面向智慧旅游物联网设备的身份认证方法的研究

杨崇宇[1]，何乐生[1, 2]，胡崇辉[1]，冯毅[1]，岳远康[1]
（1. 云南大学信息学院，云南 昆明 650091；2. 云南省智慧旅游工程研究中心，云南 昆明 650091）

摘　要：物联网化是智慧旅游业发展的重要趋势，其网络安全问题涉及政府管理端、公有云平台、设备生产商、景区方和游客等多个相关方，且物联网设备均部署在公共场所中，易受物理攻击，身份认证成为物联网安全的基础和关键。提出了基于行政申请的无证书身份认证方法，通过消息队列遥测传输（MQTT, message queuing telemetry transport）协议的消息队列机制维护设备安全状态，解决低功耗设备休眠带来的状态问题。该方法基于国家商用密码算法，保证了物联网信息安全的自主可控。性能评估显示，该方法能够有效帮助管理部门防范来自上述各方的安全威胁，身份认证平均准确率达到 99.7%，且设备嵌入式随机存取存储器（RAM）、闪存（FLASH）消耗各不超过 35 KB、30 KB，满足智慧旅游场景应用需求。

关键词：智慧旅游；物联网；身份认证；MQTT；国密算法
中图分类号：TN915.08
文献标志码：A
doi：10.11959/j.issn.2096−3750.2025.00433

Research on identity authentication methods for IoT devices in smart tourism

YANG Chongyu[1], HE Lesheng[1,2], HU Chonghui[1], FENG Yi[1], YUE Yuankang[1]
1. College of Information, Yunnan University, Kunming 650091, China
2. Yunnan Province Smart Tourism Engineering Research Center, Kunming 650091, China

Abstract: The internet of things (IoT) is a key trend in smart tourism, involving multiple stakeholders like government management, public cloud platforms, device manufacturers, scenic areas, and tourists. IoT devices, often deployed in public spaces, are vulnerable to physical attacks, making identity authentication critical for security. A certificate-free identity authentication method based on administrative applications was proposed, using MQTT protocol message queues to maintain device security status, addressing issues with low-power devices in sleep mode. Based on national cryptographic algorithms, secure and controllable IoT information was ensured. Performance evaluations show that it effectively helps prevent security threats, achieving an average authentication accuracy of 99.7%, with embedded RAM and FLASH usage not exceeding 35 KB and 30 KB, suitable for smart tourism applications.

Key words: smart tourism, IoT, identity authentication, MQTT, Chinese cryptographic algorithm

0　引言

随着信息技术的快速发展，旅游物联网的应用已经成为了现代旅游业的重要部分[1-2]。智慧旅游物联网是一种融合了物联网技术与旅游业的新型模式，能够提高旅游业的经济效益，增强游客的满意度和忠诚度[3-4]。然而，旅游物联网的持续发展与物联网设备数量的激增也带来了新的挑战。

在旅游物联网中，存在着复杂的参与者和不同的利益诉求。在此背景下，除了关注攻击者[5-7]以外，还必须考虑政府机构作为管理端的权利保障，预防设备生产商、公有云平台和游客的安全违规行为。例如，设备生产商未经许可私自添加设备或多台设备使用同一身份连接至服务器。在这些复杂的

收稿日期：2024−09−08；修回日期：2024−10−29

安全考虑中，强化身份认证机制以保障管理端的管理权显得尤为关键。

此外，旅游物联网设备往往部署在公共或半公共场所，攻击者较易接近这些设备，从而进行侧信道攻击。智慧旅游中的设备需要长时间运行且不间断地进行数据传输和处理，这为攻击者提供了充足时间来观察和分析可能的侧信道数据。若旅游物联网设备被侧信道攻击击破，游客个人隐私将被大范围侵犯，景区的声誉将遭到破坏。而目前的研究尚未充分探讨如何通过身份认证技术全面提升智慧旅游物联网的整体安全。

其次，消息队列遥测传输（MQTT, message queuing telemetry transport）协议[8-11]是旅游物联网最常用的协议。其中Broker是MQTT协议中的核心组件，负责管理客户端之间的消息通信[12-14]。在智慧旅游物联网中，往往选择第三方公有云平台作为Broker，数据直接流转到公有云服务器，存在安全隐患。特别在身份认证方面，可能导致数据遭到篡改或伪造，增加未经授权访问和数据泄露的风险。

最后，随着物联网网络环境日益复杂[15-17]，存在信号强度或游客数量激增导致数据丢包的可能性，从而影响身份认证的准确性。如何保证认证准确性成为一项巨大的挑战[18-20]。若设备安全状态被误判，会导致正常服务中断，进而影响游客的体验。

1　相关工作

近年来，国内外研究者深入研究了MQTT网络环境下物联网设备的身份认证问题。在网络空间安全领域，数字证书是身份认证的常用方法。数字证书通过认证机构颁发，确保通信双方的身份，并对数据进行加密保护。然而，在物联网环境中，传统的数字证书认证机制存在显著的缺点。物联网设备计算能力和存储空间较为有限，这使得处理和管理数字证书变得不切实际。数字证书的生成、存储、更新和撤销过程都需要较高的计算和存储资源，这对于资源受限的物联网设备来说负担过重。因此，在物联网环境下，研究者越来越关注无证书认证（CA, certificateless authentication）的方式，例如，基于预共享密钥方案或对称加密技术，能够有效减少计算和存储开销。这种方法不仅提高了身份认证的效率，还能更好地适用于物联网设备，成为了物联网身份认证研究的重点方向。

