面向轻量级物联网设备的高效匿名身份认证协议设计

王振宇[1]，郭阳[1]，李少青[1]，侯申[2]，邓丁[1]
（1. 国防科技大学计算机学院，湖南 长沙 410000；2. 信息工程大学，河南 洛阳 471003）

摘 要：针对现有方案中复杂安全原语不适合资源受限的物联网设备的问题，基于物理不可克隆函数（PUF）为物联网设备设计了一种轻量级高效匿名身份认证协议。通过形式化安全模型和 ProVerif 协议分析工具，证明该协议满足信息传输机密性、完整性、不可追踪和前向/后向保密等 13 种安全属性。与近几年认证方案的性能对比分析表明，该协议在设备端与服务器端的计算开销分别为 0.468 ms 和 0.072 ms，设备存储开销与通信开销分别为 256 bit 和 896 bit，高度适用于资源受限的轻量级物联网设备。

关键词：物理不可克隆函数；轻量级；匿名性；双向认证；物联网
中图分类号：TN918.9
文献标志码：A
DOI: 10.11959/j.issn.1000−436x.2022125

Design of efficient anonymous identity authentication protocol for lightweight IoT devices

WANG Zhenyu[1], GUO Yang[1], LI Shaoqing[1], HOU Shen[2], DENG Ding[1]
1. College of Computer Science and Technology, National University of Defense Technology, Changsha 410000, China
2. Information Engineering University, Luoyang 471003, China

Abstract: Aiming at the problem that complex security primitives in existing schemes were not suitable for resource-constrained IoT devices, a lightweight efficient anonymous identity authentication protocol for IoT devices was designed based on physical unclonable function (PUF). Through the formal security model and ProVerif tool, it was proved that the protocol satisfies 13 security properties such as information confidentiality, integrity, un-traceability, and forward/backward secrecy. Compared with existing relevant protocols, the computing overhead of the protocol on the device side and the server side is 0.468 ms and 0.072 ms respectively, and the device storage and communication overheads are 256 bit and 896 bit respectively, which is highly suitable for lightweight IoT devices with limited resources.

Keywords: physical unclonable function, lightweight, anonymity, mutual authentication, Internet of things

0 引言

物联网（IoT, Internet of things）是一种新兴通信网络，基于互联网实现任何对象的互连互通，包括传感器、标签和智能设备[1-2]。物联网在人们生活和工作等各个应用场景都扮演着重要角色，包括智慧城市、智慧商城、智慧银行、智能农业以及家庭自动化等[3]。然而，物联网中存在大量体积小、硬件处理能力低、资源有限的终端设备，这使以用户为驱动的传统复杂安全认证协议很难在资源受限的物联网中发挥作用[4]。

为了满足物联网信息安全需求，采用轻量级安全协议是解决资源受限设备信息认证的有效方法[5]。大多数物联网设备的身份认证协议通过采用对称密钥算法、Hash 函数等轻量级安全原语，来保证信息传输的机密性、完整性以及不可否认性等安全属性[6-7]。然而，密钥存储在非易失性存储器（NVM, non-volatile memory）容易被侧信道

收稿日期：2021−12−16；**修回日期：**2022−03−15
基金项目：国家自然科学基金资助项目（No.61832018）
Foundation Item: The National Natural Science Foundation of China (No.61832018)

攻击，无法保证关键信息安全性。

物理不可克隆函数（PUF, physical unclonable function）是一种新型轻量级安全原语，不需要使用NVM、不需要为每块芯片量身定制，且对侵入式攻击有较强的灵敏性反应，是平衡认证机制安全性和软硬件开销的合适方法[8]。PUF 利用芯片在制造过程中无法控制的随机工艺偏差来产生器件独有的数字签名，其安全性来自物理无序系统的复杂性和不可预测性[9-10]。PUF 特定的“激励–响应”机制触发，不需要存储且硬件开销小，可以避免传统密钥面临的安全问题，非常适用于物联网设备安全认证[11-12]。

在射频识别技术（RFID, radio frequency identification）、无线传感器网络（WSN, wireless sensor network）等物联网应用环境中，节点资源的严重受限使计算、存储和通信开销较大的传统认证技术无法应用，因此基于 PUF 的轻量级认证技术成为该领域研究的热点[13]。Gope 等[14]考虑到 PUF 自身可靠性问题，基于理想 PUF 与带纠错机制的噪声 PUF 分别提出匿名且不可追踪的安全认证协议。Hossain 等[15]为解决 IoT 系统受设备克隆和重新编程攻击，结合 PUF 提出一种资源高效的双向身份认证协议。但是协议中加入了计算开销大的椭圆加密算法，从而不适合轻量级设备的认证。Akgün 等[16]根据 Vaudenay 安全模型对协议的隐私进行验证，同时采用 PUF 生成的密钥对来保证设备的安全存储。Zhou 等[17]结合 IoT 架构与云服务器提出了一种高效身份隐私验证方案，但是不能防御克隆攻击。Moriyama 等[18]针对攻击者通过物理攻击访问设备的内存试图破坏设备安全和隐私等问题，提出了一种可证明安全私有的 RFID 身份认证协议，以防止设备整个内存的信息泄露。Patil 等[19]结合区块链技术，以 PUF 为计算模型提出了一种隐私保护认证协议，该协议通过 PUF 构造的智能合约区块链来确保用户的隐私性和机密性。

虽然上述轻量级安全协议能解决部分物联网设备的安全传输问题，但是这些协议仍存在效率低、安全性差、功能不完善等缺点。因此，本文基于 PUF 为物联网设备提出一种高效匿名轻量级身份互认证协议，该协议可以确保关键的安全属性，包括匿名性、可用性、机密性和前向/后向保密性等。此外，该协议还可以确保攻击者不能获取 PUF 的任何激励–响应对（CRP, challenge response pair）信息，从而防止攻击者对设备中的 PUF 进行建模攻击。形式化安全模型和 ProVerif 协议分析工具证明该协议可以防御窃听、克隆、重放等多种安全威胁。相比于其他近期的安全协议，该协议具备更低的计算开销、通信开销和存储开销，适用于资源受限的轻量级物联网设备。

1 物联网设备协议认证的安全模型

物联网系统的整体结构由设备、网关、网络与服务器构成。本文构建的物联网设备认证协议的安全模型如图 1 所示。物联网设备认证协议的攻防体系由系统结构、安全威胁与安全需求三方面构成。

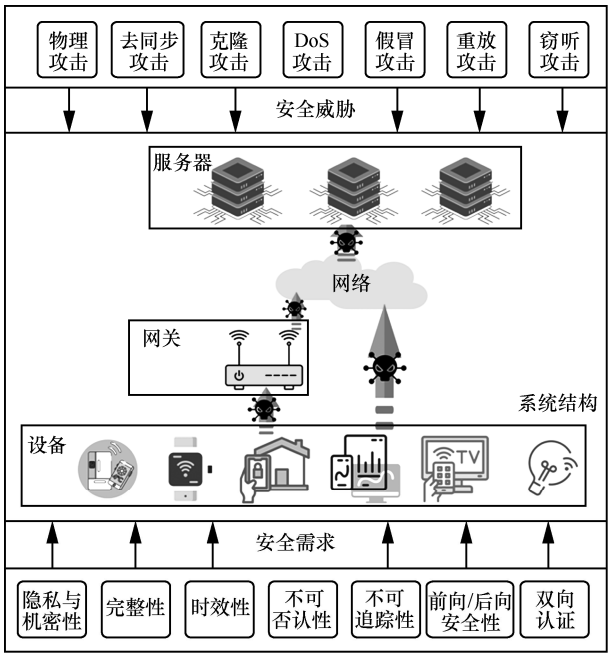

图 1 物联网设备认证协议的安全模型

物联网设备认证协议的攻防体系可表示为

物联网设备认证协议的攻防体系=*F*(系统结构、安全威胁、安全需求)

系统结构=*f*(设备、网关、网络、服务器)

安全威胁=*f*(物理、去同步、克隆、DoS、假冒、重放、窃听等攻击)

安全需求=*f*(隐私与机密性、完整性、时效性、不可否认性、不可追踪性、前向/后向安全性、双向认证)

1.1 物联网设备安全威胁

由于物联网设备和服务器之间的信道传输不安全，设备在认证过程中会面临各种安全威胁。

1) 物理攻击。攻击者通过物理攻击访问设备的内存，以获得密钥等安全信息。

2) 去同步攻击。攻击者破坏设备与服务器的某次通信过程，使共享的信息经过本次交互后不再一致，导致双方不能成功进行后续认证。

3) 克隆攻击。攻击者非法获取电子设备信息后，通过复制或假冒的方式替代被攻击的设备。

4) 假冒攻击。在设备通信过程中，攻击者假冒设备（或服务器）身份，引入新的信息，删除原有信息来与服务器（或设备）进行交互，通过对方的验证，从而达到冒充合法设备（或服务器）的目的。

