Managing Android devices in the Enterprise[1]

Understanding EMM, MDM and MAM

Authors

Jill Dove, MSc (Royal Holloway, 2015)

Geraint Price, ISG, Royal Holloway

Abstract

Mobile Devices are now ubiquitous in the enterprise. This enables flexible working and new business opportunities, but these mobile devices have also resulted in the complex problem of needing to protect and manage the enterprise data on them. An Enterprise Mobility Management (EMM) industry has grown to provide this remote management capability. There are specific challenges to remote management of Android devices, as Android fragmentation also affects its management interface. This article sets the context by summarising the complexities of contemporary mobile device management. It then focusses on the two approaches to the device management problem, which are Mobile Device Management (MDM) and Mobile Application Management (MAM), in the context of Android devices.

Introduction

IT departments have been remotely managing enterprise data on mobile devices since the early 2000s. Typically, employees accessed enterprise data (mainly email) using Blackberry devices. The devices were procured and managed by the IT department and were simple to lock-down according to the enterprise security policy, using a Blackberry management platform.

Contemporary Enterprise Mobility Management (EMM) has fundamental differences to contend with, including:

- The heterogeneous mobile device estate, which can include iOS, Android and Windows OS devices, has resulted in the enterprise needing a single platform to manage their entire estate, which is what the Mobile Device Management (MDM) industry has provided. An MDM solution manages the entire mobile device and locks it down in accordance with enterprise security policy.
- Growth of BYOD (Bring Your Own Device) and COPE (Corporately Owned Personally Enabled) has resulted in the enterprise needing to provide access to enterprise data on devices that it may not fully control. This has led to the need for containerisation solutions where enterprise data can be managed with minimal impact on the employees personal data. Mobile Application Management (MAM) is one popular type of containerisation, which may be also used in conjunction with MDM.

---

[1] This article is to be published online by Computer Weekly as part of the 2016 Royal Holloway information security thesis series. It is based on an MSc dissertation written as part of the MSc in Information Security at the ISG, Royal Holloway, University of London. The full MSc thesis is published on the ISG's website.

The article firstly describes the MDM use case, and the need for a management interface to provide remote management capability. The specific challenges of the Android management interface are then discussed. It then describes the conceptual MDM operating model and finally its implementation in a component architecture.

MAM solutions are described in terms of the three core capabilities they provide to enable managed apps: an Enterprise App Store; the ability to manage the apps' deployment lifecycle; and the ability to manage its security services.

Finally, the main differences between app-level containerisation and the alternative OS-level containerisation are discussed before briefly considering Android for Work (AFW) within this context.

Mobile Device Management

MDM solutions focus on managing the employee's entire mobile device, and enforcing its compliance with enterprise security policies, in order to provide access to enterprise resources, such as email, intranet and networked apps.

MDM solutions assume trusted users, who opt into having their mobile devices managed for the convenience of accessing enterprise resources from them, and are therefore prepared to comply with an Acceptable Use Policy. Typically, MDM solutions are not appropriate in use cases where there is a hostile user base, with users that are determined to subvert the controls that the MDM places on the managed device, for example a parental controls use case.

Management interface

When the first iOS and Android devices were released they were intended for the consumer market, and completely unsuitable for the enterprise as they could not be managed. Later versions, however, started to incorporate manageability, thus triggering the growth of the MDM industry.

As illustrated in Figure 1, for a mobile device to be manageable the OS needs to provide management functions that are accessible to a remote administrator. For example, session locking (or timeout) is a management function that a device user can configure, but the OS needs to offer a management interface to enable configuration by a remote administrator. The configuration set by the administrator must also override any user configuration.

