OVERCOMING CHANNEL BANDWIDTH CONSTRAINTS IN SECURE SIM APPLICATIONS

John A. MacDonald[1], William Sirett[2] and Chris J. Mitchell[1]

[1]Information Security Group, Royal Holloway, University of London, Egham, Surrey TW20 0EX, UK; [2]Smart Card Centre, Information Security Group, Royal Holloway, University of London, Egham, Surrey TW20 0EX, UK.

Abstract: In this paper we present an architecture based on a Java (J2SE, J2EE, J2ME and Java Card) platform supporting a secure channel from a Mobile Operator to the SIM card. This channel offers the possibility of end to end security for delivery of large data files to a GSM SIM card. Such a secure channel could be used for delivery of high value content that requires a high bandwidth channel – perhaps either rendered for user infotainment, or processed in the client Mobile Station (device and SIM card) for remote device management. Our methodology overcomes the bandwidth constraints of the SIM Toolkit Security scheme described in GSM standard 03.48. To validate our proposal we have developed code to create DRM and Web Service test scenarios utilising readily available J2ME, Java Card, J2SE and J2EE platforms, Web Services tools from Apache, the KToolBar emulator from Sun, and a Gemplus Java Card.

Keywords: J2ME, Java Card, SAT Security, SIM card, Web Service Security, DRM.

1. INTRODUCTION

Since its inception in September 1994, the SIM Application Toolkit (SAT) (3GPP TS 31.111, 2004; GSM 11.14, 2001) and SIM Toolkit Security (3GPP TS 03.48, 2001) have been used extensively. They are primarily used to securely transfer device and network management information and simple user applications (such as device independent, Operator-specific, power-on menus) to the SIM card.

These two independent concepts – the SAT and GSM standard 03.48, have been a very successful marriage (Guthery and Cronin, 2002). SAT allows applications resident within the tamper proof SIM card to initiate actions, whilst GSM standard 03.48 provides security services for any SMS message. Together they have been a critical enabler of many network management and revenue generating services deployed by GSM operators worldwide.

However the availability of large capacity SIM cards and high performance 2.5G and 3G devices means that this once-successful combination is now proving to be a constraint for the following reasons:

- GSM standard 03.48 uses SMS as the transport mechanism. SMS stands for Short Message Service, and is a way of sending a maximum of 160 characters (140 bytes) to and from mobile devices. Despite the GSM standard 03.48 allowing the concatenation of up to 255 such SMS messages to increase the payload, it is reported (Guthery and Cronin, 2002) that most operators limit this to approximately 5, i.e. a maximum payload of only 700 bytes. This is due to uncertain and indeterminate device operation when receiving such a large concatenated SMS message. With 128kB Java Card devices now routinely deployed, this bandwidth limitation is equivalent to less than 1% of the capacity of current generation SIM cards.
- Although a significant innovation in 1994, the SAT instruction list comprises only 31 proactive commands. These commands provide only limited control over the user experience, e.g. PLAY TONE, DISPLAY TEXT, GET INKEY, more appropriate for the text-based devices of the mid 1990's. The devices typical of today's 2.5G and 3G market would benefit from greater application customisation capability between device and SIM card.

A secure channel capable of downloading high bandwidth, high-value data within an application framework that provides rich control over the host device could thus be advantageous. This paper proposes such a channel.

2. THE JAVA FRAMEWORK

In recent years, Java enabled devices have become increasingly popular within the mobile market. Our proposal creates a high bandwidth secure channel for a Java platform, utilising Java Card (Chen, 2004) and J2ME (Topley, 2002) technologies.

In the GSM and UMTS system architectures, the Mobile Station (MS) may comprise two java components:

- the user device (often referred to as the handset). This typically comprises a Java runtime environment conforming to the J2ME Connected Limited Device Configuration (CLDC), complemented by additional classes from the Mobile Information Device Profile (MIDP). Java applications that run on MIDP compliant user devices are known as MIDlets.
- the SIM card provided by the network operator. The latest generation devices are typically UICC (3GPP TS 31.101, 2003) Java Cards where the SIM application (3GPP TS 31.102, 2003) is just one of the possible Java applications (ETSI TS 101 476, 2000) that the Java Card is capable of running. Java applications that run on Java Cards are known as Applets.

Recent work through the Java Community Process (JCP) has increased the utility of a mobile Java solution. The result of this technical innovation has been a rapid growth of complex, revenue generating, but largely fun-based J2ME applications within the gaming and entertainment sector. However, although some serious business applications exist (Itani and Kayssi, 2004), the Java environment has largely been ignored by the professional business and network management community because of concerns over security.

