文章编号:1001 - 9081(2006)08 - 1813 - 04

基于 Java ME 的点到点短信加密应用

杨建强
(襄樊学院 电气信息工程系,湖北 襄樊 441053)
(yjq@ mail. whut. edu. cn)

摘 要:针对短信传输的安全问题,给出了一种基于 Java ME 的短信加密应用解决方案。该方案用旧密钥加密新密钥来完成密钥的传递和更新,针对短信服务的特点,采用有效措施确保双方能够进行正常的短信通信:若对方未能收到新密钥或其确认短信,则允许重复发送新密钥及其确认短信;若双方在当前密钥生存期的 1/3 这段时间内未能及时更新密钥,则继续使用旧密钥通信。给出了密钥更新过程中特殊情况的处理方法。从安全性、可靠性方面对这些方法和措施进行了分析,说明了应用中需要注意的事项。

关键词:Java ME;短信服务;加密;密钥更新
中图分类号: TP309.7 **文献标识码**:A

Application of point-to-point SMS encryption based on Java ME

YANG Jian-qiang
(*Department of Electrical Information Engineering, Xiangfan University, Xiangfan Hubei* 441053, *China*)

Abstract: To keep SMS from interception in transmission, a solution of application of point-to-point SMS encryption based on Java ME was presented. This solution used encrypted SMS to deliver the new key. Considering the characteristics of SMS, effective measures were adopted to ensure that both sides could correctly communicate with each other through SMS. The new key SMS and its confirmation SMS were allowed to be sent again in case they were not received. Both sides would continue using the old key if the triplicate lifetime of the new key had elapsed and the new key was not updated. The means of solving special cases that occurred in the course of key updating were specified. Meanwhile, these measures were also analyzed in terms of security and reliability, and cases in need of attention were pointed out.

Key words: Java ME; Short Message Service(SMS); encryption; key update

传统的点到点短信从一部手机发出后,传送到运营商的短信中心,再发送给相应的手机,其信号就暴露在空中,存在着一定的安全隐患。利用 Java ME 技术,对发出的文本短信加密,使包括运营商在内的任何第三方都无法看到短信的内容,则为我们提供了一个传递机密信息的安全通道。不过,要真正实现安全短信通信,必须解决好密钥管理问题。我们开发的基于 Java ME 的短信加密程序较好地解决了这个问题。该程序的典型应用是在商业合作伙伴之间传递机密信息,或在关系密切的两个人之间传递敏感信息。结合我们开发的实例,本文对利用 Java ME 技术实现短信加密中的有关问题,特别是密钥管理问题进行了探讨。

1 基于 Java ME 的短信加密程序简介

Java ME 包括连接设备配置(Connected Device Configuration,CDC)体系和有限连接设备配置(Connected Limited Device Configuration,CLDC)体系,目前得到广泛支持的是 CLDC 体系[1]。我们开发的短信加密程序属于 CLDC 体系,它基于移动信息设备框架(Mobile Information Device Profile,MIDP)2.0,并使用了可选包无线消息应用编程接口(Wireless Messaging API, WMA)[2],以及第三方加密包 Bouncy Castle。该程序允许与多个用户进行加密短信通信。接收加密短信的手机号码是预先设定的,每一个号码都有一个与之关联的密钥,密钥会定期更新,新密钥通过短信(已加密)传递给对方。

基于 MIDP 和 CLDC 的 Java 程序也叫 MIDlet,一个或多个 MIDlet 组成一个 MIDlet 套件[3]。短信加密程序(MIDlet 套件)包含三个 MIDlet:

1)EncryptMIDlet 用于加密和发送短信。它也用于发送新密钥和阅读已保存的加密短信。无论是发送正常的短信还是新密钥,该 MIDlet 都会在要发送的内容前面附上一个专门的标识串然后再加密。对于正常的短信,使用“MSG”标识(以下简称此类短信为 MSG 短信),对于密钥,使用“KEY”标识(以下简称此类短信为 KEY 短信)。附上标识串的目的是便于接收方识别收到的短信是正常的短信还是新密钥。

2)DecryptMIDlet 用于接收和解密短信。它也用于保存收到的 MSG 短信。当收到 KEY 短信时,该 MIDlet 也用于回复一个仅包含“CON”串的确认短信(以下简称 CON 短信)。当收到 CON 短信时,该 MIDlet 将自动回复一个仅包含“DOE”串的确认短信(以下简称 DOE 短信)。