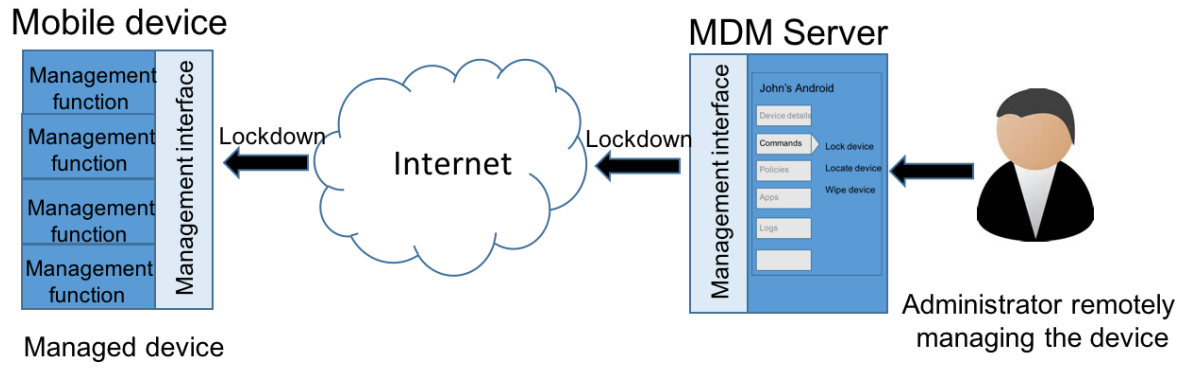

Figure 1: Administrators are able to remotely manage Managed Devices

Limited Android manageability

An Android management interface was first introduced in Android 2.2, which was called the Device Administration API. The original Device Administration API provided a minimal set of operations,

and there have only been limited changes to it in subsequent releases. This current set of operations includes:

- Authentication and session locking
    - Configure advanced password policy
    - Restrict unsuccessful authentication attempts
    - Reset password
    - Lock device, or set session timeout
- Encrypt device storage
- Disable camera
- Wipe data

Despite being minimal, this Device Administration API is still powerful, and therefore restricted to special Device Administrator apps. These Device Administrator apps form the basis of MDM solutions, effectively acting as the agent for remote administrators on the mobile device.

Fragmentation

Due to the limitations of the Device Administration API, many OEMs (such as Samsung, HTC and LG) have developed their own proprietary device management APIs, effectively extending the Device Administration API. Even within the same OEM the extensions may differ across model ranges. Typical extension features include:

- Restriction of device activities: including screen capture, copy to clipboard, ability to factory reset, use of microphone.
- Connectivity restrictions, including tethering/hotspot, NFC, Bluetooth, WiFi, USB debugging.
- Application lifecycle control – silent install, prevent install of blacklisted apps, prevent uninstall.
- Silent installation of certificates, and certificate key pairs.
- Configure services, such as VPN and Wi-Fi SSID.

In an enterprise that is able to restrict the variety of mobile device models within its estate, these extensions could provide significant management capability.

In practice, enterprises often have different Android mobile device models from multiple OEMs. This fragments the capability to manage the estate, e.g. it is only possible to restrict WiFi on some Android devices with the estate. This additional complexity may lead many enterprises to instead manage all their Android devices with the common minimal set of operations available in the Android Device Administration API.

Although Android 5 significantly extended the management interface, it was not an extension to the Device Administration API, and can only be used in conjunction with Android for Work (AFW), which is briefly described at the end of the article.

MDM Operating Model

As is now common in computer security, MDM solutions use a combination of *Prevent, Monitor and Respond* strategies to enforce device compliance with enterprise security policies.

