基于 MQTT 协议的轻量化文本信息分发技术研究

熊风光[1,2,3],陈　霖[1],韩慧妍[1,2,3],张　元[1,2,3],庞　敏[1,2,3],焦世超[1,2,3]
(1. 中北大学 计算机科学与技术,山西 太原 030051;
2. 机器视觉与虚拟现实山西省重点实验室,山西 太原 030051
3. 山西省视觉信息处理及智能机器人工程研究中心,山西 太原 030051)

摘　要:随着卫星通信网络的发展,基于卫星通信网络实现手机、平板电脑等移动通信设备之间的通信成为研究的热点。由于其采用无线电波进行信号传输,且需要经过多个中继站进行信号的转发和处理,导致存在网络延时大、丢包率高以及信道狭窄等问题。针对文本信息在卫星通信网络下的分发过程存在效率低下、可靠性不高的问题,设计一种基于 MQTT(Message Queuing Telemetry Transport)协议的轻量化文本信息分发技术。该技术使用 MQTT 协议作为消息传输协议,在文本信息分发前对 MQTT 协议进行主题设计、发布订阅机制设计、设备连接设计以及设备心跳设计,确保设备之间的连通性;在文本信息的分发过程中,设计数据校验加密算法、文本信息轻量化处理方法和离线消息存储机制,保证文本信息分发的安全性、可靠性。实验结果表明:相较于传统的基于 JSON 数据的文本信息分发技术,该技术在提高文本信息分发效率的同时,可确保信息分发的安全性、完整性和稳定性。

关键词:MQTT 协议;轻量化;文本信息分发;数据加密;离线消息存储
中图分类号:TP391　　**文献标识码**:A　　**文章编号**:1673-629X(2024)02-0090-08
doi:10.3969/j.issn.1673-629X.2024.02.014

Research on Lightweight Text Information Distribution Technology Based on MQTT Protocol

XIONG Feng-guang[1,2,3],CHEN Lin[1],HAN Hui-yan[1,2,3],ZHANG Yuan[1,2,3],
PANG Min[1,2,3],JIAO Shi-chao[1,2,3]
(1. School of Computer Science and Technology,North University of China,Taiyuan 030051,China;
2. Shanxi Key Laboratory of Machine Vision and Virtual Reality,Taiyuan 030051,China;
3. Shanxi Province's Vision Information Processing and Intelligent Robot Engineering Research Center,
Taiyuan 030051,China)

Abstract:With the development of satellite communication network,the communication between mobile devices such as mobile phones and tablet computers based on satellite communication network has become a research hotspot. Because it uses radio waves for signal transmission and needs to pass through multiple relay stations for signal forwarding and processing,it has problems such as large network delay,high packet loss rate and narrow channel. Aiming at the problems of low efficiency and low reliability of text information distribution in satellite communication network,a lightweight text information distribution technology based on MQTT (Message Queuing Telemetry Transport) protocol is designed. In this technology,MQTT protocol is used as the message transmission protocol. Before text information is distributed,subject design,publishing and subscription mechanism design,device connection design and device heartbeat design of MQTT protocol are carried out to ensure connectivity between devices. During text information distribution,data verification and encryption algorithms,lightweight text information processing methods,and offline message storage mechanisms are designed to ensure the security and reliability of text information distribution. Experimental results show that compared with traditional text information distribution technology based on JSON data,the proposed technology can improve the efficiency of text information distribution and ensure the security,integrity and stability of information distribution.

Key words:MQTT protocol;lightweight;text information distribution;data encryption;offline message store

收稿日期:2023-04-20　　**修回日期**:2023-08-23
基金项目:国家自然科学基金(62106238);山西省科技重大专项计划"揭榜挂帅"项目(202201150401021);山西省科技成果转化引导专项(202104021301055)
作者简介:熊风光(1979-),男,博士,副教授,CCF 会员(48960M),研究方向为软件开发与数据可视化。

