Towards Secure Mobile P2P Systems

James Walkerdine, Simon Lock
Computing Department
Lancaster University
Lancaster, LA1 4WA, UK
{walkerdi, lock}@comp.lancs.ac.uk

Abstract

The growth of mobile devices with near PC equivalent capabilities has brought with it the possibility of mobile P2P systems. However, the nature of mobile devices brings additional issues that need to be considered, especially for when considering system security. This paper presents our initial work on the EU funded PEPERS project that seeks to support the development of secure mobile P2P applications. A core part of this support is with the design and development of a mobile P2P security platform.

1. Introduction

The rapid growth of mobile devices with ever increasing functionality has brought with it the possibility of mobile Peer-to-Peer (P2P) systems. More recently, advances in wireless networking and mobile computing technologies, such as wireless LANs, wireless mesh networks and 3G cellular networks have further facilitated the migration of the P2P paradigm into wireless mobile computing. The combination of mobile and P2P technologies could be ideal for organisations that possess characteristics such as, decentralised management styles, geographically dispersed or highly mobile workforces, a wide range of computing and communications devices, etc.

The ad-hoc and heterogeneous nature of mobile P2P systems, however, can present significant challenges to application designers - particularly when it comes to security and privacy. Within a mobile P2P system, not only must encryption be employed, but also robust authentication procedures are required for connecting trusted and non-trusted devices with each other - a task that can be difficult in decentralised environments where connection to a trusted authority is not guaranteed.

A further problem within organisational environments is how to ensure company security policy is applied to all devices (for example personal laptops, PDA's or flash memory). Traditionally, security policies and enforcement mechanisms have predominantly been developed for centrally managed environments - additional techniques will be required to allow these to be applied to distributed, mobile and intermittently connected platforms.

The EU funded PEPERS (Mobile Peer-to-Peer Security Infrastructure) project seeks to address these concerns and will develop an infrastructure for the design, development and operational deployment of secure mobile peer-to-peer applications. A key aspect of this is the design and development of a generic mobile P2P security platform that is able to handle aspects of system functionality such as identity management, data confidentiality, authentication, access control, privacy and application validation. This paper presents an overview of the platform.

The paper is split into two main sections. We begin with a discussion of the requirements that were identified for the security platform. We then present our design of the platform and an example to illustrate how it can be utilised.

2. Requirements for a Mobile P2P Security Platform

To help in the design and evaluation of the security platform, the PEPERS consortium includes two industrial user partners who wish to use P2P technology to support their own business practices. One of these is a major international security firm who wishes to support their guards in communicating and sharing information with each other whilst on patrol. The second is a media company who wishes to allow their journalists, photographers and editors to work together in the creation of magazines. In both cases the companies wish to use P2P to support secure communication and data exchange between their personnel, who may be geographically dispersed or particularly mobile, and may only have access to mobile devices (i.e., the security guards and journalists).

One of the initial activities within PEPERS was to gather requirements to help inform the design of the security platform. This process comprised of three activities; studying scenarios of the user partner's work practices in order to identify desired functionality/ requirements; assessing available technologies and how they can impact on the platform; and drawing on our previous experiences from the P2P ARCHITECT project [1]. In the case of the latter activity, within P2P ARCHITECT we had investigated how best to support the development of dependable P2P systems. Within PEPERS we plan to build on this work, whilst focusing specifically on secure mobile P2P system development.

Requirement	Comments
Standard P2P functionality	Standard functionality expected by a P2P system/API, including: Peer Discovery, Publication, Searching, Unique Addressing, Peer Groups and Distributed storage
Standard security features	Fundamental aspects of the security platform including: Secure communication, Encryption, Authentication, Certificate management, Data integrity and Reputation mechanisms
Support the notion of data life	In some situations it is desirable for secure data (for example, door access codes) to cease existing should certain conditions apply (for example, mobile device leaving the relevant location or being on the device for too long a time)
Authentication based on temporal / geographic constraints	To provide lifetime control on temporary authorisation / delegation of authorisation by proxy to third parties
Be able to physically locate peers	Within a mobile environment not only can it be important to locate peers over the network, but also to locate them physically (for example, two guards with PDA's performing separate patrols in a warehouse)
Data fault tolerance/integrity support, failure recovery mechanisms	Utilising redundancy within the system to promote data fault tolerance. Allowing a peer to recover its data or state should it fail. This could include backup with restore, or roll back mechanisms
Discover detailed mobile peer information	Not just whether or not a peer is active on the network, but also the peers functionality and 'mobile' properties (for example, battery life, memory, etc). This is particularly important for mobile P2P systems (for example, you do not want to route messages to a mobile peer that is about to shut down due to low power)
Support different network topologies	P2P systems can be built on different type of underlying network topology. The security platform should be flexible enough to support this.
Device and protocol independent	Able to operate on a range of mobile devices (Mobile, PDA, Laptops, etc) and over a range of operating systems. Able to utilise different underlying protocols and technology (for example, JXTA).
Provision of reliable logging mechanism	To help support accountability and auditability the platform should provide secure logging mechanisms. Logs should possess access rights and be replicated across peers to ensure redundancy. Logging should be tamper proof.