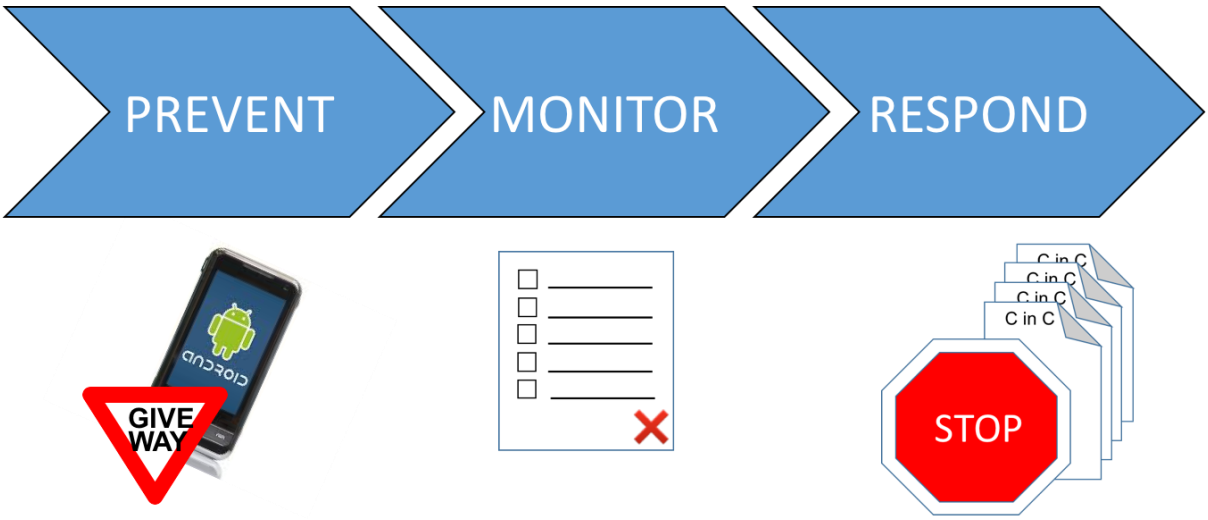

Figure 2: Prevent, Monitor and Respond strategy to EMM

Prevent: The MDM solution *attempts* to lock-down a managed device by enforcing security polices which restrict the device usage, but this may not be sufficient.

TWO REASONS PREVENTING ALONE IS INSUFFICIENT

Not all user actions are preventable on a managed device. For example, device rooting, or the installation of public apps on a device when only the Android Device Administration API is available.

A non-compliant user could remove the device from management at any time, using a number of techniques, including:

- Physically removing the MDM Agent.
- Deactivating the MDM Agent as a Device Administrator.
- Subverting the MDM Agent so it reports false information.
- Blocking communication between the MDM Server and the device.
- Factory resetting the device

Monitor: The managed device is continually monitored for its compliance status with respect to the security policies, and its last known contact with the MDM server.

Respond: The type of response, whether automated or manual, will depend on the severity of the non-compliance. For example, if it is detected that the device has been rooted then there may be an immediate automated response. Whereas if a device is out of contact for an extended period of time, there may need to be a manual process to establish if there is a legitimate reason for this, e.g. the user is on an extended absence from work and has their device switched off. Typical respond options, in approximate order of severity include:

1. Send a message to the user, informing them of the non-compliance and requesting they take action, otherwise there will be a more severe response.
2. Lock user out of device.
3. Prevent the user from gaining further access to any remote enterprise resources.
4. Wipe enterprise resources from the device (apps and their data).
5. Wipe the entire device.

MDM Architecture

An MDM solution consists of a Device Administrator app installed on the mobile device, which is the MDM Agent, and server-side components including the MDM Server and Gateway Proxy. There are four processes enabled by these components, which are described below:

Enrolment creates a managed device

Before a mobile device can be managed it needs to be enrolled with the MDM solution. Enrolment typically involves the user downloading and installing the MDM Agent from the Google Play Store, and then entering their identification and authentication details as assigned by their administrator.

Upon successful enrolment, the user is instructed to activate the MDM Agent as a Device Administrator, and confirm that they accept their device will be remotely managed. The MDM Agent is now able to access the Android Device Administration API, and manage the device.

Managed device Reporting and Apply Security Controls

There are two ongoing processes that are applicable to any managed device, these are:

- Apply Security controls: At the request of the administrator, the MDM Server informs the MDM Agent of the security controls to be applied to the device. Security controls include: lock down policies (such as password complexity), configuration settings (such as WiFi access points) and commands (such as wipe device).
- Reporting: The MDM Agent sends reports on the state of the device back to the MDM Server, from where the administrator can monitor it.

