# 面向手机短信的隐私保护方案①

沈薇薇, 熊金波, 黄阳群, 姚志强

(福建师范大学 软件学院, 福州 350108)

**摘 要**: 针对手机短信存在的用户隐私泄露问题, 设计一种面向手机短信的隐私保护方案. 本方案结合非对称加密技术, 通过使用信息接收者的公钥对原始短信进行加密获得短信密文, 并结合预设的生命周期信息封装成短信自毁对象(Message Self-destructing Object, MSO)并通过运营商发送给接收者, 接收者接收到 MSO 后对其生命周期进行验证, 只有当前时间处于其生命周期内时, 才能进一步使用其私钥对短信密文进行解密获取原始短信内容, 一旦超过 MSO 的生命周期, 则 MSO 将被自动删除以保护用户隐私安全. 实验分析表明, 本方案能够有效保护用户手机短信的隐私安全, 实现生命周期控制并自动删除过期短信, 并且对硬件系统要求低, 开销合理,适合在人们日常生活中进行推广使用.

**关键词**: 手机短信; 隐私保护; 数据加密; 生命周期; 自动删除

## Privacy Protection Scheme for Short Message

SHEN Wei-Wei, XIONG Jin-Bo, HUANG Yang-Qun, YAO Zhi-Qiang

(Faculty of Software, Fujian Normal University, Fuzhou 350108, China)

**Abstract**: In order to protect the privacy of short message, a privacy protection scheme for short message is proposed in this paper, which is based on asymmetric encryption. In this security system, the message is first encrypted by the receiver's public key to obtain message ciphertext. Then the message ciphertext is combined with the life time information and encapsulated into Message Self-destructing Object (MSO). At last, the MSO is sent to the receiver through the operator. After getting the MSO, the receiver can use the private key to decrypt the ciphertext to obtain the original message only when the current time is in its life time, and after its life period, the system will automatically delete the MSO to protect the user's privacy. The experiments show that this scheme can effectively protect the privacy of short message and remove the expired MSO. And the system has low performance overhead, it is suitable to be used widely in people's daily life.

**Key words**: short message; privacy protection; data encryption; life time; self-destruction

近年来, 智能手机的广泛普及给人们生活带来极大的便利与改善, 移动互联网的快速发展和各种手机第三方应用软件的推陈出新带给用户不少新的体验, 一些使用流量的"免费通信软件"如微信、QQ 等即时通讯软件越来越深受广大用户的青睐.然而运营商所提供的短信服务依旧占有极大的市场份额, 自始至终都是人们日常生活中不可替代的通信服务之一. 因此随着网络通信的信息安全成为当下研究热点, 手机短信的安全性也越来越受到人们重视, 其存在的安全隐患有如下三点:

(1) 短信内容被拦截、窃取. 由于手机第三方应用程序来源极为广泛, 下载途径多样, 其安全可靠性均无法保障, 甚至可能携带有木马、病毒等对手机的短信进行拦截、窃取等非法操作, 这将造成用户隐私泄露问题;

(2) 现有的手机自带的通用短信系统多数为明文显示, 并且没有设定身份验证, 因此一旦用户丢失手机, 手机内存储的涉及个人隐私的短信随时都可能被

① 基金项目:国家自然科学基金(61370078,61402109);福建省自然科学基金(2015J05120);福建省教育厅科研项目(JA14091)
收稿时间:2015-08-03;收到修改稿时间:2015-09-17

不法分子利用, 对用户造成严重威胁;

(3) 运营商可能因国家安全或追查刑事犯罪需要向政府或司法机关提供用户的通信记录内容, 因此运营商并非绝对可信,可能向他人泄露用户的个人隐私.