0　引　言

随着移动互联网的发展和普及,越来越多的人使用手机、平板电脑等移动设备进行信息获取和交流[1]。借助传统的通信网络在一些偏远地区、灾害现场、海洋、航空等特殊场景下获取信息可能会受到限制。第一,传统的通信网络受限于地理位置和网络覆盖范围,导致其信号弱或没有信号;第二,传统的通信网络在自然灾害、战争等突发事件中容易受到破坏;第三,传统的通信网络需要铺设大量的基础设施才能实现信息传输。针对上述问题,卫星通信网络[2]的价值就体现了出来,卫星通信网络可以覆盖地面基站网络无法覆盖的地方,同时,卫星通信是天上的链路,不会受到地面灾害环境因素的影响,并且不依赖于基础设施建设。在复杂的环境场景下,使用卫星通信网络进行通信设备之间的通信是一个不错的选择。卫星通信网络虽有覆盖范围大、依赖性小等优点[3],但其存在着一些缺点:其一,卫星通信网络的网络时延大、丢包率高,导致数据在传输过程中会出现不稳定情况而丢失;其二,卫星通信网络的信道窄,在传输数据信息量比较大时会增加传输时间、降低传输效率。以上问题导致卫星通信网络无法完全在通信设备中展开使用。现有文本信息分发技术是基于 JSON 数据的传统分发技术以及数据分块和重组分发技术。但是已有技术在卫星通信网络环境下分发有以下不足之处:第一,由于卫星的频谱资源有限,分发大量的文本信息可能会导致带宽瓶颈,限制数据传输的速度和容量;第二,由于信号在地面和卫星之间的传输,有可能被恶意攻击者截获、窃听或篡改;第三,数据分块和重组分发技术是将文本信息分成较小的数据块,通过卫星链路逐个传输,然后在接收端重新组装成完整的文本信息。如果某个数据块丢失或出错,整个文本信息的完整性可能会受到影响。针对上述问题,该文研究了基于 MQTT(Message Queuing Telemetry Transport)协议的轻量化文本信息分发技术。设计了数据校验加密算法、文本信息轻量化处理方法和离线消息存储机制。

1　相关工作

1.1　MQTT 协议

MQTT[4]是采用发布/订阅模式、基于 TCP/IP 协议的通信协议,具有轻量级、简单、开放等特点。它能够在低带宽或者不可靠的网络上传输数据,功耗非常低,以轻量级、简单、易于部署等优点使 MQTT 成为约束环境下的理想通信协议[5]。

1.2　GZIP 压缩算法

GZIP 算法是一种基于 DEFLATE 算法[6]的数据压缩算法,这是 HTTP 标准中使用的压缩方案之一,通过减少网络传输的字节大小来最小化延迟。它采用了霍夫曼编码和 LZ77 算法对数据进行压缩,可以获得更高的压缩比和更快的解压速度[7]。GZIP 广泛应用于各种领域,包括文件压缩、网页压缩、日志文件压缩等。它在 Web 服务器中常用于对传输的内容进行压缩,以减少数据传输的带宽消耗和提高用户访问速度。

1.3　AES 加密算法

AES(Advanced Encryption Standard,高级加密标准)是一种安全可靠的对称加密算法,其密钥长度可达到 128 位、192 位或者 256 位,具有安全性高、速度快、可靠性高等特点[8]。AES 被认为是一种安全可靠的加密算法,已经通过了多个密码学标准的审查和认可,应用于保护敏感数据的安全性,包括网络通信、数据存储和加密文件等领域。

2　文中方法

2.1　总体架构

针对文本信息在卫星通信网络下的分发过程存在安全性弱、效率低下、可靠性不高的问题,该文开展轻量化文本信息分发技术研究。主要工作如下:

(1)基于 MQTT 协议进行设备通信准备工作,建立可靠的数据链路连接;

(2)针对文本信息安全性弱的问题,将 AES 加密算法与 CRC 校验工具结合使用,对加密的文本信息在分发至接收端进行解密后执行 CRC 校验,检验数据是否被篡改;

(3)针对文本信息分发效率低下的问题,对分发的文本信息进行轻量化操作,即利用 GZIP 压缩算法对待分发的信息进行压缩;

(4)针对文本信息分发过程中可靠性不高的问题,设计离线消息存储机制,确保文本信息可以被接收端接收。

轻量化文本信息分发流程如图 1 所示。文本信息由服务端分系统进行分发,应用端分系统接收并展示。首先进行创建主题、创建发布订阅机制、设计设备心跳和设备连接等过程,连通服务端分系统与应用端分系统的通信状态。然后服务端准备向应用端发送文本信息,服务端生成文本信息校验码、文本信息进行加密处理、文本信息进行轻量化处理、文本信息序列化。经过处理后的文本信息准备分发,分发之前判断应用端用户是否在线,若不在线,则将文本信息进行离线存储,待用户上线之后发送;若在线,则将文本信息分发至应用端。应用端在接收文本信息之后,进行文本信息反序列化、数据解密、数据完整性校验等一系列操作,数据完整性校验用于判断数据在分发的过程中是否被恶意篡改,若文本信息被篡改,则通知服务端文本信息被