Authorise access to enterprise resources

When a user attempts to access an enterprise resource from an app on their mobile device, the app connects via a Gateway Proxy, which verifies the access is authorised. A factor in this authorisation decision will be the compliance status of the device, which is information that the Gateway Proxy receives from the MDM Server. Therefore, non-compliant devices may be unable to access enterprise resources until the user takes action to restore the device's compliance status.

Figure 3 summarises the key components and information flows between them, the purpose of each is described below:

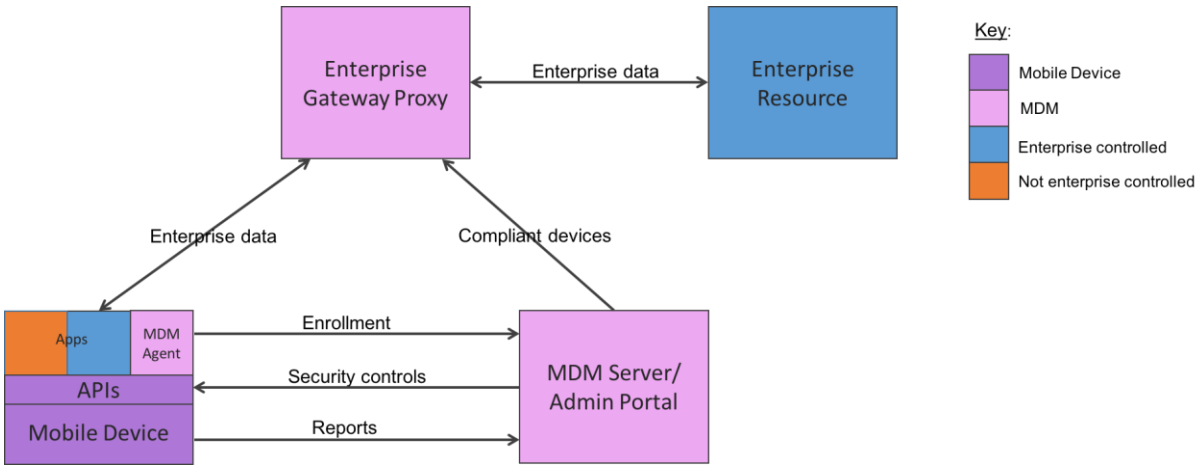

Figure 3: MDM Conceptual Architecture Model.

MDM Server and Admin Portal

This could be implemented on a server within the enterprise's own network or on a hosted cloud. MDM vendors offer hosted cloud solutions that can be both shared between multiple enterprises (shared SaaS) or dedicated to a single enterprise (dedicated SaaS).

An administrator accesses the MDM Server via the Admin Portal to carry out a number of tasks, including: user management, security control assignment, and compliance monitoring and response.

The administrator assigns security controls to mobile devices via the Admin Portal, but the MDM Server is responsible for the actual distribution of the security controls to the mobile device using a push notification service, such as Google Cloud Messaging (GCM).

MDM Agent

The MDM Agent has multiple functions, which include:

- receiving a security control message from the MDM Server, verifying its integrity, and executing the relevant Android API to apply the security control to the device.
- querying the Android APIs to gather state information about the device, which is reported to the MDM Server. Information collected typically includes:
    - Device information, e.g. model, Android version, IMEI etc.
    - Details of individual apps installed on the device.
    - Security policies that the Android OS is enforcing.
    - Location data
    - OS integrity status.
- providing a reliable communication path with the MDM Server.

Gateway Proxy

The Gateway Proxy is typically implemented on premise, as its purpose is to control access to enterprise resources.

