计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2022,31(12):169−177 [doi: 10.15888/j.cnki.csa.008852]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

MQTT-SE 数据加密传输算法①

晁喜斌, 郭　锋, 武传坤

(临沂大学 信息科学与工程学院, 临沂 276000)
通信作者: 郭　锋, E-mail: guofeng_g@lyu.edu.cn

摘　要: 随着日新月异的高新技术不断发展, 物联网、大数据、人工智能交叉融合, 深度关联. 物联网全面融入了我们的生活、工作、社会发展等方方面面. 而物联网目前最广泛、最主流的协议当属 MQTT 协议, 低开销低带宽的先天优势促成了海量物联网设备接入网络. 但在万物互联时代大背景下, “自由可控, 安全可信”是行业发展的理念和标准. 目前很多研究者提出了从 MQTT 出发设计安全算法的方案, 但发现“基于 MQTT 的数据加密传输算法”该论文的核心算法存在密钥泄露的风险, 为此指出了其核心算法的缺陷并提出 3 种新的 MQTT-SE 算法. 分别是基于对称加密的 MQTT-SE 算法、基于公钥的 MQTT-SE 算法、基于公钥证书的双向认证 MQTT-SE 算法. 从而达到 MQTT 传输在低效能环境下的基础上达到高性能安全加密传输的目的.

关键词: 物联网 (IoT); 数据加密; MQTT; 公钥证书; 低效能; 信息安全; 隐私保护

引用格式: 晁喜斌,郭锋,武传坤.MQTT-SE 数据加密传输算法.计算机系统应用,2022,31(12):169–177. http://www.c-s-a.org.cn/1003-3254/8852.html

MQTT-SE Algorithm for Data Encryption Transmission

CHAO Xi-Bin, GUO Feng, WU Chuan-Kun

(School of Information Science and Engineering, Linyi University, Linyi 276000, China)

Abstract: With the rapid development of high-tech with each passing day, the cross fusion and deep correlation among the Internet of Things, big data, and artificial intelligence are implemented. The Internet of Things is fully integrated into all aspects of our life and work as well as social development. At present, the most widely used and mainstream protocol of the Internet of Things is the message queuing telemetry transport (MQTT) protocol, whose inherent advantages of low overhead and low bandwidth have contributed to the access of a large number of Internet of Things devices to the network. However, in the era of the Internet of Everything, “freedom, controllability, safety, and credibility” are the concepts and criteria of industrial development. Many researchers have proposed MQTT-based design schemes for security algorithms. Regarding the paper titled “Data encryption transmission algorithm Based on MQTT”, however, its core algorithm is found to be at risk of key leakage. Therefore, this study points out the defects of this core algorithm and proposes three MQTT-SE algorithms respectively based on symmetric encryption, public key, and mutual verification of public key certificates. These algorithms can achieve the purpose of high-performance and safe encryption transmission even in a low performance MQTT transmission environment.

Key words: Internet of Things (IoT); data encryption; message queuing telemetry transport (MQTT); public key certificate; low performance; information security; privacy protection

1　引言

在物联网飞速发展的时代背景下, 物联网相关应用覆盖了工业、医学、生活、商业、社会发展等领域中, 其中多数应用都涉及数据的管理、监测、保密等

① 基金项目: 山东省重大科技创新工程 (2019JZZY010134)
收稿时间: 2022-04-03; 修改时间: 2022-05-09; 采用时间: 2022-05-28; csa 在线出版时间: 2022-07-22