篡改并且在应用端不进行展示;若文本信息没有被篡改,则提取文本信息,进行文本信息展示。

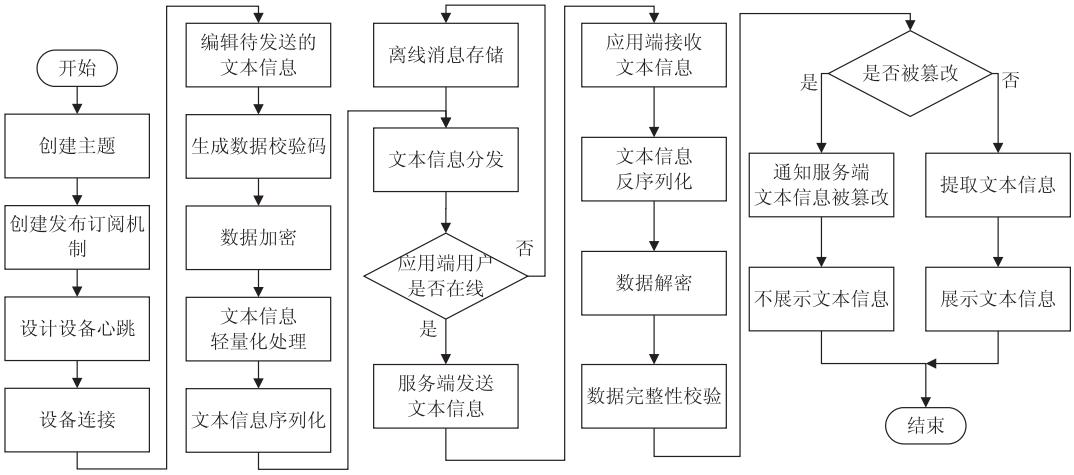

图 1 轻量化文本信息分发流程

2.2 设备通信准备工作

2.2.1 主题设计

主题[9]本质上是一个使用 UTF-8 编码的字符串,用来对发布的消息进行区分,在 Publish 报文中,随着消息载体一起发布。主题可以是由单一级别的主题组成,例如 user1,也可以是由多个级别的主题组成,例如 qingliang/clientUser/user1,主题的各个级别之间使用“/”进行分割。主题是区分大小写的,user 和 User 是两个不同的主题。

客户端在订阅主题时,可以选择订阅特定的主题消息,也可以通过使用通配符订阅若干个主题消息。通配符只能在订阅主题时使用,而不能在发布消息时使用。主题通配符有两种存在形式:单级通配符、多级通配符。单级通配符使用“+”表示,例如 qingliang/+/user,单级通配符可以代替一个主题层级。多级通配符使用“#”表示,例如 qingliang/#/user,多级通配符可以代替任意数量的主题层级。

在轻量化文本信息分发技术的研究中,设计了两种主题,分别如下:

(1)登录主题:用于应用端登录的主题,服务端订阅;

(2)信息分发主题:用于服务端向应用端下发文本信息的主题,客户端订阅。

2.2.2 发布订阅机制

发布订阅(Publish/Subscribe)机制[10]是一种消息转发机制,在这种消息转发机制下,消息分发的完整流程分为三步:订阅-发布-转发。同一客户端既可以是消息的发布者,也可以是消息的订阅者。订阅-发布流程如图 2 所示,客户端 3 订阅主题 qingliang/user/user2、客户端 4 订阅主题 qingliang/user/user1,客户端 1 向主题 qingliang/user/user1 发布消息“10”、客户端 2 向主题 qingliang/user/user2 发布消息“20”。由于客户端 3 订阅主题与客户端 2 发布消息的主题匹配成功,客户端 3 会接收到由客户端 2 发布并且由服务器转发的消息“20”。同理,客户端 4 会接收到由客户端 1 发布并且经过服务器转发的消息“10”。

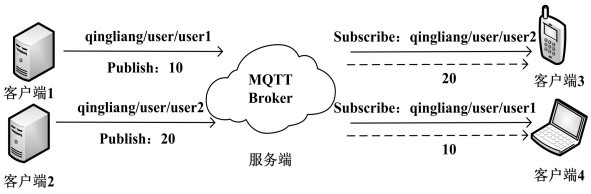

图 2 发布订阅机制

2.2.3 设备连接设计

客户端设备与服务端建立可靠的连接是后续一切服务的基础,MQTT 客户端之间想要实现发布订阅机制、通信等服务,必须要通过 MQTT 服务端,因此客户端与服务端建立连接是一项很重要的工作。

MQTT 客户端与服务端在建立连接的过程中,客户端先向服务端发送 CONNECT 报文,服务端收到报文后,向客户端发送 CONNACK 报文。CONNECT 报文包含了 clientId, cleanSession, keepAlive, username, password 等数据信息,其中 clientId 是客户端的唯一标识;cleanSession 有 true 和 false 两个值,如果设置为 true,则表示当服务端向客户端发送消息时,无论客户端是否收到,都不会再次发送,这种情况适用于发送不重要的报文,如果设置为 false,则表示服务端会接收客户端的确认消息,服务端会保存信息继续发送,直至接收到客户端的确认收到消息,这种情况适合重要的报文,并且在此情况下 Qos 的设置要大于 0;keepAlive 是用于服务端实时监测客户端的连接情况。CONNACK 报文包含了 returnCode,sessionPresent 等数据信息,其中 returnCode 是连接的状态码,用来表示客户端的连接情况,返回“0”表示客户端与服务端连接成功,返回非“0”值表示连接失败;sessionPresent 表示

