窄带物联网下的安全门锁密钥可靠更新方案

刘梦君[1,3] 沙 涛[1] 李 丹[2] 刘树波[2]
(湖北大学计算机与信息工程学院 武汉 430062)[1]
(武汉大学计算机学院 武汉 430072)[2] (湖北大学教育学院 武汉 430062)[3]

摘 要 在窄带物联网(Narrow Band Internet of Things,NB-IoT)通信系统中,设备间的数据通信以无连接的UDP(User Datagram Protocol)报文方式传输。在不可靠的UDP传输机制下,密钥的可靠更新成了安全门锁机制研究中的难点。文中设计了一个无连接通信链路上的密钥可靠更新方案,该方案利用智能门锁密钥更新的特点,通过精心设计的密钥传输交互机制,使门锁设备通过UDP协议获取密钥并且可靠地完成密钥更新。理论分析和原型实验的结果表明,该方案能够可靠地更新密钥,并具有较小的通信开销和计算开销。

关键词 窄带物联网,密钥更新,安全通信,UDP,可靠传输
中图法分类号 TP273 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2019.04.022

Reliable Security Lock Key Updating Scheme over Narrow Band Internet of Things

LIU Meng-jun[1,3] SHA Tao[1] LI Dan[2] LIU Shu-bo[2]
(School of Computer and Information Engineering,Hubei University,Wuhan 430062,China)[1]
(School of Computer,Wuhan University,Wuhan 430072,China)[2]
(School of Education,Hubei University,Wuhan 430062,China)[3]

Abstract In narrow band internet of things communication system,the data among devices are transmitted by connectionless UDP protocol. Based on the unreliable UDP protocol,it is difficult to reliably update the keys for the security door system. This paper designed a reliable key updating scheme over connectionless communication link. This scheme makes use of the characteristics of smart lock key updating,and adopts the carefully designed interactive key transmission mechanisms,so as to make lock devices get keys by UDP protocol and achieve key updating reliably. Theoretical analysis and prototype experiment results show that the proposed scheme holds small communication and computation overhead while reliably updating the key.

Keywords NB-IoT,Key updating,Secure communication,UDP,Reliable transmission

1 引言

随着移动互联网的快速普及以及互联网+的深入发展,人们越来越依赖智能化的产品服务,智能门锁即是典型代表之一[1-2]。通常意义上的智能门锁包括卡片锁、指纹锁、密码锁、声纹锁、人脸识别锁等。这些门锁存在如下缺点:1)价格昂贵、能耗大、安装部署不便,如指纹锁、声纹锁和人脸识别锁需要有供电源;2)管理手段落后,如卡片锁和密码锁等均没有远程管理接口,运维费用高。因此,亟需能够同时解决上述问题的新型智能门锁。

窄带物联网为解决上述两方面问题带来了希望。基于蜂窝的窄带物联网是基于蜂窝网络构建的,所需带宽约为180 kHz,可部署于原有的2G,3G,4G网络基站上,部署成本低,因此一经推出,就受到了各行业及三大电信运营商的强力推崇,并且其作为一种国产化标准,得到了国家政策层面的大力支持。NB-IoT聚焦于低功耗、广覆盖(LPWA)物联网(IOT)市场。NB-IoT通信技术由于具备低功耗、广覆盖、低成本、大容量等优势,可被广泛应用于多种垂直行业,如物流跟踪、智能路灯、智能水电气表、智能垃圾桶、智慧农业等。这些设备大都被广泛且大面积地应用在无人值守的环境下,在完成各自功能的同时,可与锁管理员持续不断地通信,从而在安装、部署、运营成本较低的情况下,实现了设备的高效管理[3-5]。

不同于一般意义上的智能设备,智能门锁是家居安全的最后一道防线。如何在提供便捷的门锁产品服务的同时保障门锁的高安全性,是设计智能门锁时必须考虑的首要问题。通常,针对具备远程管理功能的智能门锁(如具备GPRS和

到稿日期:2018-03-12 返修日期:2018-04-23 本文受国家自然科学基金面上项目(41671443),湖北省自然科学基金项目(2017111112010003),湖北省教育厅自然科学基金项目(2017111310010003),武汉市科技局应用基础研究计划资助项目(2016010101010024)资助。
刘梦君(1988-),男,博士,讲师,主要研究方向为移动/无线网络、移动社交/分布式系统上的安全与隐私,E-mail:lmj_whu@163.com;沙 涛(1996-),男,主要研究方向为物联网、信息安全,E-mail:st_1996@foxmail.com(通信作者);李 丹(1981-),男,博士生,主要研究方向为数据挖掘、信息安全;刘树波(1970-),男,博士,教授,博士生导师,主要研究方向为物联网安全与隐私保护、数据隐私挖掘与发布。

WiFi 通信接口的门锁[6]),用户开启门锁设备的权限都是由管理员通过远程在线分配密钥决定的[7]。考虑到安全性,这些与具体用户关联的开锁密钥都必须定期更新。采用 GPRS 或 WiFi 通信方式的智能门锁虽然能够可靠地更新用户的开锁密钥,但需要较大的通信开销,难以长时间运行。采用 NB-IoT 通信方式的智能门锁虽然极大地降低了通信开销,大大延长了智能门锁的工作时间;但 NB-IoT 通信技术低功耗运营模式的设计,使得它采用了不可靠的 UDP 通信协议来传输数据,从而导致该方式将大概率地面临门锁密钥更新失败的问题。而门锁密钥一旦更新失败,将导致用户无法开锁,从而严重影响用户体验。

