Journal of Computer Applications
计算机应用, 2021, 41(5): 1378 - 1385
ISSN 1001-9081
CODEN JYIIDU
2021-05-10
http://www.joca.cn

文章编号:1001-9081(2021)05-1378-08
DOI:10.11772/j.issn.1001-9081.2020060985

基于代理重加密的消息队列遥测传输协议端到端安全解决方案

谷正川[1,2*],郭渊博[1],方　晨[1]
(1. 战略支援部队信息工程大学 密码工程学院,郑州 450002; 2. 中国人民解放军77562部队,西藏 日喀则 857000)
(*通信作者电子邮箱zhengchuan_g@163.com)

摘　要:针对消息队列遥测传输(MQTT)协议缺乏保护物联网(IoT)设备间通信信息的内置安全机制,以及MQTT代理在新的零信任安全理念下的可信性受到质疑的问题,提出了一种基于代理重加密实现MQTT通信中发布者与订阅者间端到端数据安全传输的解决方案。首先,使用高级加密标准(AES)对传输数据进行对称加密,以确保数据在整个传输过程中的机密性;然后,采用将MQTT代理定义为半诚实参与方的代理重加密算法来加密传输AES对称加密使用的会话密钥,从而消除对MQTT代理的隐式信任;其次,将重加密密钥生成的计算工作从客户端转移到可信第三方,使得所提方案适用于资源受限的IoT设备;最后,使用Schnorr签名算法对消息进行数字签名,以提供数据来源的真实性、完整性和不可否认性。与现有MQTT安全方案相比,所提方案用和不提供端到端安全性的轻量级方案相当的计算和通信开销获取了MQTT通信的端到端安全特性。

关键词:消息队列遥测传输;安全性;密码学;代理重加密;高级加密标准;数字签名
中图分类号:TP309　　**文献标志码:**A

End-to-end security solution for message queue telemetry transport protocol based on proxy re-encryption

GU Zhengchuan[1,2*], GUO Yuanbo[1], FANG Chen[1]
(1. *College of Cryptography, Information Engineering University of Strategic Support Force, Zhengzhou Henan* 450002, *China*;
2. *PLA* 77562 *Troop, Shigatse Tibet* 857000, *China*)

Abstract: Aiming at the lack of built-in security mechanism in Message Queue Telemetry Transport (MQTT) protocol to protect communication information between the Internet of Things (IoT) devices, as well as the problem that the credibility of MQTT broker is questioned in the new concept of zero trust security, a new solution based on proxy re-encryption for implementing secure end-to-end data transmission between publisher and subscriber in MQTT communication was proposed. Firstly, the Advanced Encryption Standard (AES) was used to symmetrically encrypt the transmitted data for ensuring the confidentiality of the data during the transmission process. Secondly, the proxy re-encryption algorithm that defines the MQTT broker as a semi-honest participant was adopted to encrypt the session key used by the AES symmetric encryption, so as to eliminate the implicit trust of the MQTT broker. Thirdly, the computation of re-encryption key generation was transferred from clients to a trusted third party for the applicability of the proposed scheme in resource-constrained IoT devices. Finally, Schnorr signature algorithm was employed to digitally sign the messages for the authenticity, integrity and non-repudiation of the data source. Compared with the existing MQTT security schemes, the proposed scheme acquires the end-to-end security features of MQTT communication at the expense of the computation and communication overhead equivalent to that of the lightweight security scheme without end-to-end security.

Key words: Message Queue Telemetry Transport (MQTT); security; cryptography; proxy re-encryption; Advanced Encryption Standard (AES); digital signature

0　引言

物联网作为一项改变数据共享格局的技术,近年来伴随着无线通信技术的发展掀起了新的发展浪潮。据全球移动通信系统协会(Global System for Mobile Communications Association, GSMA)预测,2025年全球物联网终端连接数量将达到250亿[1]。物联网将数字世界与物理世界相映射,通过物联网连接到互联网的数百亿设备在共享信息的过程中可能生成数万亿条信息。如何保证这些信息的安全性是一项基本挑战。消息队列遥测传输(Message Queue Telemetry Transport, MQTT)作为物联网中流行的应用层协议之一,因它在资源和计算上的占用空间小而得到广泛应用[2]。但是MQTT协议中只规范了一些可选的弱防护机制,如使用客户端ID以及用户名/口令对客户端进行身份验证,这种程度的安全性机制不能满足物联网中不断增长的安全性需求。为提高MQTT的安全

收稿日期:2020-07-08;**修回日期:**2020-11-11;**录用日期:**2020-11-13。
基金项目:信息保障技术重点实验室基金资助项目(614211203010417)。
作者简介:谷正川(1989—),男,重庆人,硕士研究生,主要研究方向:物联网安全、零信任网络; 郭渊博(1975—),男,陕西周至人,教授,博士,主要研究方向:网络安全、态势感知; 方晨(1993—),男,安徽安庆人,博士研究生,主要研究方向:人工智能安全、隐私保护。

性,协议给出了一些建议,如客户端认证使用X. 509客户端证书、底层传输协议使用安全传输层协议(Transport Layer Security, TLS)代替纯传输控制协议(Transmission Control Protocol, TCP)。但这些机制的安全性成本超出了物联网受限设备在计算能力和内存等方面的可接受范围。因此,开发一种具备高安全性且适用于物联网中资源受限设备的解决方案,成为MQTT安全性研究的热点。

另一个问题是MQTT代理的可信任性越来越受到质疑。MQTT通信模型属于发布/订阅范式,其通信依赖的核心是MQTT代理。零信任是一种新型安全框架,其核心理念是"从不信任并始终验证",即内部和外部网络都不可信[3]。2020年2月美国国家标准与技术研究院(National Institute of Standards and Technology, NIST)发布的零信任架构(第2版草案)[4]再次强调:零信任是一种以资源保护为核心的网络安全范式,其前提是信任从来不是隐式授予的,而是必须进行持续评估。因此,即使MQTT代理位于受防火墙保护的内部网络也是不可信的。而且,随着云技术的发展,MQTT代理服务器也从本地转移至云端。阿里云、腾讯云等云服务商提供的MQTT云服务在提供便利时也带来了代理服务器端数据安全的巨大隐患。