Shapsough等[21]提出了通过传输层安全协议（TLS）/数据报传输层安全协议（DTLS）增强消息传输的安全性，但没有探讨MQTT中端对端认证机制。Bisne等[22]提出了基于密钥策略基于属性加密（KP-ABE）和高级加密标准（AES）加密的MQTT协议安全机制，但此方案存在计算开销较大的问题，难以应用于资源有限的旅游物联网设备中。Calabretta[23]等提出了基于口令认证密钥协议的MQTT设备身份认证方案，但只考虑了Broker对设备的身份认证，未能保证端到端的安全性。Imghoure[24]等提出了使用椭圆曲线加密算法的MQTT安全认证方案，虽然安全性足够，但是只考虑了第三方攻击者，未考虑Broker和设备生产商可能的违规操作。Buetas等[25]提出了一种基于加密安全卡的MQTT安全认证方案，但是增加了成本，且难以标准化。Lesjak等[26]虽然考虑了MQTT协议中Broker的不可完全信任性，使用了TLS芯片提高安全性，但是也存在兼容性问题和供应链问题。谷正川等[27]提出了基于代理重加密的MQTT协议安全认证方案，保证了端到端的安全认证，但是缺乏了Broker对设备的身份认证。

综上所述，现有研究并未考虑设备生产商及Broker存在的安全隐患，对智慧旅游场景下的特有问题考虑不足，适用性不强。对此，本文提出了一种面向智慧旅游物联网设备的认证方法。

2　智慧旅游物联网系统模型

本文所针对的“一部手机游云南”网络结构模型如图1所示。“一部手机游云南”是由云南省政府与腾讯公司联合打造的旅游智慧平台，已成为云南旅游产业转型升级的新引擎、云南数字经济发展的重要标志、中国智慧旅游的重要标杆。

其网络架构由管理端服务器、设备生产商制造的物联网设备、作为Broker服务器的公有云服务器3个主要部分组成。管理端服务器由政府机构搭建，作为整个设备身份认证的核心，享有最高信任等级。管理端服务器保存了所有物联网设备的公钥，负责对设备进行身份认证。公有云服务器凭借其先进的计算和存储能力，扮演MQTT协议中的Broker角色，负责接收物联网设备消息并将其分发给相应的管理端服务器。物联网设备由设备生产商

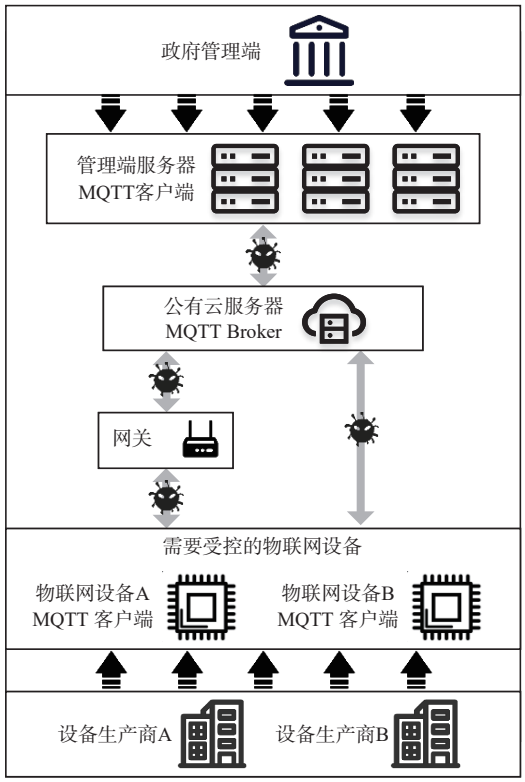

图1 “一部手机游云南”网络结构模型

生产，分布在智慧旅游场景中的各个节点，负责收集数据、执行控制任务以及与云端进行通信，应完全受政府机构的监管。此网络结构利用了公有云服务器的高效率和弹性特性，可以有效地处理大量来自物联网设备的数据交换需求，同时保证数据处理的速度和稳定性。

3 设备认证的设计与实现

为了防止设备生产商未经授权私自添加设备，采用基于行政申请的无证书一机一密策略，即每个设备连接至MQTT服务器所需的身份信息与密钥对是独一无二的，确保每个物联网设备节点具有唯一的安全标识，保障了设备连接的唯一性和密钥的安全性，防止未经授权设备接入网络。

同时，通过使用动态密钥机制，确保了管理端对物联网设备使用期限的管理，有效应对了对称加密密钥可能遭受的侧信道攻击，降低了密钥泄露造成损失的可能性。

此外，在现有智慧旅游物联网系统中，身份认证和数据传输通常由公有云服务器管理，因其强大的计算资源能够处理大量并行数据，从而降低管理和运营成本。然而，这种方式认证安全性较低，且设备身份认证完全依赖第三方公有云平台，难以保障政府管理端的管理权。为此，在现有身份认证流程的基础上提出了一种结合国密算法的新型身份认证机制，该机制不仅有效保障了管理端的权利，还显著提高了安全性，防止了公有云服务器可能存在的数据窃取和监听风险。

最后，使用MQTT协议虽然可以建立客户端与服务器之间的持久会话，对低功耗物联网设备有利，但也存在设备掉线引发身份认证失败的情况，进而导致合法设备被误判为非法设备。为了解决这一问题，设计了安全状态机制，在整个身份认证流程中持续有效，确保设备的安全状态得到准确评估，避免设备因外部因素被误判为非法设备，增强身份认证过程的抗干扰能力。

本节所使用的符号定义见表1。

表1　　符号定义

符号	定义
count	身份认证次数
$seed_1$,$seed_2$	随机数种子
random(·)	生成随机数的函数
PK,SK	SM2算法公钥、私钥
||	字符串连接符
Hmac(·)	Hmac算法
Authen_Req	身份认证请求
SM3(·)	SM3杂凑算法
$SM2_G(\cdot)$	SM2密钥派生算法
$SM2_S(\cdot)$,$SM2_V(\cdot)$	SM2签名/验签算法
$SM2_E(\cdot)$,$SM2_D(\cdot)$	SM2加/解密算法
key,key_{new}	SM4算法密钥
$SM4_E(\cdot)$,$SM4_D(\cdot)$	SM4加/解密算法
$C\{\cdot\}$	密文