The fundamental problem is that the MIDlet runs within the Java implementation of the user device. The user device is unlikely to be trusted by the Operator to hold network level components and functions that protect valuable network assets. This distrust is likely to get worse as devices move from traditionally closed proprietary operating systems to more open operating systems capable of performing the file manipulation required by advanced 2.5G and 3G services. Securing a J2ME application currently requires the security keys, certificates and user identities to be stored within the user device. Many institutions within the Mobile Operator and Financial Service sectors are likely to consider this to be an unacceptable security risk.

In the GSM/3GPP mobile architecture, security and trust resides in two locations, the network HLR and the Operator issued tamper-resistant SIM card. The threat model is well researched and has resulted in the security services model at the heart of the GSM and 3GPP design (Hillebrand, 2002). What is needed is a methodology to extend this trust to the MIDlet environment.

3. THE SAT SECURITY FRAMEWORK

Our proposal builds on the dual capabilities of SMS Security and SIM Application Toolkit (SAT). The former is defined in Security Mechanisms for SIM stage 2 (3GPP TS 03.48, 2001). It provides end to end security services for an SMS message going to or coming from the SIM card. The SAT API allows a SIM card application to be informed of events (referred to as *event download*) by the user device, and to issue commands (referred to as *proactive command*) to the user device.

We use the proactive command `SET_UP_EVENT_LIST` to register for the `SMS_PP` event. On occurrence of such an event, or when commanded by the *Protocol Identifier* of the SMS *Mandatory Header*, the received SMS is passed on to the SIM application as a compound TLV (Tag Length Value) in the data field of an `ENVELOPE` APDU command. The SMS's *Command Header* specifies how the payload data is secured. The SIM application's response to the `ENVELOPE` command is then returned to the sender in a *Response Packet*. By using this approach, and by concatenating five SMS messages, it is possible to securely deliver around 700 bytes of data from Server to SIM card, receiving a proof of delivery in acknowledgement. We use this capability to securely transfer the Operator domain certificate and long term symmetric keys necessary to establish and secure our high bandwidth channel to the SIM card.

4. THE PROPOSED SECURE DATA TRANSFER TECHNIQUE

The MIDP 2.0 specification (JSR-118 JCP, 2002) introduces the concept of domains within a J2ME implementation (Block and Wagner, 2003). A Domain Protection Root Certificate controls application access to a domain. Any application within a domain enjoys a set of unique permissions and access to restricted and sensitive APIs provided by that domain. Before an application can be over the air (OTA) loaded into the Operator domain it must be digitally signed. The signature is checked against the SIM card resident root certificate and, if authorised, the application is loaded into the Operator domain of the untrusted device.

The Security and Trust Services API (JSR-177 JCP, 2004) provides an Operator domain J2ME application with the ability to access a connected trusted element (i.e. a SIM card within our scenario). Our proposal involves creating a J2ME and Java Card *Security Agent* application that is capable of implementing a secure high bandwidth channel between Server and SIM card endpoints. At no time does the J2ME application have access to any of the enabling cryptographic keys or functions. The bandwidth of the secure channel created by the *Security Agent* is only limited by the 2G/2.5G/3G network and the data rate resulting from the `ENVELOPE` APDU command.

The J2ME element of the *Security Agent* benefits from the processing power and I/O capabilities of the user device and has direct access to the results of secure SIM card computations executed by the Java Card Applet. Serious business applications such as DRM, e-commerce and securing web services can now be implemented by combining such J2ME and Java Card *Security Agent* applications. These business applications will additionally benefit from device vendor independence and potentially rapid rollout from OTA distribution and installation.

5. PROTOCOL

Full details of our protocol are provided elsewhere (MacDonald et al., 2004). It uses both symmetric and asymmetric cryptographic techniques to provide the authentication, integrity and confidentiality services required to support a secure high data bandwidth channel from Server to SIM card. Our protocol has been designed on the assumption that the user device and SIM card are pre-issued and in the field. We assume that neither user device nor SIM card contain pre-installed application code to create the desired secure high bandwidth channel.

We choose to use symmetric rather than asymmetric cryptography for authentication and key agreement. Performance is critical in a mobile system and overhead must always be minimized wherever possible (Blanchard and Trask, 2002). The long term secret key K_{SC} shared by the Server and the SIM card, and used to support the secure channel to the SIM, is confidentially distributed from the Server to the SIM card endpoint with authentication and integrity services provided by GSM standard 03.48.