Table 1 - A selection of key Requirements for the Mobile P2P Security Platform

As a result of these activities, a set of platform requirements were identified, a high level overview of which are presented in Table 1.

As can be seen, aside from the typical P2P functionality and security requirements, the elicitation process also revealed some interesting additional requirements. Of particular importance was the notion of data life and activity logging. The user partners wanted to make sure that security critical data could not remain on a device when outside the parameters of its intended use. For example, door access codes would only exist on the device within a certain time frame and within a certain location (the locality of the door). Likewise the users activities performed whilst using the system should also be securely and reliably logged, not only to support accountability but also to help detect tampering with the system.

The use of mobile devices also resulted in additional requirements that deal with determining their functional attributes and their physical location. This was particularly important for the security firm who wanted to be able to track the location of their guards. Likewise it is also important from a general P2P perspective where knowledge of a device's ability (and of those in the locality) can influence decisions such as where data should be routed, stored, etc. For example, to avoid routing messages via a device with low battery or limited bandwidth.

3. Designing a Mobile P2P Security Platform

A key issue in the development of security mechanisms for ad-hoc mobile P2P systems is the adaptation of existing (largely centralised) security approaches. Although the distributed and decentralised nature of many P2P systems can provide benefits such as resilience and reliability, the fact that the majority lack centralised control presents a challenge.

Existing work within the area of P2P security has tended to focus on individual aspects. Berket et al. [2] developed a distributed PKI mechanism for ensuring confidentiality, communication integrity and access control within P2P based information sharing systems. Agarwal et al. [3] developed the Secure Group Layer to support secure distributed communications. Likewise, JXTA/Poblano [4], Tran et al., [5], Ye et al. [6] have all attempted to tackle various aspects of P2P security. Although these developments are all beneficial they are limited in that they do not provide a complete security solution.

If secure mobile P2P applications are to be developed then it would be beneficial if a single, inclusive security toolkit could be drawn upon. Within PEPERS we are aiming to achieve this by developing a secure mobile P2P platform.

When designing a secure P2P runtime platform for mobile devices it is critical to take into account the properties and impact of the underlying technology. The use of mobile technology as a foundation for the

platform presents a number of interesting challenges, with some of the key ones being:

- *Communication cost* - Unless a special arrangement has been made with a network provider, using the mobile phone network to communicate data between peers will cost money. As a consequence, this form of communication should be kept to a minimum within any platform that is designed. Alternatives include using Wi-Fi and Bluetooth, however these possess more limited range capabilities.
- *Battery life* - Battery life is critical in mobile devices and needs to be maximised. Any form of computational or communication activity requires additional CPU cycles, which in turn draws on battery power. To prolong battery life such activities must be minimised and the design of any platform needs to reflect this.
- *Resources and capabilities* - Available mobile technology can differ significantly in the amount of resources they possess and in the capabilities they offer. Limited resources can impact on the feasibility of providing certain functionality within the platform. The design of the platform should therefore provide flexibility, allowing for lightweight implementations to be built where required.

The consideration of issues such as these can impact not only on the core design of the platform, but also on the choice of underlying topology that it would utilise. Certainly within a mobile system, adopting a fully decentralised topology is likely to be a costly approach (in terms of battery life) due to the additional overhead required for routing messages, managing system security, etc. Likewise mobile devices are not ideal for performing computationally heavy activities (for example, key generation for use in encryption), and so the platform may benefit from the inclusion of one or more super peers within the network. To cope with the different possible topologies that may be used, within PEPERS we are designing our platform to be topology independent.

4. The PEPERS Runtime Platform

Based on the identified requirements, a design for a more inclusive secure mobile P2P platform has been developed. The intention is for developers to build their own P2P applications on top of this platform and utilise the services it offers. As shown in the requirements, a key objective is for it to be generic enough to allow its use on different devices (mobile, PDA's, Laptops, etc), on different OS's (Symbian, Win CE, Linux etc), and being flexible enough to work with different protocols and topologies.