为解决上述问题,本文消除对MQTT代理的隐式信任,将其定义为半诚实参与方。通过结合Schnorr签名[5]、高级加密标准(Advanced Encryption Standard, AES)和代理重加密三种密码技术,提出一种适用于受限物联网设备的MQTT端到端数据安全传输解决方案,其主要工作有以下几点:

1)使用轻量级代理重加密算法进行会话密钥加密传输,结合AES对称加密算法确保MQTT数据传输的端到端安全性;

2)将代理重加密密钥的生成工作由发布者端转移至可信中心,避免在发布者客户端设备上进行大量复杂计算和密钥存储;

3)采用Schnorr签名对发布消息进行数字签名,订阅者不需要存储发布者公钥也可验证消息来源真实性、完整性和不可否认性。

1　相关工作

在假设MQTT代理可信或经过身份认证后可信的前提下,许多研究通过在MQTT发布/订阅者客户端与MQTT代理间使用非对称密码体制协商会话密钥或安全传输会话密钥。然后使用会话密钥对传输数据进行对称加密来实现发布者与订阅者间的数据机密性。

Calabretta等[6]提出一种基于增强的口令认证密钥交换(Augmented Password-Authenticated Key Exchange, AugPAKE)协议的MQTT安全通信方案,客户端与代理间通过特定主题进行4次信息交互完成AugPAKE协议过程,实现客户端与代理间的相互认证,并协商出后续数据加密传输的会话密钥,使每个节点(无论是发布者还是订阅者)都与代理间维持一个会话密钥。文中明确表示,将代理视为信任的实体,负责解密和加密MQTT有效载荷,对交换数据具有完全的可见性。尽管作者建议数据由代理以加密格式存储,只有当代理收到授权(和验证)的订阅请求时才能对数据进行解密和重新加密。但攻击者仍然可以通过攻击代理或与代理共谋来窥探MQTT通信中传输的数据。另外,该方案中4次MQTT信息交互产生了较大的额外通信开销。

文献[7]中提出了一种基于MQTT协议的分级安全框架,使用Rabin加密密码系统[8]、椭圆曲线集成加密方案(Elliptic Curve Integrated Encryption Scheme, ECIES)[9]和带伽罗瓦消息验证码的计数器模式运行高级加密标准(Advanced Encryption Standard in Galois/Counter Mode, AES-GCM)加密方案为不同敏感程度的数据提供不同级别的安全性。所有加解密采用轻量级密码方案,并将密码方案嵌入正常的MQTT发布/订阅通信流中,无需使用额外的安全性开销。但该方案也是通过MQTT代理加密和解密发布/订阅者间传输的数据,同样存在代理是否可信的问题。

文献[10]中,作者试图通过将MQTT与通用异步接收器/发送器和Rime协议栈一起用作传输患者医疗数据的协议,创建一个安全的端到端物联网环境。使用具有256位密钥的AES-GCM实现端到端数据机密性,但是需要通过基于哈希消息认证码的密钥派生函数和椭圆曲线上的Diffie-Hellman(D-H)密钥交换算法管理系统中不同实体间的密钥。同时,路由需要解密含有多个传感器加密数据的聚合数据包,分别进行验证后使用预加载密钥重新聚合加密再发送给数据订阅者。

文献[11-12]中对代理的隐式信任不作强制要求,分别通过基于身份密码学和基于代理重加密实现端到端的数据安全性。文献[11]中提出一种基于身份密码术的会话密钥协商方案。用户与提供传感器数据的物联网网关间通过基于身份密码术的公钥加密进行相互认证,并在一个往返时间内协商出可用于后续通信加解密的会话密钥。该方案在提供端到端数据安全性的同时,还减少了使用基于公钥基础设施(Public Key Infrastructure, PKI)的公钥加密完成身份认证和会话密钥协商的方案中的证书存储开销。但基于身份密码术的公钥体制对于计算和存储能力要求较高,难以适用于资源受限的物联网设备。Kim等[12]提出了一种物联网环境下在轻量级设备上使用常规密码算法共享和管理数据的方法,该方案的实现结合了代理重加密体制[13-14]。代理重加密是一种典型的密文共享方案,通过代理服务器将一个用户的密文转换为另一个用户可以解密的密文,整个过程中不泄露用户的私钥和明文信息。文献[12]通过使用代理重加密减少数据共享过程中单个节点的加密计算负担,每个节点执行加密和创建重加密密钥,然后将加密密文和重加密密钥发送到代理服务器,由代理服务器生成允许其他节点解密的密文。在传统的点对点数据共享方式中,提供数据的节点需要与所有使用该数据的节点间分别进行一次加解密操作。因此,与传统数据共享方式相比该方案减少了数据共享过程中的加解密次数;但该方案使用的是基于椭圆曲线密码系统的代理重加密算法,算法中包含双线性映射。双线性映射的计算开销远大于椭圆曲线上的标量乘法[15],并不是轻量级密码算法的首选。此外,在该方案中每个节点除执行数据加密外还需要为每个共享数据的节点生成重加密密钥,操作时间将随传感器节点的数量成正比例增加。

因此,本文提出了一种基于代理重加密实现MQTT端到端安全性的解决方案。该方案在确保MQTT数据传输端到端安全性的同时,通过采取以下两点措施来适应资源受限的物联网环境:1)使用AES加密传感数据,只对AES加密使用的会话密钥进行代理重加密,大量减少非对称加解密操作;2)将传统代理重加密框架[13,16-17]中重加密密钥的生成操作由发布者转移到包含有密钥生成中心(Key Generation Center, KGC)的

可信中心，从而减少发布者端的计算量和密钥存储量。此外，本方案中采用Schnorr签名确保数据完整性、来源的真实性和不可否认性。

2 MQTT端到端安全解决方案

本章主要介绍为MQTT服务提出的基于代理重加密安全框架，以及使用的符号和加密方法、方案架构和具体流程的各个阶段。

2.1 使用的符号和密码学

表1中定义了所提方案中使用的符号。

根据效率和安全性，本文选择了以下密码学加密方案：

AES对称加密，以及基于改良的计算Diffie-Hellman（modified Computational Diffie-Hellman，mCDH）难题的代理重加密算法和Schnorr签名算法[13]。