因此,本文将研究在现有 NB-IoT 不可靠的 UDP 数据传输方式下,以尽可能小的计算开销和通信开销来完成用户开锁密钥的可靠更新。本文第 2 节介绍了相关领域的研究;第 3 节介绍了系统模型及其所解决的问题;第 4 节介绍了本文设计思想及详细的设计流程;第 5 节对本文所提供的方案进行了仿真实验;最后总结全文。

2 相关研究

在窄带物联网通信系统中,密钥更新是确保网络通信信息不被窃取的基础,是智能家居安全门锁管理最主要的功能,也是物联网系统(Internet of Things System)通信安全研究的热点之一[8-10]。然而,物联网系统不能无限地增加带宽和系统资源,在某些情况下嵌入式硬件设备具有计算能力低、存储空间小、能量有限等不足。特别地,由于能量有限,为了追求更长的工作时间,会要求物联网通信系统硬件设备在与锁管理员通信时尽可能地减少资源消耗,而锁管理员会不定期地发送新密钥以确保与硬件设备之间的通信安全[11]。

虽然采取 TCP/IP 协议能可靠地传输密钥,但是 TCP 协议在建立连接的过程中占用的系统开销大、传输速率低,并不能很好地满足占用系统资源小、实时性、高效性等需求。为此,有研究人员提出了 FAST-TCP[12],BIC TCP[13] 等改进方法,并给出了很多有益的 TCP 改进建议。但是 NB-IoT 网络不支持使用 TCP 协议,因此上述改进建议仍然不能很好地满足在长距离复杂网络中传输密钥时的低系统开销、实时性、高效性等需求。

NB-IoT 网络使用的 UDP 是一种无连接的传输层协议,提供面向事务的简单不可靠信息传送服务,具有速率高、系统开销小等优点。但是 UDP 是不可靠传输协议,在数据传输过程中往往会发生丢包,而密钥传输过程被要求是完整、可靠的。为此,研究人员也提出了 RBUDP[14],UDT[15],AUDP[16] 等方法来改善 UDP 在传输中的不稳定性;但是这些方法需要手动设置发送率,并不适用于密钥更新的应用环境,并且如果设定的传送速率超出当时的网络状况,则将发生大量的重传,从而造成资源浪费和拥堵[17]。

针对现有方案的不足和窄带物联网系统中密钥更新的特点,本文提出了一种能完整地实现高速、低开销且稳定更新设备中的密钥的方案,称为安全门锁可靠更新方案,用于可靠、高速地满足锁管理员与智能锁硬件设备的密钥更新需求。

3 系统模型与问题定义

3.1 系统模型

智能锁系统中主要有 3 类实体:用户、智能锁、管理员(通常由一台管理员运维的服务器来执行角色功能),如图 1 所示。按照功能,可以将实体划分为密钥管理者和密钥需求者,用户和智能锁是密钥需求者,锁管理员为密钥管理者。考虑到安全性,用户发送给智能锁的开锁指令需要使用特定密钥加密,只有在密钥匹配的情况下,智能锁才能获得有效的开锁指令。而在某些特定场合下(如政府公租房和个人出租房),房屋管理方或者所有者需要把控门锁密钥权限,即对相关用户开锁权限进行管控,这就需要对用户的开锁密钥进行更新。为了降低功耗,通信时使用 NB-IoT,NB-IoT 在实际中提供的是 UDP 无连接报文传输服务。用户与锁管理员交互的途径众多,如现场更新、4G、因特网等,因此用户与管理员之间的密钥更新问题可以很容易且可靠地得到解决。由此可见,智能锁密钥更新的关键在于管理员与用户设备之间的密钥更新。

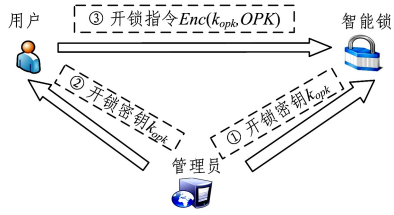

图 1 智能锁系统架构图

Fig. 1 System architecture of smart lock

3.2 安全模型

系统中,锁管理员是可信的,即他会确保发送给用户和锁的密钥正确无误,且不会绕开系统管理机制与用户或者锁合谋,发送不应被用户或锁得到的密钥,也不会将密钥泄露给无关的第三方。用户是半诚实的,即用户会遵守系统密钥更新方案,但他会尽可能地利用系统给予的现有信息在无密钥或者密钥过期的情况下进行开锁。锁是安全可控的,不会被外界所劫持。本文不考虑恶意用户的主动攻击行为,如用户暴力开锁或者恶意实施 DOS 攻击等,所有对系统密钥更新的攻击只设定在对密钥更新和数据传输的通信过程中,如窃听、中间人攻击等。

3.3 问题描述

对于系统中的 3 个实体,即锁管理员(密钥管理者,Lock Manager,LM)、智能锁(Smart Lock,SL)和用户(User,U)。因为智能锁用户的某些因素改变,所以锁管理员需要通过一套可靠且运行于 NB-IoT 网络上的密钥更新机制来更新智能锁和用户的密钥。智能锁一般通过干电池供电,因此在 NB-IoT 提供的 UDP 无连接报文服务上不能使用计算和通信开销过大的密钥更新方案。同时,智能锁的第一功能是使有正确钥匙的用户正常开锁且无正确的钥匙用户无法开锁。