3.1 安全状态机制

规定每个设备都存在着一个对应的变量count，用于记录设备重新身份认证的次数，若count达到预设阈值，则设备安全状态被标记为非法。此外，为了提高系统鲁棒性和身份认证准确性，本文引入了“复活”机制。允许非法状态的设备在一定条件下恢复为合法状态，从而确保设备能在误判情况下恢复正常运行。其中，设计超时状态与数据错误状态的目的是在合法状态和非法状态中增加一个缓冲区，保证设备安全状态判断的容错率，避免智慧旅游场景下可能存在其他因素的影响。设计频繁上线状态可有效避免设备生产商在不同设备上烧录相同身份信息，即未经许可的情况下私自添加设备。

这种设计不仅增强了整个安全系统对错误的鲁棒性，也提供了一种有效的安全保障措施，以应对智慧旅游环境下的复杂挑战。其安全状态转换关系如图2所示。

3.2　密钥获取阶段

此阶段设备生产商提交纸质《申请表》以获取设备身份信息，申请表包含设备的基本信息及相关证明文件，通过受控的行政申请流程实现密钥交互。其密钥对生成过程如下

$$SK = SM3(deviceName||random(seed_1)) \quad (1)$$

$$PK = SM2_G(SK) \quad (2)$$

将随机数与设备唯一标识符deviceName作为SM3算法的输入数据，保证了密钥对的唯一性和不可预测性。

此外，国密SM2算法的密钥派生算法基于椭圆曲线离散对数问题，其密钥对的计算具有单向性，在计算能力有限的情况下，反向求解公钥对应的私钥是不现实的，这使得私钥的安全性得到了保障。并且生成的私钥SK仅由设备生产商保存，防止了私钥的泄露和未授权访问。

密钥获取阶段如图3所示。

3.3　第一次身份认证阶段

在此阶段，公有云作为Broker，负责对设备进行身份认证。设备在获得政府部门审批之后，将获取deviceName、deviceSecret、productKey作为身份信息，从而能够连接至公有云服务器，实现Broker对终端的身份认证。Broker对设备身份信息进行验证的计算过程如下

$$ClientID = deviceName\&productKey||timestamp||Hmacmethod \quad (3)$$

$$username = deviceName\&productKey \quad (4)$$

$$password = Hmac(deviceSecret, deviceName\&productKey\ productKey||deviceName||productKey||timestamp) \quad (5)$$

公有云在接收到身份信息之后，根据数据库中保存设备身份信息进行相同计算过程得到ClientID′、username′、password′，将其与设备上发的ClientID、username、password进行对比，以此判断设备的身份信息是否正确。只有经过公有云身份认证的设备才能接入MQTT网络，才可实现与管理端服务器的通信。

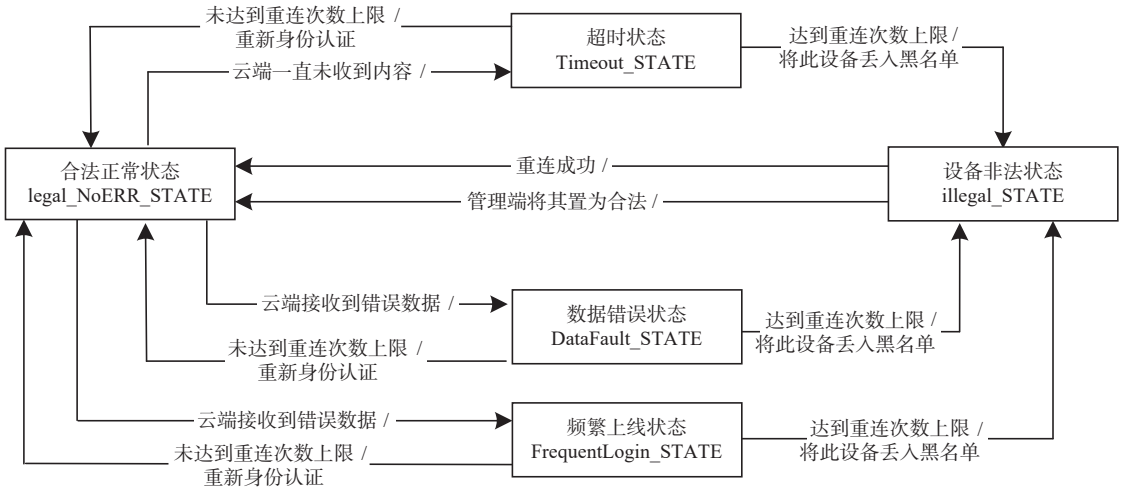

图2　安全状态转换关系

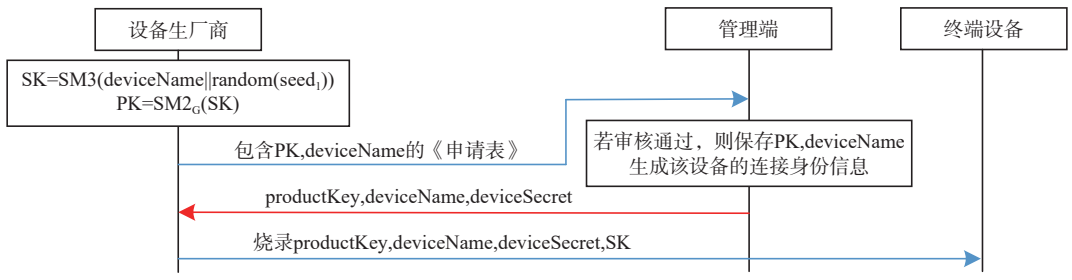

图3　密钥获取阶段

第一次身份认证阶段如图4所示。

3.4 第二次身份认证阶段

此阶段实现管理端对物联网设备的身份认证。由终端设备向管理端发送身份认证请求Authen_Req，管理端接收到Authen_Req之后生成随机码R。设备通过对随机码R进行签名并生成签名值S，管理端再对生成的签名值进行验签，以确认设备的合法性。通过这一系列步骤，管理端能够有效地验证设备的身份，确保只有经过授权的设备才能接入智慧旅游物联网系统。在此阶段中，各变量的计算过程如下