问题, 而数据安全更是关键. 物联网通信协议中最主流的数据传输协议则是 MQTT (message queuing telemetry transport), 本质是一种工作在 TCP/IP 协议族上基于发布/订阅范式的消息协议. 在设计车辆网络管控系统的过程中, 发现车辆在上传运行数据和自身状态数据时需要对数据进行加密传输, 同时在接收授权中心下发控制指令的过程中, 要保证指令的准确和真实, 因此必须将安全传输算法与 MQTT 协议相结合, 但与哪些安全传输算法相结合要考虑特定物联网场景下的设备性能. 物联网的感知终端往往性能参差不齐. 我们知道海量物联网设备中有绝大多数是性能低下、网络带宽受限的设备终端, 而 MQTT 协议刚好满足这些需求. 有优势的同时也不可避免存在缺陷, MQTT 协议本身在传输数据的时候不对数据进行加密, 也没有相应的安全机制. 这一缺陷在当今时代是不被人们所容忍的, 工业、生活、经济上的数据保密和信息安全格外被关注. 这就使得我们在不放弃主流 MQTT 协议前提下, 设计相应的加密算法, 从而保证协议继续在物联网安全领域广泛地被应用. 2010 年国家出台《国务院关于加快培育和发展战略性新兴产业的决定》, 标志着物联网被列入国家发展战略, 科研人员和国家相关单位都在这方面一直尝试和努力. 钱玉磊[1] 提出了如何抵御各个层面上的 Dos/DDoS 攻击, 提供可靠服务保障, 但没有在数据传输层、数据链路层提出安全保护方案. 邢赛楠[2] 从 3 个方面提出保护安全的解决方案: 认证、鉴权以及数据加密. (1) 认证: 用户名和密码. (2) 鉴权: 服务端对客户端进行分类分级, 也可称之为 ACK 鉴权. (3) 数据加密: 传输层采用 TLS 进行加密, 数据链路层采用 AES 加密. 但并不是所有的物联网设备都部署在能够提供安全保护的传输层、链路层的安全环境中. 而且也没有给出 ACK 鉴权的具体技术方案. Barata 等[3] 提出了健康数据收集、传输、可视化系统, 但没有对数据传输进行保护, 病人的隐私更得不到保障. 李勇[4] 提出应用于 UDP协议之上的 MQTT-SN 协议[5], 它是基于传感器网络环境的 MQTT 协议, 不适用 4G/5G 移动通信网络和传统宽带网络的 MQTT 协议. Patel 等[6] 提出了基于 MQTT 的用户验证框架, 用户设备和传感设备在连接之前没有考虑假冒连接, 同时共享密钥的传输容易被截获. Shin 等[7] 提出了基于 MQTT 的安全框架 AugMQTT, 该框架通过 AugMQTT 协议生成会话密钥, 但 AugPAKE 协议建立在 D-H 密钥交换的基础上, 其本质就是复杂版的 D-H 密钥交换实现, 因此在交换密钥的过程中无法抵抗“中间人”攻击, 非法 Client 端或者 Server 端可以冒充“中间人”. 巫钟兴[8] 提出了基于软硬件协作的数据加密传输方案, 这使得方案必须在硬件支持下才可部署, 大大缩小了适用范围和运行成本. Lesjak 等[9] 提出了基于硬件的 TLS 客户端系统来保障 MQTT 安全性, 该类型的方案同样存在于成本与证书管理的问题. Venkata 等[10] 设计了一种轻量级传输方法 LWTM. 该算法分为控制层和数据层, 控制层负责密钥生成参数, 加密数据块在消息中的位置, 加密数据块的大小、位置. 加密数据可通过 TCP/UDP 等不安全传输, LWTM 提高了解密效率, 降低了计算机能耗. 高锐强等[11] 论证了基于 TLS/SSL 协议传输数据是在牺牲性能换取安全性, 但这同样也对物联网的设备的最低性能有了限制. 刘文浩等[12] 提出了一种无双线性对的无证书两方密钥协商方案, 该方法优点是避免了在证书管理方面占用大量资源的缺陷, 不过仍需要第三方帮助生成通信过程中的部分密钥. 谷正川等[13] 提出了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, 该方案引入了 AES 对称加密、代理重加密算法、Schnorr 签名算法等手段, 通过实验对比, 该方案对资源受限设备的适用程度并不高. 于振中等[14] 提出的 MQTT-EA 算法看似是基于前人研究的又一次提升, 但仔细论证, 不难发现其中有很多漏洞和弊端[15], 需要进一步改进. 为此通过分析 MQTT-EA 的算法中存在的不足, 提出 3 种可靠的加密传输算法.

2 MQTT-EA 算法的分析

文献 [14] 给出了 MQTT-EA 的流程图如图 1. 我们首先借助图 1 去分析文献 [14] 中 MQTT-EA 整个加密传输算法过程, 存在以下具体问题: (1) 在 Step 1 中, Client 端向 Server 端发送 CONNECT+deviceID 控制报文. Server 端在接收到验证 deviceID, 在这一步中, 非法 Client 端完全可以窃取“deviceID”进行假冒攻击. (2) 在 Step 3 中, 如果非法 Client 端携带窃取的 deviceID, 同时若知道密钥生成算法 f, 在交换 ServerKey 和 ClientKey 两个变量后, 仍然可以效仿 Server 端, 利用自定义密钥生成算法 f(ServerKey, ClientKey) 生成会话密钥 SessionKey. 在第 2 节结尾处提出对算法的保密在商业领域已经被抛弃, 并给出了证明. 因此 Step 3 中会话密钥是可通过假冒连接而非法获得.

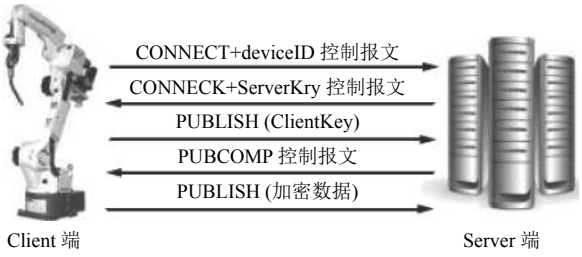

图 1 MQTT-EA 流程图

文献 [14] 在安全性分析中谈到, 在模拟攻防的过程中存在如下安全漏洞: 假设有敌手 A, A 能将自己的设备号 device-A 加到白名单, 并且能够监听 Client 端和 Server 端通信的任意阶段, 但是 A 不知道加密密钥. 一个安全系统除了密钥等关键参数需要保密外, 不应该假定某些算法是保密的, 因为实践证明这种保密不会长久. 自从 1977 年 DES 密码算法问世后, 通过对算法的保密来实现信息保密的方式基本被商业领域抛弃. 我们通过举一个实际意义的例子来证明该观点. 第二代移动通信系统 GSM 中使用的用于产生会话密钥的密钥生成算法 A3 (类似于本文中的密钥生成算法), 尽管设计之初被作为高级商业机密对待, 而且以硬件的形式在 SIM 卡中实现, 但还是被研究人员通过蛛丝马迹恢复出算法的详细信息, 并公布到互联网上. 在本文中, 密钥生成算法通过软件实现, 则很容易被破解, 例如通过对硬件或者设备终端进行反编译就可破解其算法[16].