具体地,在某一时刻 t,锁管理员 LM 发现用户 U_i 之前的开锁密钥 k'_{U_i} 发生了变化,需要将其更新为当前的新密钥 k_{U_i}。假定智能锁设备 SL_i 与锁管理员 LM 之间更新密钥的通信开销为 $Comm_{LM\to SL_i}$,则本文的问题可以归结为寻求一种资源受限 UDP 传输机制上的密钥更新方案 $Update(k'_{U_i}\to k_{U_i},k'_{SL_i}\to$

$k^t_{SL_i}$：$LM \rightarrow U_i$，$LM \rightarrow SL_i$），Min（$Comm_{LM \rightarrow SL_i}$），使得 $k^t_{U_i} = k^t_{SL_i}$。

该安全门锁密钥更新过程的目标为：

1）通过 NB-IoT 网络提供的资源受限 UDP 传输服务，以尽可能低的开销完成 LM 与智能锁之间的密钥更新；

2）整个密钥更新过程只涉及 LM、智能锁及用户的交互，不需要第三方服务器的参与；

3）确保用户能够开锁。

需要特别注意的是，为了避免密钥更新过程对用户开锁造成影响，用户与锁的密钥更新时间一般选择为用户活动较少的时间点，如凌晨 3 点。

4　窄带物联网下的安全门锁密钥可靠更新方案

4.1　主要思想

智能锁（SL）的密钥更新管理需要使用网络资源受限的 NB-IoT 网络，其使用的 UDP 服务容易丢失密钥更新包，因此本文从两个方面来解决密钥可靠更新问题。一方面，由于用户手机（U）可与锁管理员（LM）可靠地进行密钥更新，为了确保用户能够正常开锁，用户的密钥被设计为迟滞更新，即为了保证用户手机密钥和智能锁密钥的一致性，手机密钥必须在锁密钥更新成功之后进行。另一方面，用户密钥的更新报文类型需要尽可能地少，且尽可能地包含密钥信息。为了达到上述目的，锁管理员发送给智能锁的报文被设计为两种：1）初始密钥更新报文（$Key\_Pack \sharp 1$）；2）确认密钥更新报文（$Key\_Pack \sharp 2$）。智能锁密钥更新流程如图 2 所示。锁管理员一旦收到智能锁确认信息（Ack），便会发送唯一一个确认密钥更新报文，并停止更新过程，否则会重传初始密钥来更新报文；但为了避免因收不到确认信息或者锁未收到更新报文而无限重传，对初始密钥报文的传输次数加以限制。锁只要收到了一种密钥更新报文，就会停止更新。在极端情况下，如果锁未成功更新密钥，它仍然使用旧密钥向锁管理员发送门锁运行信息，锁管理员则会开启新一轮的密钥更新。使用上述方案，既可以确保用户在密钥更新失败的情况下正常开锁，又能以较少的报文传输量完成密钥更新，从而在 NB-IoT 网络环境下对智能锁系统实现可靠的密钥更新。

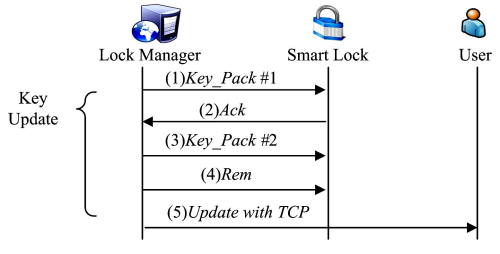

图 2　智能锁密钥更新流程图

Fig. 2　Flowchart of key updating in smart lock system

为了安全且可靠地更新 SL_i 中的密钥并确保授权用户能够开锁，本文给出了两种实现方案。第 1 种方案即安全门锁密钥可靠更新方案（Reliable Security lock Key Update，RSKU）通过精心设计的密钥更新过程，在保护密钥更新过程安全可靠的同时确保用户能够正常开锁，实现了上述设计思想。但理论上，这种设计方案由于重传次数是预先设定的，在网络信号极差时并不能保证密钥能够更新成功。为了解决该问题，对第 1 种方案进行改进，提出了第 2 种方案——动态安全门锁密钥可靠更新方案（Dynamic and Reliable Security lock Key Update，DRSKU）。第 2 种方案增加了根据网络状况动态地改变重传次数的功能，既减少了通信开销，又提高了更新成功率。

4.2　安全门锁密钥可靠更新方案

根据设计思想给出具体的方案流程，相关符号如表 1 所列。假设 LM 对于第 i 个智能锁 SL_i 需要发送更新的密钥为 $k^t_{SL_i}$。使 LM 首先向各个需要更新密钥的 SL 发送必要的初始密钥以更新报文（$Key\_Pack \sharp 1$）；SL_i 接收到 LM 发送的信息后（bn_i 表示是否接收成功），LM 需要知道 SL_i 是否接收到每个初始密钥的更新报文，即需要判断 $bn_i = 1$ 是否成立。为此，SL_i 通过发送确认信息（Ack）回应 LM，bm_i 表示 LM 接收到 Ack 的结果，因此 LM 能够对 $bm_i = 0$ 对应的密钥进行重传。但是，如果没有一种保障机制来保证重传超过一定限制后能够中断此操作，在网络情况极端恶化的情况下，系统将会陷入无限循环重传，造成系统崩溃。为此，通过设定最大重传次数 $N_{\max}$ 来限制重传次数。对 $bm_i = 1$ 对应的智能锁 LM 发送确认密钥更新报文（$Key\_Pack \sharp 2$），如果重传次数不超过 $N_{\max}$，$bm_i > 0$ 成立，则此次密钥更新成功，否则更新失败，但一般情况下不能保证 $Key\_Pack \sharp 2$ 一定送达，为此设定 SL_i 在一段时间后使用当前的 $k^t_{SL_i}$ 给 LM 发送信息。