3)AddPhoneMIDlet 用于设置接收加密短信的手机号码(预设号码)、与该号码相关联的密钥以及密钥的更新时间等信息。这些信息存储在由记录管理系统(Record Management System,RMS)管理的记录库(Record Store,RS)中。RMS 是 MIDP 提供的一种用于持久性存储和检索数据的机制[3]。AddPhoneMIDlet 只在添加或删除预设号码时使用。

短信加密程序中有四个 RS,其中一个用于存放收到的

收稿日期:2006 - 02 - 15;**修订日期**:2006 - 04 - 27 **基金项目**:湖北省教育厅科学技术研究资助项目(Q200625002)
作者简介:杨建强(1971 -),男,湖北枣阳人,副教授,硕士研究生,主要研究方向:信息安全、无线网络技术.

MSG 短信,另外三个存放有密钥。存放有密钥的三个 RS 分别是:1)mainRS,它存放通信双方已经确认的密钥;2)acceRS,它存放暂时还没有得到确认的新密钥。这两个 RS 的结构是一样的,每条记录都包括五个字段:预设号码,密钥,下一次密钥更新的启始时间,密钥更新间隔和主号码标识(如图 1 所示);3) timeRS,其每条记录包括 3 个字段:预设号码(PhoneNumber),CON 短信或 DOE 短信的时间戳(TimeStamp),上一个密钥生存期内的旧密钥(Key)。

PhoneNumber	Key	InitialTime	Interval	MasterID
48 bit	128 bit	64 bit	7 bit	1 bit

图 1 记录格式

短信加密程序使用了 MIDP 2.0 新增的 Push(推送)特征[4]。Push 使 DecryptMIDlet 能够在手机收到来自预设号码的短信时才被激活,其他时间则可以不运行。

2 短信加密程序中的密钥管理

密钥管理包括密钥的产生、存储、备份、装入、分配、保护、更新和销毁等[5]。本文只就几个重要方面进行探讨。

2.1 密钥的产生

用户每添加一个新的预设号码,AddPhoneMIDlet 都会要求用户给出一个与该号码相关联的密钥(初始密钥),如果忽略它,则使用默认密钥。默认密钥仅仅为最初的短信传递,特别是新密钥的传递提供一种相对安全的通道。在为新添加的号码分配一个初始密钥时,通信双方只需在特定时间内各自输入协商好的 8 个字符,AddPhoneMIDlet 自动把它们转换成对应的初始密钥。所有的密钥都有生命周期,在为某个号码更新密钥时,EncryptMIDlet 利用用户所输入的字符串,当前时间,以及一个序列数和一个随机数来产生新的密钥[1]。为增强密钥的不可预测性,用户至少随机输入 8 个字符。

2.2 密钥的存储和保护

如前所述,短信加密程序中的密钥存储在 RS 中,这种存储是持久的,即不会因为 MIDlet 的关闭或手机断电而丢失。在 MIDP 1.0 中,RS 只能由创建它的 MIDlet 套件访问,MIDP 2.0 也允许 RS 在多个 MIDlet 套件之间共享[3]。在短信加密程序中,我们把四个 RS 都设置成只能由创建它们的 MIDlet 套件访问,这保证了存储在 RS 中的密钥不会被其他程序访问到。

2.3 密钥的更新

2.3.1 有关说明

在为短信加密程序添加新的预设号码时,除了需要给出与该号码相关联的密钥,还需要给出密钥的更新间隔和更新启始时刻,以及确定通信双方哪一方号码是主号码。所有这些信息都需要双方事先确定。更新间隔以天为单位,默认更新天数为 3 天。为避免影响正常通信,默认更新启始时刻为深夜 0 点。AddPhoneMIDlet 依据所给的更新间隔和更新启始时刻计算出下一次密钥更新的启始时间。主号码的作用是:当双方同时产生新的密钥时,主号码方产生的密钥将优先得到采用。当 MasterID 为 1 时,表示新号码为主号码。AddPhoneMIDlet 把得到的有关数据写入到 mainRS 中。