$$R = \mathrm{random}(\mathrm{seed}_2) \tag{6}$$

$$S = \mathrm{SM2}_S(\mathrm{SK}, \mathrm{SM3}(R)) \tag{7}$$

$$\mathrm{SM2}_V(\mathrm{PK}, S, R) = \mathrm{TURE?} \tag{8}$$

$$C\{\mathrm{key}\} = \mathrm{SM2}_E(\mathrm{PK}, \mathrm{key}) \tag{9}$$

$$\mathrm{key} = \mathrm{SM2}_D(\mathrm{SK}, C\{\mathrm{key}\}) \tag{10}$$

在身份认证通过之后，管理端会向设备发送国密SM4算法的密钥key，而终端设备也会返回应答信号以保证密钥获取的成功，此密钥将在数据通信过程中使用。

第二次身份认证阶段如图5所示。

3.5 安全通信阶段

此后进入安全通信阶段，使用SM4算法加密通信的所有数据，包括物联网终端采集的信息以及管理端下发的控制指令等。

对于数据发送端

$$C\{\mathrm{msg}\} = \mathrm{SM4}_E(\mathrm{key}, \mathrm{msg}) \tag{11}$$

对于数据接收端

$$\mathrm{msg} = \mathrm{SM4}_D(\mathrm{key}, C\{\mathrm{msg}\}) \tag{12}$$

安全通信阶段如图6所示。

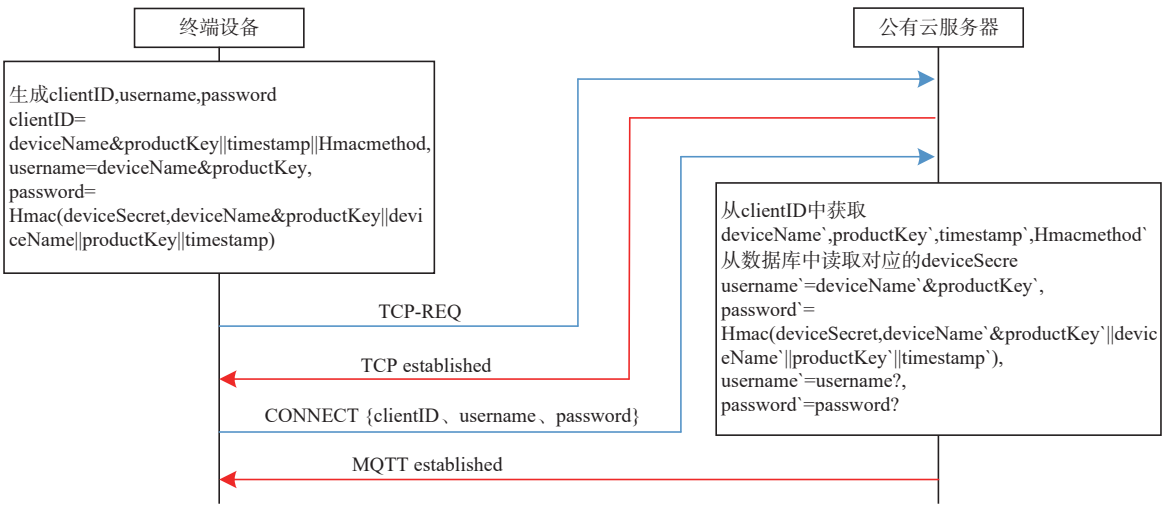

图4　第一次身份认证阶段

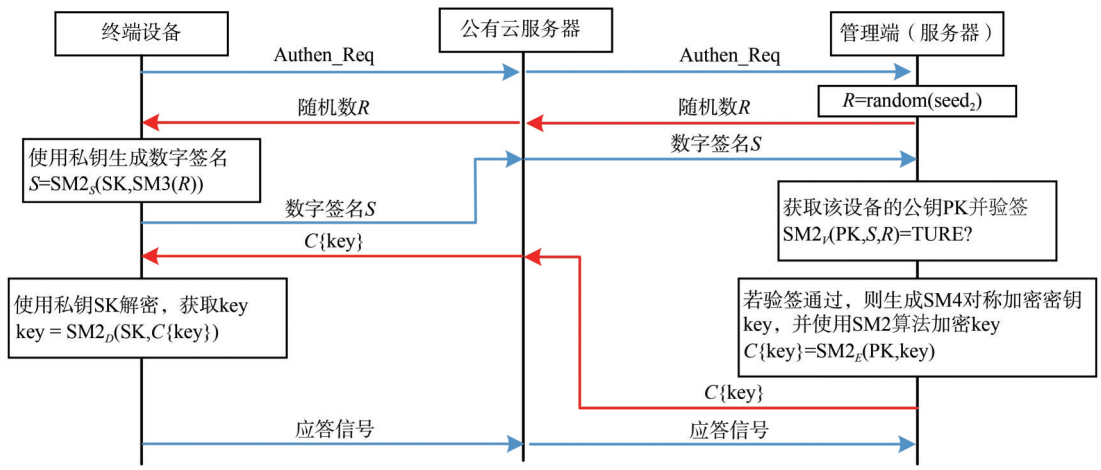

图5　第二次身份认证阶段

3.6　密钥更新阶段

为了实现管理端对设备使用时间的管控，设定每个SM4算法密钥的可用时间，不断更新SM4算法密钥，并使用SM2算法加密密钥，保证密钥的安全更新。

$$C\{\text{key}_{\text{new}}\} = \text{SM2}_E(\text{PK}, \text{key}_{\text{new}}) \tag{13}$$

$$\text{key}_{\text{new}} = \text{SM2}_D(\text{SK}, C\{\text{key}_{\text{new}}\}) \tag{14}$$