AES用于数据资源加解密，代理重加密算法保护AES加密密钥，Schnorr算法对信息进行数字签名防篡改。

表1 所提方案使用的符号

Tab. 1 Notations used in the proposed scheme

符号	定义	符号	定义	符号	定义
(D, v)	签名信息	M	明文	Sig	签名
$\|\|$	报文拼接	m	传输消息	SK	会话密钥
B	代理	P	发布者	SK_i	i的私钥
C_m, C_p, C_s	密文	PK_i	i的公钥	SK_{i_sig}	i的签名私钥
Dec	非对称解密	PK_{i_sig}	i验证签名公钥	T	主题
Dec_AES	AES解密	ReEnc	代理重加密	t	时间戳
Enc	非对称加密	ReKeyGen	重加密密钥生成	TC	可信中心
Enc_AES	AES加密	RK	重加密密钥	$UUID_i$	i的通用唯一识别码
KDF	密钥派生函数	$RK_{p\to s}$	P到S的重加密密钥	Ver	签名验证
KeyGen	密钥生成函数	S	订阅者	$\xleftarrow{R}$	从中随机选择

2.2 方案概述

方案框架如图1所示，模型由4个实体组成：发布者（Publisher，P）、代理（Broker，B）、订阅者（Subscriber，S）和可信中心（Trusted Center，TC）。

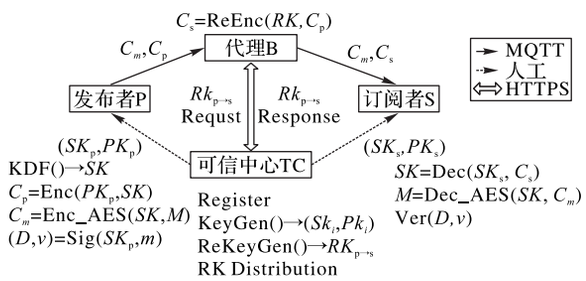

图1 MQTT端到端安全解决方案

Fig. 1 Solution for MQTT end-to-end security

可信中心的设立与放弃信任MQTT代理的做法并不矛盾。MQTT代理需要与众多的发布/订阅者客户端设备进行信息交互，资源受限客户端设备的低安全防护级别扩大了MQTT代理的攻击面。在这种情况下，选择信任MQTT代理并对其进行安全加固的成本是极高的，并且MQTT代理上需要客户端设备完全响应的强安全约束条件也是客户端设备难以承受的。本文设立的可信中心只是扩展了传统无证书代理重加密系统模型[18]中KGC实体（如图2）的功能，而可信中心只响应代理的重加密密钥请求。因此在代理重加密系统模型中建立可信中心比安全加固代理的实用性要高。

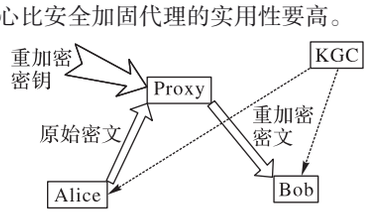

图2 无证书代理重加密

Fig. 2 Certificateless proxy re-encryption

发布者可以感知数据并生成按主题分类的消息。所有发布者将消息发送给代理，代理将它们转发给最终用户（订阅者）。下面分别介绍各实体的组成与功能。

2.2.1 发布者

发布者通过密钥派生函数（Key Derivation Function，KDF）生成会话密钥，并利用会话密钥对感知数据进行AES加密；同时，使用自己的公钥对会话密钥进行加密。最后，使用签名密钥签名需要发送的消息。

2.2.2 代理

代理组成如图3所示。MQTT核心模块实现与发布/订阅者在MQTT协议标准下通信；安全套接字层超文本传输协议（Hyper Text Transfer Protocol over Secure socket layer，HTTPS）通信模块实现代理与可信中心间安全的超文本传输协议通信；重加密模块实现发布者会话密钥密文的重加密。最后，将重加密密文和感知数据密文重新封装到有效载荷，发送给数据的订阅者。

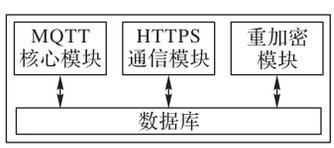

图3 代理结构

Fig. 3 Architecture of broker

2.2.3 订阅者

订阅者使用主题向代理订阅数据，收到代理发送的消息后，订阅者首先解密经代理重加密的密文获取会话密钥，使用该会话密钥解密有效载荷得到明文信息。最后进行消息签名验证，验证通过则接受消息，否则丢弃。

2.2.4 可信中心

可信中心结构如图4所示，主要包含注册管理中心、密钥生成中心、重加密密钥生成器和数据库。发布/订阅者设备由设备所有者通过客户端向可信中心注册，注册成功后由可信中心的KGC为每个设备生成公私钥对和签名密钥对。可信中心根据设备注册的属性（发布/订阅）和主题进行匹配，为每

对的发布者–订阅者对等方预生成重加密密钥,并将生成的重加密密钥通过安全的方式发送给代理。由于本文主要分析发布者与订阅者端到端的数据安全,对于设备的认证授权与访问控制方案不进行讨论。

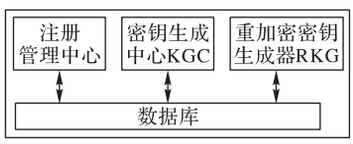

图4　可信中心结构

Fig. 4　Architecture of trusted center

2.3　具体流程

2.3.1　系统初始化与设备注册

系统初始化与设备注册流程如图5所示。

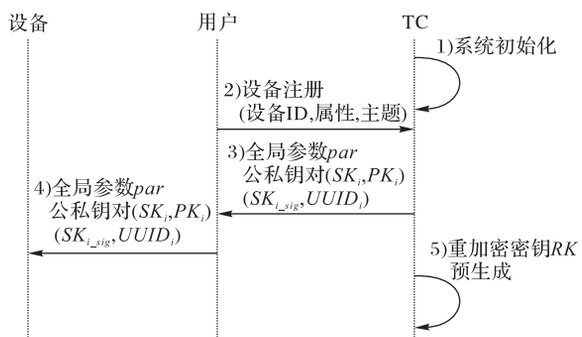

图5　系统初始化与设备注册流程

Fig. 5　Process of system initialization and device registration

1)系统初始化:GlobalSetup(1^k) → (par)。

全局设置算法GlobalSetup()将安全参数k作为输入。它输出全局参数par:(q, G, g, H_1, H_2),其中素数q,阶数为q的有限循环群G,g是G的生成元,哈希函数H_1:$\{0,1\}^* \rightarrow \mathbf{Z}_q^*$,哈希函数$H_2$:$G \rightarrow \mathbf{Z}_q^*$。

