# Securely Share – A Confidential Document Sharing System for Android Devices

# Introduction

## motivation

As the popularity of mobile communications devices increases, there is a growing tendency to use these as a convenient means of reviewing and revising documents on-the-move. Where these documents are of a confidential nature, particular attention must be paid to the fact that mobile devices are more vulnerable to compromise that traditional desktops, which are usually more extensively protected by the security measures implemented as part of an organization’s internal network.

There are multiple mechanisms for keeping files secure on company servers whilst allowing employees the necessary permissions to work collaboratively with sensitive data as required. As the mobile device culture becomes more prevalent in the workplace, the addition of Mobile Device Management (MDM) applications empowers users to also access corporate data via their mobile devices whilst still allowing IT departments to retain a degree of control over data security.

Thus, it is acknowledged that maintaining the security of confidential documents can be challenging, even with the weight of a corporate IT infrastructure behind it. In this project, we seek to address the issue of allowing groups of users from *different* organizations (i.e. with no shared IT infrastructure) to collaborate securely on confidential documents and furthermore, to access these documents via a smartphone or tablet computer whilst minimizing the risk of exposing sensitive information to a potential attacker.

## Project Aims

The primary aim of this project was to implement a scheme for the secure sharing of confidential documents between small (typically < 15) groups of collaborators, subject to the following constraints:

* Groups are self-organizing and represent multiple organizations, hence they cannot draw on the support of any central IT services.
* The documents involved are confidential in nature and hence should be encrypted both in transit and at rest.
* Group members wish to be able to access documents on a mobile device which is running the Android operating system.
* The solution devised should use only well-tested cryptographic techniques and standard libraries and should minimize the amount of trust to be placed in a third-party.

In pursuit of these aims we developed a solution called Securely Share, consisting of a detailed design of the security components of the system and a prototype android application (SecurelyShare) to provide a platform on which to implement and evaluate the various security features. It was acknowledged that, in a live setting, documents would usually originate on a PC rather than on a tablet device and thus the system would also need to a PC-based component. However, within the time constraints of the project it was considered infeasible to develop a fully featured system; our solution is submitted rather as a ‘proof of concept’.

## overview of report

The subsequent chapters of this report will deal with the design, implementation and evaluation of the project. Chapter 2 introduces some of the background material on key technologies used and presents an overview of the android applications reviewed as part of our preliminary research. In the light of this research, Chapter 3 presents a detailed analysis of the problem and expands the aims outlined in XXX into a more complete project specification, including details of the threat model against which we are attempting to defend. Chapters 3 and 4 deal with the solution design and implementation Need detail of rest of chapters here

# Further Background Material

In this chapter we will introduce some key aspects of the android architecture and its security features. We will also examine the features offered by the Dropbox AP and finally we will look briefly at some of the commercial applications which were reviewed as part of our initial research and which offer some features similar to Securely Share.

## android architecture and security features

### general architecture – event driven

### android components

### android permissions

## dropbox api

## tpm?

## review of other available applications

# Project Specification

# Problem Analysis /System Specification

What were the requirements

What were the options considered

What did we finally decided to attempt

## system (prototype) requirements

## security requirements

# Solution Design

This chapter will outline the key design decisions:-

* Why Dropbox and Sync API
* Group key sharing – review options considered and state why I chose the one I did
* Protecting the keys – what were the issues and why did I use Java Keystores
* The prototype – what did I decide to include and why

What about decision to aspire to best practice and impact on development time

## available hardware

## security design

## prototype design

### android app

### admin software

# Implementation

## android app - SecurelyShare

### cryptography

### dropbox integration

### key sharing

NB Include screenshots here

# Testing

# Project Management

## risk analysis

# Evaluation

Somewhere need to include challenge of starting from zero knowledge base

Lack of familiarity with event driven architechture

Evaluation of design choices

Strengths, weaknesses

## Further work

### prototype to production

### security enhancements

## overall product evaluation

## security evaluation

## project evaluation

## project management ??

# Conclusion