当前 clientId 所对应的设备是否有未确认收到的消息,如果返回值为 true,说明此时有尚未确认收到的消息,不需要重新订阅主题,连接之后会收到保存在服务端的消息;如果返回值为 false,说明原本的订阅关系已消失,需要重新订阅主题。

2.2.4　设备心跳设计

设备心跳设计[11]是实现客户端与服务端建立可靠连接的关键。为了确认客户端与服务端的连接状态,除了客户端正常向服务端发送消息之外,在没有发送消息的空闲时间,客户端会在心跳时间主动向服务端发送消息,这条消息称为 PINGREQ(心跳请求),如果服务端收到了客户端的消息,则认为客户端处于在线状态,同时会回复一条消息,这条消息称为 PINGRESP(心跳响应),反之则处于离线状态。

由于心跳时间是在客户端设定的,为了让服务端知道客户端的心跳时间,客户端需要将该时间通知给服务端。在 2.2.3 设备连接时的 CONNECT 报文中有一个字段是 keepAlive,这个字段就是用于设置心跳时间并通知服务端。

心跳机制是否会触发是根据客户端发送消息的频率来决定的。第一,客户端在心跳时间之内向服务端发送消息,此时服务端不需要心跳请求也可以知道客户端是在线的,这个时候心跳机制没有触发;第二,客户端在心跳时间之内没有向服务端发送消息,此时客户端需要发送心跳请求给服务端,这个时候触发了心跳机制。服务端在 1.5 倍的心跳时间之内没有收到客户端发送的心跳请求或者普通消息,如公式 1,则认为客户端与服务端已经断开了连接。

$$stime > 1.5 \times ctime \tag{1}$$

其中,ctime 是客户端心跳时间,stime 是服务端监测时间。

2.3　轻量化文本信息分发技术研究

2.3.1　数据校验加密算法

数据校验和加密算法是保护数据安全的重要措施。数据校验算法[12]可以检测到数据在传输过程中是否被篡改,确保数据的完整性;数据加密算法[13]可以将数据转换为一种难以理解的格式,防止数据被篡改,确保数据的机密性。数据在发送方要先校验后加密,接收方先解密后校验。该文将循环冗余校验算法(CRC)与 AES 加密算法相结合以确保数据在传输过程中的完整性和机密性。

数据校验算法是用于检测数据在传输或存储过程中是否被篡改或损坏的技术。常见的数据校验算法包括奇偶校验、循环冗余校验、校验和、哈希函数等。CRC 可以高效地检测数据是否被篡改,不会占用过多的系统资源或带来太大的通信延迟;CRC 与其他数据校验算法相比具有更强的检测能力。因此,该文的数据校验算法采用循环冗余校验算法(CRC)。

数据加密是保护数据安全的重要措施[14]。数据加密算法可以将数据转换为一种难以理解的格式,防止数据被篡改,确保数据的机密性。数据加密算法分为以下三类:对称加密算法、非对称加密算法、散列函数算法。由于对称加密具有加密和解密速度快、加密效率高等优点,该文选择对称加密算法中的 AES 进行加密。加密的具体流程如图 3 所示。

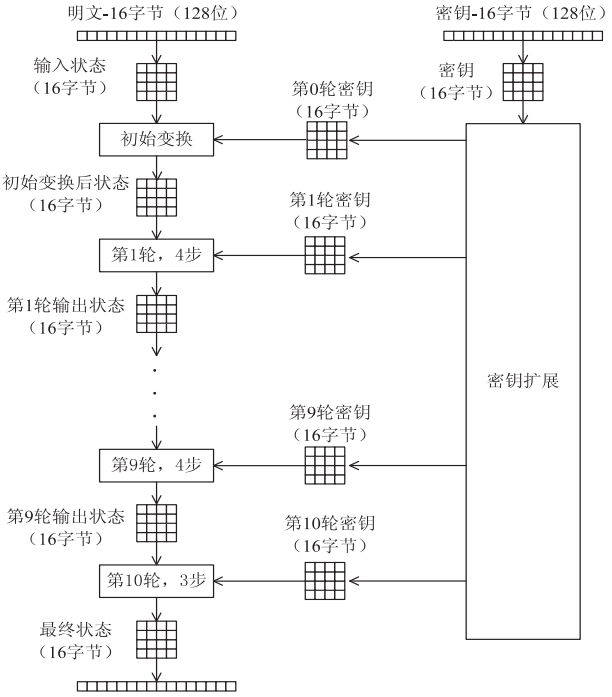

图 3　数据加密流程

(1)字节代换。

每一轮的第一个阶段都是从字节代换开始,这一步依赖于非线性 S-box,将原始矩阵通过 S-box 进行变换。将 4×4 矩阵的每一个字节都进行字节代换,高 4 位作为 x 值,低 4 位作为 y 值,通过 x , y 值在 S-box 中找到对应的目标值进行替换,得到一个新的矩阵。