5) 重放攻击。攻击者通过窃听并收集设备与服务器之间的通信信息，然后在某个时间重放这些消息给另一方，来通过消息接收方的验证。

6) 窃听攻击。攻击者通过窃听服务器与设备之间的通信信道，非法获取通信的秘密信息。

1.2 物联网设备安全需求

为了保证物联网设备与服务器的安全认证，协议需满足以下安全需求。

1) 隐私与机密性。协议既要保护设备通信数据的机密性，同时也要保护用户身份、位置等敏感信息的隐私性。

2) 完整性。为了确保数据的完整性，协议必须有能力检测未授权方的数据操作，防止对物联网系统中的通信数据进行修改。

3) 时效性。包括物联网在内的所有信息系统，必须及时和正确地反馈合法用户的数据请求。安全协议要确保设备在遭受攻击的情况下，仍然可以完成数据的正常获取和安全传输。

4) 不可追踪性。为了更好地保护用户的隐私，协议应该支持不可追踪性，即攻击者无法从截获的消息中追踪用户的行为。

5) 前向/后向安全性。前向安全性意味着前一会话密钥的信息不能帮助攻击者获得后面认证的会话密钥，从而保证未来通信的安全性。同时，协议还应提供后向安全性，即一个安全信道的泄露不会损害先前信道的安全性。

6) 双向认证。为了减少通信开销，协议不提供在线注册中心，即不需要在线注册中心来实现相互认证。

2 数学理论知识

本节介绍用于保护信息隐私性和完整性的安全原语，包括物理不可克隆函数、PUF 建模攻击假设和单向 Hash 函数假设。

2.1 物理不可克隆函数

原生 PUF 可以形式化地定义为将一个有限空间 $C=\{0,1\}^{l_1}$ 中选出的激励 c 映射为一个有限空间 $R=\{0,1\}^{l_2}$ 中输出 r 的物理系统，其中映射的随机属性 k 取决于设备制造过程中的偏差

$$\mathrm{PUF}_k : C \to R$$

由激励 $c \in C$ 生成响应 $r \in R$ 的操作也可以表示为

$$r \leftarrow \mathrm{PUF}_k(C)$$

2.2 PUF 建模攻击假设

2.1 节定义的 PUF 为 $\{0,1\}^{l_1} \to \{0,1\}^{l_2}$，其中 PUF 输入激励 c 的长度为 l_1，而得到输出响应 r 的长度为 l_2。PUF 的建模攻击可以通过下面的激励−响应模型来确定，该游戏包括 2 个阶段。

阶段 1　攻击者 $\mathcal{A}$ 获取 PUF 的大量 CRP 子集 $[(C_1,R_1),(C_2,R_2),\cdots,(C_i,R_i)]$，然后对获得的 CRP 子集进行学习、训练并建模为 $\mathrm{PUF}_{\mathcal{A}}()$。

挑战　攻击者 $\mathcal{A}$ 随机选择一个不在以前查询序列中的激励 C_x。

阶段 2　攻击者 $\mathcal{A}$ 用建立的 $\mathrm{PUF}_{\mathcal{A}}()$ 模型查询激励 C_x。

响应　攻击者 $\mathcal{A}$ 猜测 PUF 的输出响应 R_x'，并且与真实的 PUF 响应 $R_x=\mathrm{PUF}(C_x)$ 进行比对。如果 $R_x'=R_x$，则表明攻击者 $\mathcal{A}$ 建模攻击成功。

2.3 单向 Hash 函数假设

定义 Hash 函数为 $\{0,1\}^x \to \{0,1\}^h$，其函数输入为可变长 x，输出为固定长串 h，h 被称为输入 x 的 Hash 值，记为

$$h=H(x)$$

安全 Hash 函数需满足以下几个基本条件。

1) 输入 x 可以是任意长度，输出数据串为固定长度。

2) 反向计算困难，即给出一个 Hash 函数值 h，很难找出特定输入 x，使 $h=H(x)$。

3) 对任何给定的分组 x，找到满足 $x \neq y$ 且 $H(x)=H(y)$ 的 y 在计算上是不可行的，满足抗弱碰撞性；同时，找到任何满足 $H(x)=H(y)$ 的 (x,y) 在计算上是不可行的，满足抗强碰撞性。

3 协议机制

本文基于 PUF 的物联网设备提出高效匿名的身份认证协议。该协议可分为三部分，包括设备注册阶段、双向认证阶段、固件更新阶段。其相关符号说明如表 1 所示。

表 1 符号说明

符号	含义
PID_D^i	设备 D 在第 i 轮认证的伪随机身份
ID_D	设备 D 的真实身份
$\mathrm{ID}_{\mathrm{TM}}$	设备 D 临时认证身份
$\mathrm{CRP}(C_i,R_i)$	第 i 轮认证激励–响应对
$\mathrm{PUF}_D(\cdot)$	设备 D 中的物理不可克隆函数
$H(\cdot)$	单向 Hash 函数
$\oplus$	异或操作
$\|$	级联操作
a_i,b_i	伪随机数

3.1 协议的基本假设

本文基于 PUF 提出了隐私匿名保护的轻量级认证协议，其基本假设如下。

1) 每个物联网设备都嵌入了 PUF，任何篡改 PUF 的攻击都会更改设备的输入与输出，使设备无法工作。

2) 嵌入设备中的 PUF 满足可靠性、不可预测性、不可克隆性等性能指标，不需要采用纠错机制来保证 PUF 的安全属性，即理想的安全 PUF。

3) 攻击者可以对特定的强 PUF 结构进行数学建模，并通过机器学习算法对其 CRP 进行预测。攻击者可以通过物理攻击等手段访问设备端中存储的信息。

4) 假定设备的注册阶段在安全通信环境中进行，数据传输绝对安全。在设备认证与固件更新阶段，攻击者可以对传输的消息进行窃听、伪造或篡改等攻击。

3.2 设备注册阶段

为了保证信息安全传输，设备需要先在服务器进行注册，其注册阶段步骤如图 2 所示，具体说明如下。

Step1 $S\rightarrow D:\{C_i,\mathrm{ID}_D,C_{\mathrm{TM}},\mathrm{ID}_{\mathrm{TM}}\}$

在设备注册阶段，服务器首先给设备分配一个身份 ID_D，并随机产生一个激励 C_i。为防止设备在认证过程中遭受去同步攻击和 DoS 攻击，服务器会随机产生临时应急激励 C_{TM} 和临时认证身份 $\mathrm{ID}_{\mathrm{TM}}$，并将信息 R_1 发送给设备端。

Step2 $D\rightarrow S:\{R_i,\mathrm{PID}_D^i,R_{\mathrm{TM}}\}$

1) 当设备接收到 ID_D 时，为了更好地隐藏身份，生成一个伪随机身份 PID_D^i。

2) 设备将接收到的 (C_i,C_{TM}) 通过 PUF 计算出响应 (R_i,R_{TM})，然后将 PID_D^i 和 $\mathrm{ID}_{\mathrm{TM}}$ 存储于设备中。设备将信息 R_2 发送给服务器。

Step3 服务器存储信息

服务器将 CRP 子集 (C_i,R_i)、$(C_{\mathrm{TM}},R_{\mathrm{TM}})$ 和设备身份信息 $(\mathrm{ID}_D^i,\mathrm{PID}_D^i,\mathrm{ID}_{\mathrm{TM}})$ 存储到数据库。

3.3 双向认证阶段

双向认证阶段实现物联网设备与服务器之间的相互认证，步骤如图 3 所示，具体说明如下。

Step1 $D\rightarrow S:\{\mathrm{PID}_D^i,A_i\}$

设备端生成一个随机数 a_i，并选择第 i 轮认证的伪随机身份 PID_D^i 以及临时身份 $\mathrm{ID}_{\mathrm{TM}}$。设备计算 $A_i=a_i\oplus\mathrm{PID}_D^i\oplus\mathrm{ID}_{\mathrm{TM}}$，并发送消息 M_1 到服务器 S。

