UNIVERSITY OF KUOPIO

End-to-End encryption of SMS messages

Marko Hassinen

Marko.Hassinen@cs.uku.fi

University of Kuopio, Department of Computer Science

UNIVERSITY OF KUOPIO

Content

- Motivation
- The Mobile Java environment
- Application structure
- Design details
- Analysis

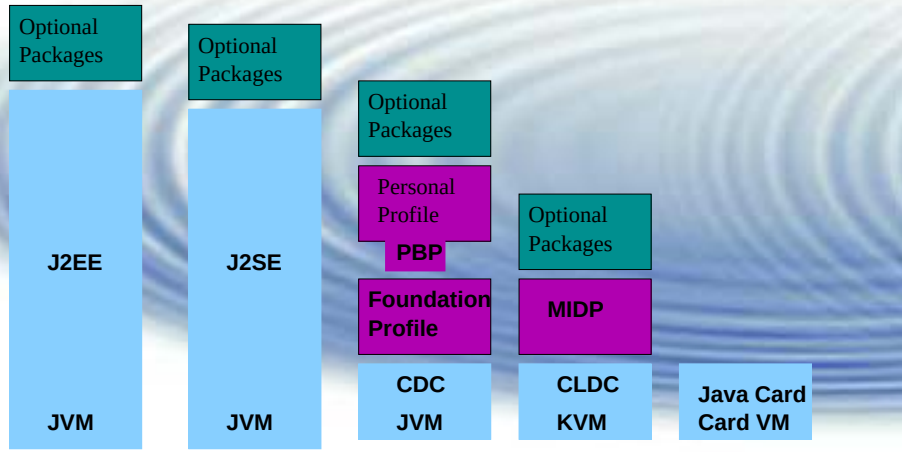
UNIVERSITY OF KUOPIO

Motivation

- Confidential data
- Wrong numbers
- Lost devices
- Curious operators
- Unencrypted traffic

UNIVERSITY OF KUOPIO

Java environments



PBP = Personal Basis Profile

UNIVERSITY OF KUOPIO

J2ME

- Java Micro Edition
- Small Java enabled Devices (Mobile phones, PDAs, Set-top boxes)
- Configurations
- Profiles
- Optional APIs (packages) like WMA etc.

UNIVERSITY OF KUOPIO

Configurations

- VM and class libraries (minimal core)
- CDC Connected Device Configuration
  - Set-Top boxes, High-end PDAs
  - Network-connected consumer and embedded devices
- CLDC Connected Limited Device Configuration
  - Devices with more limited processing power
  - Mobile phones etc.

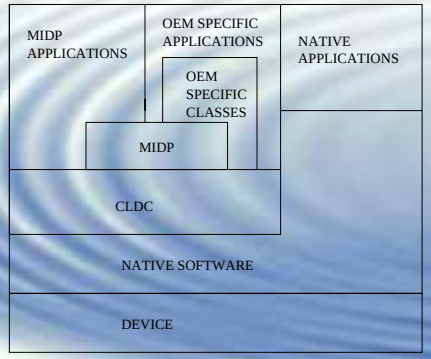
UNIVERSITY OF KUOPIO

Profiles

- Profiles refine the java environmet suitable to the device.
- Mobile Information Device Profile (MIPD)
  - JSR 37, JSR 118
  - Profile for mobile devices

UNIVERSITY OF KUOPIO

MIPD



UNIVERSITY OF KUOPIO

MIDlet

- J2ME applications are called MIDlets
- An application that conforms to the CLDC and MIDP API's
- Written in Java, compiled, pre-verified and packed into a JAR package
- Installation into the device is device dependent
- Our security solution is a MIDlet

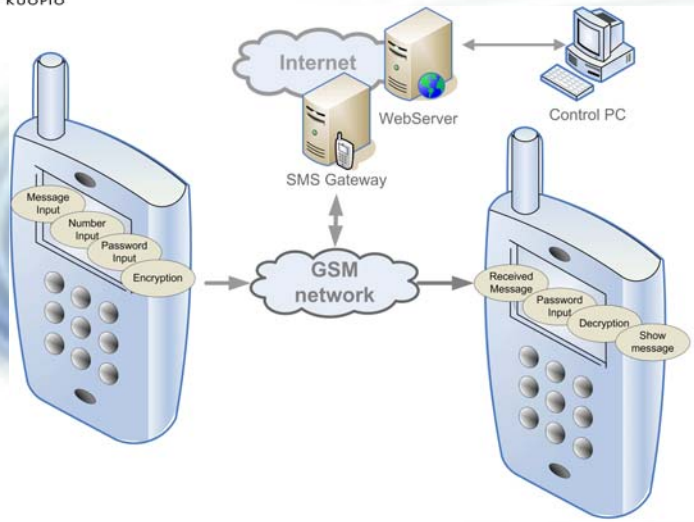
UNIVERSITY OF KUOPIO

Why Java

- Portability
- Standards
- Security
- Ready packages

UNIVERSITY OF KUOPIO

Application overwiev



UNIVERSITY OF KUOPIO

Application provides

- Privacy of message traffic
- Authentication (not strong)
- Safe storage
  - Sent and received messages
  - Contacts

UNIVERSITY OF KUOPIO

Simulator screenshots

UNIVERSITY OF KUOPIO

WMA

- Wireless Messaging API
- Specification JSR-120
- GCF (Generic Connection Framework)

```
MessageConnection clientConn =
  (MessageConnection)Connector.open(protocol + outgoingaddress);
```

- For SMS messages the protocol is `sms://`.
- The outgoing address is of the form `number:port`, for example `+5550000:4321`.