表 1　符号说明

Table 1　Symbol descriptions

符号	说明
P_s	密钥更新成功率
n	发送密钥更新的总数
$k^t_{SL_i}$	需要更新的新密钥
$N_{\max}$	设定的最大重传次数
T_{prop}	传播时延
T_{w}	设定的等待确认信息时间
T_{RT}	重传时间
$T_{\text{Key\_Pack}\sharp 1}$	初始密钥更新信息发送时间
T_{Ack}	确认信息发送时间
$T_{\text{Key\_Pack}\sharp 2}$	确认密钥更新信息发送时间
BN	锁是否收到 LM 信息的布尔集
BM	LM 是否收到确认信息的布尔集
bn_i	每一个更新过程中锁收到的更新信息布尔集
bm_i	每一个更新过程中 LM 收到的确认信息布尔集
$T_{\text{total}i}$	每一个密钥更新所需时间
T_{avg}	平均每一个密钥的更新时间

通过上述方法，LM 发送 $k^t_{SL_i}$ 后就能使 U 与 SL_i 使用 $k^t_{SL_i}$ 开锁，否则它要么会因为未收到 Ack 而认定 SL_i 未收到 $Key\_Pack \sharp 1$，从而开始重传，要么因为 SL_i 之后发来的信息中使用的 $k^t_{SL_i}$ 而认定更新失败。其中隐含的原理为：

1）无论密钥是否更新成功，LM 通过 SL_i 发送的 Ack 及上报信息来确定 SL_i 使用何种密钥；

2）SL_i 通过 LM 是否发送 $Key\_Pack \sharp 2$ 来决定是否使用 $k^t_{SL_i}$，这使 LM 能够保证 U_i 使用的 $k^t_{U_i}$ 与 SL_i 使用的 $k^t_{SL_i}$ 始终保持一致，即授权用户在所有情况下都能够正常开锁。

该方案的具体实现过程如算法 1 所示。

算法 1 安全门锁密钥可靠更新方案 RSKU()

Input:锁管理员 LM 将要更新的新密钥 $k_{SL_i}^t$

Output:对应智能锁 SL_i 的密钥更新结果(True or False)

1. Key_Pack#1←AES($k_{SL_i}^t$),新密钥经过加密后形成初始更新密钥报文
2. bn_i←send(SL_i←LM,SL_i←Key_Pack#1)
3. if bn_i==False:
4. if 当前重传次数 N<N_{max}:
5. 重新执行步骤 2,重传信息 Rem
6. else:
7. return False
8. else:
9. bm_i←send(LM←SL_i,LM←Ack)
10. if bm_i== False:
11. if N<N_{max}:
12. 重新执行步骤 2,重传信息 Rem
13. else:
14. return False
15. else:
16. send(SL_i←LM,SL_i←Key_Pack#2)
17. return True

本节提出的窄带物联网下的安全门锁密钥可靠更新方案中更新 U_i 的密钥 $k_{U_i}^t$ 时使用 3G/4G 网络中的 TCP 协议,此处不再赘述。窄带物联网下 SL 的密钥 $k_{SL_i}^t$ 的更新工作主要分为 4 个阶段:1)新密钥生成阶段,该阶段必须保证新密钥的随机性且新密钥不可被攻击者获得,本方案通过生成一个随机数 Random 与时间戳,分别使用哈希函数生成两个 64bit 的哈希值,组成一个 128bit 的密钥;2)LM 发送初始密钥更新报文阶段,LM 通过 SL_i 发送的信息进行密钥更新,将对应的 SL_i 对要更新的新密钥先以旧密钥加密算法加密,然后通过 UDP 协议发送出去并等待传回确认信息;3)确认阶段,首先将 SL_i 接收到的更新信息组成一个布尔集,再对集合中的每一个元素进行判断,若元素值为 1,则对应的 SL_i 将会给 LM 发送确认信息,LM 将收到的确认信息组成一个新的布尔集后与发送密钥更新数的集合进行比对,若两个值不相等则进行重传;4)重传阶段,LM 将收到的确认信息与最开始的初始密钥更新报文进行比对,若已发送的初始密钥更新报文未收到确认信息,则重复阶段 1),若经过 N_{max} 重传后还是没有收到确认信息,则判定此次密钥更新失败;若收到确认信息则结束重传,并给收到确认信息对应的设备发送确认密钥以更新报文,锁如果未收到确认密钥更新报文,则会在一段时间后用旧密钥发送信息,并告知 LM 更新未完成,双方继续使用旧密钥,以保证密钥的一致性,同时进行新一轮的密钥更新协商。