2)用户注册设备。

用户通过客户端向TC注册设备,注册内容包括:设备标识(如物理地址)、设备属性(发布/订阅)、设备主题(设备在MQTT通信中发布/订阅该主题)。

3)公私钥对生成:KeyGen(i) → (sk_i, pk_i)。

注册成功后,TC使用密钥生成算法KeyGen()为设备i生成公私钥对(sk_i, pk_i)。密钥生成算法选择$x_i \xleftarrow{R} \mathbf{Z}_q^*$,并设$sk_i = x_i$,$pk_i = g^{x_i}$。

发布/订阅者设备的密钥生成如下:

随机选择$x_{i1}, x_{i2}, x_{i3} \xleftarrow{R} \mathbf{Z}_q^*$,设$sk_{i1} = x_{i1}$,$pk_{i1} = g^{x_{i1}}$;$sk_{i2} = x_{i2}$,$pk_{i2} = g^{x_{i2}}$;$sk_{i3} = x_{i3}$,$pk_{i3} = g^{x_{i3}}$。设置$PK_i = (pk_{i1}, pk_{i2})$,$SK_i = (sk_{i1}, sk_{i2})$;$SK_{i\_sig} = sk_{i3}$,$PK_{i\_sig} = pk_{i3}$。生成设备的公私钥对($SK_i$, PK_i),签名密钥对(SK_{i_sig}, PK_{i_sig})。由TC统一生成设备通用唯一识别码(Universally Unique Identifier, UUID),并设置$UUID_i = PK_{i_sig}$。

此过程后,发布者获得公私钥对(SK_p, PK_p)、签名密钥对(SK_{p_sig}, $UUID_p$),订阅者获得公私钥对(SK_s, PK_s)、签名密钥对(SK_{s_sig}, $UUID_s$);

4)安全参数返回。

注册成功后,用户必须将可信中心生成的系统参数、设备UUID、设备签名密钥和设备公私钥对通过安全通道(如手工嵌入设备安全固件、使用设备可信平台模块安全存储等)加载到设备中。

5)重加密密钥预生成:ReKeyGen(SK_i, PK_i, PK_j) → $RK_{i \rightarrow j}$。

算法按以下步骤生成重加密密钥:

① $w_1, w_2 \xleftarrow{R} \mathbf{Z}_q^*$;$w_3 \in \mathbf{Z}_q^*$;

② $Z = pk_{s2}^{w1}$;

③ $w_3 = sk_{p1} - sk_{p2} \cdot w_2 \bmod q$;

④ $rk_1 = w_3$, $rk_2 = H_2(Z^{sk_{s1}}) \cdot w_2 \bmod q$;

⑤ $rk_3 = pk_{p1}^{w1}$;

返回重加密密钥$RK_{p \rightarrow s} = (rk_1, rk_2, rk_3)$。

2.3.2　数据传输

数据传输流程如图6所示。

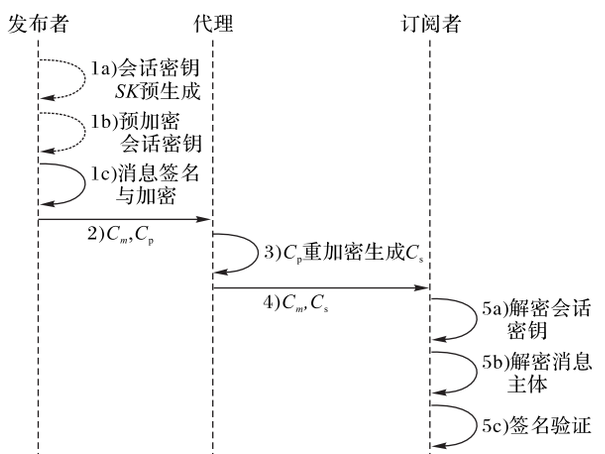

图6　数据传输流程

Fig. 6　Data transmission process

1)发布信息生成。

1a)会话密钥预生成:KDF() → SK。

发布者使用KDF生成会话密钥SK。该算法可以使用计时器触发,也可以使用计数器触发。计时器触发方式中,根据设置的会话密钥生存周期进行定期生成;计数器触发方式中,根据限制会话密钥加密次数进行密钥生成。对于最敏感的数据可以实行最高安全级别的防护,将会话密钥设置为一次完整的数据发送一更换。

1b)预加密会话密钥:Enc(PK_p, SK) → C_p。

① $r \xleftarrow{R} \mathbf{Z}_q^*$;

② $C_{p1} = g^r$, $C_{p2} = pk_{p2}^r$, $C_{p3} = pk_{p1}^r \cdot SK$;

返回密文$C_p = (C_{p1}, C_{p2}, C_{p3})$。

1c)消息签名与加密。

消息主体为$\{T, UUID, M, t, sig\}$。其中T为发布数据对应的主题;$UUID$是设备在TC注册后获取的统一标识,同时也充当用于验证设备签名信息的公钥;M代表可用于承载发布数据资源的主体信息;t为时间戳;sig的生成方法如下:

① $u \xleftarrow{R} \mathbf{Z}_q^*$;$D = g^u$;

② $e = H_1(T, UUID_p, M, t, D)$;

③ $v = (u + SK_{p\_sig} \cdot e) \bmod q$;

返回签名$sig = (D, v)$。

消息加密:C_m = Enc_AES(SK,(T, $UUID_p$, M, t, sig))。消息加密采用AES对称加密,加密密钥是发布者预生成的会话密钥SK。

2)发布消息封装与发送。

发布者端消息处理完毕,生成有效载荷 $C_p \| C_m$,并将该发布消息发送到代理。

3)代理重加密:$\mathrm{ReEnc}(RK_{p\to s}, C_p) \to C_s$。

在发布/订阅者与代理建立连接阶段,使用设备UUID作为客户端标识。代理收到发布消息可根据发布者 $UUID_p$ 和相应主题的订阅者 $UUID_s$ 进行重加密密钥 $RK_{p\to s}$ 的查找或请求。

代理对 C_p 重加密的计算方法如下:

$C_{s1} = C_{p1}^{rk1}, C_{s2} = C_{p2}, C_{s3} = C_{p3}, C_{s4} = rk_2, C_{s5} = rk_3$;