Step2 $S\rightarrow D:\{T_S,C_{Ni},C_{Di},R_D^i,\mathrm{Auth}_D\}$

1) 服务器对设备发送的请求信息进行认证，如果服务器不能搜索到伪随机身份 PID_D^i，则拒绝设备认证。同时，服务器启动紧急认证模式，使用临时认证身份 $\mathrm{ID}_{\mathrm{TM}}$ 重新对服务器进行认证请求。在应急模式下，服务器将选择临时应急激励 $(C_{\mathrm{TM}},R_{\mathrm{TM}})$，并且提供新的伪随机的身份 PID_D^i 给设备进行下一轮认证。临时认证身份 $\mathrm{ID}_{\mathrm{TM}}$ 和临时应急激励 $(C_{\mathrm{TM}},R_{\mathrm{TM}})$ 被使用之后，服务器需要将这些信息从数据库中删除。该机制可以有效抵御 DoS 攻击和去同步攻击，从而保证服务器的匿名性和不可追踪性。

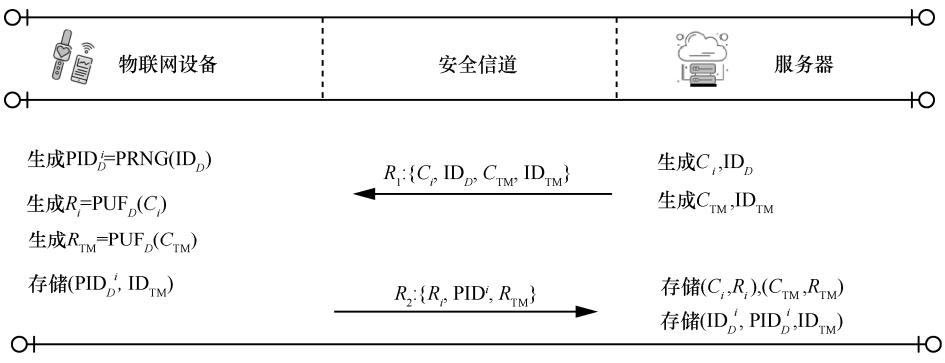

图 2 设备注册阶段步骤

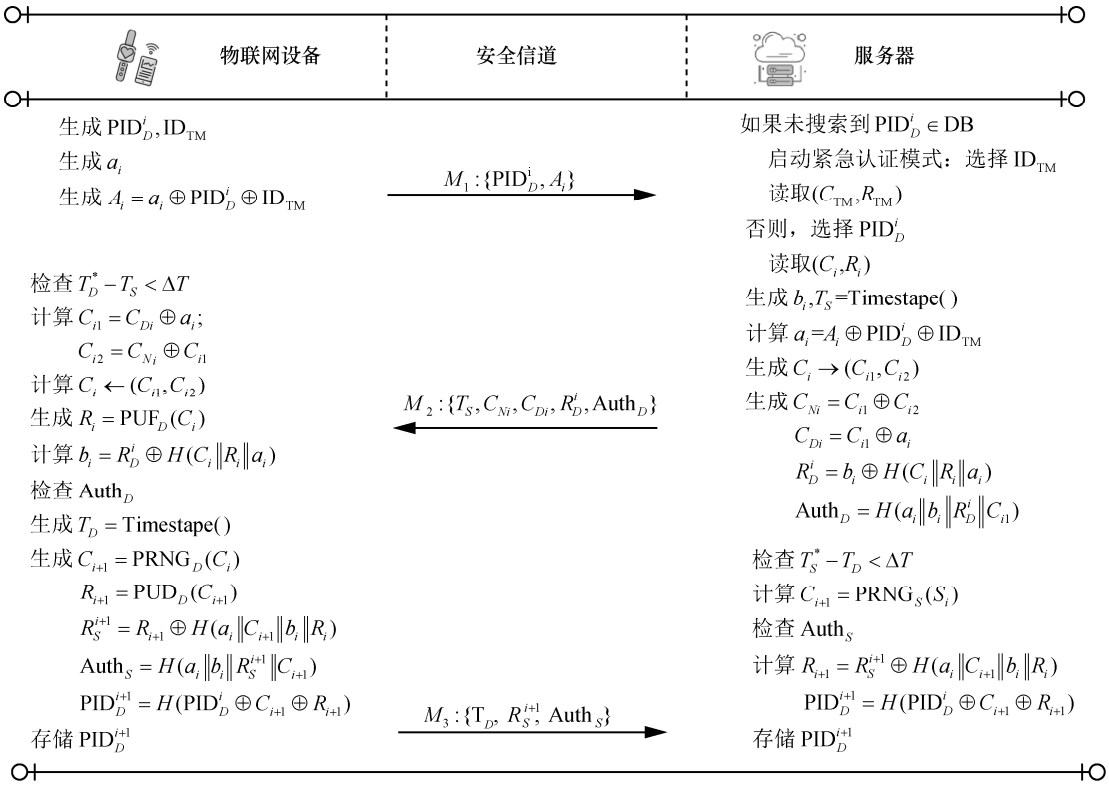

图 3　认证阶段步骤

2) 如果服务器成功识别伪随机身份 PID_D^i，则记录该时刻的时间值 T_S，同时从存储器中读取 (C_i, R_i) 进行下一步认证。服务器从传输信息 A_i 解析出随机值 a_i，并且产生一个随机值 b_i。

3) 服务器将选取的激励 C_i 采用按位选取方式分成两段 C_{i1} 与 C_{i2}，然后通过解析出的随机数 a_i 计算出 $C_{Ni} = C_{i1} \oplus C_{i2}$ 与 $C_{Di} = C_{i1} \oplus a_i$。服务器计算出 R_D^i 与 $Auth_D$，并且将消息 M_2 发送给设备。

Step3 $D \rightarrow S : \{T_D, R_S^{i+1}, Auth_S\}$

1) 设备首先检查传输时延是否在允许的时间间隔 ΔT 内，即 $T_D^* - T_S < \Delta T$。ΔT 是服务器与设备完成多次认证的平均时间阈值（经验值），防止消息被重放。设备记录该时刻的时间值 T_D。

2) 设备从接收的信息中恢复 PUF 的激励 C_{i1} 与 C_{i2}，然后按位选取方式拼接成 C_i。设备输入激励 C_i 到 PUF 中得到响应 R_i，并解析出随机数 b_i。设备将计算值 $Auth_D$ 与服务器发送值 $Auth_S$ 进行比对，若相等，则设备对服务器的认证成功；否则，设备对服务器的认证失败。

3) 设备将激励 C_i 作为种子来产生新的激励 C_{i+1}，并且计算新的响应值 $C_{i+1} = PRNG_D(C_i)$ 与 $R_{i+1} = PUF_D(C_{i+1})$。同时，设备计算 R_S^{i+1} 和认证信息 $Auth_S$，并且将信息 M_3 发送给服务器。

Step4 服务器认证信息

服务器首先检查传输时延是否在允许时间间隔 ΔT 内，即 $T_S^* - T_D < \Delta T$。服务器将 $Auth_S$ 与设备发送的 $Auth_D$ 进行比对，若相等，则服务器对设备的认证成功，进入固件更新阶段；若不相等，则服务器对设备的认证失败。

3.4　固件更新阶段

当设备完成认证后，利用 Hash 函数更新伪随机身份 PID_D^{i+1}，即 $PID_D^{i+1} = H(PID_D^i \oplus C_{i+1} \oplus R_{i+1})$，并且将新的伪随机认证身份 PID_D^{i+1} 存储到设备中。服务器先更新 CRP(C_{i+1}, R_{i+1})，然后计算并存储新一轮 PID_D^{i+1} 到服务器中。

4　形式化安全分析

本节通过随机预言模型和形式化工具 Proverif 对协议进行安全性分析。

4.1　形式化安全模型

为了证明本文所提协议的安全性，在随机预言模型下实现用户相互认证。假设存在一个多项式时

间攻击者 $\mathcal{A}$，它可以访问设备和服务器传输信道之间的所有消息，以及所有公共参数。在该协议中，符号 $\prod_P^i$ 表示协议参与者 P 的第 i 次会话，参与者 P 包括设备 D 和服务器 S，分别表示为 $\prod_D^i$ 和 $\prod_S^i$。攻击者 $\mathcal{A}$ 可以在多项式时间内进行以下查询，来破坏协议的安全性。

1) Send($\prod_S^i, p_1, m_1$)。该查询序列表示攻击者 $\mathcal{A}$ 假冒合格设备 D，发送消息 p_1 给服务器，并且接收服务器返回消息 m_1。

2) Send($\prod_D^i, p_2, m_2$)。该查询序列表示攻击者 $\mathcal{A}$ 假冒合格服务器，发送消息 p_2 给设备，并且接收设备返回消息 m_2。

3) Execute($\prod_S^i, \prod_D^i, m$)。该查询模拟攻击者 $\mathcal{A}$ 监听设备 $\prod_D^i$ 和服务器 $\prod_S^i$ 之间通信信道的能力，并拦截信道上会话信息 m。