Figure 1 provides a diagrammatic overview of the proposed PEPERS Platform. Not only has the design been informed by the identified requirements, but also from the structures of existing P2P based platforms (such as JXTA). Principally a layered design has been used, with the security layer representing the most significant part of the platform. To help simplify the design of this layer, it has been broken down into a set of modules that represent key functionality that the runtime platform should provide (for example, encryption, authentication, etc). Interfaces for each module will be well defined allowing them to communicate with each other should one or more be used. Such interfaces will typically be APIs that can be accessed from within the security platform or, where

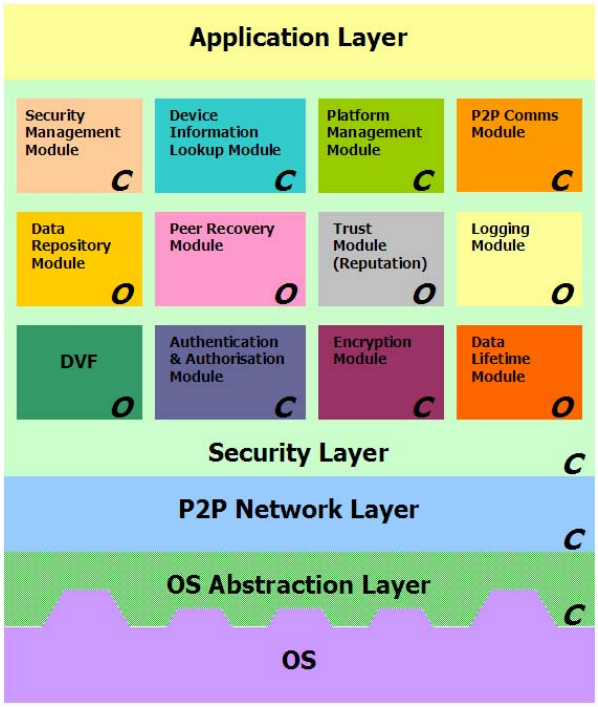

Figure 1 - High-level design overview of the PEPERS runtime platform

applicable, externally from the application layer. The modules have been categorised as being core (C), because they are core to the platforms operation, and optional (O), because they represent additional functionalities.

The development of a high-level design is necessary for a number of reasons. Firstly, it assists in the process of developing an inclusive security platform, but more importantly it provides the flexibility to work with existing security technologies and developments. It is not feasible, nor desirable, for PEPERS to develop all the technologies that would be required within a mobile security platform. Instead we want to draw on existing techniques and bring them together under a common umbrella. A consequence of this is that, certainly at this stage, for some modules desired functionality descriptions may be more important than detailed designs - to allow for the identification and selection of suitable existing technologies.

We will now provide a brief overview of this high-level design.

OS Abstraction Layer

The Operating System Abstraction Layer aims to bridge the gap between the underlying OS and the PEPERS runtime platform. It can be viewed as providing two key roles:

- *Abstracting OS access to ensure platform independence* - a requirement was for the platform to run on a variety of operating systems. The OS abstraction layer brings all operating systems up to the same standard and thus aids portability of the higher layers of the framework.
- *Expose relevant functionality that is provided by the underlying OS* - this is to reflect the fact some of the desired functionality may already by provided by the OS.

P2P Network Layer

As has been discussed in our previous work [7], numerous topologies and protocols exist upon which P2P systems can be built. These types of topology can provide different advantages and disadvantages depending on the requirements of the desired system. Because it is foreseen that different styles of topology may be adopted, the PEPERS runtime platform needs to be generic and flexible enough to cope with either.

This layer therefore seeks to provide a generic abstraction over the different P2P protocols and topologies that could be used within a mobile P2P environment. A common interface would be provided to the Security layer that could be accessed by the relevant modules (for example, the P2P Communication Module).

Security Layer

The Security Layer represents the bulk of the PEPERS platform. It contains the core functionality for providing a secure mobile P2P framework, on which applications can then be built.

Because of the breadth of functionality provided by the Security Layer, it has been broken down into a modular structure. The use of a modular approach provides a number of advantages:

- **Expandability** - new modules can be added at a later date.
- **Customisability** - the fact that some modules are optional means that they can be excluded from an implementation if, for example, they are not needed or device storage space is limited.
- **Interoperability** - the modular nature means that applications can interact with them independently or, potentially, modules can be overridden by similar functionality provided by the application. For example, if the application developers wants to use different encryption methods than what is provided by the PEPERS platform.
- **Modularity** – by separating functionality into modules and policing communication between those modules, it is possible to achieve a higher level of security. The interfaces between the modules act as firebreaks so that if one becomes compromised it is less likely to propagate to the other modules.