因此正确的安全模型是假设攻击者也掌握密钥生成算法, 但不知 Server 和 Client 之间的密钥. 只有这样才称得上符合当代物联网发展的加密算法. 当文献 [14] 中前提假设不成立的时候, 很多步骤都将是不安全的.

我们再从攻击者能力和假冒攻击两个角度去分析安全模型.

(1) 攻击者能力假设

ClientID 不是秘密信息, 攻击者可以获得, 因为无论在终端破解硬件, 还是在数据通过 TCP/UDP 协议传输的过程中均可以被截获. Server 和 Client 都需要使用一个密钥生成算法 $f()$. 虽然这个算法是自定义的, 但需要在 Server 和 Client 之间共享.

(2) 假冒攻击

攻击者假冒一个合法的 ClientID, 又能破解其密钥生成算法, 表明整个 MQTT-EA 流程没有任何障碍, 同时在基于前面的安全假设前提下, 假冒攻击可以成功.

基于前者的模型和存在问题, 提出一种新的加密算法, 该算法又细分为 3 种具体的加密方案, 针对不同的应用场景发挥其算法优势, 可以选择单独使用也可以选择配合使用.

3 一种新的 MQTT-SE 算法

本文针对 MQTT-EA 存在假冒攻击、密钥泄露的风险提出一种对 MQTT 安全加强的协议, 称为 MQTT-SE (MQTT security enhancement).

3.1 基于对称加密的 MQTT-SE 算法

文献 [14] 中的设计思想是 Client 端将 ClientKey 发送到 Server 端. (2) Server 端将已知的 Serverkey 和 Clientkey 作为变量, 利用算法 f(ServerKey, ClientKey) 生成会话密钥 SessionKey. 由于 ServerKey、ClientKey 在传输的过程中可以被监听, 再加上会话密钥生成算法容易被破解. 这种安全算法不能提供较高的安全性. 我们提出基于对称加密的 MQTT-SE 算法, 该算法的特点表现在对称密钥的选择以及会话密钥的协商.

如何完成会话密钥共享, 同时保证会话密钥在 Client 端和 Server 端之间安全传输. 会话密钥协商的具体步骤如下.

(1) Client→Server: CONNECT+ClientID+$R1$, 其中 $R1$ 是 Client 产生的随机数.

(2) Server 执行如下步骤.

1) 根据 ClientID 找到他们之间共享的对称密钥 K, K 被预先植入设备内.

2) 产生随机数 $R2$.

3) 计算 SessionKey=$f(K, R1, R2)$.

4) Server→Client: CONNECT+$R2$.

(3) Client 收到上述数据后, 执行如下步骤.

1) 计算 SessionKey=$f(K, R1, R2)$.

2) 使用 SessionKey 加密如下数据: ClientID||data, 然后将密文传给 Server.

(4) Server 收到上述数据后, 使用 SessionKey 解密数据, 比较 ClientID 是否正确. 若不正确, 则中断连接; 否则存储数据 data. 如果 Server 需要向 Client 发送数据, 则直接使用 SessionKey 进行加密, 后续加密无须包括 ClientID, 当会话密钥 SessionKey 传输完毕以后, 可以根据网络环境和设备性能选取加密算法, 例如 AES、国密 SM4 等.

以上算法执行全过程可由图 2 详细说明.

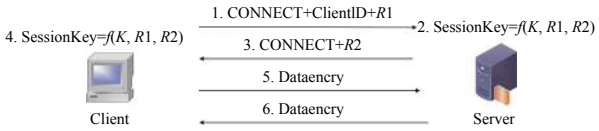

图 2 基于对称加密的 MQTT-SE 算法

由于每次会话密钥 SessionKey 的生成都依赖 $R1$ 和 $R2$, $R1$ 与 $R2$ 都会随机生成, 每次连接都是用新的 SessionKey, 保证每次加密数据的密钥都是不同的, 增加了截获密钥的攻击成本; 基于对称加密 MQTT-SE 算法中的 K 是预先植入设备中, 在设备出厂之前就已经植入, 正是由于这个特点, 本算法适用一个物联网系统中的设备均来自于同一出厂商, 生产商在设备出厂之前就植入密钥 K, 避免了密钥 K 传输过程中泄露的风险.

3.2 基于公钥的 MQTT-SE 算法

由于物联网设备样式多种多样, 一个完整的物联网系统中所包含的设备往往来自多个出厂商, 所以基于对称密钥的 MQTT-SE 算法就不太适用, 因为共享密钥 K 无法预先植入不同厂家的物联网设备中. 所以为了解决这个问题, 提出了基于公钥的 MQTT-SE 算法. 公钥的使用一般通过公钥证书来实现. 公钥可以关联用户信息、机构签名. 给不同类型的物联网设备分发公钥证书, 解决了不同出厂商下的物联网设备无法预先植入共享密钥 K 的弊端.