具体地,在基于 NB-IoT 的网络中建立以下机制。

1)锁管理员 LM。锁上传信息后,LM 开始准备更新所需的新密钥 $K=\{k_{SL_1}^t,k_{SL_2}^t,\cdots,k_{SL_i}^t\}$,其中 $i=n$。为了保证密钥的安全性,按式(1)生成密钥 $k_{SL_i}^t$:

$$k_{SL_i}^t=h(Random())h(t_n) \tag{1}$$

其中,$Random()$为一个随机函数,其产生一个 8 位小数并且 $Random()\in[0,1]$;t_n 为 LM 时间的时间戳;$h()$为哈希散列函数。本文采取 SHA3-64 哈希算法随机生成密钥,SHA3 为新一代加密算法,也是目前安全强度最高的哈希算法,其 64bit 模式将会生成一个 64bit 的哈希值,对随机值及时间戳分别进行 SHA3-64 运算,得到的两个 64bit 哈希值组装成 128bit 的新密钥。

2)锁管理员 LM。新密钥生成后,LM 向智能锁 SL 传送第一次密钥 $K=\{k_{SL_1}^t,k_{SL_2}^t,\cdots,k_{SL_i}^t\}$,其中 $i=n$。密钥更新包中标识其为初始密钥更新报文,并等待锁的回应。初始密钥更新报文包记为 $Key\_Pack\#1$,组成如下:

$$Key\_Pack\#1=UIDE_{k_{SL_i}^t}(t_s,k_{SL_i}^t,ID_i)C \tag{2}$$

我们对 UDP 协议进行了一些改进,在其首部增加了一个 UID 字段,在其尾部增加了一个 C 字段。具体地,UID 字段为 2 个字节,用于标识设备的唯一 ID,以保证智能锁 SL 信息与 LM 信息的一致性;C 字段为校验字段,存放 16 位 CRC 校验码;t_s 为发送信息的时间;$k_{SL_i}^t$ 是即将更新的新密钥;ID_i 为当前密钥更新的标识号;$E()$为 AES 对称密码算法,对每一个智能锁设备 SL 进行密钥更新时,使用之前协商好的密钥 $k_{SL_i}^t$ 作为 AES 加密算法的密钥对报文中的新密钥及密钥更新识别号进行加密。

3)智能锁 SL。对于所有进行密钥更新的设备,定义一个布尔集合 BN 来表示是否接收到 $Key\_Pack\#1$,$BN=\{bn_1,bn_2,\cdots,bn_i\}$,其中 $i=n$。用户收到并成功更新密钥后,对 $bn_i=1$ 对应的设备发送确认信息报文,并等待 LM 发送的确认密钥更新报文。确认信息记为 Ack,组成如下:

$$Ack=UIDE_{k_{SL_i}^t}(t_s,M)C \tag{3}$$

其中,M 为确认信息,包含必要的设备和密钥对应的信息,用于识别发出的某个密钥更新包是否被接收。

4)锁管理员 LM。对于 LM 收到 SL 发送的确认信息 Ack,定义一个布尔集合 $BM=\{bm_1,bm_2,\cdots,bm_i\}$,其中$bm_i=1$ 代表已接收到 Ack。设集合 $F_1=\{k_1\wedge bm_1,k_2\wedge bm_2,\cdots,k_i\wedge bm_i\}$,其中 k_i 代表对应设备是否发送 $Key\_Pack\#1$,$k_i=1$ 代表已发送。LM 对 $\forall i\in[1,n]$,$k_i\wedge bm_i=1$ 所对应的智能锁设备 SL 发送第二次更新密钥包,密钥包中标识其为确认密钥更新报文,记为 $Key\_Pack\#2$,其组成如下:

$$Key\_Pack\#2=UIDE_{k_{SL_i}^t}(t_s,ID_i)C \tag{4}$$

为保证确认密钥更新报文能够被 SL 接收,无论 SL 收到与否,$Key\_Pack\#2$ 都会再发送一次;同时为了节约通信开销,$Key\_Pack\#2$ 只包含密钥更新标识号 ID_i 和时间信息,用于识别当前密钥更新的阶段,而不包含新密钥。SL 收到后,便知道 LM 收到了确认信息,无需再发送确认包,双方可以明确使用新密钥不会出错。

5)锁管理员 LM。设集合 $F_2=\{k_1\oplus bm_1,k_2\oplus bm_2,\cdots,k_i\oplus bm_i\}$。对于 $\forall i\in[1,n]$,若 $k_i\oplus bm_i=1$,则表明 LM 没有收到锁的确认信息 Ack,继续重传信息 Rem。Rem 的组成如下:

$$Rem=UIDE_{k_{SL_i}^t}(t_s,k_{SL_i}^t,ID_i,RT)C \tag{5}$$

智能锁收到 Rem 后,将会给 LM 发送确认信息 Ack。LM 只有在接收到 Ack 后,才能传送 $Key\_Pack\#2$。在 Rem 中增加 RT 字段,用于标识重传次数,LM 一共只会传 N_{max} 次

初始密钥更新报文，如果 $RT>N$ 时传输的还是初始密钥更新报文，则表明此次密钥更新失败，双方继续使用原密钥。

若 SL 间隔一段时间再次用旧密钥发送状态信息，则表明密钥更新未成功，两者再次开启密钥协商。

4.3　动态安全门锁密钥可靠更新方案

如上节所述，在密钥更新的过程中存在传输效率、密钥更新成功率及平均更新时间 3 个方面的要求。以 AUDP[16] 作为对比，采用 AES/ECBPKCS5 的加密方式生成密钥[18-19]，其密钥数据部分为 16 字节。通过公式计算得到 AUDP 和本文提出的算法的传输效率。在密钥更新成功率和平均更新时间方面，通过修改 T_w 和 N 的值进行实验，来动态达到满足用户需求的目的。