Mobile Application Management

As employees started to bring their own devices into the enterprise, it was no longer feasible to lock-down the device functionality to the extent that the enterprise required. There were some early attempts to apply desktop virtualisation technologies to mobile devices, which minimised the lock-

down requirements, as enterprise data was never actually stored on the device. The fundamental disadvantage of desktop virtualisation technologies is that the mobile device has to always be online in order to access the enterprise data, and there are still too many scenarios in which this is not possible.

Containerisation offered a different solution, providing separation between enterprise and personal data on the device. In theory this enables the enterprise to manage its enterprise data on a device to which the user otherwise retains control. In practice, most enterprises will use it in conjunction with potentially less restrictive MDM security controls.

App-enabled containerisation is provided by Managed Application Management (MAM) solutions. An MAM provides an Enterprise App Store and the capabilities to manage both the apps' deployment lifecycle and its security services.

Enterprise App Store

The Enterprise App Store provides a catalogue of all the apps available to enterprise users, which includes both:

- Enterprise Apps: These are apps that the enterprise has developed for its own purpose, and does not want to make available on Google Play, therefore they are hosted by the Enterprise App Store.
- Public Apps: This is a whitelisting of apps available on a Public App Store such as Google Play. The Enterprise App Store only provides meta-data for these whitelisted apps, including the download location of the app on Google Play.

Managed app deployment lifecycle

The managed app deployment lifecycle can include approving public apps for the catalogue, adding enterprise apps to the catalogue, forcing app installation on devices, and removing apps (and their data) from devices.

The ability to manage apps on a mobile device, including forcing app installation, removing apps (and their data) and preventing app installation is dependent on the proprietary device management API, as the Android Device Administration API provides no app management functionality. Therefore, an MAM solution (partially) achieves compliance with the app installation security policy by using the Monitor and Respond approach. Thus, it may not be possible to stop a user downloading a non-whitelisted app, but their device may be rendered non-compliant it they do, potentially preventing further access to enterprise resources.

Figure 4 extends the conceptual model to include MAM.