公钥加密的信息只能由与之相对应的私钥解密, 基于公钥, 可设计如下 MQTT-SE 算法步骤.

(1) Client→Server: CONNECT+Certificate_Client+$R1$, 其中 $R1$ 是 Client 产生的随机数.

(2) Server: 执行如下步骤.

1) 验证 Certificate_Client 的正确性.

2) 产生随机数 $R2$, 计算 SessionKey=$R1 \oplus R2$.

3) 使用 Client 的公钥 pk_Client 加密 $R1$ 与 $R2$ 的异或, 得到 $c=E_{\text{pk\_Client}}$(SessionKey).

4) 将下述数据发送给 Client: CONNECT+$R2$+c.

(3) Client 收到上述数据后, 执行如下步骤.

1) 使用自己的私钥解密 c, 得到 SessionKey.

2) 验证 SessionKey $\oplus R2 = R1$ 是否成立, 若不成立, 则中断连接.

3) 使用 SessionKey 加密如下数据: ClientID||data, 然后将密文传给 Server.

(4) Server 收到上述数据后, 使用 SessionKey 解密数据, 比较 ClientID 是否正确, 然后存储数据 data.如果 Server 需要向 Client 发送数据, 则直接使用 SessionKey 进行加密, 后续加密无须包括 ClientID.

以上算法执行全过程可由图 3 详细说明.

图 3 基于公钥的 MQTT-SE 算法

公钥证书的 Certificate_Client 的引入, 可以避免非法 Client 端利用有效的 ClientID 实现假冒连接. 安全性较第 3.1 节中的“基于对称加密的 MQTT-SE 算法”有了很大提升, 但公钥证书的管理对设备终端的性能有了一定的要求.

3.3 基于公钥证书的双向认证 MQTT-SE 算法

上述方案第 3.2 节中只使用了 Client 的公钥, 任何一个人都可以假冒 Server 来执行这个协议, 因为该协议不需要 Server 提供任何参数. 它的缺陷只满足了单向认证, 所以提出了基于公钥证书的双向认证 MQTT-SE 算法.

如果没有选择双向认证, 整个过程就不会引入 Server 的证书, 因此任何一方都可以假冒 Server, 和 Client 建立连接并协商 SessionKey, 实现数据传输, 从而非法 Server 端就可以欺骗 Client.

(1) Client→Server: 发送数据给服务端, 内容包括 CONNECT+Certificate_Client, 同时 Client 产生的随机数 $R1$, Server 产生随机数 $R2$.

(2) Server 执行如下步骤:

1) 验证 Certificate_Client 的正确性, 并用 Client 的公钥加密 $R2$, 即 $C_2=E_{\text{pk\_Client}}(R2)$.

2) 向 Client 端发送 Certificate_Server+ C_2.

(3) Client 执行如下步骤.

1) Client 验证 Certificate_Server 的正确性, 并用 Client 的私钥解密 C_2, 得到 $R2$.

2) 计算 $C_1=E_{\text{pk\_Server}}(R1)$, 并把 C_1 发送至 Server 端.

(4) Server 用自己的私钥解密 C_1, 到 $R1$, 并计算 SessionKey=$R1 \oplus R2$, 用 Client 的公钥加密会话密钥 Sessionkey, 即: $c=E_{\text{pk\_Client}}$(SessionKey), 将密文 c 发送给 Client 端.

(5) Client 收到上述数据后, 执行如下操作.

1) 使用自己的私钥解密 c, 得到了 SessionKey.

2) 验证 SessionKey⊕$R2$=$R1$ 是否成立, 成立则连接成功, 若不成立, 则中断连接.

(6) Server 与 Client 之间的数据传输, 则直接使用 SessionKey 作为会话密钥, 选择合适的加密算法进行加密, 从而保证整个通信过程安全稳定.

以上算法执行全过程可由图 4 详细说明.

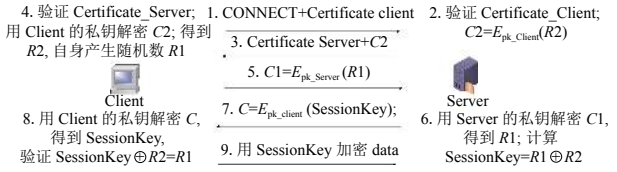

图 4 基于公钥证书的双向认证 MQTT-SE 算法

该协议的优点实现了双向认证, 避免了 Client 端和 Server 端的任意一端的假冒连接, 最终以 SessionKey 为密钥选择适合的加密算法加密数据, 从而在保证双向认证的基础上实现了数据保密传输.

该协议的缺点是: 该通信在发送公钥和验证签名的同时, 需要有 3 次额外数据通信服务, 增大了通信开销, 对 Client 端、Server 端的计算性能、通信能力有了一定的要求.

4 实验分析

第 3 节提出了实现 MQTT-SE 算法的 3 种具体方案, 分别是基于对称加密算法的 MQTT-SE 算法、基于公钥的 MQTT-SE 算法、基于公钥证书的双向认证 MQTT-SE 算法. 我们分别对每个算法做安全理论分析和实验分析.