1)平均更新时间应被限定在一个范围内。所有用户对平均更新时间都有一个最大容忍度，在容忍范围内算法可以随意优化；过长的更新时间会使用户感到难以容忍，不适用于实时性要求高的应用场景，从而影响算法性能。

2)密钥更新成功率应尽可能的高。更新成功率是判定此算法能否稳定地实现锁管理员 LM 与智能锁 SL 之间可靠更新密钥的一个重要衡量指标，但是也需要一个平衡点，使得算法成功率提升的利大于算法平均更新时间增加的弊。基于这两点要求，定义算法评估函数 $Eval$ 为：

$$Eval=(1-P_s)*T_{\text{avg}} \tag{6}$$

在之后的实验中，通过算法评估函数 $Eval$ 来寻求 T_{w} 和 N 的最优解。

首先，锁管理员 LM 对所有的智能锁设备发送对应的初始密钥更新报文包 $Key\_Pack\sharp 1$，定义设备集合为 K，其中 k_i 代表 LM 对某一个智能锁是否发送 $Key\_Pack\sharp 1$。锁接收到 LM 发送的信息组成 BN，对于 $\forall bn_i\in BN$，规定 1 代表接收到，0 代表未接收到，对于 $\forall bn_i\in BN\wedge bn_i=1$，锁将给 LM 发送确认信息；LM 将接收到的确认信息组成 BM，对于 $\forall bm_i\in BM\wedge bm_i=1$，LM 将会对相应的锁发送第二次密钥更新信息，代表整个密钥更新过程完成。可以简单地推导出该过程的总更新时间为：

$$T_{\text{total}i}=T_{\text{Key\_Pack}\sharp 1}+T_{\text{prop}}+T_{\text{Ack}}+T_{\text{prop}}+T_{\text{Key\_Pack}\sharp 2}+T_{\text{prop}} \tag{7}$$

这个过程中还有一类元素为 $bm_i\in BM\wedge bm_i=0$。对于这类元素，LM 会逐一地比对 $k_i\in K$，$\forall i\in[1,n]$ 和 $bm_i\in BM$，$\forall i\in[1,n]$，若 $k_i\in K\cap bm_i\in BM=\varnothing$，则代表密钥更新过程中发生丢包，LM 将会对锁重新发送第一次的密钥更新信息，最大的重传数不超过给定值 $N_{\max}$。重传花费的时间为：

$$T_{\text{RT}}=\sum_{i=1}^{N}T_{\text{w}}+T_{\text{prop}}+T_{\text{Key\_Pack}\sharp 1} \tag{8}$$

在实际密钥更新过程中肯定会发生密钥更新信息或确认信息丢失的情况，因此将重传花费的时间代入式(2)得到完整的密钥更新时间计算公式：

$$T_{\text{total}i}=3\times T_{\text{prop}}+T_{\text{Key\_Pack}\sharp 1}+T_{\text{Ack}}+T_{\text{Key\_Pack}\sharp 2}+\sum_{i=1}^{N}T_{\text{w}}+T_{\text{prop}}+T_{\text{Key\_Pack}\sharp 1} \tag{9}$$

其中，$N\leqslant N_{\max}$，$T_{\text{total}i}$ 为每个密钥更新所花费的总时间，如果密钥更新未成功，则 $N=N_{\max}$。在得到每一个密钥更新所花费的时间后，可以推导得到密钥的平均更新时间：

$$T_{\text{avg}}=\frac{\sum_{i=1}^{n}T_{\text{total}i}}{|K|} \tag{10}$$

其中，$|K|$ 为集合中元素的总数。首先对 BM 元素值为 1 的元素建立集合 $S=\{s_1,s_2,\cdots,s_j\mid\forall s_j\in BM\wedge s_j=1\}$，然后计算成功率。密钥更新成功率可以表示密钥更新成功数与密钥更新总数的比值，即：

$$P_s=\frac{|S|}{|K|} \tag{11}$$

将式(10)和式(11)代入式(6)，即可得到完整的评估函数表达式：

$$Eval=(1-\frac{|S|}{|K|})\times\frac{\sum_{i=1}^{n}(3\times T_{\text{prop}}+T_{\text{Key\_Pack}\sharp 1}+T_{\text{Ack}}+T_{\text{Key\_Pack}\sharp 2}+\sum_{i=1}^{N}T_{\text{w}}+T_{\text{prop}}+T_{\text{Key\_Pack}\sharp 1})}{|K|} \tag{12}$$

该函数表明了安全门锁密钥可靠更新方案的优势和损失的带权定量评估结果。基于这个函数，本文提出动态安全门锁密钥可靠更新方案，如算法 2 所示。

算法 2　动态安全门锁密钥可靠更新方案 DRSKU()

```
Input:RSKU()算法的结果 result
Output:LM 和 SL_i 之间的最大重传次数
1. if(result==False):
2.    if(N>=N_max):
3.       return N_max=N_max+1
4.    else:
5.       RSKU()
6. else:
7.    if(N <N_max):
8.       return N_max=N_max-1
```

其中，N 代表当前重传的次数，$N_{\max}$ 代表最大重传次数。当 N 超过最大重传次数时，$N_{\max}$ 对应增加 1，反之则减少 1。我们可以通过修改函数中的参数，来动态决定密钥更新中的最大重传次数和等待确认时间，以使此密钥更新方案最优化。