STEP 1 *Install MIDlet into Operator Domain, and Applet into SIM*

The first step is to prepare the SIM card so that the MIDlet can be installed within the Operator domain of the J2ME device. The MExE (3GPP TS 23.057, 2003) security framework, like other specialist services and applications that use the mobile network purely as a transport mechanism, relies on signature verification before the MIDlet can be installed within the target domain. We use GSM standard 03.48 to securely transfer the Operator Domain public key certificate Cert_{OP_DOM} to the SIM card. The MExE J2ME implementation on the user device receives the signed MIDlet. Successful verification of the signature using the public key in Cert_{OP_DOM} provides data origin authentication and integrity of the MIDlet JAD and JAR files. The

Security Agent MIDlet is installed in the Operator domain of the user device with full JSR 177 permissions, allowing APDU commands to be issued to SIM card resident Applets. To initiate installation of the Java Card *Security Agent* Applet, the MIDlet starts an http session with the server, and supplies it with the SIM card's unique identifier. The server responds with the SIM card Applet code, integrity protected with a MAC computed using the shared secret K_{SC}. The MIDlet *Security Agent* then transfers this data to the SIM card via the `Envelope` APDU command. An on-card installer application verifies the MAC and hence the origin authentication and data integrity of the Applet. If there is any discrepancy the installation process ceases; otherwise the *Security Agent* Applet is securely installed. This results in the creation of an applet instance and its registration with the Java Card runtime environment.

Note that neither MIDlet nor Applet carry any secret keys or other private data. Hence code encryption is not necessary. Integrity services to protect the MIDlet and Applet against virus insertion attack whilst in transit are required and are provided by the use of Digital Signatures and MACs respectively.

STEP 2 *Perform mutual entity authentication*

At some time later, i.e. after the http session of STEP 1 has closed and both Applet and MIDlet are installed, the Operator may choose to securely download bulk data from the Server to the SIM card. Before this begins, both endpoints verify each other's identity by means of a mutual entity authentication protocol. We use a three-pass mutual authentication protocol based on MACs and nonces, as specified in ISO/IEC 9798-4 (ISO/IEC9798-4, 1999).

STEP 3 *Set up session keys to protect bulk data transfer*

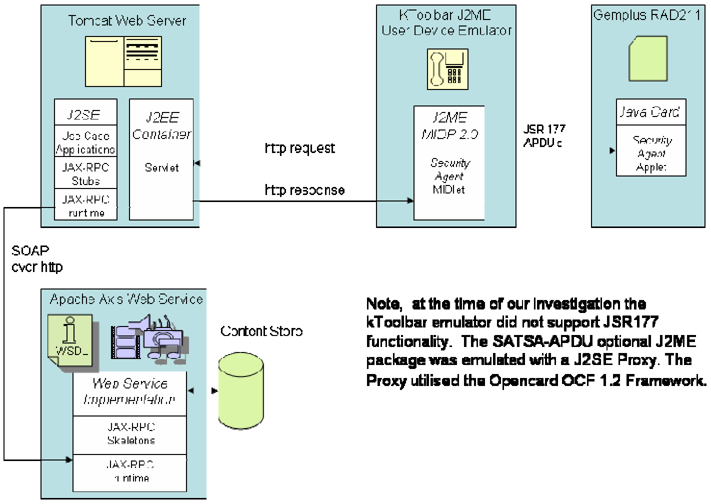
Following mutual entity authentication, both Server and SIM card derive session Integrity (IK) and Confidentiality (CK) keys to provide security services to protect the bulk data transferred between Server and SIM card. Both Server and SIM card Applet will contain identical functions $f1$ and $f2$ to calculate the session cipher and integrity keys using the nonces r_S and r_C exchanged as part of the authentication protocol in step 2, and the long term shared secret K_{SC}, as follows:

$$CK = f1_{K_{SC}}(r_S \| r_C) \text{ and } IK = f2_{K_{SC}}(r_S \| r_C).$$

Once session keys have been established, the bulk data may be transferred between Server and SIM, encrypted for confidentiality with *CK* and concatenated with a MAC computed using *IK* for data origin authentication and integrity.