4.1 对基于对称加密的 MQTT-SE 算法安全分析

在 Client 端连接 Server 端的过程中, 首先考虑的是假冒攻击, 其次就是密钥在传输的过程中是否有泄露的风险.

4.1.1 假冒攻击

由于本应用场景所需要的物联网设备是同一出厂商的物联网设备, 其中设备在出厂前都被植入密钥 K, 并做了详尽的密钥登记工作, 假设某一非法 Client 端窃取了有效的 ClientID、$R1$, 通过初步与 Server 端传输可以获得 $R2$, 但由于无出厂前的内置密钥 K, 无法实现 SessionKey=$f(K, R1, R2)$ 计算, 但 Server 端可完成 SessionKey=$f(K, R1, R2)$, Client 端加密的数据无法让 Server 端正常解密, 因此 Client 端无法与 server 端建立正常连接.

4.1.2 密钥安全

由于密钥 K 是同一厂家预先植入设备内, 避免了密钥 K 在网络空间中传输, 非法 Client 端自然无法在传输的过程中侦听或截获密钥 K.

4.2 对基于公钥证书的 MQTT-SE 算法安全分析

在客户端和服务器交互的过程中, 比较容易常见的两种攻击, 一种是假冒攻击, 另一种就是非法监听.公钥证书的提出是针对无法预先植入密钥 K, 不同出厂商下的物联网设备.

4.2.1 假冒攻击

(1) A 通过非法渠道获得 Certificate_Client_A, 并自身随机产生随机数 $R1'$.

(2) Server 执行如下步骤: 验证 Certificate_Client_A 的正确性, 发现其真实身份和证书所标识的身份不一致, 则拒接连接.

4.2.2 密钥安全

(1) 由于 SessionKey 是由 $R1⊕R2$ 得到, 而 $R1$ 与 $R2$ 由 Client、Server 分别产生的随机数, SessionKey 的建立需要 $R1$ 和 $R2$ 共同参与, 目的是避免一方一直独自掌握 SessionKey 建立的权限, 增加协商密钥的不确定性, 从而提高密钥的安全性.

(2) 由于在使用 Client 的公钥加密 pk_Client 加密 $R1$ 与 $R2$ 的异或, 得到 C=E_{pk_Client}(SessionKey), 同时 Server 端把 CONNECT+$R2$+c 发送给 Client, 使用自己的私钥解密 c, 得到 SessionKey; 紧接着验证 SessionKey⊕$R2$ = $R1$ 是否成立, 成立则使用 SessionKey 加密如下数据 ClientID||data, 然后将密文传给 Server, 解密后比对 ClientID 和 $R1$, 正确, 则成立; 若不成立, 则中断连接.

(3) 该过程中, SessionKey 是被 pk_Client 加密后以密文 c 的形式发送 Client 端, 即使截获密文 c 也无法解密出 SessionKey, 当然也无法破解连接成功后的密文.

4.3 基于公钥证书的双向认证 MQTT-SE 算法安全分析

单向认证只能保证连接过程中 Client 端不被假冒, 在需要较高安全需求的应用背景下需要实现双向认证, 从而禁止非法 Server 端“欺骗”Client 端.

4.3.1 假冒攻击

(1) Client→Server 发送 CONNECT+Certificate_Client 给 Server 端, Server 验证 Certificate_Client 的正确性, 实现了对 Client 的身份确认, 保证了 Client 的真实性.

(2) Server 产生随机数 $R2$, 计算 $C_2=E_{pk\_Client}(R2)$, 保证 $R2$ 以密文的形式传输, 同时把 Server 的公钥证书 Certificate_Server 和 C_2 发送至 Client 端.

(3) Client 验证 Certificate_Server 的正确性, 确认了 Server 的身份, 保证了 Server 端的真实性; 并用 Client 的私钥解密 C_2, 得到 $R2$ 并存在 Client 端.

(4) 借助公钥证书 Certificate_Client、Certificate_Server 可以避免 Client 端和 Server 端的假冒连接.

4.3.2 密钥安全

在保证 Client 端和 Server 端均为真实可靠的前提下, 会话密钥的约定过程如下.

(1) Server 端计算 $C_2=E_{pk\_Client}(R2)$ 并发送至 Client 端, 以密文形式传输 $R2$, $R2$ 是安全的.

(2) Client 端计算 $C_1=E_{pk\_Client}(R1)$ 并发送至 Server 端, 同样以密文的形式把 $R1$ 传输至 Server 端.

(3) 此时 Server 端和 Client 端均存储了 $R1$, $R2$, Server 端计算 SessionKey=$R1\oplus R2$, 同时以密文形式 $c=E_{pk\_client}$(SessionKey) 传输至 Client 端.

(4) Client 端可以通过自己的私钥解 c, 得到 SessionKey, 为了保证 SessionKey 是由最初的 $R1$ 与 $R2$ 的异或得到, 再验证 SessionKey$\oplus R2=R1$ 是否成立, 成立则说明能保证 SessionKey 的机密性和完整性. 保证了会话密钥的安全, 就能保证加密数据传输的安全性. SessionKey=$R1\oplus R2$; $c=E_{pk\_client}$(SessionKey) 同时 Server 端把 CONNECT+$R2$+C 发送给 Client, 使用自己的私钥 c, 得到 SessionKey; 紧接着验证 SessionKey $\oplus R2 = R1$ 是否成立. 以密文 c 的格式传输, 是避免网络抓包导致密钥泄露. SessionKey 的协商受 $R1$、$R2$ 影响, 保证会话密钥每次在连接过程中都是随机变化的, 避免会话密钥的一成不变, 即使偶然的一次密钥泄露也不会影响后续的连接. 因此可以保证了密钥的安全性.