4) Reveal($\mathcal{A}$)。该查询可模拟对手发起 DoS 攻击。在该模式下，允许攻击者 $\mathcal{A}$ 阻塞协议通信并且中断服务器和设备之间的同步。

5) Corrupt($\prod_D^i, K$)。该查询序列模拟攻击者 $\mathcal{A}$ 破坏设备的能力，并访问内存中的机密信息。

6) Hash(M)。攻击者 $\mathcal{A}$ 使用消息 M 进行 Hash 查询，查询 $\prod_P^i$ 中 Hash 列表 L_h 中是否存在消息 M。

7) Query(CRP)。该序列是查询认证过程中 PUF 生成的 CRP，若查询成功，则返回消息 (C_i, R_i) 到列表 L_C；否则，将生成一个随机值 $C_i, R_i \in \{0,1\}^n$ 重放给 $\mathcal{A}$，并将元组 (C_i, R_i) 存放在列表 L_C。

8) Test($\prod_P^i$)。该查询用于衡量会话密钥信息的语义安全性。攻击者 $\mathcal{A}$ 可以向 $\prod_P^i$ 发送一个测试查询，如果 $c=1$，则将实际会话密钥信息返回给 $\mathcal{A}$；如果 $c=0$，则返回一个随机位字符串。

语义安全性概念。在测试中，需要区分协议中的真实会话信息和随机信息。允许攻击者 $\mathcal{A}$ 对设备或服务器执行多次测试查询，最后 $\mathcal{A}$ 输出一个猜测位 c'。将 Pr[Succ] 定义为 $\mathcal{A}$ 测试成功的概率，则 $\mathcal{A}$ 打破身份验证和密钥协商协议 P 的语义安全性方面的优势为 $\mathrm{Adv}_p(\mathcal{A}) = |2\Pr[\mathrm{Succ}] - 1|$。如果 $\mathrm{Adv}_p(\mathcal{A})$ 可以忽略不计（对于任何足够小的 $\varepsilon > 0$，有 $\mathrm{Adv}_p(\mathcal{A}) < \varepsilon$），则证明所提协议安全。

4.2 形式化安全证明

引理 1 在所提协议中，攻击者 $\mathcal{A}$ 调用 Send()、Execute()、Corrupt() 等查询序列不能获得设备中任何机密信息。

证明 在协议的认证过程中，$\mathcal{A}$ 调用 Send()、Execute() 序列得到消息 $\mathrm{PID}_D^i, R_D^i, R_S^{i+1}$。在协议的认证过程中，PUF 产生的 $\mathrm{CRP}(C_i, R_i), (C_{i+1}, R_{i+1})$ 是协议的安全密钥参数，并且这些 CRP 通过 Hash 函数进行保护。由于 Hash 函数的单向性，攻击者 $\mathcal{A}$ 窃听了消息 R_D^i, R_S^{i+1}，仍然不能获得 $\mathrm{CRP}(C_i, R_i), (C_{i+1}, R_{i+1})$。在所提协议中，设备不需要存储任何安全隐私信息。攻击者 $\mathcal{A}$ 调用 Corrupt() 序列从设备中获取存储信息，只能是伪随机识别身份 PID_D^i。在协议中只具备 PID_D^i 信息，无法通过服务器的认证，并且由于 PUF 自身的特性，任何破坏 PUF 的尝试都会改变其 CRP 子集，最终将使设备变得无用。

证毕。

引理 2 在所提协议中，攻击者 $\mathcal{A}$ 不能获取 PUF 的 $\mathrm{CRP}(C_i, R_i)$。

证明 在协议认证过程中，PUF 所产生的 $\mathrm{CRP}(C_i, R_i)$ 是安全认证过程中的密钥信息。协议认证的起始激励 C_i 不是由设备端产生的，而是从服务器的数据库中读取的，防止攻击者 $\mathcal{A}$ 调用 Corrupt() 指令来读取激励 C_i。为防止 PUF 的 CRP 被建模，服务器将激励 C_i 采用按位选取方式分成 C_{i1} 与 C_{i2}，解析出随机数 a_i 并计算 C_{Ni} 与 C_{Di}。随机数 a_i 被加密为 $A_i = a_i \oplus \mathrm{PID}_D^i \oplus \mathrm{ID}_{\mathrm{TM}}$，从而 $\mathcal{A}$ 不能解析出随机数 a_i 和信息 C_{i1} 与 C_{i2}。同样，由引理 1 可知，协议通过 Hash 函数进一步将安全参数 $\mathrm{CRP}(C_i, R_i), (C_{i+1}, R_{i+1})$ 的信息保存到 R_D^i, R_S^{i+1}。因此，攻击者不能获取 PUF 的 $\mathrm{CRP}(C_i, R_i)$ 信息。

证毕。

引理 3 攻击者 $\mathcal{A}$ 调用 Hash() 查询序列不能获取协议中的重要信息。

证明 攻击者 $\mathcal{A}$ 获取协议中 Hash 函数的参数，调用 Hash() 查询序列来实现对协议的攻击。

$H_1(C_i \| R_i \| a_i) - \mathrm{Query}$。攻击者 $\mathcal{A}$ 保留一个列表 L_{h1}，其形式为 (C_i, R_i, a_i, h_1^i)。列表 L_{h1} 初始值为空，当攻击者接收到消息 (C_i, R_i, a_i) 进行 Hash() 查询时，攻击者检查元组是否存在于列表 L_{h1} 中。若该元组在列表中，则返回消息 $h_1^i = H_1(C_i \| R_i \| a_i)$；否则，随机选择 $h_1^i \in (0,1)^n$ 并且将 (C_i, R_i, a_i, h_1^i) 插入列表

L_{h1}。由引理 1 与引理 2 可知，攻击者 $\mathcal{A}$ 不能获取 PUF 参数 $(C_i,R_i),(C_{i+1},R_{i+1})$ 和随机数 a_i，从而 $\mathcal{A}$ 会随机选择 $h_1^i \in (0,1)^n$ 返回到列表 L_{h1} 中。

调用 Hash() 序列的其他函数与之类似。因此，攻击者调用 Hash() 查询序列不能得到协议中的重要信息。

证毕。

引理 4　所提协议可以防御 DoS 攻击和去同步攻击。

证明　在所提协议中，为了防御 DoS 攻击和去同步攻击，在注册阶段加入临时认证身份 $\mathrm{ID_{TM}}$ 与临时应急 CRP $(C_{\mathrm{TM}},R_{\mathrm{TM}})$。假设攻击者 $\mathcal{A}$ 在认证过程中调用了 Reveal($\mathcal{A}$) 序列，从而导致服务器不能接收消息 $A_3:\{T_D,R_S^{i+1},\mathrm{Auth}_S\}$，不能更新下一次认证的 CRP (C_{i+1},R_{i+1})。为了防止服务器的资源被攻击者耗尽，则服务器需要设备端使用临时认证身份 $\mathrm{ID_{TM}}$ 重新进行认证请求。一旦服务器接收到的消息为 $\mathrm{ID_{TM}}$（代替 PID_D^i），服务器将发送临时激励 CRP $(C_{\mathrm{TM}},R_{\mathrm{TM}})$，并且继续这一轮的协议认证。通过该机制可以防御 DoS 攻击和去同步攻击。

证毕。

引理 5　所提协议可以防御物理攻击和克隆攻击。

证明　攻击者 $\mathcal{A}$ 可以通过调用 Corrupt() 指令来进行物理攻击，获取存储在设备中的敏感信息。因此，安全的认证协议不应在内存中存储任何秘密。然而，现在大多数密码算法认证协议都是以密钥的形式存储一个或多个机密值在内存当中，从而会导致密钥泄露。根据引理 1 和引理 2，本文所提协议在设备端存储任何密钥信息。另外，PUF 利用芯片在制造过程中无法控制的随机工艺偏差，从而使具有相同光刻掩模的制造商无法对这些签名进行物理上的复制。PUF 自身所具备的不可克隆特性保证设备可以防御克隆攻击。根据 3.1 节内容，所提协议要求每个设备都嵌入了 PUF。因此，所提协议可以防御物理攻击和克隆攻击。

证毕。

定理 1　所提协议可以防御机器学习。

证明　虽然 PUF 的应用成功解决了密钥存储在片上 NVM 中的不安全性，但是 PUF 会同时产生大量的 CRP 子集。攻击者 $\mathcal{A}$ 通过获取 PUF 的大量 CRP，利用已知的 CRP 子集模拟并训练模型，进而来预测未知的 CRP。在该协议机制中，攻击者 $\mathcal{A}$ 可以通过下面的训练模型来获得 CRP 子集进行攻击。