6. PROOF OF CONCEPT PROTOTYPE IMPLEMENTATION

To validate our proposal we have constructed the Proof of Concept model of Figure 1, based on readily available open source tools:



Note, at the time of our investigation the kToolbar emulator did not support JSR177 functionality. The SATSA-APDU optional J2ME package was emulated with a J2SE Proxy. The Proxy utilised the Opencard OCF 1.2 Framework.

[

Figure 1. Proof of Concept Prototype Implementation

- A J2EE Servlet web application performs the Mobile Operator function and is packaged as a WAR file (Web Application Archive) for easy deployment on a Tomcat Apache Web Server.
- The J2ME Client is emulated by the Wireless KToolbar (Sun Microsystems, 2003) from Sun Microsystems, running our *Security Agent* MIDP 2.0 MIDlet on the reference J2ME implementation.
- The SIM card function is provided by a Gemplus GemXpresso RAD 211 Java Card with crypto package, connected to our demonstration environment via a USB card reader.
- A Web Service application communicates with the Mobile Operator function using SOAP over http. We used the jax-rpc API together with tools from Apache Axis to create the service WSDL and deploy the Web Service on a Tomcat Server.

The demonstration environment of our proof of concept model is implemented in J2SE. J2SE provides the necessary Java Swing classes for monitoring the various use case applications tested on our model. The model is designed so that each phase of a specific use case is initiated manually and monitored by visual feedback through the use of J2SE's GUI `LayoutManager` class and `ActionListener` interface.

A framework that provides a high bandwidth secure channel between Server and SIM card is a significant enabler for application deployment. For demonstration purposes we have deployed Digital Rights Management and Web Services Security platform applications onto this framework. We now review how these operations leverage the proposed high bandwidth channel and framework; full details are provided in MacDonald and Mitchell (2004a) and MacDonald and Mitchell (2004b).

6.1 Proof of Concept DRM Applications

Digital Rights Management is an attempt to use technology to limit piracy and copyright violation of digital media (Litman, 2001). DRM solutions typically separate the Digital Asset from the Rights Object. Often the Asset is encrypted with a secret key. A separately delivered Rights Object includes both the secret key for decryption of the Asset and the user permissions. The user must therefore have both the Digital Asset and the Rights Object to render the digital asset. Without the Rights Object, the Digital Asset may be peer to peer distributed, and transferred from device to device.

Our framework is ideally suited to such a content centric DRM application. Typically the user device would be notified of the receipt of such an encrypted Digital Asset by the asset's MIME type. This would invoke the *Security Agent* to store the encrypted Digital Asset in the, relatively plentiful, device memory, and then securely fetch the Rights Object from the Rights Fulfilment Server. The Rights Object would be securely transferred to the SIM card via our high bandwidth channel. The Digital Asset is recoverable only by the entity that holds the Rights Object.

At some time later, upon user request, the Digital Asset would be transferred (perhaps streamed) to the SIM card for authorisation, where it is decrypted and streamed back to the device for consumption and rendering. The Rights Object, comprising the root decryption key of the Digital Asset and the current user permissions, would always reside on the secure SIM card. Such an implementation greatly reduces the network resource cost incurred by the practice of streaming each rendering instance of the Digital Asset over the WAN 2.5G and 3G network.

6.2 Proof of Concept Web Services Security Application

Our framework can be extended to provide a mobile Operator endorsed authentication and payment platform for web services. For this vertical application the Server Servlet also provides the stub to the remote *Web Service* which is packaged as a WAR file and deployed on the Tomcat Server. Described by its WSDL we use the JAX-RPC API from Apache Axis to create the stubs to the service interface. Communication between Servlet and Web Service is according to the SOAP protocol using http as the transport mechanism.

In this application a high level user discovery process is provided by the J2ME *Security Agent*. User service selection initiates the mutual authentication step concluding with the creation of the high bandwidth secure channel between Server and SIM card. The Server may now issue an authentication token followed by an authenticated payment token when the user decides to consume the service. The authenticated payment token is exchanged for the web service, and the content associated with the service provided to the Server using SOAP over http. The service content may now be securely transferred to the SIM card via the high bandwidth J2ME and JavaCard *Security Agent* channel. This implementation provides:

- the user with a high level service discovery interface plus anonymity from Web Service providers;
- the Mobile Operator with a pivotal role and revenue generating opportunity in the provision of a web services security and payment platform;
- the Content Provider with a secure, scaleable distribution channel.

Note that, whilst it is possible to use the J2ME and Java Card *Security Agent* to create a secure high bandwidth channel, it may not be desirable to use these entities for service rendering and consumption. Extending connectivity to the personal area network of the J2ME device is particularly straightforward given the availability of the SAT `OPEN CHANNEL` proactive command.