4.4 传输性能的比较

对于计算能力受限、又有数据安全传输需求的设备. 要么对传统 Publisher-Broker-Subscriber 加以改进, 要么将 Publisher-Broker-Subscriber 与相适应的加密算法相结合. 基于 MQTT 的车辆网络管控系统充当实验环境来验证本文所提出的安全传输算法. 通过多次实验分析, 其实验结果如表 1 所示, 服务器设备为山东省重大科技创新工程项目中自行设计的系统, 名称为车辆网络管控系统. 由表 1 得出结论, 基于对称加密的 MQTT-SE 算法和基于公钥的 MQTT-SE 算法具有时间性能优势. 同时横向比较 4 种数据加密传输算法, 从传输数据包的数量和耗时两个指标进行分析, 分析结果如图 5 所示. 将新提出的 MQTT-SE 算法下的 3 种具体安全加密算法和 MQTT 协议相结合并在相同的实验环境中做性能测试. 并与文献 [3] MQTT-EA 算法相比较. 由图 5 得出结论, 基于公钥证书的双向认证 MQTT-SE 算法中由于客户端和服务器需要双向认证, 传输相同数量数据包的情况下则和 MQTT-EA 性能相似. 基于对称加密的 MQTT-SE 算法与基于公钥的 MQTT-SE 算法性能优于前两种算法, 并且基于公钥的 MQTT-SE 算法的性能更优一点.

表 1 4 种加密算法及相关参数的对照

类别	MQTT-EA	基于对称加密的MQTT-SE	基于公钥的 MQTT-SE	基于公钥证书的双向认证MQTT-SE
安全密级	不安全	较安全	安全	很安全
加密耗时 (ms)	512	387	441	537

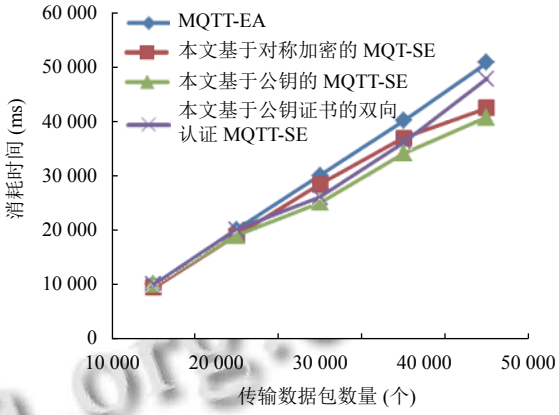

图 5 4 种加密传输算法的性能比较

4.5 MQTT-SE 改进效果

截止目前, 大量研究者基于 MQTT 协议提出了各具特色的数据加密传输算法或方案. 但仍可粗略分为基于 SSL 安全协议类型、基于软硬件协作类型、基于 MQTT-SN 类型、基于公钥证书类型等, 为更好体现本文所提出的 MQTT-SE 算法的改进效果, 则将 MQTT-SE 算法与上述不同类型下的典型算法做横向对比, 对比结果如表 2 所示, 测试环境均为车辆网络管控系统.

由表 2 可知, 本文提出的基于对称加密的 MQTT-SE 算法在部署成本、硬件要求、传输性能、稳定性、维护成本、可扩展性等方面都是优于其他算法, 缺点则是安全性一般. 本文基于公钥 MQTT-SE 算法, 能够拥

有和表 2 中其他加密传输算法同样高的安全等级, 同时可以兼容中/低功耗传感器, 该特点直接拓宽了本算法的应用场景, 同时在完成系统部署后, 表现出良好的稳定性、后期维护成本较低, 可扩展性高. 在传输性能、证书管理、双向认证等方面也优于其他算法.

4.6 MQTT-SE 算法的有效性

虽然基于 MQTT 协议的数据加密传输算法有很多, 但目前在本领域来看, 文献 [13] 中“基于代理重加密的数据队列遥测安全传输协议”和文献 [6] 中“一种在通用物联网模型中基于 MQTT 的新型安全框架”被大多数研究者所认可, 一定意义上可代表本领域最新研究现状. 为详细说明本文提出的 MQTT-SE 算法的有效性, 从两个大的方面去分析: (1) 并发处理能力, (2) 数据服务能力.