1) 选择一个有效的认证环境，服务器 S 与设备 D 可以在该环境中通信。

2) 攻击者 $\mathcal{A}$ 可以多次调用 Send()、Execute()、Hash()、Corrupt() 等操作。完成这些操作后，将消息返回给攻击者　。

3) 攻击者调用 Query(CRP) 命令来获取 PUF 的 CRP。

4) 如果攻击者 $\mathcal{A}$ 可以猜测 PUF 的正确 CRP，则认为设备中的 PUF 能被建模攻击，攻击者赢得此次游戏，否则失败。

在该游戏模型中，如果攻击者 $\mathcal{A}$ 能通过 n 轮认证之后获取 PUF 的大量 CRP，则认为攻击者能对 PUF 进行建模攻击。由引理 1 可知，攻击者 $\mathcal{A}$ 调用 Send()、Execute()、Hash() 等操作都只能获取伪随机身份 PID_D^i，并且所提协议没有存储任何 CRP 信息在设备中。因此，攻击者 $\mathcal{A}$ 不能获取到任何 CRP 信息。

攻击者在认证过程中调用 Send()、Execute() 操作，可以获得有关 PUF 的 CRP 子集的信息，即 C_{Ni},C_{Di},R_D^i。由引理 2 可知，激励 C_i 采用按位选取方式将其分成 C_{i1} 与 C_{i2}，并通过 C_{Ni} 与 C_{Di} 加密保护，从而不能获取 PUF 的激励信息 C_i。同时由引理 3 可知，调用 $H_1(C_i \| R_i \| a_i)-\mathrm{Query}$ 序列，列表 L_h 中不存在攻击者收到的消息 (C_i,R_i,a_i)，从而不能解析出 PUF 的 CRP (C_i,R_i)。

攻击者在认证过程中获取消息 R_S^{i+1}，但激励 C_{i+1} 是以上一次激励 C_i 为种子通过伪随机数发生器 PRNG 产生的。因为激励 C_{i+1} 不会被传输，也没有存储于设备，所以攻击者不能获取更新后激励 C_{i+1}。最后，因为 Hash 函数具有单向性，所以由消息 $\mathrm{Auth}_S=H(a_i \| b_i \| R_S^{i+1} \| C_{i+1})$ 不能得到更新后的响应 R_{i+1}。攻击者 $\mathcal{A}$ 不能获取 PUF 的任何 CRP 信息，不能对设备中的 PUF 进行建模攻击。因此，该协议机制可以防御机器建模攻击。

证毕。

定理 2　在所提协议中，设备的每轮认证是不可追踪的。

证明　在物联网设备认证过程中，如果攻击者无法对服务器的两次有效身份认证进行关联，则认

为设备是无法追踪的。为证明该协议具备不可追踪性，通过调用安全模型的指令来构建一个训练模型。假设攻击者 $\mathcal{A}$ 是多项式时间算法，其规则如下。

1) 选择 2 个有效的设备 D_1 和 D_2，以及一个可以正常通信的服务器 S。

2) 攻击者在服务器 S 与设备 D_1 和 D_2 认证过程中可以多次调用以下操作：Send()、Execute()、Hash()、Corrupt()。在完成这些操作之后，将消息返回给攻击者 。

3) 攻击者在 2 个设备中随机挑选一个设备 $D_i(i=0,1)$，并在服务器 S 与设备 D_i 调用以下操作：Send()、Execute()、Hash()、Corrupt()，从而学习到更多信息。

4) 攻击者通过训练的结果来猜测设备 D'。若 $D'=D_i$，则攻击者赢得此次游戏，否则失败。

在这次游戏模型中，通过语义安全性的概念，攻击者能成功猜测出设备的概率为 $\mathrm{Adv}_p(\mathcal{A})=|2\Pr[D'=D_i]-1|$。如果 $\Pr[D'=D_i]=\frac{1}{2}$，则 $\mathrm{Adv}_p(\mathcal{A})=0$，攻击者 $\mathcal{A}$ 赢得比赛的优势为零，表明该协议中设备是不可追踪的。

根据引理 1 与引理 2 可知，攻击者在第 i 轮认证过程中，无法获取 PUF 产生的 CRP (C_i,R_i) 与 (C_{i+1},R_{i+1})，从而不能计算第 i+1 轮的伪随机身份信息 PID_D^{i+1}。并且攻击者 $\mathcal{A}$ 不能将 PID_D^i 与 PID_D^{i+1} 进行关联，只能随机猜测设备 D_i，此时成功的概率 $\Pr[D'=D_i]=\frac{1}{2}$。则攻击者 $\mathcal{A}$ 成功猜测出设备的概率 $|\mathrm{Adv}_p(\mathcal{A})=|2\Pr[D'=D_i]-1|=0$。因此，所提协议机制具备不可追踪性。

证毕。

定理 3 所提协议在保证设备具有向后不可追踪性的前提下，仍具有向前的保密性。

证明 定理 2 证明了协议具有不可追踪性，在定理 2 的游戏模型基础上，重新构建新的游戏模型。则攻击者模型如下。

1) 选择 2 个有效的设备 D_1 和 D_2，以及一个可以正常通信的服务器 S。

2) 攻击者在服务器 S 与设备 D_1 和 D_2 可以调用以下操作：Send()、Execute()、Hash()、Corrupt()，并将消息返回给攻击者 。

3) 攻击者在 2 个设备中随机挑选一个设备 $D_i(i=0,1)$。

4) 攻击者 $\mathcal{A}$ 调用 Corrupt() 指令获得当前数据，即当前临时身份。

5) 攻击者 $\mathcal{A}$ 通过训练的结果来猜测设备 D'。若 $D'=D_i$，则攻击者赢得此次游戏，否则失败。

在该模型中的步骤 4)，可以获得的当前临时身份 $\mathrm{PID}_D^{i+1}=H(\mathrm{PID}_D^i\oplus C_{i+1}\oplus R_{i+1})$ 是通过上一个伪随机身份 PID_D^i 的 Hash 值所产生的。根据引理 3 和 Hash 函数的单向性可知，不能得到随机身份 PID_D^i。此外，在该协议中 PUF 所产生的每一个 CRP 都是随机产生的，(C_i,R_i) 与 (C_{i+1},R_{i+1}) 之间毫无关联，彼此相互独立，结合引理 2，攻击者 $\mathcal{A}$ 不能从第 i+1 轮信息 PID_D^{i+1} 推测出第 i 轮信息 PID_D^i。

该协议没有存储任何 CRP 以及机密信息在设备中，并且所使用的认证信息都是一次性的。即使攻击者 $\mathcal{A}$ 获得当前身份信息和 CRP，但 $\mathcal{A}$ 仍然无法通过窃听或者物理攻击等手段来跟踪设备信息。攻击者将不能准确猜测出该设备，则成功的概率为 $\Pr[D'=D_i]=\frac{1}{2}$。因此，所提认证协议在保证设备具有向后不可追踪性的前提下，仍具有向前的保密性。

证毕。

定理 4 所提协议可以防御假冒攻击。

证明 在协议认证机制中，攻击者尝试假冒为合法设备与服务器认证，则攻击者可以通过下面的训练模型来进行攻击。

1) 选择一个有效的认证环境，服务器 S 与设备 D 可以在该环境中通信。

2) 攻击者可以多次调用以下操作：Send()、Execute()、Hash()、Corrupt()。在完成这些操作之后，将消息返回给攻击者 。

3) 攻击者 $\mathcal{A}$ 调用 Send($\prod_S^i, p_1, m_1$) 指令来伪装成合法设备，与服务器进行认证。

4) 如果攻击者可以与服务器相互认证，则攻击者赢得此次游戏，否则失败。

5) 攻击者调用 Send($\prod_D^i, p_2, m_2$) 指令来伪装成合法服务器，与设备进行认证。

6) 如果攻击者可以与设备相互认证，则攻击者赢得此次游戏，否则失败。

当攻击者伪装成合法设备时，需要发送有效伪随机身份 PID_D^i 和正确的认证消息 R_S^{i+1} 与 Auth_S。因为 $R_S^{i+1}=R_{i+1}\oplus H(a_i\|C_{i+1}\|b_i\|R_i)$ 的生成需要有效

信号 (a_i, C_{i+1}, b_i, R_i)。由引理 1 与引理 2 可知，攻击者 $\mathcal{A}$ 不能得到 PUF 的 CRP $(C_i, R_i), (C_{i+1}, R_{i+1})$。并且由引理 2 可知，协议中的随机数 (a_i, b_i) 不能被攻击者 $\mathcal{A}$ 解析。通过引理 3 可知，信息 (a_i, C_{i+1}, b_i, R_i) 不能被获取，从而调用 Hash() 指令不能获取关键信息。因此在该协议机制中，攻击者 $\mathcal{A}$ 不能假冒成合法设备与服务器认证。