(2)行位移。

字节代换之后执行状态的下一步是行位移,这一步是在每行中循环地将状态字节向左移动。第一行的字节保持不变,第二行的字节向左循环移动一个字节,第三行的字节向左循环移动两个字节,第四行的字节向左循环移动三个字节。新状态矩阵没有改变大小,仍然是原来的 16 字节,但是改变了状态字节的位置。

(3)列混合。

行位移的下一步是列混合,这一步只在前 9 轮中使用,第 10 轮是没有这一步的。这一步是将输入的矩阵左乘一个给定的 4×4 矩阵,得到一个新的矩阵,见公式 2。由于列混合的运算是在 GF(2^8)域上的运算,所以运算方式和平时的十进制运算方法是有所区别

的。公式 3 计算第一行数据、公式 4 计算第二行数据、公式 5 计算第三行数据、公式 6 计算第四行数据。

$$\begin{bmatrix} K_{0,0} & K_{0,1} & K_{0,2} & K_{0,3} \\ K_{1,0} & K_{1,1} & K_{1,2} & K_{1,3} \\ K_{2,0} & K_{2,1} & K_{2,2} & K_{2,3} \\ K_{3,0} & K_{3,1} & K_{3,2} & K_{3,3} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{bmatrix} \tag{2}$$

$$K_{0,j} = (2 * S_{0,j}) \oplus (3 * S_{1,j}) \oplus S_{2,j} \oplus S_{3,j} \tag{3}$$

$$K_{1,j} = S_{0,j} \oplus (2 * S_{1,j}) \oplus (3 * S_{2,j}) \oplus S_{3,j} \tag{4}$$

$$K_{2,j} = S_{0,j} \oplus S_{1,j} \oplus (2 * S_{2,j}) \oplus (3 * S_{3,j}) \tag{5}$$

$$K_{3,j} = (3 * S_{0,j}) \oplus S_{1,j} \oplus S_{2,j} \oplus (2 * S_{3,j}) \tag{6}$$

给定的矩阵中只有 01,02,03 三种数据,接下来对这三种数据的运算进行描述。所有的运算都要转成二进制运算,01 乘以任何数都是本身;02 的二进制是 00000010,02 与其他数据相乘的规则见公式 7;03 的二进制是 00000011,03 与其他数据相乘的规则见公式 8。

$$(00000010) * (a_7a_6a_5a_4a_3a_2a_1a_0) = \begin{cases} (a_6a_5a_4a_3a_2a_1a_00), a_7 = 0 \\ (a_6a_5a_4a_3a_2a_1a_00) \oplus (00011011), a_7 = 1 \end{cases} \tag{7}$$

$$(00000011) * (a_7a_6a_5a_4a_3a_2a_1a_0) = [(00000010) \oplus (00000001)] * (a_7a_6a_5a_4a_3a_2a_1a_0) = [(00000010) * (a_7a_6a_5a_4a_3a_2a_1a_0)] \oplus (a_7a_6a_5a_4a_3a_2a_1a_0) \tag{8}$$

(4)轮密钥加。

列混合的下一步是轮密钥加,这一步是 AES 算法中最关键的阶段,能够在加密数据期间提供更多的安全性,创建密钥与密文之间的关系。轮密钥加的第一次运算(即初始变换)是将明文矩阵与原始密钥矩阵进行异或运算,剩下 10 次的轮密钥加运算是将经过密钥扩展产生的新密钥与上一步产生的新矩阵进行异或运算。

(5)密钥扩展。

AES 算法是基于 AES 密钥扩展进行加密和解密的算法,密钥扩展是 AES 算法中重要的步骤。每一轮都会产生一把新密钥,新密钥用于每一轮的加密。密钥的大小是 16 字节(K0 ~ K15),前四个字节 K0 ~ K3 表示为 W0,接下来的四个字节 K4 ~ K7 表示为 W1,以此类推。

每一轮密钥扩展运算规则如下:如果 i 不是 4 的倍数,那么第 i 列由式 9 确定;

$$W[i] = W[i-4] \oplus W[i-1] \tag{9}$$

如果 i 是 4 的倍数,那么第 i 列由式 10 确定。

$$W[i] = W[i-1] \oplus T(W[i-1]) \tag{10}$$

公式 10 的函数 T 由 3 部分组成:字循环、字节代换和轮常量异或。字循环是将 1 个字中的 4 个字节循环左移 1 个字节;轮常量异或是将经过前两步得到的结果同轮常量 Rcon[j]进行异或,其中 j 表示轮数。