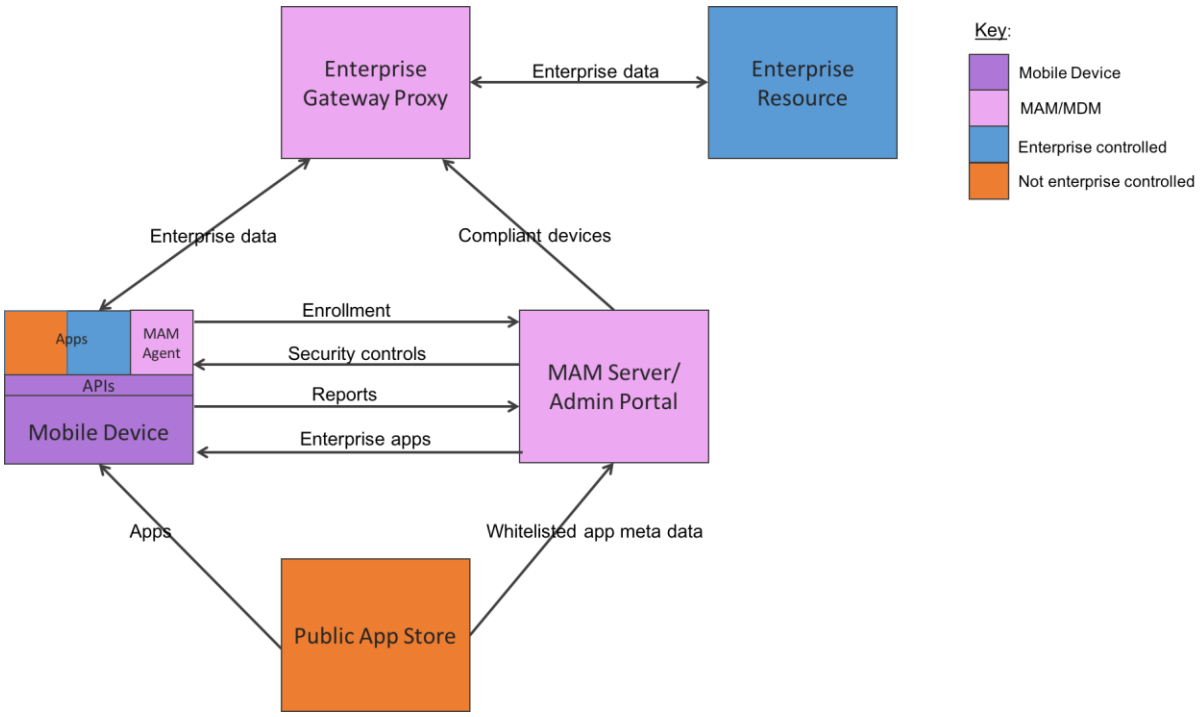

Figure 4: MAM and MDM Conceptual Architecture Model.

Security services

Before describing managed app security services it is useful to consider how security services apply to apps in general.

Although the Android platform architecture ensures that apps are sandboxed from each other, therefore prohibiting them from accessing each other's data, there are many other ways an app can potentially leak information. This could be as simple as a user capturing an app screen (saving the screen to the shared gallery), copying app data onto clipboard (and then into another app), or opening a file in an unauthorised (potentially malicious) app. Bundled apps may also behave in a similar manner.

Similarly, most apps do not have app-level access control, neither do they encrypt their data at rest. Therefore, an attacker with physical access to the mobile device potentially has access to the app data, especially if there are no other device-level security controls in place. Finally, if apps do not encrypt data in transit, they enable an attacker to eavesdrop on, or modify the data in transit over the network, which is relatively simple with open WiFi networks.

Managed apps address these vulnerabilities by preventing data from being saved to shared areas, in addition to providing a number of app-level security services, which can include:

- The ability to enable Data Loss Prevention (DLP) functions, such as disable clipboard, prevent screen capture, and files from being opened in unauthorised apps.
- User access control, for which the user has to provide an app specific password, and is only authorised to access the app if the device is compliant with the security policies. There are also single-sign-on extensions for a defined group of managed apps.
- Encryption of data at rest.
- Enable per-app Virtual Private Network (VPN), which protects data in transit. Unlike device-level VPNs, a per-app VPN creates a secure tunnel from the app to the Gateway Proxy, preventing potentially malicious apps on the device from attacking the enterprise.

- The ability to deploy configuration settings together with the app. These will typically be the domain names of remote enterprise resources to which the app connects. This simplifies access for the user, and is a deterrent from attempting to access invalid resources.

Approaches

There are two approaches to developing a managed app, both of which are tied to an MAM vendor. These are use of a Software Development Kit (SDK) or app wrapping.

The SDK approach involves a developer using an MAM vendor provided SDK, and is a suitable approach when the app is a new build. The APIs within the SDK provide the security services that the app needs to be manageable.

The wrapping approach takes an existing app, and wraps it in a layer of MAM vendor code to provide the security services. The wrapping is also intended to intercept any forbidden interactions that the app might attempt, and handle them gracefully. However, it is not possible to wrap an app from a public app store, as only the unsigned binary code from the original developer can be wrapped.

The disadvantage with the managed app approach to containerisation is that managed apps are intrinsically tied to a specific MAM vendor, which may result in vendor lock-in for the enterprise.

Alternative approaches to containerisation

OS-enabled solutions are an alternative approach to containerisation, but the disadvantage is that the enterprise needs to restrict the Android devices it supports, for example, to specific Samsung models, or Android devices with a modified Android OS.

Android for Work (AFW) was introduced in Spring 2015 and is available in several versions:

For devices running Android 5 and above, Google introduced an extended management interface, providing features such as: Data Loss Prevention restrictions, Chrome browser restrictions, Wi-Fi and VPN configuration and application controls. This API is accessible in two modes:

- Work Managed device, where the entire device is locked-down in accordance with enterprise security policies.
- Work Profile, where there is a workspace and a personal space, and this separation is enforced at the OS-level.