当攻击者伪装成服务器时，通过调用 Send $(\prod_D^i, p_2, m_2)$ 指令。服务器需要生成 (C_i, R_i) 和认证信息 $R_D^i = b_i \oplus H(C_i \| R_i \| a_i)$, $\text{Auth}_D = H(a_i \| b_i \| R_D^i \| C_{i1})$。由引理 1～引理 3 可知，攻击者 $\mathcal{A}$ 不能解析出随机数 (a_i, b_i) 与 CRP (C_i, R_i)，从而不能与设备相互认证。因此，攻击者不能伪装成一个合法服务器与设备来进行相互认证。

证毕。

定理 5　所提协议可以防御重放攻击

证明　在协议认证机制中，攻击者 $\mathcal{A}$ 可以按定理 4 的训练模型来进行重放攻击，$\mathcal{A}$ 成功获得合法设备的身份验证信息，并收集该信息进行下一轮的身份验证，以欺骗合法设备与服务器。

在协议的第 i 轮认证中，攻击者调用 Send()、Execute()、Hash()、Corrupt() 指令获得关键信息 $(\text{Auth}_D, R_S^{i+1}, \text{Auth}_S, T_D^i, T_S^i)$。在第 i+1 轮认证时，$\mathcal{A}$ 调用 Send$(\prod_D^i, p_2, m_2)$ 指令来对设备进行攻击。设备在第 i+1 轮的时间为 T_D^{i+1}，而攻击者所保留的仍然是上一轮的时间值 T_S^i，从而设备与服务器的认证时间戳大于临界时间，即 $T_D^{i+1} - T_S^i > \Delta T$。设备会直接返回 Reject，表示该次认证失败。同理，服务器在第 i+1 轮的更新时间为 T_S^{i+1}，攻击者 $\mathcal{A}$ 调用 Send$(\prod_S^i, p_1, m_1)$ 指令来对服务器进行重放攻击也会失败。

为防御重放攻击，所提协议提供了双重保护机制。攻击者 $\mathcal{A}$ 在游戏模型中的第 i 轮会获取随机数 (a_i, b_i)。但在第 i+1 轮认证过程中，随机数也被更新为 (a_{i+1}, b_{i+1})。随机数的更新能防止重放攻击。因此，该协议通过时间戳与随机数更新 2 种机制双重防御重放攻击。

证毕。

4.3　形式化工具分析

ProVerif[20]是一种在形式模型中广泛用于验证密码协议的工具。ProVerif 支持许多加密原语，包括对称/非对称加密、数字签名、散列函数、Diffie-Hellman 密钥协议。ProVerif 能够证明多种安全属性，以及去同步、重放、窃听、假冒等各种攻击类型。本节使用 ProVerif 工具来证明协议的隐私身份验证属性。

在 Dolev-Yao 模型[21]下，该协议由设备和服务器 2 个进程并行运行。图 4 和图 5 分别为协议设备和服务器在注册阶段与认证阶段的运行代码。event DeviceStarted (IDi)表示设备开始认证，event DeviceAuthed (xAuthD)表示设备结束认证。同样，event ServerStarted (xPIDi)和 event ServerAuthed (xAuthS) 分别表示服务器开始认证和结束认证。

```
(*  IoT Device  *)
let IoTDevice =
    let Rid = puf(Ci) in
    let RTMd = puf(CTM) in
    let PIDi = prng(IDi) in
    let m1 = (Rid, RTMd, PIDi) in
    out (ch1, m1);  (*  IoT Device => Cloud Server [private]. *)
    event DeviceStarted(IDi);  (* Start Device Auth *)
    new ai: bitstring;
    let Ai = xor(ai, xor(PIDi, IDTM)) in
    out (ch2, Ai);  (*  IoT Device => Cloud Server *)
    in (ch2, (xTb: bitstring, xCNi: bitstring, xCDi: bitstring,
    xRDi: bitstring, xAuthD: bitstring));
    let Ci1' = xor(xCDi,ai) in
    let Ci2' = xor(xCNi,Ci1') in
    let Ci' = match(Ci1', Ci2') in
    let Rid' = puf(Ci') in
    let bi' = xor(xRDi, h(con(Ci', con(ai,Rid')))) in
    let AuthD' = h(con(ai, con(bi', con(xRDi, Ci1')))) in
    if AuthD' = xAuthD then
    event DeviceAuthed(xAuthD);  (* End Device Auth *)
    new Ta: bitstring;
    let Ci2 = prng(Ci') in
    let Ri2d = puf(Ci2) in
    let Ri2S = xor(Ri2d, h(con(ai, con(Ci2, con(bi', Rid'))))) in
    let AuthS = h(con(ai, con(bi', con(Ri2S , Ci2)))) in
    let PIDinew = h(xor(PIDi, xor(Ci2, Ri2d))) in
    let m4 = (Ta, Ri2S, AuthS) in
    out(ch2, m4). (*  IoT Device => Cloud Server *)
```

图 4　设备运行代码

```
(* Cloud Server *)
let CloudServer =
    in (ch1, (xRid: bitstring, xRTMd: bitstring, xPIDi: bitstring));
    in(ch2, (xAi:bitstring));
    event ServerStarted(xPIDi);  (* Start Server Auth *)
    new bi: bitstring;
    new Tb: bitstring;
    let ai' = xor(xAi,xor(xPIDi, IDTM)) in
    let (Ci1: bitstring, Ci2: bitstring) = algorithm1(Ci) in
    let CNi = xor(Ci1, Ci2) in
    let CDi = xor(Ci1, ai') in
    let RDi = xor(bi, h(con(Ci, con(ai',xRid)))) in
    let AuthD = h(con(ai', con(bi, con(RDi, Ci1)))) in
    let m3 = (Tb, CNi, CDi, RDi, AuthD) in
    out (ch2, m3); (* Cloud Server  => IoT Device *)
    in(ch2, (xTa: bitstring, xRi2S: bitstring, xAuthS: bitstring));
    let Ci2' = prng(Ci) in
    let Ri2d' = xor(xRi2S, h(con(ai', con(Ci2', con(bi, xRid))))) in
    let PIDSnew = h(xor(xPIDi, xor(Ci2', Ri2d'))) in
    let AuthS' = h(con(ai', con(bi, con(xRi2S, Ci2')))) in
    if AuthS' = xAuthS then
    event ServerAuthed(xAuthS). (* End Server Auth *)
```

