standard of security as that proposed in other papers' database encryption schemes.

That being said, Pope et al.'s (2011) work has not been tested widely, and as much as their tests for performance on various systems revealed that CryptDB adds 'only' 26% of an overhead to query processing in some cases (lower in others), this number is yet to be further analysed – for instance, does it make for a worthwhile trade-off in a system that requires both high security for its data, as well as quick access to it?

7. Anticipated outcomes

This project should yield a mobile application for one to report crime to authorities (in this case,

CPS) covertly. The interface is expected to aid users in scenarios where reporting a crime openly could result negatively for them. Moreover, it must do so reliably, confirming an alert without leaving any recognisable trace of someone having attempted to make a report.

Furthermore, the protocol put forward by Kayayurt and Tuglular (2006) will be analysed and evaluated. We will evaluate in terms of performance, especially that of the key exchange algorithm, and also how the protocol affects the application as a whole. By the end of this, we would like to have an idea of the possible attacks on the proposed protocol and ways to fix the protocol to avoid them.

Finally, an analysis of CryptDB in the context of crime reporting will be produced, focussing on how well the system performs under this encryption scheme, as well as suggestions for or – preferably – implementation of improvements (with regard to "improvements", the expectation is that the backend will be modified so that CPS officers are able to use the application to go about their work, with the encryption overhead not hampering it significantly).

7.1. Expected Impact

In the bigger picture, this work will contribute towards building a system that would greatly help the people of this country – and hopefully even beyond – as people will be empowered to report crime safely, have their reports transmitted and kept most safely , all the while guaranteeing prompt delivery of those reports to authorities too – neither the system's performance nor security will have been compromised for the other. It is hoped that more work will be done on what we deliver, to bring it the idea to actual existence and use in community, and effectively reduce the large number of heinous crimes that go unreported.

7.2. Success Factors

The most positive outcome for this project will be a system that effectively allows a victim to report a crime covertly without any trace of the communication. The report should be carried quickly and securely to the CPS system and should acquire a prompt response from them.

8. Project Plan

8.1. Division of work

Thabo Ndlovu will focus on designing the interface for the mobile crime reporting app. Nina Otsweleng will work on the secure transmission of the crime reports between the mobile device and database. Lastly, Tami Maiwashe will design and develop the backend.

8.2. Risks

Risk	Probability	Impact	Mitigation
Loss of a team member for a significant length of time (e.g. illness)	L	H	Not only will we take good care of our health, but we will also develop each part of the project such that interdependencies are minimal. Furthermore, we will hold meetings weekly to discuss our progress so as to stay in tune with one another's work, as opposed to working in isolation.
Scope creep	M	H	We will continuously revisit our scope definition (as initially outlined in this proposal) to tighten it and evaluate how relevant our work is so far.
The three parts can't be integrated	M	M	As mentioned briefly earlier, we will make efforts to stay abreast with one another's work, and this will include constantly sharing our expectations of what each part will feed into the other, as well as explanations of our parts' output.
Failure to complete the project on time	L	H	We will make ourselves accountable within the team and to our supervisor. We will use our project timeline to guide our work, and we will also use it to document new goals.
Poor cooperation from CPS	M	M	We will communicate with this organisation professionally at all times, and be sure to motivate well for this project.