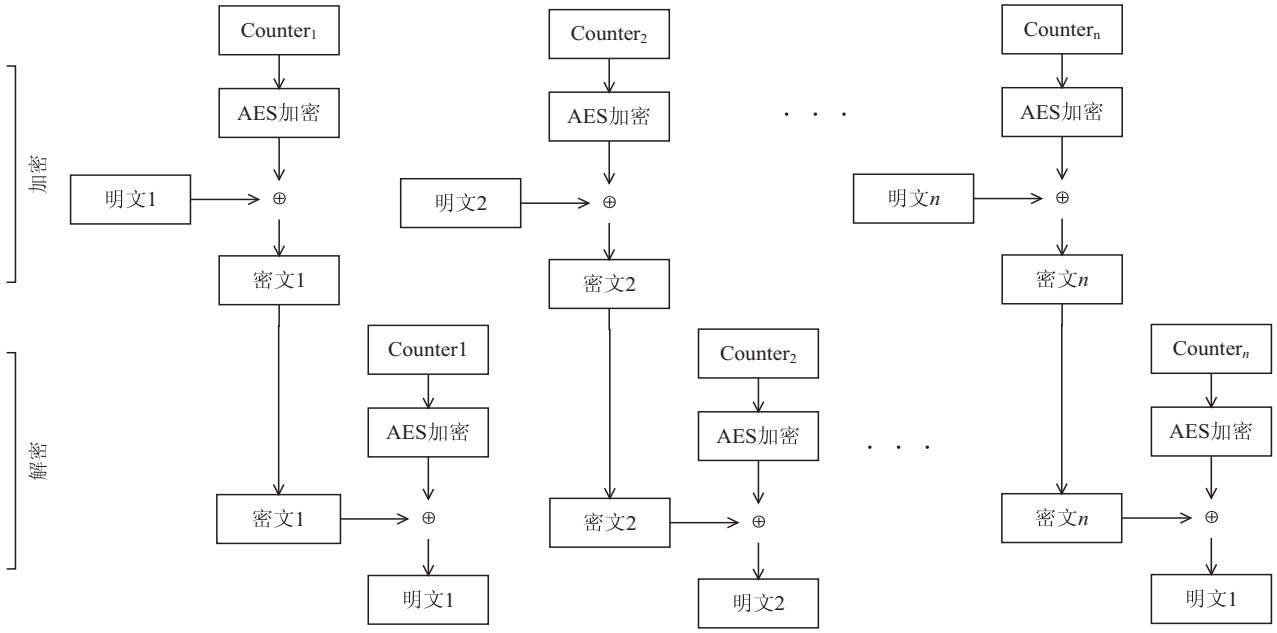

图 4 文件加密解密过程

上述加密过程是针对 16 字节的数据,但实际上文件大小不会是 16 字节。对于一个文件的加密解密过程如图 4 所示,将一个文件分成 n 个 16 字节长度的明文数据,定义一个 16 字节的初始向量 counter1,对

counter1 进行 AES 加密,得到的数据和明文 1 进行异或得到密文 1;counter1 加 1 得到 counter2,对 counter2 进行 AES 加密,得到数据和明文 2 进行异或得到密文 2,以此类推,一直计算到最后一个密文 n ,将 n 个密文拼接在一起就是文件对应的密文。解密过程与加密过程类似,counter1 进行 AES 加密后得到的数据与密文 1 进行异或得到明文 1,counter2 进行 AES 加密后得到的数据与密文 2 进行异或得到明文 2,以此类推,一直计算到最后一个明文 n ,将 n 个明文拼接在一起就是解密后的文件数据。在这种加密解密过程中,双方互相需要知道 counter1 的值和 AES 的密钥。

2.3.2　文本信息轻量化处理

为了减轻网络带宽压力、提高传输效率,在文本信息分发之前需要进行轻量化处理。提出使用 GZIP 算法[15]结合文本信息的特点进行压缩。

轻量化文本信息分发技术结合 GZIP 算法进行文本信息压缩的流程如图 5 所示。首先,提取准备分发的文本消息,将提取到的消息使用 GZIP 压缩算法进行压缩,得到更小的空间占用;接着,将压缩后的消息转换为字节流数组,将字节流数组封装至待分发的消息体中;然后,将消息分发至应用端,应用端接收到消息之后进行转码解压缩;最后,将得到的消息进行完整展示。

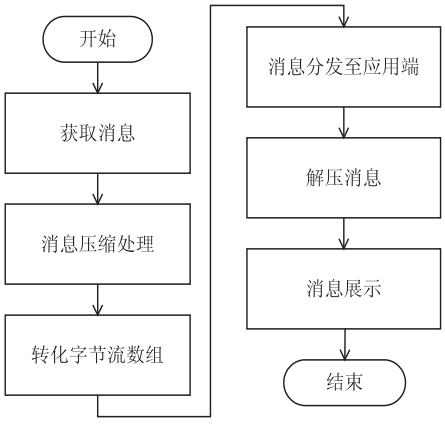

图 5　文本信息轻量化处理流程

2.3.3　离线消息存储

由于不稳定的网络环境,应用端设备会出现异常掉线的情况。在应用端设备异常掉线或者应用端设备用户没有登录的期间,可能会有应用端用户订阅的主题消息发布,为了保证应用端用户上线之后能够正常收到离线时发布的消息,系统需要将这部分离线的消息存储起来,等到应用端用户上线之后再将消息推送给应用端用户,以确保消息的可靠传输。