表 2 MQTT-SE 算法与其他加密传输算法的对照

类别	MQTT-SN[4,5]	基于SSL安全协议[2,11]	基于软硬件协作[8,9]	本文基于对称加密的MQTT-SE	本文基于公钥的MQTT-SE
部署成本	低	一般	高	低	低
硬件要求	低功耗传感器	高功耗传感器	高功耗传感器	低功耗传感器	中/低功耗传感器
传输协议	MAC/IP/UDP层	TCP层	TCP层	TCP层	TCP层
传输性能	较快	一般	较慢	快	快
证书管理	无需	需要	无需	无需	无需
稳定性	不稳定	稳定	一般	稳定	高
维护成本	一般	高	高	低	低
可扩展性	低	低	一般	高	高
双向认证	无	无	无	有	有

(1) 并发处理能力

将不同的数据加密算法部署在相同的服务器上, 并模拟不同数量的客户端去和服务器建立连接, 并观测服务器的 CPU 使用率的变化情况. 在保证算法为唯一可变变量的前提下, 如果 CPU 使用率越高, 则说明处理并发的能力越弱; 反之, 处理并发的能力越强. 文献 [13] 算法、文献 [6] 算法及本文 MQTT-SE 算法的并发处理能力如图 6 所示.

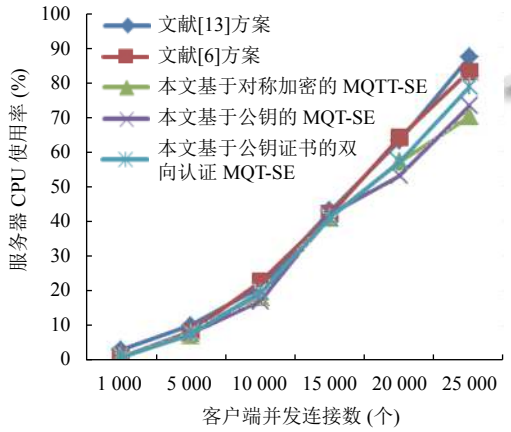

图 6 5 种加密传输算法并发能力大小比较

(2) 数据服务能力

文献 [13] 中既使用 AES 对数据进行对称加密, 还在代理端引入了重加密算法, 最后使用 Schnorr 签名算法对消息进行签名, 整个过程对客户端和服务器的计算性能和网络资源都有一定的要求, 该方案对资源受限设备的适用程度并不高, 尤其在数据加密/解密、密钥管理、数据重连等方面在对比试验中表现并不出色. 文献 [6] 中以代理预共享的公钥加密数据, 设备与代理通过公钥证书来验证彼此, 并通过访问控制策略去管理主题“topic”的分配权限, 该方案在会话密钥的管理、相互认证过程中占用了大量网络资源和时间开销. 为了更好分析文献 [13]、文献 [6]、本文提出的 MQTT-SE 数据加密传输算法之间的优缺点, 把上述算法与方案分别部署到山东省重大科技创新工程项目下的“车辆网络管控系统”中去, 编写特定的测试程序, 借助测试工具. 在保证只有算法不同其他均相同的前提下, 对数据加/解密、密钥管理、认证时间、数据传输延迟、断线重连等性能指标进行实验对比, 对比结果如图 7 所示.

由图 7 可知, 在同样的测试环境中, 本文基于对称加密的 MQTT-SE 算法在加解密、数据延迟方面都优于文献 [13] 方案和文献 [6] 方案. 本文基于公钥的 MQTT-SE 算法与本文基于公钥证书的双向认证 MQTT-SE 算法在密钥管理、认证时间等方面表现都优于文献 [13]、文献 [6] 中的方案.

4.7 MQTT-SE 应用场景

对于不同性能的物联网设备可从第 3 节中选取适

合自身应用场景的数据加密算法.

场景 1. 例如收集温度、湿度、光照等受限 Client 端, 前提是 Client 端和 Server 端来自同一设备出厂商, 并在出厂前植入了共享密钥 K, 避免了在两端传输的过程中被侦听的风险. 这种应用场景下可以选择基于对称加密的 MQTT-SE 算法, 该算法满足一定的安全性, 同时可以满足数据的传输效率, 有较好的数据处理能力.

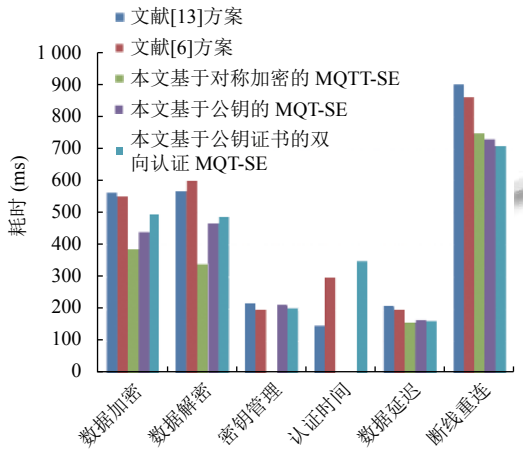

图 7 5 种加密传输算法数据服务能力比较

场景 2. 若一个物联网系统中的 Client 和 Server 并不来自同一出厂商, 则预先植入共享密钥 K 是不现实的, 所以选择基于对称加密的 MQTT-SE 算法是不恰当的; 因此, 当具有一定请求权限能力的 Client 端, 为了保证请求指令的安全性, 可以选择基于公钥的 MQTT-SE 算法, 因为该 Client 端的资源、性能较富足, 可以满足公钥证书正常工作的基本性能需求, 且对安全性有较高的要求.