为了减少这些安全隐患, 保护手机用户信息的隐私安全, 研究人员开始关注移动终端下数据隐私安全保护方面的问题:如 Enck 等[1]设计一种信息流跟踪系统(TaintDroid), 通过污点标记敏感数据的方式检测手机第三方应用程序是否存在隐私侵权行为, 该方法可以甄别手机第三方应用程序是否存在拦截、窃取手机短信等非法操作, 为用户辨别恶意软件提供参考; 文献[2]就是采用这种实验方式并提出一种基于 Android 的严格管理敏感数据的操作系统 CleanOS, 该系统通过监测 RAM 和存储卡上的敏感信息(密码, email 等), 并对其进行加密, 一旦这些敏感信息不被应用程序所使用便对其进行驱除, 确保用户的隐私信息不会被其他软件恶意获取, 避免造成用户隐私信息泄露的问题; 但该设计主要是针对移动应用软件所涉及的隐私数据如用户密码, 账号等进行保护, 并不适用于手机短信内容的隐私保护. Tung 等[3]提出一种的 Pandora Messaging 移动客户端, 实现即时移动通讯消息自毁, 有效防止信息泄露, 该设计中, 发送方和接收方各自提供一对短暂的公私密钥对, 通过这对短暂的公私密钥对的有效时间来实现消息的安全自毁, 一旦该私钥过期, 即接收方无法对消息进行解密, 进而实现信息不可读, 该方案能够实现即时通讯消息的隐私保护和消息生命周期的控制. 文献[4-10]等在不同程度上对 Vanish[11]方案进行改进, 利用分布式 Hash 表网络的自动周期更新节点的特性实现网络内容安全自毁, 但是其采用密钥不可用或部分密文丢弃达到消息内容不可读的方法实现隐私数据的安全自毁, 实际上并没有删除消息本身, 不但占用存储空间, 而且也还存在密文被破解的安全隐患.

综上所述, 为了保护手机短信的隐私安全, 需要对这些短信设定一个阅读期限, 消息接收者只有在该期限内才能阅读短信内容, 除此之外任何时间任何人都不能获取消息内容. 为此, 本文结合非对称加密技术提出一种面向手机短信的隐私保护方案, 通过对短信进行封装处理后再发送给接收者, 能够有效保护用户隐私安全并实现自动删除.

本文余下部分的组织为: 第 1 节给出方案涉及的定义, 并提出方案假设和系统模型;第 2 节首先概述方案并分阶段介绍方案流程和算法设计;第 3 节从安全性分析和实验测试结果对方案进行综合分析;第 4 节总结全文并给出下一步研究方向.

## 1 方案概述

### 1.1 概念说明

本文所用到的两个概念定义如下:

生命周期[12]. 指发送者发送的短信内容能够被接收者阅读的时间段, 若接收者请求阅读短信的当前时间早于该时间段, 则阅读请求被拒绝;若当前时间正好处于该时间段, 则阅读请求被接受并进一步处理;若当前时间晚于该时间段, 则该短信将被自动删除, 任何人都不能再次阅读. 本方案要保护短信在其生命周期内外的隐私安全.

短信自毁对象(Message Self-destructing Object, MSO). MSO 是本方案设定的一种数据包, 封装有加密后的信息内容, 并设置了该信息内容的生命周期信息, 能够有效保护信息的隐私安全.

### 1.2 方案假设

本方案做如下安全假设:

(1) 通信连接. 本方案通过运营商实现短信的收发, 因此需要确保运营商提供短信服务, 即用户所使用的手机号码应处于服务范围, 能够正常收发短信.

(2) 短信发送者和接收者可信. 短信发送者和接收者是可信的, 不会主动泄露或传播短信明文.

(3) 密钥管理中心可信. 密钥管理中心作为可信第三方服务器, 负责提供用户的公/私钥, 并且不会主动向他人泄露用户私钥信息, 并且在本方案中能够正常验证申请私钥者的身份并授权私钥.

### 1.3 系统模型

本方案的系统模型如图 1 所示, 模型包含信息封装模块、信息服务模块、信息解封装模块.

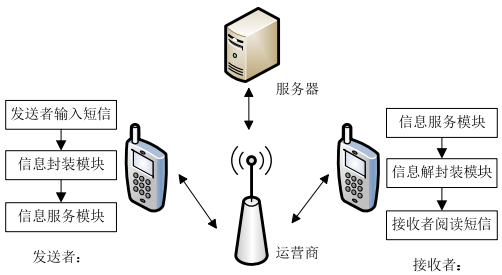


图 1 系统总体框图

信息封装模块: 信息封装模块使用非对称加密技术[13], 使用接收者的公钥对发送者要发送的原始短信进行加密获得短信密文, 接着结合发送者设定的生命周期信息进行组合封装成 MSO.

信息服务模块: 信息服务模块用于完成 MSO 的发送与接收, 确保信息封装模块所生成的MSO能通过网络传输送达信息接收者, 并在接收手机处于开机状态下进行通知, 由于本模块在本方案中只完成基本短信收发功能, 是手机基本应用之一[14], 因此在本方案算法设计上不予重点关注.

信息解封装模块:信息解封装模块首先解析接收到的 MSO, 对其生命周期进行验证, 若当前时间处于MSO 的生命周期内, 则进一步使用密钥对信息密文进行解密获取信息明文;若当前时间处于生命周期之前, 则停止对 MSO 的进一步解析;若当前时间已过期, 则该 MSO 将被自动删除.