2.3.2 密钥更新过程中密钥发送方与接收方之间的交互

密钥更新的基本过程是:密钥发送方(假定号码为 PNa)发送 KEY 短信给密钥接收方(假定号码为 PNb),密钥接收方收到 KEY 短信后回复 CON 短信给密钥发送方,密钥发送方收到 CON 短信后立即更新与密钥接收方即 PNb 通信的密钥(以下简称 PNb 的密钥),然后回复一条 DOE 短信给密钥接收方。密钥接收方收到 DOE 短信后立即更新密钥发送方即 PNa 的密钥。上述过程如图 2 所示。在当前密钥生存期的 1/3这段时间内,如果 PNa 没有收到来自 PNb 的 CON 短信,PNa 可能会给 PNb 重复发送 KEY 短信,如果 PNb 没有收到来自 PNa 的 DOE 短信,PNb 可能会给 PNa 重复发送 CON 短信。PNa 每收到一个 CON 短信都会回复一个 DOE 短信。需要说明的是,尽管可以利用短信中心反馈的状态报告替代 CON 短信和 DOE 短信,但该状态报告是可选的[6],并且 WMA 也没有提供获取该状态报告的方法[2]。

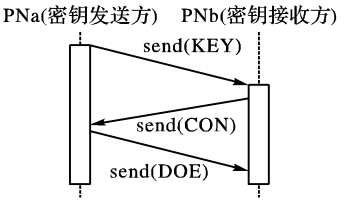

图 2 密钥发送方和接收方之间的交互

2.3.3 密钥发送方更新密钥

当 PNa 给 PNb 发送 MSG 短信时,EncryptMIDlet 立即将当前时刻与 mainRS 中的 PNb 的 InitialTime 进行比较。如果当前时刻小于 InitialTime,说明 PNb 的密钥还处于生存期内,不需要更新。如果大于 InitialTime,说明密钥需要更新,则 EncryptMIDlet 首先查看 acceRS 中是否存在 PNb 的记录,如果存在,说明或者短信加密程序已经为 PNb 生成过新密钥,或者曾经收到过来自 PNb 的新密钥,否则 EncryptMIDlet 提示用户更新密钥。如果用户同意更新,则 EncryptMIDlet 利用用户所输入的字符串等信息产生一个新的密钥并与标识串“KEY”连接起来,然后使用 mainRS 中的 PNb 的密钥(旧密钥)加密连接后的位串并发送出去。紧接着,EncryptMIDlet 为 PNb 计算下一次密钥更新启始时间,即计算新的 InitialTime。新的 InitialTime 根据下面的公式计算得到:

$$\text{新 InitialTime} = \text{旧 InitialTime} + n \times \text{Interval} \quad (1)$$

其中,旧 InitialTime 和 Interval 来自 mainRS,n 是一个使新 InitialTime 最小并且大于当前时间的整数。最后,EncryptMIDlet 把 PNb、新密钥、新 InitialTime,以及来自 mainRS 的 Interval 和 MasterID 存放到 acceRS 中。

在收到 PNb 回复的 CON 短信之前,PNa 仍然可以给 PNb 发送 MSG 短信,也能够接收来自 PNb 的 MSG 短信,短信加密程序将使用旧密钥加密或解密这些短信(包括 CON 短信)。一旦收到 CON 短信,DecryptMIDlet 首先把 PNb、CON 短信的时间戳,以及 mainRS 中的 PNb 的旧密钥存放到 timeRS 中,然后用 acceRS 中的 PNb 的记录更新 mainRS 中的 PNb 的记录,然后删除 acceRS 中的 PNb 的记录,接着自动回复一条 DOE 短信给 PNb(加密密钥来自 timeRS),最后提醒用户 PNb 的密钥已经更新。

在短信通信中,短信中心偶尔会重复发送短信(原因是短信中心发出短信后并没有收到接收方的确认信号[6]),这些重复短信时间戳是一样的。另外,短信加密程序也允许用户在没有收到对方回复的情况下重复发送 KEY 短信或 CON

短信。如果在当前密钥(这里指新密钥)的生存期的 1/3 这段时间内再次收到来自 PNb 的 CON 短信,则 DecryptMIDlet 首先把该 CON 短信的时间戳与上一次收到的 CON 短信的时间戳进行比较。如果二者不相等,则 DecryptMIDlet 自动回复一条 DOE 短信,否则不回复。如果继续收到来自 PNb 的 CON 短信,DecryptMIDlet 采用同样的处理方法。无论如何,在这段时间内,每次收到来自 PNb 的 CON 短信 DecryptMIDlet 都会自动更新 timeRS 中 PNb 的 TimeStamp 字段。当前密钥的生存期从上一次密钥更新的启始时间开始,直到跨过 Interval 为止。