输出密文:$C_s = (C_{s1}, C_{s2}, C_{s3}, C_{s4}, C_{s5})$。

4)代理转发发布消息。

重加密密文生成后,代理对消息有效载荷 $C_s \| C_m$ 进行封装并发送给订阅者。

5)订阅信息解析。

订阅者按如下步骤解析信息:

5a)解密会话密钥:$\mathrm{Dec}(SK_s, C_s) \to SK$

① $w_2 = C_{s4} \big/ H_2(C_{s5}^{sk_{s2}})$

② $SK = C_{s3} \big/ (C_{s1} \cdot C_{s3}^{w_2})$

算法正确性证明如下:

$$① w_2 = \frac{C_{s4}}{H_2(C_{s5}^{sk_{s2}})} = \frac{rk_2}{H_2(rk_3^{sk_{s2}})} = \frac{H_2(Z^{sk_{s1}}) \cdot w_2 \bmod q}{H_2(pk_{p1}^{w_1 \cdot sk_{s2}})} = \frac{H_2(pk_{s2}^{w_1 \cdot sk_{s1}}) \cdot w_2 \bmod q}{H_2(pk_{p1}^{w_1 \cdot sk_{s2}})} = \frac{H_2(g^{sk_{s2} \cdot w_1 \cdot sk_{s1}}) \cdot w_2 \bmod q}{H_2(g^{sk_{s1} \cdot w_1 \cdot sk_{s2}})}$$

$$② SK = \frac{C_{s3}}{C_{s1} \cdot C_{s2}^{w_2}} = \frac{C_{p3}}{C_{p1}^{rk_1} \cdot C_{p2}^{w_2}} = \frac{pk_{p1}^{r} \cdot SK}{g^{r \cdot rk_1} \cdot pk_{p2}^{r \cdot w_2}} = \frac{pk_{p1}^{r} \cdot SK}{g^{r \cdot w_3} \cdot g^{sk_{p2} \cdot r \cdot w_2}} = \frac{g^{r \cdot sk_{p1}} \cdot SK}{g^{r \cdot (w_3 + sk_{p2} \cdot w_2)}}$$

5b)消息解密:$\mathrm{Dec\_AES}(SK, C_m) \to (T, UUID_p, M, t, sig)$。

订阅者使用解密代理重加密密文 C_p 后获得的 SK 对 C_m 进行AES解密,即可得到消息主体 $(T, UUID_p, M, t, sig)$。

5c)签名验证。

解密出主体消息后,订阅者首先核验时间戳 t。如果时间戳核验通过,订阅者继续对消息主体的签名 $(sig = (D, v))$ 进行验证,签名验证按如下步骤进行:

① 计算:$H_1(T, UUID_p, M, t, D)$;

② 验证:$g^v = D \cdot UUID_p^{H_1(T, UUID_p, M, t, D)}$。

验证通过的情况下,订阅者接受消息主体中的数据资源 M,否则消息被拒绝。

3 性能评估与安全讨论

3.1 性能评估

分别与文献[6]中提出的基于AugPAKE协商会话密钥的方案、文献[7]中提出的基于轻量级非对称加密协商会话密钥的方案、文献[11]中提出的基于身份密码学非对称加密协商会话密钥的方案和文献[12]中提出的基于属性代理重加密数据资源的方案进行对比分析和性能评估。表2~4提供了本文方案与MQTT安全性相关的其他几个方案间的对比,比较的内容包括:性能、计算成本和通信开销。

性能表现如表2所示。性能表现对比点的选取是根据较为严格的零信任安全理念确定的。MQTT端到端的加密要求数据从发布者端到订阅者端全程处于密文状态,不允许除发布/订阅者以外的第三方获得明文。半诚实代理要求代理严格按照协议规范进行报文的处理,而不依赖于代理保守秘密。身份认证要求从消息中的某些信息能够确认发布者或订阅者的身份,但充当确认角色的可是发布/订阅者本身、代理或是其他第三方。空间解耦特性是MQTT通信协议的优势,指发布者与订阅者不需要互相了解。在部分安全性解决方案中(如文献[11]方案),为加强安全性发布/订阅者必须持有关联对方身份的秘密,从而破坏了MQTT协议本身的优势特性。

表2 不同方案性能表现对比

Tab. 2 Performance comparison of different schemes

方案	端到端加密	半诚实代理	身份认证		空间解耦
			发布者认证	订阅者认证	
文献[6]方案	×	×	√	√	√
文献[7]方案	×	×	√	√	√
文献[11]方案	√	√	√	√	×
文献[12]方案	√	√	×	√	√
本文方案	√	√	√	×	√

本文主要关注数据传输过程的安全性问题,没有对发布/订阅者与代理建立连接的过程进行约束,所以方案缺乏对订阅者认证的安全属性。这一问题将在建立MQTT认证授权机制的未来工作中进行补充。

计算开销对比如表3所示。讨论中省略相对较快的操作,如随机数生成、异或等。

表3 不同方案计算开销对比

Tab. 3 Computation overhead comparison of different schemes

方案	发布者	代理	订阅者
文献[6]方案	2SM+2M+1A+4H+1MAC+1AES	6SM+2M+8H+4MAC+4AES	2SM+2M+1A+4H+1MAC+1AES
文献[7]方案	3SM+1M+1A+1H+1AES	4SM+1A+1H+3AES	5SM+1M+2A+2H+2AES
文献[11]方案	4IBC+1AES	0	4IBC+1AES
文献[12]方案	2Bm+2SM+5E+1H	1Bm+1SM+1E	1Bm+1M+1E+1H
本文方案	4E+2M+1A+1H+1AES	1E	4E+4M+2H+1AES
增强的本文方案	4E+2M+1A+1H+1AES	3E+1M+1H	5E+5M+1A+2H+1AES

注:Bm表示双线性映射,SM表示标量乘法,E表示求(模)幂运算,M表示模乘法/除法,A表示模加/减法,H表示哈希函数,MAC表示消息身份验证码运算,AES表示AES加密/解密,IBC表示基于身份密码学的一次非对称加密/解密。

为了更加合理地对比各方案,首先对本文的解决方案进行如下补充说明:

1)发布者端密钥派生函数生成会话密钥过程的计算开销处理。密钥派生函数生成会话密钥过程比较简单,比如输入一个随机数和一个计数器当前值,然后进行一次哈希运算,即得到一个密钥。在与文献[7]方案对比时,两方案均使用密钥