TWO APPROACHES TO OS-ENABLED CONTAINERISATION

Samsung Knox: Knox is marketed as a defence-grade mobile security platform with features including a trusted boot from a hardware root of trust and a workspace container. The workspace container was enabled by adapting SELinux for the Android platform, thus providing isolation of apps and data inside of the workspace container from those outside of it.

Virtual Machine: There have been a number of attempts to apply virtualisation technologies on the device itself, but they all require a modified Android OS. Systems such as OKL4 Microvisor from Open Kernel labs and vLogix Mobile from Red Bend use type I hypervisors (bare metal) for this.

There is also a version of AFW for pre-Android 5 devices, and as such, it is unable to take advantage of the extended Android management interface. It effectively provides app-level containerisation that is independent of any specific MAM provider.

To enable AFW the enterprise has to register its domain through the Google Admin console, and arrange for each user to have a work account under this domain. Integration points are provided between the EMM solutions and Google to simplify this process. If the enterprise already has Google Apps it already has a relationship with Google, however for other enterprises this will be new.

Summary

Mobile devices in the enterprise are here to stay, and the EMM industry has an important role to play in helping enterprises protect their enterprise data.

EMM solutions are aimed at broadly compliant users, who are prepared to have their devices managed in order to access enterprise resources. EMM solutions also recognise that it is not possible to prevent the occurrence of all breaches of security policy, so use a combined Prevent, Monitor and Respond strategy.

However, the ability to fully lock-down the entire mobile device in accordance with enterprise security policy is not possible for use cases such as BYOD and COPE. In addition, the fragmented Android management interface limits the ability to manage a heterogeneous estate with a single security policy.

App-level containerisation provides the ability for an enterprise to manage enterprise data on an uncontrolled device. Although in reality many enterprises will use a “light-touch” MDM, combined with MAM. A disadvantage of MAM is that the enterprise is then tied to a specific MAM vendor. AFW offers a potential alternative to this, also providing OS-enabled containerisation for Android 5 and above devices, but requires the enterprise to enter into a direct relationship with Google.

This article has been adapted from the dissertation “Evaluation of the Suitability of the Mobility Common Criteria Protection Profiles for Enterprise Mobility Management”. Following a review of EMM solutions, the dissertation describes the Mobility Protection Profiles (PPs) released under the Common Criteria standard. The Mobility PPs are intended to be used by vendors to achieve an independent evaluation of their mobility solutions, and the dissertation analyses whether they do actually reflect EMM solutions. The dissertation concludes that broadly speaking MDM is well reflected in the PPs, whereas MAM is not.

Biographies

Jill Dove is a Principal Consultant for a global system integrator. She has a broad range of business, technical and security experience in designing and delivering systems for clients in both the government and commercial sectors. She completed an MSc in Information Security from Royal Holloway, University of London in 2015, with a dissertation that was inspired by a recent role as an Integration Architect for a cloud-based content filtering security service aimed at enterprise mobile devices. She also has an MSc in Computing Science from Imperial College, University of London

Dr Geraint Price BSc (London), PhD (Cantab) obtained his B.Sc. in Computer Science from Royal Holloway University of London in 1994 and his Ph.D. from University of Cambridge in 1999. His Ph.D. dissertation analysed the interaction between Computer Security and Fault Tolerance. From 1999 to

2001, he was a Research Associate within the University of Cambridge, working on projects related to Denial of Service attacks in networks. In November 2001, he joined the Information Security Group (ISG) as a Research Assistant to work on a project funded by PricewaterhouseCoopers on the future of Public Key Infrastructures. From late 2002 to mid 2004 he worked on a research project funded by the PKI Club at Royal Holloway. In Sept 2004 Geraint was appointed as Lecturer in Information Security. Geraint has a strong interest in the practice of information security and leads the ISG's external engagement activities with business and government. Geraint is a regular attendee, panellist and speaker at a number of industrial fora, including I-4 and the ISF.