密钥更新阶段如图7所示。

4　性能验证

本节探讨并综合评估新认证方法在多个关键维度上的表现，包括安全性、准确性、计算成本以及存储需求。特别地，对能否预防设备生产商违规行为、阻止公有云平台窃取数据进行了分析，确保该认证方法能有效抵御各种潜在的安全威胁。验证了本文方法在旅游物联网环境下的准确性。

4.1　安全性验证

4.1.1　中间人攻击

本文安全算法中通信数据使用国密SM2算法和国密SM4算法加密。对于SM2算法，攻击者若想破解加密数据，需解决离散对数问题，其复杂度为$O(2^{256})$，即在合理的计算资源下，攻击者无法有效解密或篡改消息。对于SM4算法，其密钥长度符合当前已知商用密码设备检测标准（如FIPS 140-2、ISO/IEC 19 790等）的规定长度，可有效防止暴力破解。若数据遭到篡改，则接收方不能成功解密，保证了数据的真实性与完整性，确保了身份认证过程在智慧旅游环境中的鲁棒性和可靠性。

4.1.2　DoS攻击

在智慧旅游场景下，设备可能试图通过频繁认证来耗尽服务器算力。本文安全方法对每个设备一定时间内身份认证次数进行限制，设备无法进行频繁的身份认证，即$N(t) \leqslant N_{\max}$，其中，$N(t)$表示时间t内的认证次数，$N_{\max}$为最大允许认证次数。且所依托的公有云平台具有强大的计算能力、存储能力及处理并发消息的能力，能够有效地预防拒绝服务（DoS, denial of service）攻击。

4.1.3　重放攻击

为了增强身份认证的安全性并避免重放攻击，规定每次通信的JSON报文中必须携带时间戳T_i和信号强度S_i。在S_i良好的情况下，若时间戳不在允许的时间窗口内，则认为是恶意信息。验证要求为$|T_i - T_{\text{rec}}| \leqslant \Delta T$，其中$T_{\text{rec}}$为接收端的当前时间，$\Delta T$为允许的时间窗口，确保了在通信过程中验证数据的时效性和来源的合理性。

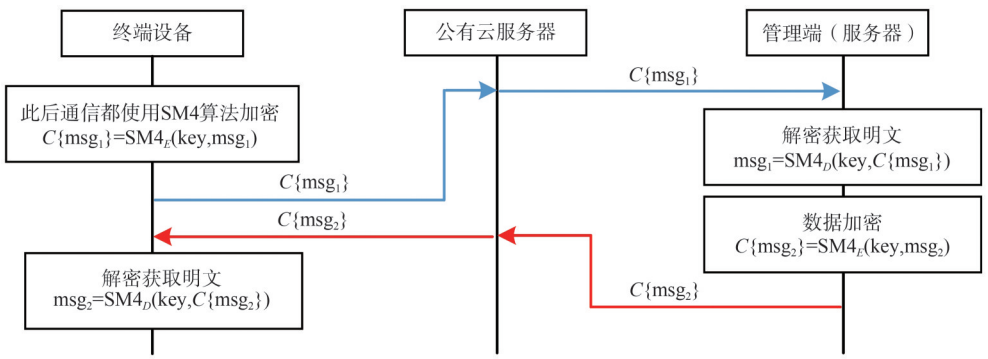

图6　安全通信阶段

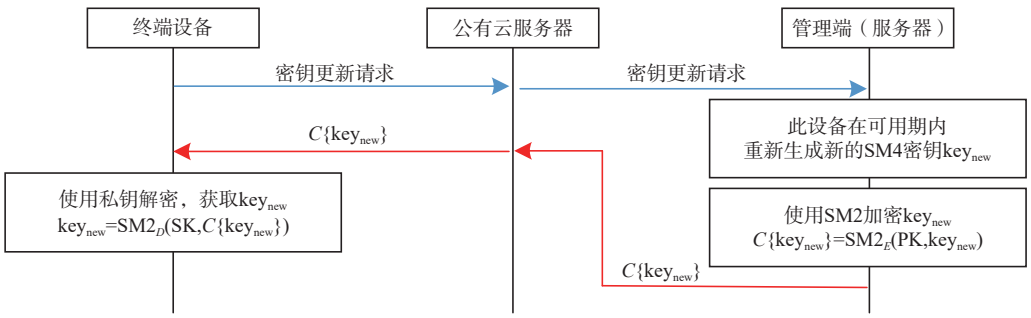

图7　密钥更新阶段

4.1.4　内部攻击

本文安全方法中管理端服务器仅保留公钥PK，确保了私钥SK的安全性，预防了潜在的内部滥用风险。此外，流经公有云服务器的所有数据都是经过加密的密文，Broker无法获得被SM4算法加密后的信息。

4.1.5　身份认证安全性

本文方法所使用的国密SM2算法基于椭圆曲线离散对数问题（ECDLP），其安全性远高于目前公有云服务器所使用的128位密钥长度的HMAC算法[28]，对于密钥猜测攻击具有很强的鲁棒性。身份认证方案对比见表2。

表2　　身份认证方案对比

算法名称	密钥猜测攻击	碰撞攻击	验签主体
HMAC-MD5	2^{128}次	2^{64}次	公有云
HMAC-SHA1	2^{128}次	2^{80}次	公有云
HMAC-SHA256	2^{128}次	2^{128}次	公有云
SM2-ECDSA	2^{256}次	无	管理端

4.1.6　侧信道攻击

在本文身份认证方法中，所提安全状态机制能够有效地限制一定时间内身份认证的次数，攻击者无法通过频繁地发送测试数据来获取物理信息，能够有效地抵挡侧信道攻击，保证了国密SM2算法密钥的安全性。

针对易被攻破的SM4加密密钥，通过动态密钥机制与SM2国密算法结合，保证其算法密钥的安全性。SM4算法的密钥更新过程如下

$$\text{key}_{\text{new}} = \text{Truncare}_{128}(\text{SM3}(\text{key} \oplus T_i)) \tag{15}$$