派生函数生成会话密钥,同时不讨论密钥派生函数预生成会话密钥的过程。在与文献[6]、[11]、[12]中方案进行对比时,将密钥派生函数生成会话密钥的计算开销等同于一次哈希运算。

2)订阅者身份认证计算量的补充。由于本文方案没有讨论订阅者身份认证,为公平对比,在订阅者端增加一次本文所采用签名算法的签名计算开销(1E+1M+1A+1H),在代理处增加一次签名验证计算开销(2E+1M+1H)。表3中增强的本文方案的计算开销表示功能补充后的总开销,本文方案的计算开销表示解决方案原本的总开销。

在提供端到端加密方面,主要与文献[11-12]中的方案进行比较。本文的解决方案总共执行:6个模乘法、9个求幂运算、1个模加法、3个哈希函数和2个AES运算。文献[11]方案需要:8个基于身份密码学的非对称加解密运算和2个AES运算。该方案本身优势是实现端到端的数据安全,并非为受限环境定制,因此8个基于身份密码学的非对称加解密运算的计算开销是受限环境中设备无法承担的。文献[12]方案需要:3个标量乘法、1个模乘法、7个求幂运算、4个双线性映射运算,其中双线性映射的计算量远远大于标量乘的计算量。因此,该方案虽然应用于物联网环境,但计算开销比本文方案要大。

与其他不提供端到端加密的文献[6]方案、文献[7]方案相比。文献[6]方案需要:10个标量乘法、6个模乘法、2个模加法、16个哈希函数、6个AES运算、6个MAC操作。文献[7]方案需要:12个标量乘法、2个模乘法、4个模加法、4个哈希函数和6个AES操作。与这些方案相比,本文方案在计算开销方面的优势不突出,但本文方案可以提供端到端的数据机密性,防止出现不诚实代理或受攻击代理泄露数据的问题。

通信开销的对比如表4所示。文献[6-7,11]中的方案中发布/订阅者与代理间的密钥协商分别需要5、1、1条消息完成,本文方案与文献[12]方案不需要额外的协商消息。

表4　不同方案通信开销对比

Tab. 4　Communication overhead comparison of different schemes

方案	发布者—代理	订阅者—代理
文献[6]方案	5+N	5+N
文献[7]方案	1+N	1+N
文献[11]方案	1+N	1+N
文献[12]方案	N	N
本文方案	N	N

注:N表示发送传感数据的消息数。

通过比较可以发现,本文的端到端加密主要优势在于不需要发布/订阅者与代理之间的密钥协商操作以及代理解密数据再加密数据的重复操作。此外,本文方案使用代理重加密算法并将验证签名的公钥定义为设备UUID随数据消息一起传输,发布/订阅者只需存储各自的公私钥对和签名生成与验证所使用的密钥对,无需存储对等方的公钥。

本文的解决方案在取得计算开销、通信开销以及密钥存储开销方面优势时也有一定的代价。与文献[7]方案相似,本文提出的解决方案将其他一些信息(主题、UUID、时间戳、签名、会话密钥密文)封装在MQTT报文的有效载荷中,压缩了数据资源的空间。但主题、UUID、时间戳、签名四部分信息的位数仅占有效载荷最大总位数的1/4,而且会话密钥密文仅在文件传输的首个数据包中发送。因此,本文提出的安全方案不影响协议正常使用。

本文中的解决方案将代理重加密过程中最复杂的重加密密钥生成过程转移到可信中心,使得发布/订阅者的计算量保持在与文献[6]方案、文献[7]方案相当的水平。同时可信中心生成重加密密钥的过程是伴随设备陆续注册进行的预处理,因此可信中心的重加密密钥生成不会成为本文方案的技术瓶颈。

3.2　算法安全性证明与协议安全性讨论

3.2.1　算法安全性证明

1)困难问题假设。

本文提出的方案的安全性基于决策性Diffie-Hellman(Decisional Diffie-Hellman, DDH)难解问题。已知g^x, g^y, $g^z\in G$,其中$x, y, z\in \mathbf{Z}_q^*$不可知,$G$为阶取$q$的循环群。判断输出与$z=xy \bmod q$是否一致,即DDH难解问题。DDH假设表示在多项式时间t内敌手A能够以如下概率解决DDH问题:

$$\mathrm{Succ}_G^{\mathrm{DDH}}(\mathrm{A})=\Pr[\mathrm{A}(g^x,g^y,g^z),z=xy \bmod q]\leqslant \varepsilon$$

其中ε是可以忽略的,则称DDH问题是(t,ε)难解的。

2)安全模型。

定义　如果敌手A在多项式时间内以可忽略的优势赢得游戏,那么提出的代理重加密算法满足选择密文攻击(Chosen Ciphertext Attack, CCA)安全性。

下面给出挑战者C和敌手A间的3个阶段组成的选择密文攻击游戏。

初始阶段　C运行KeyGen算法生成公私钥对,C将公钥发送给A,并将(SK_i, PK_i)记录在列表T_{PK}中。

阶段1　A向C发出以下系列的预言机询问:O_{SK}、O_{RK}、O_{RE}、O_{DEC},C向A返回查询结果。

私钥生成预言机O_{SK}:A输入$PK_i=(pk_{i1}, pk_{i2})$,C检索列表T_{PK}查找PK_i并将其对应的私钥$SK_i=(sk_{i1}, sk_{i2})$返回给A。

重加密密钥生成预言机O_{RK}:A输入(PK_i, PK_j),其中PK_i和PK_j都由C检索返回。C返回$RK_{i\to j}=(rk_1, rk_2, rk_3)$给A。

代理重加密预言机O_{RE}:A输入$(PK_i, PK_j, C_{\mathrm{p}})$,C返回重加密密文$\mathrm{ReEnc}(C_{\mathrm{p}}, O_{\mathrm{RK}}(PK_i, PK_j))$。

解密查询预言机O_{DEC}:A输入PK和密文C,C查询PK对应的私钥SK,执行解密算法恢复明文M并返回M。

挑战阶段　一旦A结束阶段1,将从明文空间中选择两个等长的明文消息M_0、M_1和一个公钥PK^*向C发起挑战。C选择随机比特$d\in\{0,1\}$并计算密文$C^*=\mathrm{Encrypt}(PK^*, M_d)$返回给A。