## 2　方案概述与算法

方案的基本工作原理如图 2 和图 3 所示, 分为如下 2 个阶段:信息封装阶段和信息解封装阶段;为了更详细概述本方案的算法构成, 本文使用典型的非对称加密算法 RSA[15]对方案进行说明.

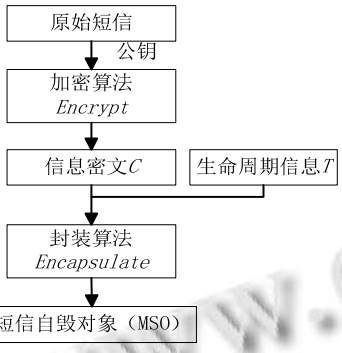


图 2　信息封装流程

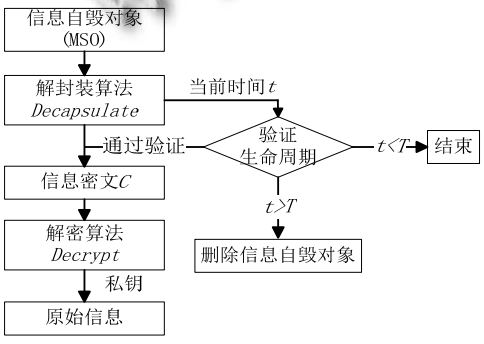


图 3　信息解封装流程

(1) 信息封装阶段

短信的封装工作由信息封装模块完成:首先使用接收者的公钥通过非对称加密算法对原始短信进行加密获得短信密文 $C$, 为了让接收者接收到 MSO 后遵循信息发送者预设的生命周期 $T$ 约束条件, 随后需要将短信密文 $C$ 与发送者指定的生命周期信息进行封装组合成短信自毁对象 MSO. 具体分为以下几个步骤:

①原始短信的输入与生命周期指定

由发送者输入短信内容明文, 并指定该短信内容的生命周期 $T$;

②原始短信的加密

通过非对称加密算法(RSA), 原始短信使用接收者的公钥对原始短信进行加密, 获得短信密文, 关键算法描述如下:

```
public static byte[] Encrypt(byte[] message,    PublicKey
pubKey)
{
      Cipher cipher = Cipher. getInstance("RSA");
      cipher. init(Cipher. ENCRYPT_MODE,    pubKey);
      return cipher. doFinal(message);
}
```

③MSO 的生成

由步骤①指定的生命周期与步骤②所得的短信密文组合封装生成 MSO, 算法描述如下:

```
public static MSO Encapsulate(byte[] Cmessage, Date
begintime, Date endtime)
{
       MSO mso = new MSO();
       byte[] LifeTime = mso. setLifetime(begintime, endtime);
       mso. associate(Cmessage, LifeTime);
       return mso;
}
```

④MSO 的发送

将步骤③所得 MSO 通过手机短信收发组件发送给接收者.

(2) 信息解封装阶段

MSO 的解封装工作由信息解封装模块完成:首先接收者接收到发送者所发送的 MSO, 对 MSO 进行解封装分离出 MSO 设定的生命周期信息 $T$, 接着获取当前时间 $t$ 进行验证, 若当前时间 $t$ 早于 MSO 的生命周期 $T$, 则停止对 MSO 的进一步解析;若当前时间 $t$ 处于 MSO 的生命周期 $T$, 则对 MSO 做进一步解析获取信息密文 $C$, 并使用接收者提供的私钥, 对所得信息密文 $C$ 进行解密, 最终获得原始短信;若当前时间 $t$ 晚于 MSO 的生命周期 $T$, 则 MSO 将被自动删除. 具体分为以下几个步骤:

①MSO 的接收

接收者通过手机短信收发组件, 获取到发送者所发送的 MSO.