其中，key_{new}为更新后的密钥，T_i为当前时间戳，$\text{Truncate}_{128}(X)$表示取哈希值X的前128位。

4.1.7　ProVerif工具验证

ProVerif是一个自动化协议分析工具，可用于验证安全协议[29]。本节使用ProVerif验证密钥交互过程的安全性，确保密钥不会被第三方截获。设备运行代码如图8所示。公有云运行代码如图9所示。管理端运行代码如图10所示。

协议查询结果如图11所示。其建模结果表明，本文所提身份认证方法具有保密性、认证性、完备性、不可否认性，且保证了管理端与Broker对设备的认证是稳定的。

```
(* Device进程 *)
let device_process(device_sk: skey, supervisor_pk: pkey) =
    new name: bitstring;
    let device_signature = hmacsha256(name, Secret) in
    event device_signs_name(device_signature);
    out(c, device_signature);

    in(c, x: bitstring);
    let random_encrypted = sm2_decrypt(x, device_sk) in
    let device_signed_random = sm2_sign(random_encrypted, device_sk) in
    event device_signs_random(device_signed_random);
    out(c, device_signed_random);

    in(c, sm4_encrypted: bitstring);
    let sm4_key_received = sm2_decrypt(sm4_encrypted, device_sk) in
    event supervisor_sends_sm4_key(sm4_key_received).
```

图8　设备运行代码

```
(* Broker进程 *)
let broker_process =
    in(c, device_signature: bitstring);
    let identity_verified = hmacsha256(device_signature, Secret) in
    event broker_verifies_identity(identity_verified);
    out(c, device_signature);

    in(c, signed_random: bitstring);
    out(c, signed_random).
```

图9　公有云运行代码

```
(* Supervisor进程 *)
let supervisor_process(supervisor_sk: skey) =
    in(c, device_signature: bitstring);
    new random_number: bitstring;
    let random_encrypted = sm2_encrypt(random_number, my_pk(supervisor_sk)) in
    event supervisor_generates_random(random_encrypted);
    out(c, random_encrypted);

    in(c, signed_random: bitstring);
    let verified_signature = sm2_verify(signed_random, random_number, my_pk(supervisor_sk)) in
    event supervisor_verifies_signature(verified_signature);

    let sm4_encrypted = sm2_encrypt(sm4_key, my_pk(supervisor_sk)) in
    out(c, sm4_encrypted).
```

图10　管理端运行代码

```
-- Query not attacker(Secret[]) in process 1.
Translating the process into Horn clauses...
Completing...
Starting query not attacker(Secret[])
RESULT not attacker(Secret[]) is true.
-- Query not attacker(sm4_key[]) in process 1.
Translating the process into Horn clauses...
Completing...
Starting query not attacker(sm4_key[])
RESULT not attacker(sm4_key[]) is true.

--------------------------------------------------------------
Verification summary:

Query not attacker(Secret[]) is true.

Query not attacker(sm4_key[]) is true.

--------------------------------------------------------------
```

图11　协议查询结果

4.1.8　安全性对比

安全性对比见表3。

4.2　可行性验证

4.2.1　准确性分析

在不同的使用环境中对本文身份认证的准确性进行验证，分别使用4个物联网设备模拟4种不同安全状况的设备，每个设备各进行1 000次身份认证。设备安全状态判断准确次数如图12所示。

将实验结果根据是否移植本文安全方法以及设备是否具有合法性将其分为4类，分别计算其身份认证准确率以验证其性能表现，不同安全状态设备身份认证准确率如图13所示。认证准确率计算式如下

$$\text{认证准确率} = \frac{\text{成功认证次数}}{\text{总身份认证请求次数}} \times 100\% \tag{16}$$

实验结果表明，本文身份认证方法在网络条件较差的情况下基本能够准确判断设备的安全状态，

表3　　安全性对比

对比项	TLS	文献[30]	文献[31]	文献[32]	本文所提方法
身份认证	√	√	√	√	√
数据加密	√	√	√	√	√
前后向安全性	√	√	√	√	√
抵抗侧信道攻击	×	×	×	×	√
抵抗中间人攻击	√	√	√	√	√
抵抗DoS攻击	√	√	√	×	√
抵抗重放攻击	√	×	√	×	√
抵抗内部攻击	×	×	×	√	√
预防生产商违规行为	×	×	×	×	√
预防Broker违规行为	×	×	√	×	√