阶段2　敌手A在额外的条件下继续发出阶段1的查询。

猜测阶段　最终敌手A返回给C一个猜测结果$d'\in\{0,1\}$,若$d'=d$,则A在游戏中获胜。

3)安全性证明。

定理　如果DDH假设成立,本文提出的代理重加密方案在随机预言机模型中安全于适应性选择密文攻击(Adaptive Chosen Ciphertext Attack, CCA2)。

证明　如果存在敌手A可以在多项式时间内以不可忽略的优势破坏CCA2安全性,那么敌手A可以构建算法来解决DDH难题。即输入$\langle G=\langle g\rangle, g^a, g^b, T\rangle$,由构建的算法确定$T=g^{ab}$是否成立。挑战者C构建以下随机预言机模型。

O_{H1}:C检查列表H_{1_list}中是否已经存在(D,α)。如果存在,C将α返回给A;如果不存在,随机选择$\alpha\in\mathbf{Z}_q^*$,将(D,α)记录

到H_{1_list}，同时返回$\alpha = H_1(D)$。

O_{H2}：C检查列表H_{2_list}中是否已经存在(D, β)。如果存在，C将β返回给A；如果不存在，选择$\beta \leftarrow \mathbf{Z}_q^*$，将$(D, \beta)$记录到$H_{2_list}$，同时返回$\beta = H_2(D)$。

C记录两个列表：K_{list}为存储公私钥对的列表；RK_{list}为存储重加密密钥的列表。

阶段1　A进行系列询问：O_{PK}、O_{SK}、O_{RK}、O_{RE}、O_{DEC}，C向A返回查询结果。

公钥生成预言机O_{PK}：输入公钥$PK_i = (pk_{i1}, pk_{i2})$，预言机设置$(pk_{i1}, pk_{i2}) = \left((g^a)^{sk_{i1}}, (g^a)^{sk_{i2}}\right)$，其中$sk_{i1}$、$sk_{i2}$是$\mathbf{Z}_q^*$中的随机数，并将$(pk_{i1}, pk_{i1}, sk_{i1}, sk_{i2})$加入列表$K_{\text{list}}$。

私钥生成预言机O_{SK}：输入公钥$PK_i = (pk_{i1}, pk_{i2})$，C在列表$K_{list}$中检索$(pk_{i1}, pk_{i1}, sk_{i1}, sk_{i2})$，并返回给A相应的私钥$SK_i = (sk_{i1}, sk_{i2})$。

重加密密钥生成预言机O_{RK}：输入PK_i、PK_j（二者均来自O_{PK}），C从K_{list}检索相应的公钥获取私钥$SK_i = (sk_{i1}, sk_{i2})$，其中$w_3 = sk_{i1} - sk_{i2} \cdot w_2 \bmod q$。然后运行重加密密钥生成算法获取重加密密钥，并返回$RK_{i \to j} = (rk_1, rk_2, rk_3)$给C。

代理重加密预言机O_{RE}：A输入$(PK_i, PK_j, C_{\text{p}})$，C返回重加密密文$\text{ReEnc}(C_{\text{p}}, O_{\text{RK}}(PK_i, PK_j))$。

解密查询预言机O_{DEC}：输入(PK_i, C_s)，C检索相应的私钥SK_j，并返回$\text{Dec}(\text{ReEnc}(C_{\text{p}}, O_{\text{RK}}(PK_i, PK_j)), sk_j)$。

挑战阶段　完成上述询问后，A向C发送PK_i'以及两条等长消息M_0、$M_1 \in \{0, 1\}$。算法A从列表K_{list}中恢复出PK_i，SK_i和密文。C选择随机比特$d \in \{0, 1\}$并按如下步骤生成挑战密文：

①$C_{\text{s1}} = g^{rw_3} = g^{bw_3}$；

②$C_{\text{s2}} = pk_{\text{p2}}^r = g^{absk_{\text{p2}}} = pk_{\text{p2}}^b$；

③$C_{\text{s3}} = pk_{\text{p1}}^r \cdot m_d = g^{absk_{\text{p2}}} \cdot m_d = pk_{p1}^b \cdot m_d$；

④$C_{\text{s4}} = H_2((pk_{\text{s2}}^{w1})^{sk_{\text{p1}}}) \cdot w_2 \bmod q = H_2(((g^a)^{sk_{s2} \cdot w1})^{sk_{\text{p1}}}) \cdot w_2 \bmod q$；

⑤$C_{\text{s5}} = pk_{\text{p1}}^{w1} = g^{aw1}$。

阶段2　敌手A在以下条件下重复询问。

①询问$O_{RK}(SK_i^*, PK_j)$；

②如果敌手A发布$\text{ReEnc}(C_{\text{p}}, RK) \to C_s$，其中$RK$由公式$\text{ReKeyGen}(par, SK_i, PK_j) \to RK$；

③询问$O_{\text{DE}}(C_s, SK_j)$。

猜测阶段　最终，A返回给C一个猜测值$d' \in \{0, 1\}$。当$T = g^{ab}$时，$C^* = (C_{\text{s1}}^*, C_{\text{s2}}^*, C_{\text{s3}}^*, C_{\text{s4}}^*, C_{\text{s5}}^*)$与$\text{ReEnc}(C_{\text{p}}, RK) \to C_{\text{s}}$相同。$T = g^{ab}$被随机数$T \in G$替换。因此敌手A无法以高于1/2的概率猜测到挑战密文中的值d。

3.2.2　协议安全性讨论

本文提出的解决方案基于安全的代理重加密算法，并支持的安全属性：

1）数据端到端的机密性。

数据资源与会话密钥在通信全程都是以密文的形式进行传输，代理也无法解密数据，只有发布者和对应的订阅者可以解密密文获得明文信息。

2）数据真实性、完整性和不可否认性。

方案通过Schnorr签名方案确保此属性。只有拥有签名密钥的发布者才能对报文进行签名。

3）重放攻击保护。

发布消息均包含时间戳，可以防止攻击者重放这些消息。

4）中间人攻击防护。

发布/订阅者通过用户向TC注册获取密钥信息，除发布者用来验证签名的公钥以设备UUID的形式会公开给代理外，其他的密钥信息即使是公钥也不对外公开，因此中间人无法窥探到以密文形式传输的报文中任何秘密信息；而且本文中的方案不涉及发布者与订阅者之间或发布/订阅者与代理之间的密钥协商过程，因此中间人无法实施攻击。危害最大的是一个合法的订阅者充当中间人角色，并且该中间人与代理合谋。代理将传输给目标订阅者的报文重加密后发送给合谋的中间人。但重加密密钥是与真实订阅者密钥对中的公钥相关联的，即使攻击者获得密文也会因缺乏对应的私钥而无法解密。