离线消息存储的设计对本系统的开发有以下优点:

(1)离线消息存储可以确保即使应用端用户处于离线状态,也能够在用户上线时接收到消息;

(2)应用端用户不用担心错过任何信息,增强用户的满意度和体验感;

(3)离线消息存储在数据库中,减少了消息的重复传输,从而降低了网络的负载,提高了传输的效率。

综上所述,该文将离线消息存储到 MySQL 数据中,并且根据应用端用户上线时的用户名关键字将离线消息推送给指定用户。具体的离线消息存储设计如下所示:

首先在 MySQL 数据库中创建一个离线消息存储表 temporary_storage,包含字段主键(id)、接收者 id(application_user_id)、消息 id(message_id)、消息内容(content)、消息类型(type)。数据库表设计如表 1 所示。

表 1　离线消息存储表 temporary_storage

列名	数据类型	注释	约束
id	int(11)	主键 id	主键
application_user_id	varchar(255)	接收者 id	
message_id	varchar(255)	消息 id	
content	text	消息内容	
type	int(11)	消息类型	

服务端用户在给应用端用户发送消息时,先将数据存储到 temporary_storage 表中,再发送给应用端用户。应用端用户收到消息后,返回至服务端一条确认收到消息,此时服务端将 temporary_storage 表中数据根据 message_id 进行删除;如果没有接收到应用端用户的确认消息,服务端则认为应用端用户处于离线状态,将数据存储至 temporary_storage 表中。当应用端用户登录时,根据用户名 application_id 关键字检查 temporary_storage 表中是否是该用户的离线消息,将属于该用户的离线消息进行补发。

3　实验结果与分析

为了验证本章消息分发技术的设计以及数据加密算法、消息轻量化处理、数据校验算法、离线消息存储的正确性、可行性以及有效性,本小节设计了如下实验,并且将本章所研究的轻量化文本信息分发技术与传统 JSON 文本信息分发技术进行对比。

(1)数据加密实验。

本节将使用的 AES 加密算法和目前流行的 MD5,SHA-256,DES,RSA 四种加密算法进行性能比较,使用 hutool 工具对五种加密算法进行加密速度上的实验对比。由于一次加密过程的计算时间具有偶然性,为了尽可能消除偶然性因素,在该实验中随机生成 10 个长度为 100 字节的字符串进行 10 次加密,记录每

次加密的时间并计算平均耗时,时间单位为纳秒。实验对比结果如表 2 所示。

表 2 数据加密算法性能对比

算法	1	2	3	4	5	6	7	8	9	10	平均
MD5	160 900	121 300	102 300	94 700	109 400	109 400	100 700	82 400	87 200	79 600	104 790
SHA-256	249 100	279 800	150 800	129 800	102 200	110 700	67 800	67 700	72 000	94 400	132 430
DES	137 100	792 800	211 100	122 700	102 900	98 700	118 800	101 500	140 700	107 100	193 340
文中加密算法	98 900	63 100	102 400	104 900	96 700	90 200	94 800	95 500	92 800	96 400	93 570
RSA	144 800	140 100	129 800	131 800	148 100	142 700	127 800	134 600	2 077 300	190 600	336 760

(2)消息轻量化处理实验。

(a)编排一条数据,采用文本信息轻量化处理技术发送给在线用户;

(b)同样的一条数据,采用传统 JSON 技术发送给在线用户;

(c)查看控制台输出的消息分发前的长度与分发后长度。

实验结果如表 3 所示。经过文本信息轻量化处理技术处理的文本信息长度小于原始文本信息。第二行是通过传统 JSON 文本信息进行分发的结果,原始文本信息为 1 745 bytes,经过 JSON 分发的文本信息为 1 755 bytes;第三行是经过该文研究的轻量化文本信息分发技术的结果,原始文本信息为 1 745 bytes,经过轻量化处理之后文本信息为 931 bytes。

表 3 文本信息分发对比

对比方式 方法	原始文本信息 大小/bytes	分发文本信息 大小/bytes	压缩率 /%
传统 JSON 文本 分发技术	1 745	1 755	100.57
轻量化信息 分发技术	1 745	931	46.65

(3)数据校验实验。

(a)应用端设备用户正常登录;

(b)一条数据,发送给应用端用户;

(c)编排一条新的数据,在后台对这条数据的字节进行修改,模拟数据被篡改,再次发送给应用端用户。

图 6 为验证结果,图 6(a)是第一条数据分发情况,第一条数据成功被应用端所接收;图6(b)是第二条数据分发情况,第二条数据没有被应用端接收,并且在服务端显示数据被篡改。