UNIVERSITY OF KUOPIO

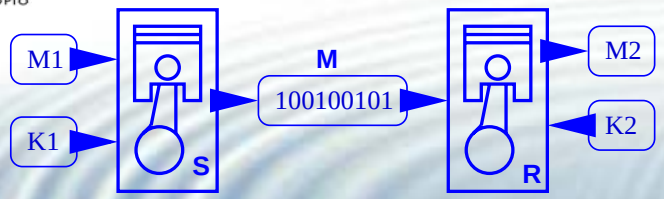
WMA

- When receiving an SMS message, the application has to open a listening connection for incoming messages.
- In SafeSMS this is done with the *openListeningSMSConn* method.

```
protected boolean openListeningSMSConn()
{
  try
  {
    conn = (MessageConnection)Connector.open(sProtocol+":"+sPort);
    conn.setMessageListener(this);
  }
  catch(Exception e)
  {
     conn = null;
  }
  return (conn != null);
}
```


UNIVERSITY OF KUOPIO

Encryption



Using a key $K1$, sender S encrypts message $M1$ and sends M. The recipient R receives message M and using the key $K2$ decrypts it into $M2$. Provided that S and R share the password, so that $K1 = K2$, and no errors happen during transmission, R will recover the message so that $M1 = M2$.

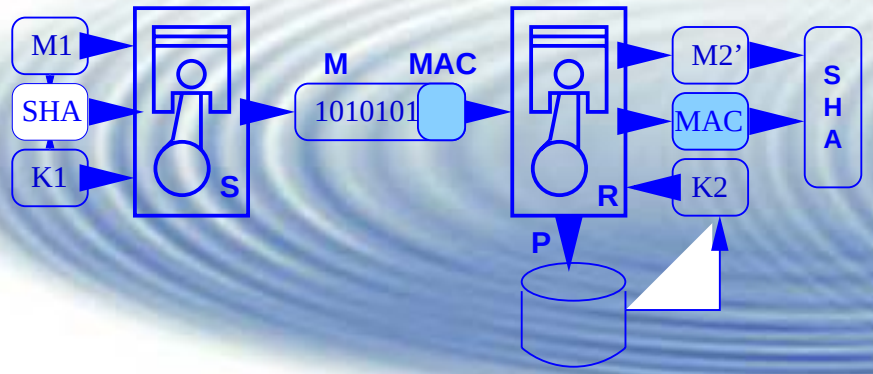
UNIVERSITY OF KUOPIO

Message Authentication

- SHA (Secure Hash Algorithm)
- Produces a 160-bit message digest
- The input is processed in 512-bit blocks
- The message has to be padded so that the length is congruent to 488 modulo 512
- A 64 bit presentation of the message length

UNIVERSITY OF KUOPIO

Message authentication



UNIVERSITY OF KUOPIO

Record Store

- Non-volatile storage
- Records can be added, deleted and replaced
- Contacts
- Sent and received messages
- All values encrypted

UNIVERSITY OF KUOPIO

Localization

- The locale, that is used in the device, can be found with the method *System.getProperty("microedition.locale")*.
- The program has five languages: Finnish, English, French, German and Italian.
- The user can choose the language.
- The language selection is stored and used in the future.

UNIVERSITY OF KUOPIO

Push registry

- MIDP 2.0 introduced a construct called *Push registry*
- Enables an application to register inbound connections to AMS (Application Management Software).
- Allows AMS to start the application on arrival of a message that is meant for the application.
- SafeSMS uses the static method of describing the Push Registry entry in the application descriptor file (JAD).

UNIVERSITY OF KUOPIO

JAD

- Java Application Descriptor
- Application details (standard)
  - JAR size
  - MIDlet name, class
  - JAR URL
  - ...
- Parameters (user defined)
  - Encryption method
  - Port
  - Language
  - MAC

UNIVERSITY OF KUOPIO

JAD

```
MIDlet-1: SafeSMS, /icons/icon.png, SafeSMS
MIDlet-Description: End to end SMS encryption
MIDlet-Jar-Size: 50334
MIDlet-Jar-URL:  http://katiska.uku.fi/~mhassine/OTA/SafeSMS.jar
MIDlet-Name: SafeSMS
MIDlet-Permissions:
 javax.microedition.io.PushRegistry,
 javax.microedition.io.Connector.sms,
 javax.wireless.messaging.sms.receive,
 javax.wireless.messaging.sms.send
MIDlet-Push-1: sms://:54321,marko.SafeSMS,*
MIDlet-Vendor: UKU
MIDlet-Version: 0.0.1
MicroEdition-Configuration: CLDC-1.1
MicroEdition-Profile: MIDP-2.0
SafeSMS-EncMethod: Blowfish
SafeSMS-Lang: Eng
SafeSMS-Port: 54321
SafeSMS-MAC: Yes
```


UNIVERSITY OF KUOPIO

Deployment

- Bluetooth
- IrDA
- Cable
- OTA (Over The Air)
- Installation device dependent

UNIVERSITY OF KUOPIO

Analysis

- Privacy and integrity
- Authentication
- PKI for non-repudiation and strong authentication
- Size, 50kB is acceptable
- Tested on actual devices, speed is acceptable