Security management module

The module administers the security for the whole runtime platform. It sets the security settings (where applicable) for the other modules within the security layer, and essentially acts as a 'front end' for these modules. This includes setting the lifetime for data files and peer group security properties.

Device information lookup module

The module allows the application (or PEPERS platform) to discover information about the device it is running on (e.g., resources, geographical location, etc). When used in conjunction with the P2P Comms module, this could be extended to discovering information about other devices within the peer group. This last point can be particularly important if decisions need to be made on where to route or backup data to. Obviously what information is made available will be dependent on the security settings of the device and PEPERS platform.

Platform management module

This module handles the general administration of the whole PEPERS runtime environment. It manages the modules within the Security Layer, allowing new ones to be integrated within the platform should they be developed. It also handles the initiation and the shutdown of the platform.

P2P Comms module

This module provides all the P2P functionality that

would be required by the platform. It handles publication, discovery, communication and general P2P management (for example, peer groups, and network organisation). This module would aim to be protocol independent in structure (and in the API it provides), and draw upon the functionality provided by the P2P Network Layer.

Data repository module

This module seeks to provide a secure and reliable data repository that is distributed across the P2P network. Within fully decentralised systems data would be replicated across peers so that it has a high availability (if peers go offline, etc). A more centralised approach could be adopted, although this will make this module largely redundant. The repository must be secure so that un-authorised access to the data cannot happen, and all access should be logged.

Peer recovery module

This module provides recovery facilities for peers should they fail. Such facilities would allow a peer to roll back to a previous state and also to retrieve lost data. For a fully decentralised system this will involve state and data information being duplicated across the network. For a more centralised system, such information could be stored in a central location.

Trust (reputation) module

This module provides trust mechanisms for the PEPERS platform. Reputation based mechanisms are the more commonly used method for representing trust within a P2P system and are likely to be most suitable for use in the runtime platform. For a fully decentralised system this will involve reputation information being duplicated across the network. For a more centralised system, such information could be stored in a central location. The Trust module, itself, would be generic in that it can be utilised not just by the runtime platform but also by the applications that are built on top. As a consequence, the Trust module also needs to be secure so that the reputation data cannot be tampered with or misused.

Logging module

This module provides a logging mechanism for the PEPERS platform that can be used to assist accountability and general security. As with the Trust module, the Logging module would be generic so that it can be utilised not just by the runtime platform but also by the applications that are built on top. It is also vital for the module to be secure - only authorised users should be able to view or edit relevant logs. Likewise logs should be encrypted to stop tampering. For the PEPERS platform all communication and internal activities of the platform would be logged. With a fully decentralised system logging information should be duplicated across the network and could utilise the Data Repository module. For a more centralised system, such information could be stored in a central location.

Dynamic Verification Framework (DVF)

Alongside the development of the mobile P2P security platform, the PEPERS project is also seeking to develop frameworks to support the static and dynamic verification of applications that are built to utilise it.

This module deals with the management of formalised descriptions of acceptable behaviour (created at design time) and the monitoring of the activities of the application/platform to ensure that it conforms to this description. The module will also have the ability to take appropriate actions when deviations from this behaviour are detected.

Authentication and Authorisation module

This module provides authentication and authorisation facilities for the PEPERS runtime platform. It supports the secure authentication of both users and devices, and enforces (role based) access control rights based on permissions for data and services. The mobile nature of the devices makes user authentication particularly important - for example, to stop a criminal who has stolen a device from retrieving security codes from it. Both centralised and distributed approaches should be supported to reflect the fact that mobile devices may not always be in contact with a central point.

Encryption module

This module provides facilities to support the encryption and decryption of data within the runtime platform. This can include data that is stored on the actual devices and also the communication between devices. It also includes support for the use of digital signatures. Both centralised and decentralised encryption mechanisms should be provided.

Data lifetime module

The notion of data lifetime was identified as an important feature from the user requirements. Data that could reside on a device may be so sensitive that it should only be accessed within certain environment conditions. For example, within a certain time period or a certain location. This module supports this feature and keeps track of the 'environmental conditions' for the device.