5　实验结果及分析

5.1　实验环境

本文在 TI 16 位微控制器 MSP430F5438A 及 Windows Server 上实现了该密钥更新方案，并评测了该方案的性能。本文实验的环境：LM 配置为 Inter(R) Xeon(R)CPU E5-2682 v4 @2.50GHz 处理器，2GB 主存，MSP430F5438A 的性能参数为 16 位超低功耗微控制器，256kB 闪存，16kB RAM；所采用的窄带网络通信模块为移远 BC95，其上行速率为 62.5kbps，下行速率为 24kbps。本文实验主要测试了本方案在不同环境下的密钥更新成功率及更新时间。

5.2　方案性能分析

5.2.1　网络状况对方案性能的影响

本文方案的密钥更新成功率主要与设定的最大重传次数 $N_{\max}$ 和 NB-IoT 中的网络丢包率有关。我们将智能锁置于不同环境下，与阿里云锁管理员进行远程通信，进行了 2000 次

通信实验，并将其平均丢包率 P_l 作为本文的参考指标。通过理论分析，更新成功率将随着丢包率的减少而增加，之后的实验结果也印证了该观点。分别在 P_l 为 0.2，0.4，0.6 的情况下对 RSKU，AUDP[16]，DRSKU 3 种方案的密钥更新成功率进行实验，实验结果如图 3－图 5 所示。

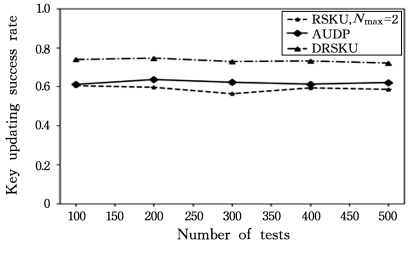

图 3 $P_l=0.6$ 时 3 种方案的密钥更新成功率

Fig. 3 Success ratio of key updating of three schemes when P_l is 0.6

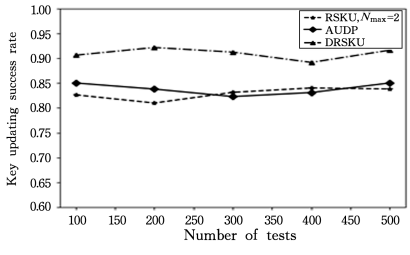

图 4 $P_l=0.4$ 时 3 种方案的密钥更新成功率

Fig. 4 Success ratio of key updating of three schemes when P_l is 0.4

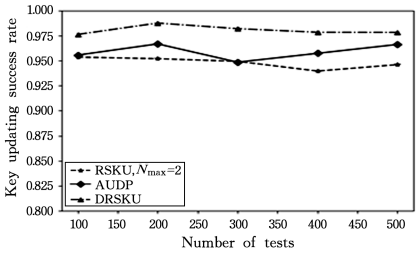

图 5 $P_l=0.2$ 时 3 种方案的密钥更新成功率

Fig. 5 Success ratio of key updating of three schemes when P_l is 0.2

从图 3－图 5 中容易看到，$N_{max}=2$ 时，RSKU 与 AUDP 方案在 NB-IoT 网路中的密钥更新成功率相近；而 DRSKU 能够根据网络状况动态地设置重传次数，从而达到最好的效果。因此，DRSKU 在 3 种方案中的成功率最高。

5.2.2 传输时延和方案评估

如前文所述，密钥更新时间是另一个影响方案性能的重要因素，为此测试了 RSKU，AUDP 和 DRSKU 3 种方案的平均密钥更新时间，测试结果如图 6 所示。

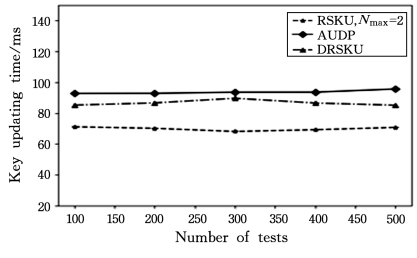

图 6 平均密钥更新时间的比较

Fig. 6 Comparison of average key updating time

从图 6 可以看出，进行密钥更新时，RSKU 与 AUDP 相比平均更新时间减少 32%，而 DRSKU 与 AUDP 相比平均更新时间减少 15%。正如前文所述，最大重传次数 N_{max}、密钥更新成功率 P_s 和网络丢包率 P_l 都将对本方案的性能产生影响。分别调整了这些参数，并通过评估函数 $Eval$ 在不同参数下进行方案的性能比较，结果如图 7 所示。

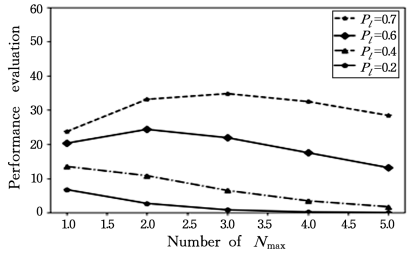

图 7 不同网络状况和最大重传次数下的性能比较

Fig. 7 Performance comparison in different network conditions and maximum retransmission times

由图 7 可知，随着网络状况的恶化（丢包率 P_l 增大），本方案性能最佳时最大重传次数 N_{max} 增加。这体现出动态安全门锁密钥可靠更新方案能够根据用户的倾向调整最大重传次数，以为用户带来最佳的使用体验。从上述结果也可看出，动态安全门锁密钥可靠更新方案的性能比 AUDP 方案的性能更好。

结束语 本文给出了一个窄带物联网下的安全门锁密钥可靠更新方案。该方案能够通过 UDP 协议安全可靠且实时地进行密钥更新，并保持密钥的一致性，从而在减少密钥更新通信开销的同时减少密钥更新的平均时延，以取得能耗与安全可靠性之间的最佳平衡。通过大量仿真实验，证明了本文方案的高效性和可靠性。将本文方案应用于日益蓬勃的各类 NB-IoT 应用场景下的远程密钥管理，是未来的研究计划。

参 考 文 献

[1] XV C E，XV F，LI X H，et al. Design of campus comprehensive access management platform[J]. Journal on Communications，2013，34(S2)：141-147. (in Chinese)
许彩娥，徐锋，厉晓华，等. 校园综合门禁管理平台的设计[J]. 通信学报，2013，34(S2)：141-147.