②MSO 的解析分离

接收者收到 MSO 后, 首先对 MSO 进行解析分离出 MSO 的生命周期信息 *T*, 便于后续对 MSO 的阅读时间段进行约束, 关键算法如下 *Decapsulation_T* 所示; 若 MSO 的生命周期验证通过后, MSO 需要进一步解析出其携带的密文信息 *C*, 关键算法如下 *Decapsulation_C* 所示:

```
public static Date[] Decapsulation_T(MSO mso)
{
    Date[] time = new Date[2];
    time[0] =mso. extract(mso). getBeginTime();
    time[1] =mso. extract(mso). getEndTime();
    return time;
}
public static Byte[] Decapsulation_C(MSO mso)
{
    Byte[] Cmessage= null;
    Cmessage=mso. extract(mso). getCmessage();
    return Cmessage;
    }
```

③MSO 生命周期的验证

获取当前时间 *t*, 并对步骤②获取到的生命周期 *T* 进行对比验证, 若 *t* 早于生命周期 *T*, 则停止对 MSO 的进一步解析;若 *t* 处于生命周期 *T*, 则进入步骤④解密获得原始短信;若 *t* 晚于生命周期 *T*, 则 MSO 将被自动删除, 关键算法描述如下:

```
Public static int Verify(Date Begintime,    Date EndTime)
{
     Date t=new Date();.get the time t from the system;
     if(t. compareTo(EndTime)>0)
          return 0;
     else if (t. compareTo(Begintime)>0&&t.
compareTo(EndTime)<0)
          return Decrypt(C);
     else if(t. compareTo(EndTime)>0)
          return delete(MSO);
    }
```

④短信密文的解密

通过步骤③对 MSO 的生命周期进行验证后, 根据接收者提供的私钥对短信密文进行解密, 获得短信明文, 关键算法描述如下:

```
public static byte[] Decrypt(byte[] Cmessage,    PrivateKey pri)
{
     Cipher cipher = Cipher. getInstance("RSA");
     cipher. init(Cipher. DECRYPT_MODE,    pri);
     return cipher. doFinal(Cmessage);
}
```

## 3 综合分析

本系统从安全性与仿真实验测试两方面对本方案进行分析.

### 3.1 安全性分析

由本文 1. 2 安全假设一节中, 发送者、接收者与密钥管理中心是可信的, 不会泄露或传播解密后的短信内容或接收者的私钥信息, 因此本方案可能存在的安全隐患在于:

①第三方软件非法拦截手机发送或接收到的短信(MSO);

②用户手机遗失, 手机内所存储的短信(MSO)被他人窥窃;

③运营商存储的用户短信内容(MSO)被他人获取;

即敌手通过非正常渠道获取完整的 MSO 并使用强力攻击企图对 MSO 进行破解获取到短信明文, 因此本方案的安全性实际可规约为本方案的算法安全性.

本方案使用 RSA 算法对原始短信进行加密获得短信密文, 并结合生命周期信息进行组合封装, 因此攻击者首先需要成功分离出短信密文和生命周期信息后能对短信密文后才能进行强力攻击或密码分析攻击, 而 RSA 算法是可证明安全的[15];因此攻击者在多项式时间内无法对短信密文进行破解获取短信内容, 即可认为理论上攻击者是无法获取到短信明文威胁用户隐私安全的.

### 3.2 仿真实验测试

本节对方案进行仿真实验测试, 使用 android 手机, 1. 2GHz 四核处理器作为实验环境, 操作系统为 Android 4. 1. 实验导入 android. telephony. SmsManager 包[16]作为本方案测试的基础框架, 在此基础上添加信息封装与解封装算法, 实现对信息的隐私保护与生命周期控制. 即系统首先使用算法 *Encrypt* 和 *Encapsulate* 对短信内容进行加密并封装成 MSO, 随后再调用 SmsManager 类[16]发送 MSO. 当用户接收到短信后, 系统调用算法 *Decapsulation_T*、*Verify*、*Decapsulation_C* 和 *Decrypt* 对 MSO 进行解析验证并解密, 最终获得原始短信明文.

实验通过对不同大小的短信进行测试, 说明本方案能够成功发送和接收 MSO, 并且实现 MSO 的生命周期控制及过期后的自动删除. 部分实验测试结果如表 1 所示: 若当前访问时间未满足 MSO 的生命周期, 则系统开销时间仅为 200ms 左右, 用于验证当前访问

时间是否满足MSO的生命周期要求;而对一条70个字符的短信进行加密封装和解封装并获取短信明文也仅需消耗低于 1 秒的时间, 相比获得的安全性与实用性而言, 该时间开销是合理的.