8.3. Resources required

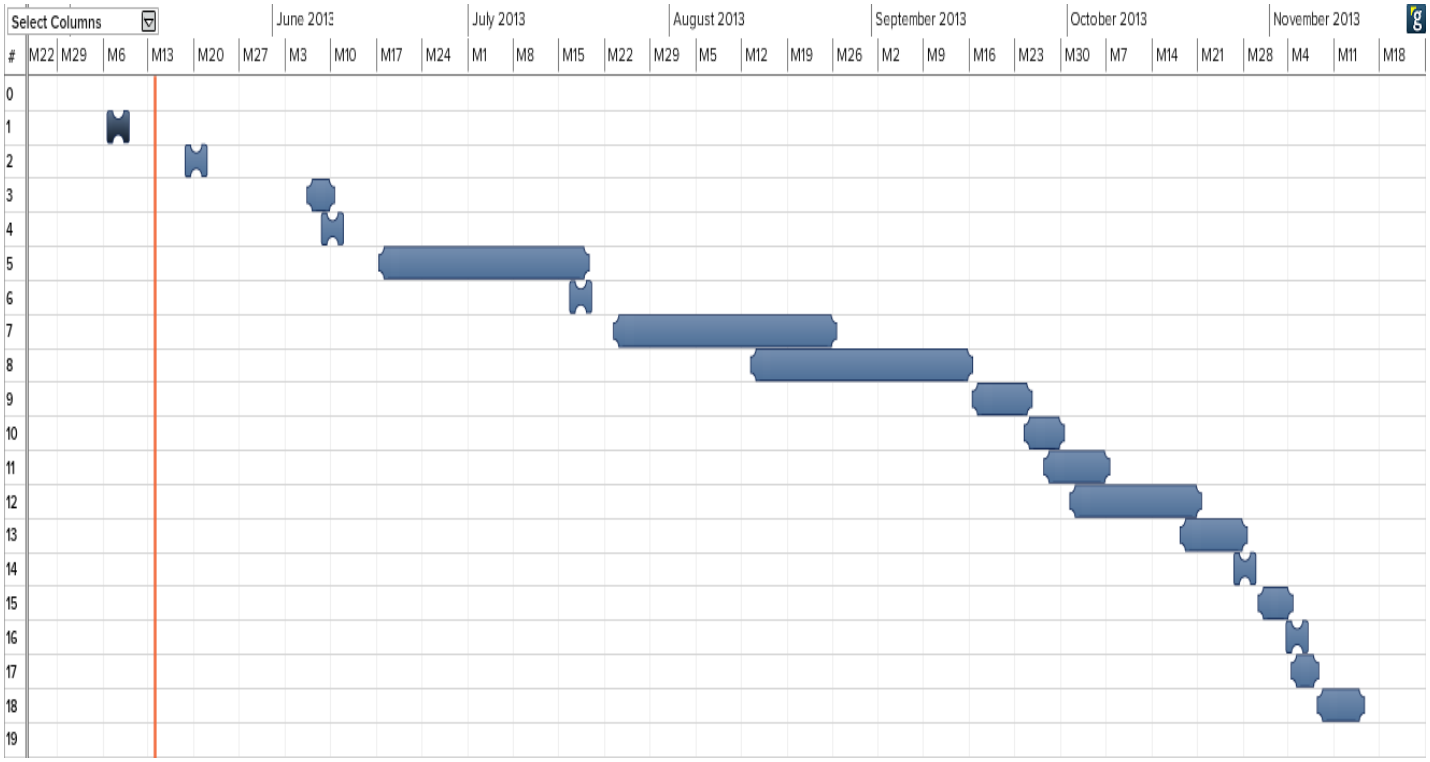
The main resources we need are Android phones to test the mobile app on, as well as a server for the backend component of the application.

8.4. Department milestones

On the next page we present the department's milestones as a table with the number of days we expect to take to reach each, the maximum number of days we will commit to the task, the date

by which we would like to have reached the milestone, and the date by which we have to be done with the task. Below the table, we represent the same information on a Gantt chart.

0	Milestone List				
1	Proposal	3 days	3 days	Wed 05/08/13	Fri 05/10/13
2	Presentation	3 days	3 days	Mon 05/20/13	Wed 05/22/13
3	Revised proposal finalization	3 days	3 days	Thu 06/06/13	Mon 06/10/13
4	Web Presence	2 days	2 days	Mon 06/10/13	Tue 06/11/13
5	Feasibility Demonstration	25 days	25 days	Mon 06/17/13	Fri 07/19/13
6	Background/Feasibility Chapter	2 days	2 days	Thu 07/18/13	Fri 07/19/13
7	Design Chapter	25 days	25 days	Tue 07/23/13	Mon 08/26/13
8	First Implementation/Performance write up	25 days	25 days	Tue 08/13/13	Mon 09/16/13
9	Final Prototype/Performance and write up	8 days	8 days	Mon 09/16/13	Wed 09/25/13
10	Implementation and Testing chapters. Implementation and Testing Completion	5 days	5 days	Tue 09/24/13	Mon 09/30/13
11	Complete Report Outline	7 days	7 days	Fri 09/27/13	Mon 10/07/13
12	Complete draft of Report	15 days	15 days	Tue 10/01/13	Mon 10/21/13
13	Report Final hand in	7 days	7 days	Fri 10/18/13	Mon 10/28/13
14	Poster	4 days	4 days	Mon 10/28/13	Thu 10/31/13
15	Web Page	4 days	4 days	Wed 10/30/13	Mon 11/04/13
16	Project Demonstrations	1 day	1 day	Tue 11/05/13	Tue 11/05/13
17	Reflective Paper	5 days	5 days	Mon 11/04/13	Fri 11/08/13
18	Project presentations	6 days	6 days	Fri 11/08/13	Fri 11/15/13