新密钥发出之后,如果 PNa 没有收到来自 PNb 的 CON 短信,分三个阶段,短信加密程序的处理方法如下:1)在当前密钥的生存期的 1/3 这段时间内,每次给 PNb 发送 MSG 短信时,EncryptMIDlet 都会提示用户"对方尚未确认更新密钥,是否重发新密钥?",如果用户同意,则 EncryptMIDlet 自动发送一条 KEY 短信。2)一旦超过当前密钥的生存期的 1/3 但没有超过当前密钥的生存期,如果再次收到来自 PNb 的 MSG 短信或者给 PNb 发送 MSG 短信,则短信加密程序首先用 acceRS 中 PNb 的 InitialTime 更新 mainRS 中 PNb 的 InitialTime,然后删除 acceRS 和 timeRS 中的 PNb 记录。3)在超过当前密钥生存期的 1/3,但没有超过当前密钥生存期的这段时间内,如果既没有收到来自 PNb 的 MSG 短信,也没有给 PNb 发送 MSG 短信,那么,一旦超过了当前密钥的生存期,如果再次收到来自 PNb 的 MSG 短信或者再次给 PNb 发送 MSG 短信,则短信加密程序首先删除 acceRS 中 PNb 的记录,然后根据公式(1)为 PNb 计算新的 InitialTime 并保存到 mainRS 中。

2.3.4　密钥接收方更新密钥

当 PNb 收到一条来自 PNa 的短信时,DecryptMIDlet 首先解密该短信(密钥来自 mainRS),然后根据短信的标识串采取相应的动作。如果标识串是"KEY",则DecryptMIDlet首先根据公式(1)为 PNa 计算新的 InitialTime,再从中减去一个 PNa 的密钥更新间隔,即得到 PNa 的上一次密钥更新的启始时间,然后将该启始时间与 KEY 短信的时间戳进行比较。如果 KEY 短信的时间戳较小,说明该 KEY 短信中的密钥是当前密钥生存期之前的其他密钥生存期内的密钥(即过期的密钥),则 DecryptMIDlet 忽略该 KEY 短信,否则 DecryptMIDlet 继续查看 acceRS 中是否存在 PNa 的记录。如果不存在,说明该 KEY 短信是第一次收到,则 DecryptMIDlet 提醒用户"对方要求更新密钥,是否同意?"。如果用户不同意,则 DecryptMIDlet 丢掉收到的密钥,并且不回复任何短信给 PNa。如果用户同意更新密钥,则 DecryptMIDlet 根据公式(1)并利用 KEY 短信的时间戳为 PNa 计算新的 InitialTime。此时,要求公式(1)中的 n 是一个使新 InitialTime 最小并且大于 KEY 短信的时间戳的整数。然后 DecryptMIDlet 把 PNa、新密钥、新的 InitialTime,以及来自 mainRS 的 Interval 和 MasterID 存放到 acceRS 中。另外,DecryptMIDlet 也把 PNa、KEY 短信的时间戳,以及来自 mainRS 中的 PNa 的旧密钥存放到 timeRS 中。所有这些操作完成之后,DecryptMIDlet 自动回复一条 CON 短信给 PNa(加密密钥来自 timeRS)。

如果 acceRS 中已经存在 PNa 的记录,则说明或者曾经收到过同样的 KEY 短信,或者已经为 PNa 生成过新密钥。如果是第一种情况,则 DecryptMIDlet 把该 KEY 短信的时间戳与上一次收到的 KEY 短信的时间戳进行比较,看其是否相等,具体处理方法与 2.3.3 小节中 PNa 收到重复的 CON 短信时的处理方法相似。如果是第二种情况(此时 timeRS 中无 PNa 的记录),则 DecryptMIDlet 首先检查 PNa 的 MasterID 是否为 1,如果为 1,则 DecryptMIDlet 删除 acceRS 中 PNa 的记录,即删除本地产生的新密钥,然后按 acceRS 中不存在 PNa 的记录的情况处理收到的 KEY 短信。如果 PNa 的 MasterID 为 0,则忽略收到的 KEY 短信。上述流程如图 3 所示。