表 1 部分实验测试情况表(测试时间:2015/09/04)

| 短信大小(字符) | 生命周期 | 封装耗时(ms) | 发送/接收结果 | 解封装耗时(ms) | 访问结果 |
|---|---|---|---|---|---|
| 10 | 2015/01/01-2015/05/01 | 684 | 成功发送/接收 | 217 | (已过期)短信被删除 |
| 10 | 2015/10/01-2015/11/01 | 683 | 成功发送/接收 | 216 | (可访问时间未到)拒绝访问 |
| 10 | 2015/07/01-2015/10/01 | 675 | 成功发送/接收 | 426 | 成功访问 |
| 70 | 2015/01/01-2015/05/01 | 895 | 成功发送/接收 | 226 | (已过期)短信被删除 |
| 70 | 2015/10/01-2015/11/01 | 857 | 成功发送/接收 | 221 | (可访问时间未到)拒绝访问 |
| 70 | 2015/07/01-2015/10/01 | 884 | 成功发送/接收 | 754 | 成功访问 |

## 4 结语

为了保护手机短信的隐私安全, 本文结合非对称加密技术提出一种面向手机短信的隐私保护方案. 通过实验分析表明, 本方案能够对用户发送的原始短信进行加密并实现生命周期控制, 有效删除过期信息, 保障用户信息的隐私安全, 同时方案对硬件要求低, 开销合理, 适合在人们日常生活中进行推广使用. 下一步研究工作的重点主要为:

(1) 由于接收方手机号是公开的并可作为身份标识, 因此下一步工作拟结合基于身份加密(IBE)[17]算法应用到本方案中, 设计出更安全简便的方案;

(2) 针对本方案实际开发出一款可在生活工作中进行推广应用的手机应用程序, 将研究成果应用到实践中, 切实保护用户隐私安全.

## 参考文献


1 Enck W, Gilbert P, Chun B, Cox L, Jung J, Mcdaniel P. TaintDroid: An information-flow tracking system for realtime privacy monitoring on smartphones. Proc. of the 9th USENIX Conference on Operating Systems Design and Implementation USENIX Association. 2010. 1–6.

2 Yang T, Ames P, Bhamidipati S, Bijlani A, Geambasu R, Sarda N. CleanOS: Limiting mobile data exposure with idle eviction. Proc. of the 10th USENIX Symposium on Operating Systems Design and Implementation, OSDI. 2012, 12. 77–91.

3 Tung T, Lin L, Lee D. Pandora messaging: An enhanced self-message-destructing secure instant messaging architecture for mobile devices. 2013 27th International Conference on Advanced Information Networking and Applications Workshops. IEEE. 2012. 720–725.

4 熊金波,姚志强,马建峰,李凤华,刘西蒙,李琦.基于属性加密的组合文档安全自毁方案.电子学报,2014,42(2):366–376.

5 姚志强,熊金波,马建峰,李琦,刘西蒙.云计算中一种安全的电子文档自毁方案.计算机研究与发展,2014,51(7): 1417–1423.

6 Wang G, Yue F, Liu Q. A secure self-destructing scheme for electronic data. Journal of Computer and System Sciences, 2013, 79(2): 279–290.

7 Xiong J, Yao Z, Ma J, Liu X, Li Q. A secure document self-destruction scheme with identity based encryption. International Conference on Intelligent Networking & Collaborative Systems. 2013. 239–243.

8 熊金波,姚志强,马建峰,李凤华,刘西蒙.面向网络内容隐私的基于身份加密的安全自毁方案.计算机学报,2014,37(1): 139–150.

9 王丽娜,任正伟,余荣威,韩凤,董永峰.一种适于云存储的数据确定性删除方法.电子学报,2012,40(2):266–272.

10 Xiong J, Liu X, Yao Z, Ma J, Li Q, Geng K. A secure data self-destructing scheme in cloud computing. IEEE Trans. on Cloud Computing, 2014, 2(4): 448–458.

11 Geambasu R, Kohno T, Levy A, Levy H. Vanish: Increasing Data Privacy with Self-Destructing Data. USENIX Security Symposium. 2009. 299–316.

12 Xiong J, Li F, Ma J, Liu X, Yao Z, Chen P. A full lifecycle privacy protection scheme for sensitive data in cloud computing. Peer-to-Peer Networking and Applications, 2015, 8(6): 1025–1037.

13 Zhou X, Zhao F, Zeng D. Research of asymmetric encryption technology. Journal of Sichuan University of Science & Engineering, 2010.

14 刘必刚.Android 通信模块的设计与优化[硕士学位论文].武汉:武汉理工大学,2010.

15 Rivest R, Shamir A, Adleman L. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 1978, 21(2): 120–126.

16 SmsManager. http://android.toolib.net/reference/android/telephony/SmsManager.html [2013-05-24].

17 Boneh D, Franklin MK. Identity-based encryption from the weil pairing. Proc. of the 21st Annual Int. Cryptology Conference on Advances in Cryptology. 2001. 213–229.