(a)数据分发被应用段接收

(b)数据分发未被应用端接收

图 6 数据分发校验验证结果

(4)离线消息存储实验。

(a)应用端设备用户没有登录,处于离线状态;

(b)编排一条数据,发送给离线的用户;

(c)应用端用户登录账号,查看消息接收状态。

图 7 为验证结果,图 7(a)表明用户离线时,数据接收失败,服务端显示用户不在线;图 7(b)表明应用端用户登录之后,数据成功接收,并且在服务端显示接收成功。

(a)用户离线下的分发失败

(b)用户登录后分发成功

图 7 离线消息分发结果

实验结果分析:

(1)数据加密算法的实验过程及结果表明,AES 在所对比的算法中具有较快的加密速度,证明了加密

算法的先进性。

(2)轻量化处理验证过程及结果表明,文本信息通过轻量化处理技术分发后的长度明显小于消息分发前的长度,可见文本信息在发送时成功地进行了轻量化处理,证明了轻量化处理的可行性。

(3)数据校验算法的验证过程及结果表明,没有被篡改的数据可以被应用端用户成功接收,已经发生篡改的数据是不可以被应用端用户接收的,在服务端的分发结果界面可以看到实时每条数据的接收情况,证明了数据校验的有效性。

(4)离线消息存储的验证过程及结果表明,当用户离线时,已经发送的消息被暂存至数据库,待用户在线时,用户可以成功接收到该用户对应的离线数据,证明了离线消息存储机制的正确性、可靠性。

4　结束语

针对文本信息分发过程中存在的效率低下、可靠性不高的问题,研究了基于 MQTT 协议的轻量化文本信息分发技术。在文本信息分发过程中,为了保证数据的安全性、完整性和稳定性,提出了数据校验加密算法以及离线消息存储等方法。同时,为了适应网络带宽、提高分发效率,提出了对文本信息进行轻量化处理的方法。

但当前研究工作仍有待改进,文本信息分发技术采用的数据加密算法属于传统加密算法,在未来可以基于人工智能等新技术对加密算法有更深层次的研究,从而应对未来可能面临的各种安全威胁。

参考文献:

[1] 共论中国互联网发展[J]. 中国信息界,2022(6):19-21.

[2] KODHELI O, LAGUNAS E, MATURO N, et al. Satellite communications in the new space era: a survey and future challenges[J]. IEEE Communications Surveys & Tutorials, 2020,23(1):70-109.

[3] 梁　程. 卫星通信网络面临的安全威胁及防范分析[J]. 网络安全技术与应用,2023(3):8-10.

[4] SONI D,MAKWANA A. A survey on mqtt: a protocol of internet of things (iot)[C]//International conference on telecommunication, power analysis and computing techniques (ICTPACT-2017). India:[s. n.],2017:173-177.

[5] 姬广龙,孙　丹,王珍珍,等. 关于 MQTT 通信协议的分析与研究[J]. 物联网技术,2023,13(1):63-66.

[6] PURWANTO R, PAL A, BLAIR A, et al. PhishZip: a new compression-based algorithm for detecting phishing websites [C]//2020 IEEE conference on communications and network security (CNS). Avignon: IEEE, 2020:1-9.

[7] LEDWON M, COCKBURN B F, HAN J. High-throughput FPGA-based hardware accelerators for deflate compression and decompression using high-level synthesis[J]. IEEE Access,2020,8:62207-62217.

[8] ABDULLAH A M. Advanced encryption standard (AES) algorithm to encrypt and decrypt data[J]. Cryptography and Network Security,2017,16:1-11.

[9] TANTITHARANUKUL N, OSATHANUNKUL K, HANTRAKUL K, et al. MQTT-topics management system for sharing of open data[C]//2017 international conference on digital arts, media and technology (ICDAMT). Thailand: IEEE, 2017:62-65.

[10] 王玉杰,马旭东,房　芳,等. 基于发布订阅机制的分布式通信中间件设计[J]. 工业控制计算机,2022,35(6):111-113.

[11] 陈文艺,梁宁宁,杨　辉. 基于 MQTT 的物联网网关双向通信系统设计[J]. 传感器与微系统,2022,41(8):100-103.

[12] 张正龙,张小华,李冀明,等. 基于 CRC32 的数据校验的研究和应用[J]. 科学咨询,2011(2):62-63.

[13] 林銮云. 数据加密技术在计算机软件安全中的应用研究[J]. 无线互联科技,2022,19(23):90-92.

[14] 严　凡. 数据加密技术在计算机网络信息安全中应用分析[J]. 网络安全和信息化,2023(2):116-118.

[15] 李　博,袁兴峰,李　隆. 一种基于 GZIP 的压缩与高效解压系统[J]. 电子设计工程,2021,29(8):48-52.