Above: The department's milestones and for each milestone: the number of days we expect to take, the maximum number of

days we will commit to it, the date by which we will aim to have reached it, and the date by which we are required to have completed the task

Below: The same milestones and our expected schedule on a Gantt chart

8.5. Team milestones

In addition to the department's milestones, we would like to have completed the following tasks by the dates listed:

- Speak to CPS *12 July*
- Develop individual components *9 August*
- Test individual components *16 August*
- Integrate three components *30 August*
- Obtain ethical clearance *30 August*
- Test whole system *6 September*
- Final 'tweaks' *16 September*

9. Conclusion

This paper outlined the project *Cry-Help* that we will be working on for the next six months. It is hoped that it will yield a mobile-crime reporting app with a secure protocol for the transmission of data and a well-protected database system that also performs well enough in the context of police services.

10. References

Arapinis, M., Mancini, L., Ritter, E., Ryan, M., Golde, N., Redon, K., & Borgaonkar, R. (2012, October). New privacy issues in mobile telephony: fix and verification. In Proceedings of the 2012 ACM conference on Computer and communications security (pp. 205-216). ACM.

Bouganim, L., & Guo, Y. (2009). Database Encryption. In S. Jajodia, & H. van Tilborg, *Encyclopaedia of Cryptography and Security* (pp. 1-9). Springer.

Elovici, Y., Waisenberg, R., Shmueli, E., & Gudes, E. (2004). A Structure Preserving Database Encryption Scheme. *SDM* , 28-40.

Jøsang, A., & Sanderud, G. (2003, January). Security in mobile communications: challenges and opportunities. In Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003-Volume 21 (pp. 43-48). Australian Computer Society, Inc.

Kayayurt, B., & Tuglular, T. (2006). End-to-end security implementation for mobile devices using TLS protocol. Journal in Computer Virology, 2(1), 87-97.

Lasley, J.R. & Palombo, B.J. 1995. When crime reporting goes high-tech: An experimental test of computerized citizen response to crime. *Journal of criminal justice.* 23(6):519-529.

Misra, S. K., & Wickamasinghe, N. (2004). Security of a mobile transaction: a trust model. Electronic Commerce Research, 4(4), 359-372.

Mynttinen, J. (2000, November). End-to-end security of mobile data in GSM. InTik-110.501 Seminar on Network Security. Helsinki University of Technology

Nielsen, J. (2000, March 19). *Why You Only Need to Test with 5 Users.* Retrieved from http://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/

Popa, R. A., Redfield, C. M., Zeldovich, N., & Balakrishnan, H. (2011). CryptDB: Protecting Confidentiality with Encrypted Query Processing. *SOSP* (pp. 1-16). Cascais: ACM.

RSA Security. (2013). *Securing Data at Rest: Developing a Database Encryption Strategy.* Retrieved May 3, 2013, from RSA Security: http://www.rsa.com/products/bsafe/whitepapers/DDES_WP_0702.pdf

Satchell, C. & Foth, M. 2011. Welcome to the jungle: Hci after dark. *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems.* ACM. 753.

Sesay, S., Yang, Z., Chen, J., & Xu, D. (2004). A Secure Database Encryption Scheme. *IEEE* , 49-53.

Shmueli, E., Vaisenberg, R., Elovici, Y., & Glezer, C. (2009). Database Encryption - An Overview of Contemporary Challenges and Design Considerations. *SIGMOD Record* (pp. 29-34). Providence: ACM.