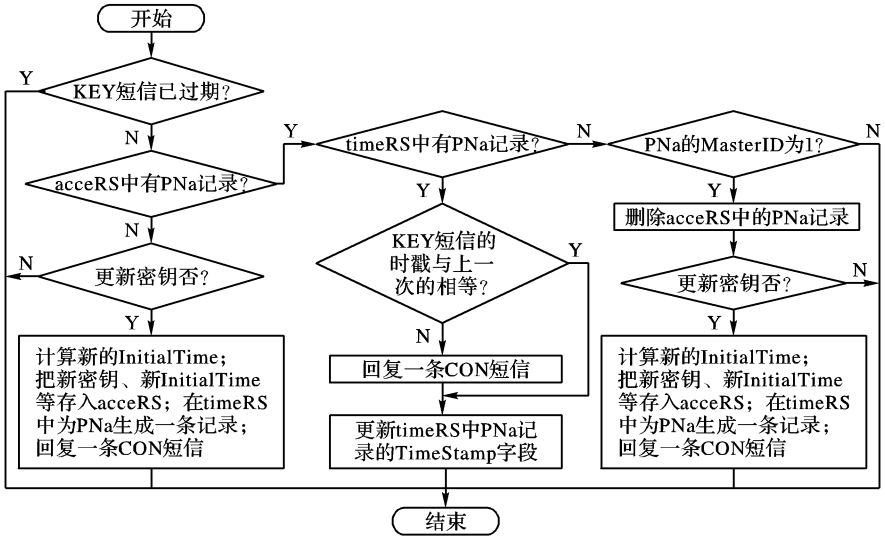

图 3　收到 KEY 短信时的处理流程

在收到 PNa 回复的 DOE 短信之前,PNb 仍然可以给 PNa 发送 MSG 短信,也能够接收来自 PNa 的 MSG 短信,短信加密程序将使用旧密钥加密或解密这些短信(包括 DOE 短信)。一旦收到 DOE 短信,DecryptMIDlet 首先用 acceRS 中 PNa 的记录更新 mainRS 中 PNa 的记录,然后删除 acceRS 中的 PNa 的记录,最后提醒用户 PNa 的密钥已经更新。此后,所有发往和来自 PNa 的 MSG 短信都将使用新密钥加密和解密。如果没有收到 DOE 短信,在当前密钥生存期的 1/3 这段时间内每次给 PNa 发送 MSG 短信时 EncryptMIDlet 都会提示用户重发 CON 短信。一旦超过当前密钥生存期的 1/3,其处理方法与 2.3.3 小节中 PNa 没有收到 CON 短信时所采用方法相同,在此不再赘述。

不管怎样,无论是密钥发送方还是密钥接收方,一旦超过当前密钥生存期的 1/3,都会忽略对方发来的 KEY、CON 或

DOE 短信,并且 DecryptMIDlet 或 EncryptMIDlet 也会在收到或给对方发送短信时自动删除 timeRS 中的包含有对方号码的记录(如果存在的话)。

2.3.5 密钥更新过程中特殊情况的处理说明

1) PNa/PNb 无法解密除首次收到的 CON/DOE 短信以外的其他 CON/DOE 短信

一旦收到来自 PNb 的 CON 短信并且回复了 DOE 短信,默认情况下,PNa 就用新密钥解密来自 PNb 所有短信。如果 PNb 在回复 CON 短信之后并没有收到来自 PNa 的 DOE 短信,它可以继续给 PNa 发送 CON 短信,以便收到 DOE 短信并更新密钥。由于 CON 短信是用旧密钥加密的,因此,除非 PNa 还没有收到过来自 PNb 的 CON 短信,否则 PNa 将无法解密这些 CON 短信。对于这个问题,短信加密程序的处理办法是:首次收到来自 PNb 的 CON 短信之后,在新密钥的生存期的 1/3 这段时间内,每收到一条来自 PNb 的短信,PNa 首先用新密钥解密该短信,然后看其前 3 个字符是否是"MSG",如果是,则认为是 MSG 短信,如果不是,则再次用旧密钥重新解密,在网络和手机都正常工作的情况下,它应该是一条 CON 短信。

另外,由于 PNb 可能在首次收到 DOE 短信之前又发送了 CON 短信,因而有可能继续收到其他 DOE 短信。与 CON 短信相似,除首次收到的 DOE 短信外,PNb 无法解密来自 PNa 的其他 DOE 短信。对此,短信加密程序采用与处理其他 CON 短信类似的方法。如果是 DOE 短信,则忽略它。