5. Implementing and Using the PEPERS Platform

The previous sections have provided a high-level design overview of the PEPERS platform. We have begun work on providing a more detailed design of some of these modules and this will be done along side initial implementation work. The PEPERS platform is to be implemented to work with Symbian OS 9 and will

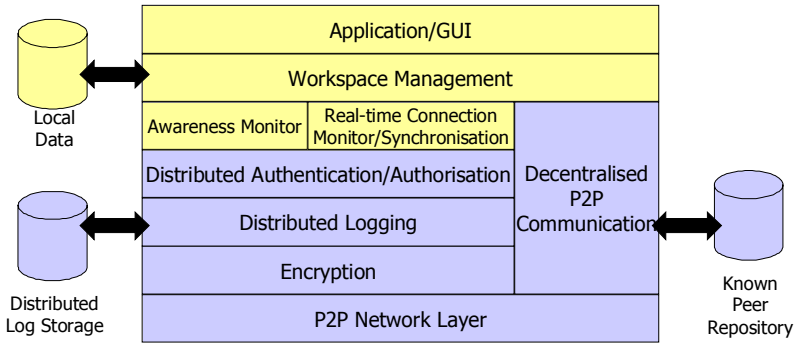

Figure 2 - A Shared Workspace Reference Architecture

be deployed and evaluated on a range of Symbian based mobile devices. That being said, the core aspects of the framework will be implemented in a platform independent manner allowing for their transference to other platforms at a future date. Where possible, existing technologies and developments will be utilised, with the mobile version of JXTA being used as a basis for P2P communication. Symbian OS also provides a powerful set of security features and these will be exploited by the PEPERS platform.

As we have already mentioned, the intention is for developers to utilise the PEPERS platform as a foundation for their own secure mobile P2P applications. To support this, within PEPERS there is also a second stream of work that focuses on supporting the development of secure mobile P2P applications. As part of this we have designed a development methodology that guides the developer through the various stages of the development process (requirements elicitation, design, implementation, etc), and a set of application reference architectures that can inform the design process.

The reference architectures provide guidance to developers on what modules (and structure) are typically required for different types of P2P mobile application domain and network topologies. Although these reference architectures can be utilised independently, they also indicate which modules of the PEPERS platform can be drawn upon to provide specific functionality. Figure 2 shows one of the reference architectures we have developed for the "Shared Workspace" application domain. The darker coloured modules indicate those that could have functionality provided for by the PEPERS platform.

Alongside this, tool support is also to be provided that will help guide designers through the development process, and help inform their decisions with regards to suitable network topologies and reference architectures. The development methodology, reference architectures and supporting tools will be described in a future paper.

6. Summary and Future Work

This paper has presented our work within the PEPERS project, which seeks to support the development of secure mobile P2P applications. A core component of the project is the development of a secure mobile P2P platform that can be used as a basis for the development of mobile P2P applications. This platform encapsulates a range of core functionalities that developers can then utilise within their own applications.

Initial work has focused on identifying requirements for this platform and the development of a high-level design - both of which have been presented in this paper. The focus will now move on to further detailing parts of the design and implementing the platform for use on Symbian OS based devices. The platform and development support before will then be used within two real world user partner developments.

The evaluation of the developed platform, along with the supporting application development methodology, will be documented in due course.

7. Acknowledgements

This work has been funded by the European Commission within the PEPERS project (FP6-26901)

8. References

[1] The EC funded P2P ARCHITECT project (IST-2001-32708). More information can be found at the URL http://www.atc.gr/p2p_architect/index.htm

[2] Berket, K., Essiari, A., Muratas, A., PKI-Based Security for Peer-to-Peer Information Sharing, In proceedings of P2P 2004, Zurich, Switzerland

[3] Agarwal, D. A., Chevassut, O., Thompson, M. R., Tsudik, G., An Integrates Solution for Secure Group Communication in Wide-Area Networks, in Proceedings of the 6th IEEE Symposium on Computers and Communications, Hammamet, Tunisisa, July 3-5, 2001, pp22-28

[4] Chen, R., Yeager, W., Poblano - A Distributed Trust Model for Peer-to-Peer Networks, Sun White Paper, http://www.jxta.org/docs/trust.pdf

[5] Tran, H., Hitchens, M., Varadharajan V. and Watters P., 2005, A Trust based Access Control Framework for P2P File-Sharing Systems, Proceedings of the 38th Hawaii International Conference on System Sciences, http://csdl.computer.org/comp/proceedings/hicss/2005/2268/09/22680302c.pdf

[6] Ye, S., Makedon, F., Ford, J., Collaborative Automated Trust Negotiation in Peer-to-Peer Systems. In the proceedings of P2P'2004. Zurich, Switzerland, August 2004

[7] Walkerdine, J., Melville, L., Sommerville, I., Dependability Properties of P2P Architectures. In the proceedings of IEEE P2P 2002, Sweden, 5th-7th September 2002