7. CONCLUSION

In this paper we have introduced a novel approach to securely transfer large data files from an application server to the mobile device SIM card. Our approach is based on a Java solution and overcomes a potential bandwidth restriction of the current GSM standard 03.48 and SAT Security process. We present a protocol and methodology that allows the secure channel to be created on capable, but unprepared, devices and SIM cards that are already issued. We have modelled our proposed solution and protocol using open source tools and indicate how it can be extended to apply to future application implementations such as DRM and Web Services.

ACKNOWLEDGEMENTS

The work of the first author was funded by Telefonica Móviles, España, S.A.U., Pza. de la Independencia, Madrid, Spain. The work of the second author was funded by the Smart Card Centre, Royal Holloway, University of London. This support is gratefully acknowledged.

REFERENCES

3GPP TS 03.48 (2001). *Technical Specification Group Terminals; Security Mechanisms for the SIM application toolkit; stage 2*. http://www.3gpp.org.

3GPP TS 23.057 (2003). *Technical Specification Group Terminals; Mobile Execution Environment (MExE); Functional description; Stage 2*. http://www.3gpp.org.

3GPP TS 31.101 (2003). *Technical Specification Group Terminals; UICC-terminal interface; Physical and logical characteristics*. http://www.3gpp.org.

3GPP TS 31.102 (2003). *Technical Specification Group Terminals; Characteristics of the USIM application*. http://www.3gpp.org.

3GPP TS 31.111 (2004). *Technical Specification Group Terminals; USIM Application Toolkit(USAT).* http://www.3gpp.org.

Blanchard, C. W. and Trask, N. (2002). Wireless security. In Temple, R. and Regnault, J., editors, *Internet and Wireless Security*, number 4 in BT Exact Communications Technology Series, chapter 9, pages 146-170. IEE, London.

Block, C. and Wagner, A. C. (2003). *MIDP 2.0 Style Guide*. Addison-Wesley, London.

Chen, Z. (2004). *Java Card Technology for Smart Cards*. Addison-Wesley, London.

ETSI TS 101 476 (2000). *Digital cellular telecommunication system (Phase 2+); Subscriber Identity Module Application Programming Interface (SIM API); SIM API for Java Card; Stage 2 (GSM 03.19).* ETSI, http://www.etsi.org.

GSM 11.14 (2001). *Digital cellular telecommunications system (Phase 2+); Specification of the SIM Application Toolkit for the Subscriber Identity Module-Mobile Equipment (SIM-ME) interface*. ETSI, http://www.etsi.org.

Guthery, S. B. and Cronin, M. J. (2002). *Mobile Application Development with SMS & the SIM Toolkit*. McGraw-Hill.

Hillebrand, F. (2002). *GSM and UMTS: The creation of global mobile communications*. John Wiley & Sons, Ltd.

ISO/IEC 9798-4 (1999). *Information technology – Security techniques – Entity authentication – Part 4: Mechanisms using a cryptographic check function*. International Organization for Standardization, http://www.iso.org, 2nd edition.

Itani, W. and Kayssi, A. (2004). J2ME application-layer end-to-end security for m-commerce. *Journal of Network & Computer Applications*, 27:13-32.

JSR-118 JCP (2002). *Mobile Information Device Profile, v2.0 (JSR-118)*. Sun Microsystems, http://java.sun.com.

JSR-177 JCP (2004). *Security & Trust Services API (SATSA) (JSR-177)*. Sun Microsystems, http://java.sun.com.

Litman, J. (2001). *Digital Copyright*. Prometheus Books, New York.

MacDonald, J. A. and Mitchell, C. J. (2004a). Content centric DRM for mobile vertical market. Information Security Group, Royal Holloway, University of London – Internal paper.

MacDonald, J. A. and Mitchell, C. J. (2004b). Web services security platform using mobile operator credentials. Information Security Group, Royal Holloway, University of London – Internal paper.

MacDonald, J. A., Sirett, W. G., and Mitchell, C. J. (2004). Establishing a security context between server & SIM: A 3 pass mutual AKE protocol with signature & MAC. Information Security Group, Royal Holloway, University of London – Internal paper.

Sun Microsystems (2003). *Wireless Toolkit, Version 2.1*. Sun Microsystems, http://java.sun.com/products/j2mewtoolkit.

Topley, K. (2002). *J2ME In a Nutshell*. O'Reilly.