2) 连续收到来自 PNa 的多个不同的 KEY 短信

由于短信存在被延迟接收的可能,如果 KEY 短信的有效期超过了密钥的更新间隔,PNb 有可能会连续收到来自 PNa 的多个不同的 KEY 短信。这些 KEY 短信中的密钥分属于不同的密钥生存期。短信加密程序通过用上一次密钥更新的启始时间与 KEY 短信的时间戳进行比较来辨别收到的密钥是否属于过期的密钥。如果属于过期的密钥,则忽略收到的 KEY 短信。详情参见 2.3.4 小节的有关论述。

3) 发送新密钥后不久收到对方发来的新密钥

当前时刻大于 InitialTime 时,通信双方都能够产生新的密钥并发送给对方,因此很容易出现一方发出新密钥后又收到对方发来的新密钥的情况。如果出现这种情况,主号码方产生的密钥将优先得到采用。详情参见 2.3.4 小节的有关论述。

3 安全性、可靠性分析

短信加密程序使用 AES 算法加密和解密短信,密钥 128bit,其安全保障是非常可靠的。如果想增加密钥的长度,只需更改 RS 中的密钥字段的长度和修改少量代码即可。短信加密程序定期更新密钥,并且用旧密钥加密新密钥来进行密钥传递,这延长了破译密钥所需的时间。在密钥更新过程中,短信加密程序利用用户所输入的字符串(至少 8 个字符),当前时间,以及一个序列数和一个随机数来产生新的密钥,这种方法显著增强了密钥的不可预测性,增加了密码分析的难度。另外,通信双方的身份完全可以由彼此的短信号码确认,此时,网络运营商充当了可信任的第三方的角色。需要说明的是,在密钥更新过程中,尽管可以利用短信的时间戳和一个 nonce 来防止重放攻击[5],不过,考虑到目前这种攻击的可能性非常小,我们并没有在程序中使用这些技术。

短信的时间戳指出了短信到达短信中心的时间,精确到秒[6]。在密钥更新过程中,发送方发送 KEY 短信之后,立即根据本地时间计算新的 InitialTime。接收方根据收到的 KEY 短信的时间戳计算新的 InitialTime。一般情况下这两个 InitialTime 是一样,因为 KEY 短信从发送方传到短信中心的时间通常是非常短的。但是,如果 KEY 短信要经过多个短信中心,那么就有可能会因短信堵塞和传输延迟而导致 KEY 短信的时间戳与发送方计算新的 InitialTime 时所依据的本地时间有较大的时间差。如果该时间差超过了密钥更新间隔,就会造成双方所计算的新的 InitialTime 不相同,进而影响密钥的更新和正常通信。不过,由于通常该时间差与密钥更新间隔相比微不足道,因此,一般情况下它对正常通信没有影响。除了短信延迟带来的上述时间差,密钥发送方、密钥接收方,以及短信中心的时钟之间的不同步也可能会带来上述问题。不过,基于同样的理由,它们对正常的通信也几乎没有影响。

目前手机短信服务还只是尽力传送,并不保证传递的准确性和即时性[6]。为防止因短信丢失或接收方处于服务区外等原因而导致接收方无法正确和及时地收到 KEY、CON 以及 DOE 短信,短信加密程序允许用户重复发送 KEY 或 CON 短信。另外,若时间超过了当前密钥生存期的 1/3,而密钥发送方因短信延迟或丢失等原因没有能够及时更新密钥,则密钥发送方和密钥接收方都会更新对应的 InitialTime 并继续使用旧密钥,从而使双方能够继续使用相同的密钥通信。

在 2.3.5 小节中,我们给出了密钥发送方 PNa 无法解密除首次收到的 CON 短信之外的其他 CON 短信的处理办法。一般情况下这种办法是可靠的,但是由于新密钥往往不同于旧密钥,因此可能会出现这样的情况:用旧密钥加密的 CON 短信恰好被新密钥解密成 MSG 短信。此时,发送 CON 短信的一方将因无法收到对方回复的 DOE 短信而不能更新密钥。由此带来的后果是,一旦超过当前密钥生存期的 1/3,通信双方将使用不同的密钥与对方通信,从而无法正确地解密彼此的短信。如果遇到这种情况,唯一的解决办法是通信双方使用 AddPhoneMIDlet 删除各自的 mainRS 中的对方号码的记录,然后重新添加对方的号码及协商好的密钥等。需要说明的是,旧密钥加密的 CON 短信恰好被新密钥解密成 MSG 短信这种情况非常少见。