5）隐私保护。

本文的解决方案中，唯一代表发布/订阅者身份的公开信息是UUID，但该UUID真正代表的设备和用户的身份只有TC知道。发布/订阅者相互间不知道对方身份信息，代理也不了解发布/订阅者的真实身份。

6）发布者认证。

发布者在发布信息中加入签名，订阅者只要验证签名就可以确定UUID的真实性。在有需要的情况下（如安全事件发生），订阅者可以将UUID提交给TC就可以认证发布者的真实身份。

4　结语

本文提出了一种将代理重加密算法用于MQTT通信的安全解决方案，为MQTT通信模型提供端到端的数据安全性。方案降低了对代理的信任度要求，可以很好适应基于云服务的应用环境。所提解决方案使用有效的代理重加密方案并进行修改，以适应资源受限的物联网环境。通过与该领域的相关工作进行比较，证明了所提解决方案的安全消息开销很小。

下一步工作主要填补解决方案中对客户端设备接入认证和发布/订阅请求授权的空白，结合代理重加密密钥的生成，制定与本方案契合的认证授权机制。

参考文献（References）

[1] 物联网安全创新联合实验室. 物联网终端安全白皮书(2019)[EB/OL]. [2019-12-25]. http://www.caict.ac.cn/kxyj/qwfb/bps/201911/P020191115523217021278.pdf.（IoT Security Lab. IoT device security white paper(2019)[EB/OL]. [2019-12-25]. http://www.caict.ac.cn/kxyj/qwfb/bps/201911/P020191115523217021278.pdf.）

[2] BRYCE R, SHAW T, SRIVASTAVA G. MQTT-G: a publish/subscribe protocol with geolocation[C]// Proceedings of the 41st International Conference on Telecommunications and Signal Processing. Piscataway: IEEE, 2018: 1-4.

[3] GILMAN E, BARTH D. Zero Trust Networks: Building Secure Systems in Untrusted Networks[M]. Sebastopol, CA: O'Reilly Media, 2017: 1-2.

[4] ROSE S, BORCHERT O, MITCHELL S, et al. Zero Trust Architecture (2nd Draft): NIST SP 800-207[S]. Washington, DC: National Institute of Standards and Technology, 2020-02-13.

[5] SCHNORR C P. Efficient identification and signatures for smart cards[C]// Proceedings of the 1989 Conference on Theory and Application of Cryptology, LNCS 435. New York: Springer, 1989:

239-252.

[6] CALABRETTA M, PECORI R, VELTRI L. A token-based protocol for securing MQTT communications[C]// Proceeding of the 26th International Conference on Software, Telecommunications and Computer Networks. Piscataway: IEEE, 2018: 1-6.

[7] MALINA L, SRIVASTAVA G, DZURENDA P, et al. A secure publish/subscribe protocol for Internet of things[C]// Proceedings of the 14th International Conference on Availability, Reliability and Security. New York: ACM, 2019: No. 75.

[8] RABIN M O. Digitalized signatures and public-key functions as intractable as factorization[R]. Cambridge: Massachusetts Institute of Technology, 1979.

[9] SHOUP V. A proposal for an ISO standard for public key encryption (version 2.1)[EB/OL]. [2020-05-20]. https://eprint. iacr. org/2001/112. pdf.

[10] MATHUR A, NEWE T, ELGENAIDI W, et al. A secure end-to-end IoT solution[J]. Sensors and Actuators A: Physical, 2017, 263: 291-299.

[11] PENG W, LIU S, PENG K, et al. A secure publish/subscribe protocol for Internet of Things using identity-based cryptography[C]// Proceedings of the 5th International Conference on Computer Science and Network Technology. Piscataway: IEEE, 2016: 628-634.

[12] KIM S, LEE I. IoT device security based on proxy re-encryption[J]. Journal of Ambient Intelligence and Humanized Computing, 2018, 9(4): 1267-1273.

[13] DENG R H, WENG J, LIU S, et al. Chosen-ciphertext secure proxy re-encryption without pairings[C]// Proceedings of the 7th International Conference on Cryptology and Network Security, LNCS 5339. Berlin: Springer, 2008: 1-17.

[14] BLAZE M, BLEUMER G, STRAUSS M. Divertible protocols and atomic proxy cryptography[C]// Proceedings of the 1998 International Conference on the Theory and Applications of Cryptographic Techniques, LNCS 1403. Berlin: Springer, 1998: 127-144.

[15] 徐鹏,崔国华,雷凤宇. 非双线性映射下一种实用的和可证明安全的IBE方案[J]. 计算机研究与发展, 2008, 45(10):1687-1695. (XU P, CUI G H, LEI F Y. An efficient and provably secure IBE scheme without bilinear map[J]. Journal of Computer Research and Development, 2008, 45(10):1687-1695.)

[16] DO J M, SONG Y J, PARK N. Attribute based proxy re-encryption for data confidentiality in cloud computing environments[C]// Proceedings of the 1st ACIS/JNU International Conference on Computers, Networks, Systems and Industrial Engineering. Piscataway: IEEE, 2011: 248-251.

[17] NUÑEZ D, AGUDO I, LOPEZ J. Proxy re-encryption: analysis of constructions and its application to secure access delegation[J]. Journal of Network and Computer Applications, 2017, 87: 193-209.

[18] 吴世坤. 代理重加密体制研究及其应用[D]. 成都:电子科技大学, 2016: 28-29. (WU S K. Research on proxy re-encryption cryptography and its application[D]. Chengdu: University of Electronic Science and Technology of China, 2016: 28-29.)

This work is partially supported by the Foundation of Science and Technology on Information Assurance Laboratory (614211203010417).

GU Zhengchuan, born in 1989, M. S. candidate. His research interests include Internet of Things security, zero trust network.

GUO Yuanbo, born in 1975, Ph. D., professor. His research interests include network security, situation awareness.

FANG Chen, born in 1993, Ph. D. candidate. His research interests include artificial intelligence security, privacy protection.