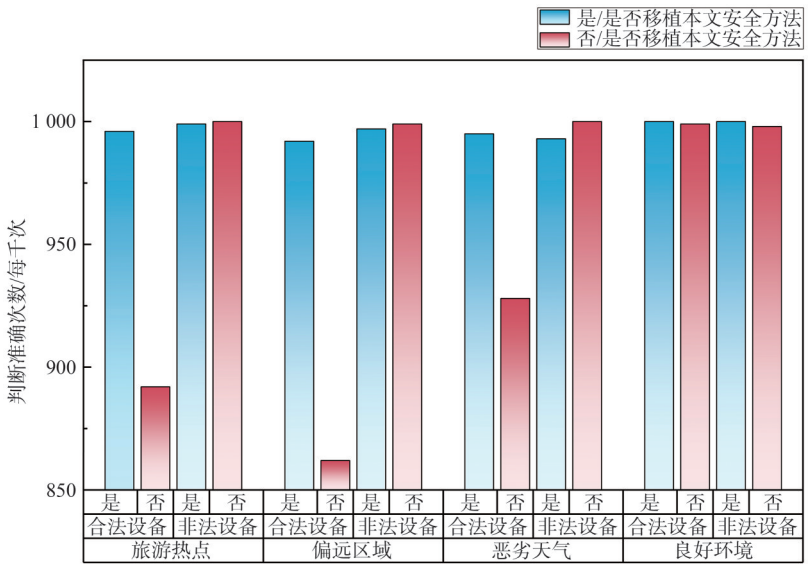

图12　设备安全状态判断准确次数

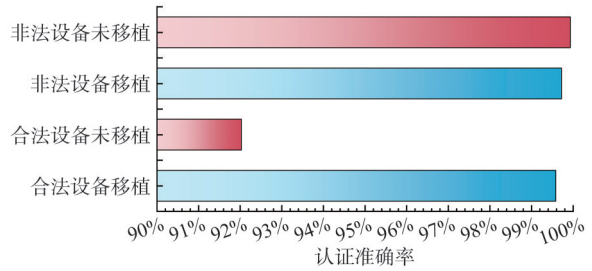

图13　不同安全状态设备身份认证准确率

身份认证准确率达到99.65%，高于未移植本文安全方法的95.98%。虽然对于非法设备身份认证的准确率略有降低，但是在合法设备身份认证误判方面有了显著提高。

4.2.2　计算开销

本节主要讨论本文认证方法与其他相关方案在计算开销方面的比较。不同安全认证方法计算开销对比见表4。其中E表示指数运算，H表示哈希运算，AES表示一次AES算法加解密运算，SM4表示一次SM4算法加解密运算，SM2S表示一次SM2算法签名验签运算，SM2E表示一次SM2算法加解密运算，MAC表示消息认证码运算。

表4　　不同安全认证方法计算开销对比

方法	计算开销
文献[22]	KP-ABE+AES
文献[23]	2AES+4MAC+6E+8H
文献[27]	AES+14E+4H
本文所提方法	MAC+SM2S+SM2E+SM4

文献[22]所使用的KP-ABE算法是一种复杂的公钥加密算法，其开销远超过AES、SM4等对称加密算法。文献[23]和文献[27]所提方法都多次使用了复杂的指数运算，其计算开销较大。与现有MQTT协议下的身份认证方法相比，本文所提方法虽没有突出的计算开销表现，但是在增加了安全性

基础上所增加的计算开销较小。

4.2.3 资源占有

在自主设计制造的智慧旅游设备上进行测试，该设备使用 AIR32F103RPT6 芯片作为主控芯片。此主控芯片负责数据采集与报警功能，物联网设备内存使用情况见表 5。

表 5　物联网设备内存使用情况

内存类型	标准容量/KB	使用情况/KB	占比
RAM	96	34.58	36.02%
FLASH	256	28.71	11.21%

从表 5 中可以看出，设备在执行身份认证任务时，资源消耗量适中，仍有足够的剩余资源来处理其他任务。这表明该认证方法适用于各种智慧旅游物联网设备，能够保证物联网设备应对复杂的多任务处理需求和大规模并发数据通信。

5 结束语

旅游物联网中，如何保证网络安全是一个巨大的挑战。本文提出了一种面向智慧旅游的物联网设备身份认证方法。通过结合国密算法与安全状态机制，解决了旅游环境下可能存在的安全问题，保障了管理端的管理权利。在“一部手机游云南”旅游智慧平台上进行了测试，结果表明，所提安全认证方法解决了设备生产商与 MQTT Broker 服务器存在的安全威胁，对设备合法性的判断准确率更高且硬件资源总消耗均少于 40%，适合智慧旅游环境下资源受限的嵌入式设备。

未来，将考虑为资源受限的设备设计更为高效的加密与认证协议，平衡安全性与性能开销。并考虑使用物理不可克隆函数为每个终端设备生成独有的数字签名，减少硬件开销。此外，将进一步研究如何优化消息队列管理策略来提升信息传输的稳定性和效率。

参考文献：

[1] WANG W, KUMAR N, CHEN J X, et al. Realizing the potential of the Internet of things for smart tourism with 5G and AI[J]. IEEE Network, 2020, 34(6): 295-301.

[2] GUO X D, WANG Y X, MAO J Q, et al. Towards an IoT enabled tourism and visualization review on the relevant literature in recent 10 years[J]. Mobile Networks and Applications, 2022, 27(3): 886-899.

[3] “十四五”文化和旅游发展规划[N]. 中国文化报, 2021-06-03(2).
14th Five-Year Plan cultural and tourism development plan[N]. China Culture News, 2021-06-03(2).

[4] 宋瑞. 中国旅游发展笔谈: “十四五”时期我国旅游业发展展望[J]. 旅游学刊, 2020, 35(6): 1.
SONG R. Discussion forum of China tourism development: prospects of China’s tourism development during the 14th-Five-Year Plan period[J]. Tourism Tribune, 2020, 35(6): 1.

[5] BUTUN I, ÖSTERBERG P, SONG H B. Security of the Internet of things: vulnerabilities, attacks, and countermeasures[J]. IEEE Communications Surveys & Tutorials, 2020, 22(1): 616-644.

[6] SERROR M, HACK S, HENZE M, et al. Challenges and opportunities in securing the industrial Internet of things[J]. IEEE Transactions on Industrial Informatics, 2021, 17(5): 2985-2996.

[7] AIREHROUR D, GUTIERREZ J, RAY S K. Secure routing for Internet of things: a survey[J]. Journal of Network and Computer Applications, 2016, 66: 198-213.

[8] AL-FUQAHA A, GUIZANI M, MOHAMMADI M, et al. Internet of things: a survey on enabling technologies, protocols, and applications[J]. IEEE Communications Surveys & Tutorials, 2015, 17(4): 2347-2376.

[9] FROIZ-MÍGUEZ I, FERNÁNDEZ-CARAMÉS T M, FRAGA-LAMAS P, et al. Design, implementation and practical evaluation of an IoT home automation system for fog computing applications based on MQTT and ZigBee-WiFi sensor nodes[J]. Sensors, 2018, 18(8): 2660.

[10] ALAIZ-MORETON H, AVELEIRA-MATA J, ONDICOL-GARCIA J, et al. Multiclass classification procedure for detecting attacks on MQTT-IoT protocol[J]. Complexity, 2019, 2019(1): 6516253.

[11] JALOUDI S. Communication protocols of an industrial Internet of things environment: a comparative study[J]. Future Internet, 2019, 11(3): 66.