场景 3. 如果是远程网络授权中心和本地控制中心进行通信指令的传输, 则对安全性和双方的真实性都有极高的要求, 如果采取基于公钥的 MQTT-SE 算法, 只能实现对 Client 端单向认证, 无法辨别 Server 端的真伪; 对于这一类应用场景则可以优先考虑基于公钥证书的双向认证 MQTT-SE 算法, 在两者计算性能、网络资源都充足的场景下, 算法多项性能指标几乎不受影响, 同时能保证传输数据 (授权指令) 的安全性和完整性.

如果在一个庞大的物联网系统中, 可能存在以上多种应用场景, 则可以将几种加密算法协同部署.

5 结论与展望

本文基于实际设计并完成的车辆网络管控系统项目, 提出了在终端设备受限又需要数据安全传输的应用场景下, 如何保证设备终端与网络平台进行数据安全传输. 将 MQTT 协议与数据加密算法相结合, 提出一种新型的 MQTT-SE 算法. 该算法实现简易, 传输效率高, 虽然无法达到绝对的安全, 但安全是相对而言的, 至少在一定程上极大提高了假冒攻击、窃取密钥、破解密文的门槛, 本文提出的基于对称加密的 MQTT-SE 算法、基于公钥的 MQTT-SE 算法、基于公钥证书的双向认证 MQTT-SE 算法, 具有一定的普适性和通用性, 且已经投入使用. 可以根据特定的应用场景而做出选择, 基于对称密钥的加密算法既可以提供身份验证, 又可以支持加密传输, 对绝大部分受限设备都比较受用. 对于基于公钥的 MQTT-SE 算法, 在当代物联网、智能家居、智能医疗等领域逐渐被大家所接受, 因此对于一些计算性能稍富足的应用场景则可以选择基于公钥证书的双向认证 MQTT-SE 算法.

参考文献

1 钱玉磊. 基于 MQTT 的安全通信服务器的研究与实现 [硕士学位论文]. 沈阳: 中国科学院研究生院 (沈阳计算技术研究所), 2015.
2 邢赛楠. MQTT 传输安全问题浅析. 科技与创新, 2018, (1): 17–18.
3 Barata D, Louzada G, Carreiro A, *et al*. System of acquisition, transmission, storage and visualization of pulse oximeter and ECG data using Android and MQTT. Procedia Technology, 2013, 9: 1265–1272. [doi: 10.1016/j.protcy.2013.12.141]
4 李勇. 无线传感器网络 MQTT-SN 协议安全机制的研究 [硕士学位论文]. 重庆: 重庆邮电大学, 2018.
5 Sadio O, Ngom I, Lishou C. Lightweight security scheme for MQTT/MQTT-SN protocol. Proceedings of the 2019 6th International Conference on Internet of Things: Systems, Management and Security (IOTSMS). Granada: IEEE, 2019. 119–123. [doi: 10.1109/IOTSMS48152.2019.8939177]
6 Patel C, Doshi N. A novel MQTT security framework in generic IoT model. Procedia Computer Science, 2020, 171: 1399–1408. [doi: 10.1016/j.procs.2020.04.150]
7 Shin S, Kobara K, Chuang CC, *et al*. A security framework for MQTT. Proceedings of 2016 IEEE Conference on Communications and Network Security (CNS). Philadelphia:

IEEE, 2016. 432–436. [doi: 10.1109/CNS.2016.7860532]
8 巫钟兴. 数据加密传输系统的研究与应用 [硕士学位论文]. 北京: 北京化工大学, 2010.
9 Lesjak C, Hein D, Hofmann M, *et al*. Securing smart maintenance services: Hardware-security and TLS for MQTT. Proceedings of the 2015 IEEE 13th International Conference on Industrial Informatics (INDIN). Cambridge: IEEE, 2015. 1243–1250.
10 Venkata SB, Yellai P, Verma GD, *et al*. A new light weight transport method for secured transmission of data for IoT. Proceedings of the 2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS). Bangalore: IEEE, 2016. 1–6. [doi: 10.1109/ANTS.2016.7947775]
11 高锐强, 朱虹, 贾立东, 等. 基于 SSL 安全协议实现工业控制通讯协议加密及认证的研究. 化工设计通讯, 2019, 45(1): 121–123. [doi: 10.3969/j.issn.1003-6490.2019.01.108]
12 刘文浩. 无双线性对的无证书公钥密码学研究 [博士学位论文]. 成都: 电子科技大学, 2010.
13 谷正川, 郭渊博, 方晨. 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. 计算机应用, 2021, 41(5): 1378–1385. [doi: 10.11772/j.issn.1001-9081.2020060985]
14 于振中, 洪辉武, 徐国, 等. 基于 MQTT 的数据加密传输算法. 计算机系统应用, 2019, 28(10): 178–182. [doi: 10.15888/j.cnki.csa.007124]
15 徐绘凯, 刘跃, 马振邦, 等. MQTT 安全大规模测量研究. 信息网络安全, 2020, 20(9): 37–41. [doi: 10.3969/j.issn.1671-1122.2020.09.008]
16 杨毅宇, 周威, 赵尚儒, 等. 物联网安全研究综述: 威胁、检测与防御. 通信学报, 2021, 42(8): 188–205. [doi: 10.11959/j.issn.1000-436x.2021124]

(校对责编: 孙君艳)