由于内存资源有限,目前大部分 Java 手机不允许多个 MIDlet 同时运行,所以短信加密程序中的每一个 RS 在任何时刻通常只能被一个 MIDlet 访问。在这类设备上使用短信加密程序不会出现数据不一致的情况。如果设备允许多个 MIDlet 同时运行,则有可能会出现数据不一至的情况,进而引起混乱。万一遇到这样的设备,建议不要同时运行 EncryptMIDlet 和 AddPhoneMIDlet(但这并不能从根本上解决问题)。我们尚未对此问题进行深入研究,不过,我们认为可以采用 Java 的同步机制处理该问题。

4 注意事项

1) 发送短信后,应该立即退出 EncryptMIDlet

在不允许多个 MIDlet 同时运行的设备上发送短信(特别是 KEY 短信)之后,应该立即退出 EncryptMIDlet,以便 DecryptMIDlet 能及时收到对方的回复。当然,如果 DecryptMIDlet 暂时没有启动,设备会缓存来自对方的短信。

2) 密钥接收方若发现对方的短信乱码,可通过主动给对方发送短信解决问题

PNa 给 PNb 发送 DOE 短信之后,将使用新密钥加密所有发往 PNb 的 MSG 短信。如果 PNb 没有收到 DOE 短信,PNb 将无法解密来自 PNa 的 MSG 短信。因为在收到 DOE 短信之前,PNb用旧密钥解密来自PNa的短信。解决该问题的方法

(下转第 1820 页)

和 V_μ,用拉格朗日插值法可以构造出多项式 $V_\mu(x)$,则攻击者将 t 代入可计算出 $V_{t\mu} = V_\mu(t)$。因此签名是可模拟的。

定理 1[6] 若基于身份的门限签名是可模拟的,且它所基于的基于身份的签名是不可伪造的,则该基于身份的门限签名是不可伪造的。

本文所提出的方案所基于的基于身份的签名方案在随机预言机模型下,能够抵抗适应性选择消息攻击和身份攻击伪造[8]。且在本方案中,规定 CES 标记 T 在每次签名算法中不能被重用,如果标记长度 l_T =80bit,则可满足 2^{80} 个签名,在随机预言模型下具有不可伪造性[1]。又结合引理 1,得出本文所提出的基于身份的可截取门限签名方案是不可伪造的。

3.6 效率分析

在签名过程中,本方案没有使用双线性对运算,而文献[7]在签名过程中使用了 4 个对运算,在验证算法中比文献[7]的方案少用了一个对运算。双线性对的运算效率是很低的,所以本方案的效率比文献[7]的方案提高了很多。

4 结语

本文在目前基于身份的密码系统的研究基础上,结合可截取签名体制,基于双线性对构造了一个基于身份的可截取门限签名方案,并证明了其在随即预言模型下能抵抗伪造。与其他基于身份的门限签名方案相比,具有更高的签名效率,在电子商务或电子政务系统中具有更高的实际应用价值。

参考文献:

[1] STEINFELD R, BULL L, ZHENG Y. Content extraction signatures [A]. Proceedings of 4th international conference on information security and cryptology (ICISC 2001) [C]. Berlin: Springer-Verlag, 2001. 285 - 304.

[2] BULL L, STANSKI P, MCG SQUIRE D. Content extraction signatures using XML digital signatures and custom transforms on - demand[A]. Proceedings of the 12th international World Wide Web conference (WWW2003) [C]. New York: ACM Press, 2003 .170 - 177.

[3] BULL L, MCG SQUIRE D, ZHENG Y. A Hierarchical Extraction Policy for content extraction signatures[J]. International Journal on Digital Libraries, 2004, 4(3): 208 - 222.

[4] SHAMIR A. Identity-based cryptosystems and signature schemes[A]. Advances in Cryptography-CRYPTO'84[C]. Berlin: Springer -Verlag , 1984, Vol 196: 47 - 53.

[5] BONEH D, FRANKLIN M. Identity based Encryption from Weil pairing[A]. Advances in Cryptography-CRYPTO 2001, LNCS2139 [C]. Berlin: Springer-Verlag , 2001. 213 - 229.