[2] ZHU H J，PAN Z F，ZHU Y L. "Internet＋" intelligent access control system[J]. Application of Electronic Technique，2017，43(3)：124-126，131. (in Chinese)
朱航江，潘振福，朱永利. "互联网＋"智能门禁控制系统[J]. 电子技术应用，2017，43(3)：124-126，131.

[3] NOUR K，MONA J，ZAHER D. Measurement-Based Signaling Management Strategies for Cellular IoT[J]. IEEE Internet of Things Journal，2017，4(5)：1434-1444.

[4] LIU Q，CUI L，CHEN H M. Key Technologies and Applications of Internet of Things[J]. Computer Science，2010，37(6)：1-4，10. (in Chinese)
刘强，崔莉，陈海明. 物联网关键技术与应用[J]. 计算机科学，2010，37(6)：1-4，10.

[5] ZHAO Z J，SHEN Q，TANG H，et al. Theory and Key Technologies of Architecture and Intelligent Information Processing for

Internet of Things[J]. Computer Science, 2011, 38(8): 1-8. (in Chinese)
赵志军，沈强，唐晖，等. 物联网架构和智能信息处理理论与关键技术[J]. 计算机科学，2011，38(8)：1-8.

[6] LUAN L X. Intelligent electronic door lock system based on GPRS and laser virtual keyboard[J]. Journal of Computer Applications, 2016, 36(S2): 319-321. (in Chinese)
栾禄祥. 基于 GPRS 和激光虚拟键盘的智能电子门锁系统[J]. 计算机应用，2016，36(S2)：319-321.

[7] PIAO Y, KIM J U, TARIQ U, et al. Polynomial-based key management for secure intra-group and inter-group communication [J]. Computers & Mathematics with Applications, 2013, 65(9): 1300-1309.

[8] LI Y N, YU Y, YANG B, et al. Privacy preserving cloud data auditing with efficient key update [J]. Future Generation Computer Systems, 2018, 76(2): 789-798.

[9] LUCA R, FRANCESCO M, JUSSI K, et al. A Semantic Publish-Subscribe Architecture for the Internet of Things[J]. IEEE Internet of Things Journal, 2016, 3(6): 1274-1296.

[10] MARC B, ALEJANDRO C, JAIME L, et al. IoT-Cloud Service Optimization in Next Generation Smart Environments[J]. Future Generation Computer Systems, 2016, 34(12): 4077-4090.

[11] ZHAO X, WU M Q, CHEN D X, et al. An Adaptive Re-Keying Mechanism for Secure Multicast[J]. Acta Electronica Sinica, 2003, 31(5): 654-658. (in Chinese)
赵欣，吴敏强，陈道蓄，等. 一个自适应的安全组通信密钥更新算法[J]. 电子学报，2003，31(5)：654-658.

[12] JIN C, WEI D X, LOW S H, et al. FAST TCP: motivation, architecture, algorithms, performance[J]. IEEE/ACM Transactions on Networking, 2006, 14(6): 1246-1259.

[13] XU L S, HARFOUSH K, RHEE I. Binary increase congestion control (BIC) for fast long-distance networks[C]// Proc. of IEEE INFOCOM. 2004: 2514-2524.

[14] HE E, LEIGH J, YU O, et al. Reliable blast UDP: predictable high performance bulk data transfer[C]// Proc. of IEEE International Conference on Cluster Computing. 2002: 317-324.

[15] GU Y H, GROSSMAN R L. UDT: UDP-based data transfer for high-speed wide area networks[J]. Computer Networks, 2007, 51(7): 1777-1799.

[16] LIU X Z, ZHOU J, LIANG D Q. Huge Data Blocks Transmission Based on UDP[J]. Telecommunication Engineering, 2012, 52(1): 96-100. (in Chinese)
刘喜作，周晶，梁德清. 基于 UDP 的大数据包可靠传输[J]. 电讯技术，2012，52(1)：96-100.

[17] LI Y M, REN Y M, LI J. Comparison and evaluation of UDP-based transport protocol performance[J]. Application Research of Computers, 2010, 27(5): 3096-3910. (in Chinese)
李一鸣，任勇毛，李俊. 基于 UDP 的传输协议性能比较与分析[J]. 计算机应用研究，2010，27(5)：3096-3910.

[18] HUANG C W, YEN C L, CHIANG C H, et al. The five modes AES applications in sounds and images[C]// Proc. of the 6th International Conference on Information Assurance and Security. 2010: 28-31.

[19] XIAO C L, ZHOU D Y, ZHANG K. CUDA based high-efficiency implementation of AES algorithm[J]. Application Research of Computers, 2013, 30(6): 1907-1909. (in Chinese)
夏春林，周德云，张堃. AES 算法的 CUDA 高效实现方法[J]. 计算机应用研究，2013，30(6)：1907-1909.