图 5　服务器运行代码

协议查询结果如图 6 所示，每个查询由虚线分隔。结果表明，服务器与设备之间的认证是稳定的，能防御各种类型攻击，且会话密钥对模拟攻击者具有稳健性。因此，设计的解决方案通过形式验证是安全的。

```
-- Query not attacker(PIDSnew[]) in process 1.
Translating the process into Horn clauses...
Completing...
Starting query not attacker(PIDSnew[])
RESULT not attacker(PIDSnew[]) is true.
-- Query not attacker(PIDinew[]) in process 1.
Translating the process into Horn clauses...
Completing...
Starting query not attacker(PIDinew[])
RESULT not attacker(PIDinew[]) is true.
-- Query inj-event(DeviceAuthed(id)) ==> inj-event(DeviceAuthed(id)) in process 1.
Translating the process into Horn clauses...
Completing...
Starting query inj-event(DeviceAuthed(id)) ==> inj-event(DeviceAuthed(id))
RESULT inj-event(DeviceAuthed(id)) ==> inj-event(DeviceAuthed(id)) is true.
-- Query inj-event(ServerAuthed(id)) ==> inj-event(ServerStarted(id)) in process 1.
Translating the process into Horn clauses...
Completing...
Starting query inj-event(ServerAuthed(id)) ==> inj-event(ServerStarted(id))
RESULT inj-event(ServerAuthed(id)) ==> inj-event(ServerStarted(id)) is true.

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

Query not attacker(PIDSnew[]) is true.

Query not attacker(PIDinew[]) is true.

Query inj-event(DeviceAuthed(id)) ==> inj-event(DeviceAuthed(id)) is true.

Query inj-event(ServerAuthed(id)) ==> inj-event(ServerStarted(id)) is true.

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

图 6 协议查询结果

5 性能分析

由于大部分物联网设备体积小、移动需求高且安全性能低，因此本文协议需要从安全性能、计算开销、存储与通信开销等方面进行评估。本节将本文协议与近几年的认证协议（Wang 等[22]、Patil 等[19]、Gope 等[14]、Bian 等[23]、Qureshi 等[24]、黄可可等[25]）的性能进行比较分析。

5.1 安全性能

本节在安全性能方面将本文协议与其他协议进行对比分析，结果如表 2 所示。攻击者可以通过物理攻击来获取存储设备中的信息，文献[19,22-25]协议都将关键信息存储在设备中，从而不可以防御物理攻击。根据定理 1，所提协议可以防止攻击者获取 PUF 的 CRP 子集，从而防御建模攻击。在文献[14,19,22,24-25]协议中，攻击者可以通过窃听、修改、假冒等攻击获取 PUF 的 CRP，从而不能防御建模攻击。文献[14,23]协议中的设备存储了新旧 2 个身份，攻击者很容易通过物理攻击获取当前身份信息，跟踪到前一轮或下一轮的认证信息，因此协议不具备不可追踪性。文献[25]协议在 PUF 的基础上，只采用了异或与移位操作，不能保证信息安全传输。在现今的密码学中，简单的异或操作是容易被破解的[26-27]，并且移位操作的计算复杂度很低。因此，协议不能很好地防御假冒攻击、线上/线下密码猜测攻击以及 DoS 攻击等。文献[19,22]协议遭受去同步攻击，导致服务器和设备的认证信息不一致，从而双方相互认证不同步。

5.2 计算开销

本节通过 Python 编写程序在设备和服务器之间实现本文协议。网络交互是通过使用抽象 TCP 客户端/服务器连接的套接字来完成的。服务器等待与指定 IP 地址上的设备连接。一旦设备成功与服务器建立连接，该协议就会执行一个相互认证会话。本文在移动设备与服务器实现该协议，在 MIRACL 单元库中仿真协议中的安全原语操作。在 FPGA 下运行 128 位 SRPUF 的运行时间[28]。本文中相关操作的运行时间如表 3 所示。其中，T_H 为 Hash 函数的运行时间，T_P 为 PUF 模块的运行时间，T_M 为 MASK/UNMASK 函数的运行时间，T_F 为辅助函数的运行时间，T_{SK} 为对称加密算法的运行时间，$T_{FE.GEN}$ 为纠错函数生成的运行时间，$T_{FE.REP}$ 为纠错函数恢复的运行时间。

表 4 描述了协议计算开销的性能比较，包括 PUF、Hash 函数、伪随机数发生器、对称加密算法等安全方

表 2 安全性能的分析与比较

协议	双向认证	不可追踪性	可扩展性	前向/后向安全性	数据完整性	不可克隆性	DoS 攻击	重放攻击	物理攻击	建模攻击	假冒攻击	中间人攻击	去同步攻击
Wang 等[22]	√	√	×	√	√	√	√	√	×	×	√	√	×
Patil 等[19]	√	√	×	√	√	√	×	√	×	×	√	√	×
Gope 等[14]	√	×	√	√	√	√	√	√	√	×	√	√	√
Bian 等[23]	√	×	×	√	√	√	×	√	×	√	√	√	√
Qureshi 等[24]	√	√	√	√	×	√	×	√	×	×	√	√	√
黄可可等[25]	√	√	×	√	×	√	×	√	×	×	×	√	√
本文协议	√	√	√	√	√	√	√	√	√	√	√	√	√

表 3 相关操作的运行时间

操作	T_H/ms	T_P/ms	T_M/ms	T_F/ms	T_{SK}/ms	$T_{FE.GEN}$/ms	$T_{FE.REP}$/ms
设备端操作	0.028	0.15	0.686	0.036	0.075	1.67	2.85
服务器端操作	0.012	—	0.332	0.013	0.038	0.82	1.43

法。本文协议在设备端使用了 4 次 Hash 函数、2 次 PUF 操作以及 2 次随机数发生器，其运行时间为 $4T_H+2T_P+2T_R\approx0.468$ ms；在服务器端仅运行 4 次 hash 函数和 2 次伪随机数发生器，其运行时间为 $4T_H+2T_R\approx0.07$ 2ms。文献[14,22-24]协议中均使用了加密算法以及纠错机制，加密算法和纠错机制运行时间长且效率低，不适用于轻量级设备。虽然文献[25]协议使用的运行操作数很少，只采用了异或与移位操作，但是安全性得不到保障。各协议在设备端与服务器端的计算开销分别如图 7(a)与图 7(b)所示。

表 4　　协议计算开销的性能比较

协议	设备端	服务器端
Wang 等[22]	$6T_H+3T_P+T_R+T_{FE.REP}$	$8T_M+T_P$
Gope 等[14]	$4T_H+2T_P+T_R+T_{FE.REP}$	$5T_H+T_R+T_{FE.REP}$
Bian 等[23]	$11T_H+T_P+T_R+T_{FE.REP}$	$10T_H+T_R+T_{FE.REP}$
Qureshi 等[24]	$5T_M+T_P+3T_R$	$5T_M+2T_P+T_{SK}$
黄可可等[25]	$2T_P+8T_F+2T_R$	$2T_R+8T_F$
本文协议	$4T_H+2T_P+2T_R$	$4T_H+2T_R$

5.3　存储与通信开销

图 7(c)与图 7(d)分别表示文献[14,22-25]协议在设备存储与通信开销方面的对比。借鉴文献[29]的实验数据，伪随机身份 PID_D^i 的字节长度为 128 bit，临时身份 ID_{TM} 的字节长度为 128 bit，$CRP(C_i,R_i)$ 的字节长度均为 128 bit，随机数 nonce 的字节长度为 64 bit，密钥的字节长度为 96 bit，辅助数据 hd 为 1 264 bit，Hash 函数的输出为 128 bit。在设备存储需求方面，所提协议只存储了 PID_D^i 与 ID_{TM}，需要 256 bit 的存储开销，远低于其他协议的存储容量。而设备认证阶段，传输开销中采用了 Hash 函数压缩字节长度，传输消息为 A_1,A_2,A_3，通信开销共 896 bit。与文献[14,22-25]协议相比，本文协议的存储与通信开销远低于其他方案。

6　结束语

面对物联网产生的海量数据给资源受限的终端设备带来信息传输的安全隐患，本文基于 PUF 为物联网设备提出一种高效匿名轻量级身份认证协议。该协议结合 PUF 和轻量级安全原语来有效地确保信息传输安全性，包括匿名性、可用性、机密性和前向/后向保密性等。该协议通过保护 CRP 子集的机密性，防止攻击者对 PUF 的建模攻击。攻击者可以通过物理攻击访问设备内存，但该协议中的设

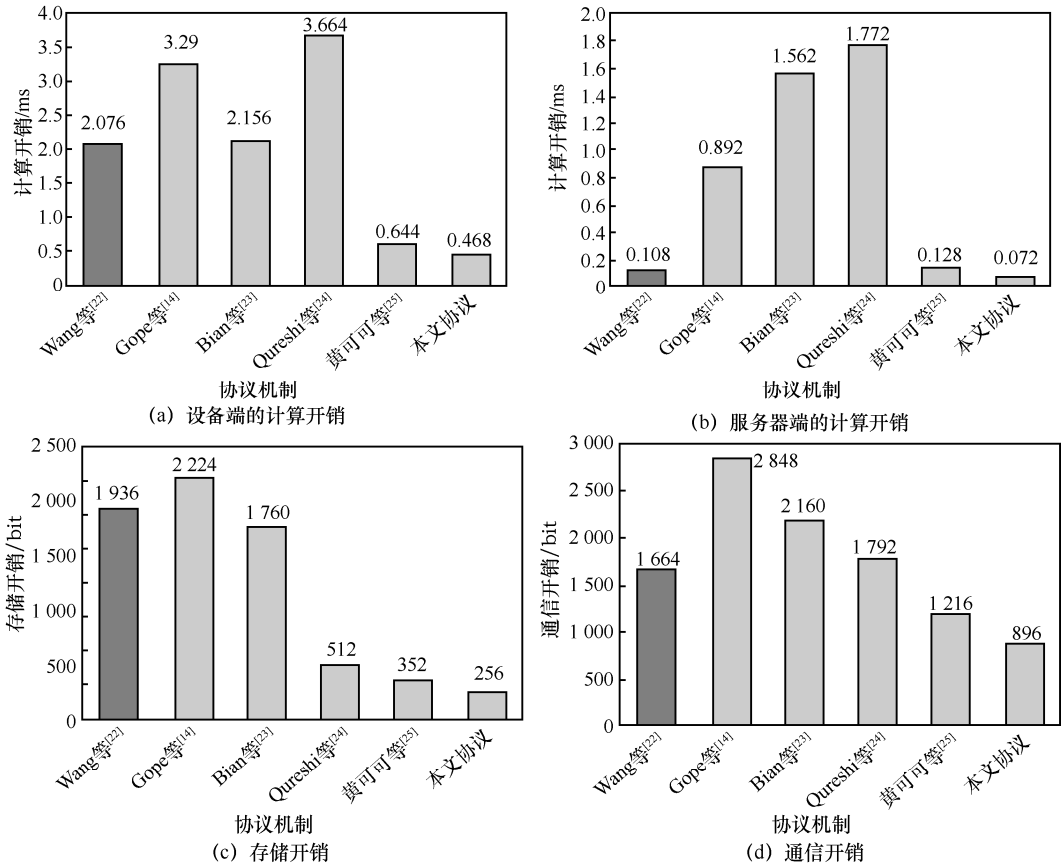

图 7　协议中各类开销对比

备不需要存储任何敏感信息。形式化安全模型和 ProVerif 协议分析工具表明，该协议满足信息传输机密性、完整性、不可追踪性、抗中间人攻击、搞建模攻击等 13 种安全属性。对比现有安全认证机制，所提协议具备计算开销、通信开销和存储开销低以及安全性能高的特点，适合轻量级物联网设备的安全通信。

参考文献：

[1] HAGHI K M, MADANIPOUR M, NIKRAVAN M, et al. A systematic review of IoT in healthcare: applications, techniques, and trends[J]. Journal of Network and Computer Applications, 2021, 192: 103-164.

[2] SHAFIQUE K, KHAWAJA B A, SABIR F, et al. Internet of things (IoT) for next-generation smart systems: a review of current challenges, future trends and prospects for emerging 5G-IoT Scenarios[J]. IEEE Access, 2020, 8: 23022-23040.

[3] BEDI G, VENAYAGAMOORTHY G K, SINGH R. Review of Internet of things (IoT) in electric power and energy systems[J]. IEEE Internet of Things Journal, 2018, 5(2): 847-870.

[4] 毅宇, 周威, 赵尚儒, 等. 物联网安全研究综述：威胁、检测与防御[J]. 通信学报, 2021, 42(8): 188-205.