[12] MISHRA B, KERTESZ A. The use of MQTT in M2M and IoT systems: a survey[J]. IEEE Access, 2020, 8: 201071-201086.

[13] VACCARI I, CHIOLA G, AIELLO M, et al. MQTTset, a new dataset for machine learning techniques on MQTT[J]. Sensors, 2020, 20(22): 6578.

[14] MISHRA B, MISHRA B, KERTESZ A. Stress-testing MQTT brokers: a comparative analysis of performance measurements[J]. Energies, 2021, 14(18): 5817.

[15] 余文科, 程媛, 李芳, 等. 物联网技术发展分析与建议[J]. 物联网学报, 2020, 4(4): 105-109.
YU W K, CHENG Y, LI F, et al. Analysis and suggestions on the development of IoT technology[J]. Chinese Journal on Internet of Things, 2020, 4(4): 105-109.

[16] ALI AL-GARADI M, MOHAMED A, AL-ALI A K, et al. A survey of machine and deep learning methods for Internet of things (IoT) security[J]. IEEE Communications Surveys & Tutorials, 2020, 22(3): 1646-1685.

[17] 陈锐, 李春强. 认证加密算法 SM4-GCM 的低成本硬件架构设

计与实现[J]. 物联网学报, 2023, 7(4): 168-179.
CHEN R, LI C Q. Design and implementation of low-cost hardware architecture for authentication encryption algorithm SM4-GCM[J]. Chinese Journal on Internet of Things, 2023, 7(4): 168-179.
[18] ROMAN R, NAJERA P, LOPEZ J. Securing the Internet of Things[J]. Computer, 2011, 44(9): 51-58.
[19] 王曼竹, 李梓琦, 陈翌飞, 等. 车联网中安全认证技术的分析与研究[J]. 物联网学报, 2021, 5(3): 106-114.
WANG M Z, LI Z Q, CHEN Y F, et al. Research and implementation of safety authentication technology in Internet of vehicles[J]. Chinese Journal on Internet of Things, 2021, 5(3): 106-114.
[20] 廖伟, 何乐生, 尹恒, 等. 一种基于Chebyshev混沌映射和CRT的ZigBee网络匿名认证方案[J]. 物联网学报, 2023, 7(4): 101-109.
LIAO W, HE L S, YIN H, et al. A ZigBee network anonymous authentication scheme based on Chebyshev chaotic mapping and CRT[J]. Chinese Journal on Internet of Things, 2023, 7(4): 101-109.
[21] SHAPSOUGH S, ALOUL F, ZUALKERNAN I A. Securing low-resource edge devices for IoT systems[C]//Proceedings of the 2018 International Symposium in Sensing and Instrumentation in IoT Era (ISSI). Piscataway: IEEE Press, 2018: 1-4.
[22] BISNE L, PARMAR M. Composite secure MQTT for Internet of things using ABE and dynamic S-box AES[C]//Proceedings of the 2017 Innovations in Power and Advanced Computing Technologies (i-PACT). Piscataway: IEEE Press, 2017: 1-5.
[23] CALABRETTA M, PECORI R, VELTRI L. A token-based protocol for securing MQTT communications[C]//Proceedings of the 2018 26th International Conference on Software, Telecommunications and Computer Networks (SoftCOM). Piscataway: IEEE Press, 2018: 1-6.
[24] IMGHOURE A, EL-YAHYAOUI A, OMARY F. ECDSA-based certificateless conditional privacy-preserving authentication scheme in Vehicular Ad Hoc network[J]. Vehicular Communications, 2022, 37: 100504.
[25] SANJUAN E B, ABAD CARDIEL I, CERRADA J A, et al. Message queuing telemetry transport (MQTT) security: a cryptographic smart card approach[J]. IEEE Access, 2020, 8: 115051-115062.
[26] LESJAK C, HEIN D, HOFMANN M, et al. Securing smart maintenance services: hardware-security and TLS for MQTT[C]//Proceedings of the 2015 IEEE 13th International Conference on Industrial Informatics (INDIN). Piscataway: IEEE Press, 2015: 1243-1250.
[27] 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[J]. 计算机应用, 2021, 41(5): 1378-1385.
GU Z C, GUO Y B, FANG C. End-to-end security solution for message queue telemetry transport protocol based on proxy re-encryption[J]. Journal of Computer Applications, 2021, 41(5): 1378-1385.
[28] WANG Q, WANG C, REN K, et al. Enabling public auditability and data dynamics for storage security in cloud computing[J]. IEEE Transactions on Parallel and Distributed Systems, 2011, 22(5): 847-859.
[29] ZHENG Y, LIU W Y, GU C Y, et al. PUF-based mutual authentication and key exchange protocol for peer-to-peer IoT applications[J]. IEEE Transactions on Dependable and Secure Computing, 2023, 20(4): 3299-3316.
[30] HUNG Y C，PIN P C. Design and implementation of efficient IoT authentication schemes for MQTT 5.0[J]. Journal of Internet Technology, 2023: 665-674..
[31] PARK C S, NAM H M. Security architecture and protocols for secure MQTT-SN[J]. IEEE Access, 2020, 8: 226422-226436.
[32] MUNSHI A. Improved MQTT secure transmission flags in smart homes[J]. Sensors, 2022, 22(6): 2174.

[作者简介]

杨崇宇（1999– ），男，云南大学信息学院硕士生，主要研究方向为物联网安全、嵌入式系统开发。

何乐生（1977– ），男，博士，云南大学信息学院副教授，主要研究方向为嵌入式系统及物联网应用、微弱信号采集和处理及其在生物电信号和射电天文信号处理等方面的应用。

胡崇辉（1997– ），男，云南大学信息学院硕士生，主要研究方向为嵌入式系统开发、物联网应用。

冯毅（1999– ），男，云南大学信息学院硕士生，主要研究方向为侧信道攻击、轻量级密码安全性分析。

岳远康（1997– ），男，云南大学信息学院硕士生，主要研究方向为侧信道攻击、轻量级密码安全性分析。