[6] BAEK J, ZHENG YL. Identity-based threshold signature scheme from the bilinear pairings[A]. IAS'04 track of ITCC'04[C]. IEEE Computer Society, 2004. 124 - 128.

[7] CHEN XF , ZHANG FG , KONIDALA DM , *et al*. New ID - based threshold signature scheme from bilinear pairings[A]. INDOCRYPT 2004, LNCS3348[C]. Berlin: Springer-Verlag, 2004. 371 - 383.

[8] CHEON JH , KIM Y , YOON HJ . A new ID - based signature with batch verification [EB/OL]. http://eprint. iacr. org /2004/131, 2004 - 05 - 31.

[9] FELDMAN P. A practical scheme for non-interactive verifiable secret sharing[A]. Proceedings of the 28th IEEE Symposium on Foundations of Computer Science[C]. IEEE, 1987. 427 - 437.

[10] GENNARO R, JARECKI S, KRAWCZYK H. Robust threshold DSS signatures[A]. Advances in Cryptology-EUROCRYPT' 96, LNCS1070[C]. Berlin: Springer-Verlag, 1996. 354 - 371.

(上接第 1816 页)

是:PNb 主动给 PNa 发送 MSG 短信。由于没有收到 DOE 短信,EncryptMIDlet 会提示用户是否重发 CON 短信。PNa 在收到重发的 CON 短信后会自动给 PNb 回复 DOE 短信。如果仍然没有收到来自 PNa 的 DOE 短信,PNb 可以继续主动给 PNa 发送 MSG 短信。当然,所有这些必须发生在当前密钥生存期的 1/3 这段时间内。

3) 短信加密程序适用于先进先出的短信传递服务

在密钥的传递和更新过程中,短信加密程序假定短信中心按照先进先出的策略传送短信。如果短信中心不保证先进先出(比如,如果为短信指定优先级[6],高优先级短信即使晚发送也可能先收到),或者采用其他策略如先进后出,则可能会引起通信混乱。解决办法:密钥发送方在发送新密钥之后暂时停止给对方发送新的短信,直到收到对方的 CON 短信,而密钥接收方也应该在收到 DOE 短信之前停止给密钥发送方发送新的短信。

5 结语

我们开发的基于 Java ME 的短信加密程序在 Sun 公司的 WTK 2.2 上测试通过。程序中没有使用 MIDP 的低级用户接口,也没有使用设备专用的 API,因此,原则上只要手机支持 MIDP 2.0 和 WMA 就可以使用该程序,并且不需要运营商提供额外的支持。除了可以在商业合作伙伴之间,以及关系密切的两个人之间传递敏感信息,我们相信,短信加密程序在密钥更新中所采用的处理方法对基于短信的手机银行、手机炒股等应用也有一定的借鉴价值。最后需要说明的是,尽管使用通话方式也可以完成密钥更新中的确认操作(即替代 CON 和 DOE 短信),并且更加可靠,但该方式未必适用于所有应用场合(如基于短信的手机银行),而且并不是每个人都欢迎这种方式,毕竟在网络和手机都正常工作的情况下,用户只需按一下键即可完成确认操作要比通话方式完成确认操作简单得多,而且节省费用。

参考文献:

[1] 杨建强,陈天煌,袁磊. 基于 CLDC 的无线 Java 安全研究[J]. 计算机应用与软件,2006, 23(3): 127 - 130.

[2] JCP. WMA - JSR 205, Wireless Messaging API(WMA) 2 for J2ME [EB/OL]. http://jcp. org/aboutJava/communityprocess/final/jsr205/index. html, 2004 - 05 - 17.

[3] JCP. MIDP - JSR 118, Mobile Information Device Profile for J2ME Version 2. 0[EB/OL]. http://jcp. org/aboutJava/communityprocess/final/jsr118/index. html, 2002 - 11 - 05.

[4] 杨建强,袁磊. 无线 Java 推送技术若干问题简析[J]. 计算机时代,2005,(3): 5 - 6.

[5] 段云所,魏仕民,唐礼勇,等. 信息安全概论[M]. 北京: 高等教育出版社, 2003. 79 - 88, 100 - 103.

[6] ETSI. GSM 03.40 v7.4.0, Digital cellular telecommunications system (Phase 2 +); Technical realization of the Short Message Service (SMS)[S]. 2000.