YANG Y Y, ZHOU W, ZHAO S R, et al. Survey of IoT security research: threats, detection and defense[J]. Journal on Communications, 2021, 42(8): 188-205.

[5] IBRAHIM A, DALKILIC G. Review of different classes of RFID authentication protocols[J]. Wireless Networks, 2019, 25(3): 961-974.

[6] LI W, LI X L, GAO J T, et al. Design of secure authenticated key management protocol for cloud computing environments[J]. IEEE Transactions on Dependable and Secure Computing, 2021, 18(3): 1276-1290.

[7] IU S M, WANG D, XU G A, et al. Practical and provably secure three-factor authentication protocol based on extended chaotic-maps for mobile lightweight devices[J]. IEEE Transactions on Dependable and Secure Computing, 2022, 19(2): 1338-1351.

[8] AO Y S, AL-SARAWI S F, ABBOTT D. Physical unclonable functions[J]. Nature Electronics, 2020, 3(2): 81-91.

[9] ELVAUX J. Machine-learning attacks on PolyPUFs, OB-PUFs, RPUFs, LHS-PUFs, and PUF-FSMs[J]. IEEE Transactions on Information Forensics and Security, 2019, 14(8): 2043-2058.

[10] 叶靖, 胡瑜, 李晓维. 非确定性仲裁型物理不可克隆函数设计[J]. 计算机辅助设计与图形学学报, 2017, 29(1): 166-171.
YE J, HU Y, LI X W. Nondeterministic logic based arbiter physical unclonable function[J]. Journal of Computer-Aided Design & Computer Graphics, 2017, 29(1): 166-171.

[11] PRADA-DELGADO M A, VÁZQUEZ-REYES A, BATURONE I. Trustworthy firmware update for Internet-of-thing devices using physical unclonable functions[C]//Proceedings of 2017 Global Internet of Things Summit (GIoTS). Piscataway: IEEE Press, 2017: 1-5.

[12] MALL P, AMIN R, DAS A K, et al. PUF-based authentication and key agreement protocols for IoT, WSNs, and smart grids: a comprehensive survey[J]. IEEE Internet of Things Journal, 2022, 9(11): 8205-8228.

[13] GU C Y, CHANG C H, LIU W Q, et al. A modeling attack resistant deception technique for securing lightweight-PUF-based authentication[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2021, 40(6): 1183-1196.

[14] GOPE P, LEE J, QUEK T Q S. Lightweight and practical anonymous authentication protocol for RFID systems using physically unclonable functions[J]. IEEE Transactions on Information Forensics and Security, 2018, 13(11): 2831-2843.

[15] HOSSAIN M, NOOR S, HASAN R. HSC-IoT: a hardware and software co-verification based authentication scheme for Internet of things[C]//Proceedings of 2017 5th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud). Piscataway: IEEE Press, 2017: 109-116.

[16] AKGÜN M, ÇAĞLAYAN M U. Providing destructive privacy and scalability in RFID systems using PUFs[J]. Ad Hoc Networks, 2015, 32: 32-42.

[17] ZHOU L, LI X, YEH K H, et al. Lightweight IoT-based authentication scheme in cloud computing circumstance[J]. Future Generation Computer Systems, 2019, 91: 244-251.

[18] MORIYAMA D, MATSUO S, YUNG M. PUF-based RFID authentication secure and private under memory leakage[R]. 2014.

[19] PATIL A S, HAMZA R, HASSAN A, et al. Efficient privacy-preserving authentication protocol using PUFs with blockchain smart contracts[J]. Computers & Security, 2020, 97: 101958.

[20] XIE Q, HU B, TAN X, et al. Robust anonymous two-factor authentication scheme for roaming service in global mobility network[J]. Wireless Personal Communications, 2014, 74(2): 601-614.

[21] DOLEV D, YAO A. On the security of public key protocols[J]. IEEE Transactions on Information Theory, 1983, 29(2): 198-208.

[22] WANG W Z, CHEN Q, YIN Z M, et al. Blockchain and PUF-based lightweight authentication protocol for wireless medical sensor networks[J]. IEEE Internet of Things Journal, 2022, 9(11): 8883-8891.

[23] BIAN W X, GOPE P, CHENG Y Q, et al. Bio-AKA: an efficient fingerprint based two factor user authentication and key agreement scheme[J]. Future Generation Computer Systems, 2020, 109: 45-55.

[24] QURESHI M A, MUNIR A. PUF-RAKE: a PUF-based robust and lightweight authentication and key establishment protocol[J]. IEEE Transactions on Dependable and Secure Computing, 2021, PP(99): 1.

[25] 黄可可, 刘亚丽, 殷新春. 一种基于 PUF 的超轻量级 RFID 标签所有权转移协议[J]. 密码学报, 2020, 7(1): 115-133.
HUANG K K, LIU Y L, YIN X C. A PUF-based ultra-lightweight ownership transfer protocol for low-cost RFID tags[J]. Journal of Cryptologic Research, 2020, 7(1): 115-133.

[26] FEISTEL H, NOTZ W A, SMITH J L. Some cryptographic techniques for machine-to-machine data communications[J]. Proceedings of the IEEE, 1975, 63(11): 1545-1554.

[27] SCHNEIER B. Applied cryptography: protocols, algorithms, and source code in C[M]. New York: Wiley, 1996.

[28] HOU S, DENG D, WANG Z Y, et al. A dynamically configurable LFSR-based PUF design against machine learning attacks[J]. CCF Transactions on High Performance Computing, 2021, 3(1): 31-56.

[29] AYSU A, GULCAN E, MORIYAMA D. End-to-end design of a PUF-based privacy preserving authentication protocol[C]//Cryptographic Hardware and Embedded Systems. Berlin: Springer, 2015: 556-576.

[作者简介]

王振宇（1995− ），男，湖南长沙人，国防科技大学博士生，主要研究方向为硬件安全、物联网协议、安全认证协议、物理不可克隆函数等。

郭阳（1971− ），男，湖南长沙人，博士，国防科技大学研究员、博士生导师，主要研究方向为微处理设计、嵌入式系统设计等。

李少青（1963− ），男，陕西西安人，国防科技大学研究员、博士生导师，主要研究方向为硬件安全、硬件木马检测、IP 核安全检测等。

侯申（1983− ），男，河南洛阳人，博士，信息工程大学讲师，主要研究方向为物理不可克隆函数设计、微处理器设计、物联网安全等。

邓丁（1993− ），男，湖南长沙人，博士，国防科技大学讲师，主要研究方向为硬件木马、物理不可克隆函数设计、高性能处理器扫描测试等。