面向边缘计算的加密协议设计

农奇宇　杨　腾
数字广西集团有限公司　广西　南宁　530003

摘　要：物联网、5G 和人工智能发展，使边缘计算成为支撑低延迟、高带宽应用的关键基础设施。但边缘计算环境有资源受限、设备异构等特性，面临严峻安全挑战，传统重量级加密协议难在边缘节点高效部署。设计安全、高效且适应性强的轻量级加密协议，是保障边缘计算生态安全的核心需求。本文剖析了边缘计算场景下的安全威胁模型与性能约束，梳理现有轻量级密码学原语研究进展，提出融合国密算法 SM4 与改进型椭圆曲线密码的混合式轻量级加密协议 EdgeLightSec。该协议优化密钥协商、采用高效双向认证流程、引入重放攻击防护策略，在保证前向保密性和抗中间人攻击能力的同时，降低计算复杂度与通信开销，为构建边缘计算安全体系提供可行路径。在部署层面，可为各接入单位部署专用安全计算服务器硬件设备，与对应的虚拟机节点建立安全通信链路，并提供本地数据隐私保护与加密存储功能。

关键词：边缘计算；轻量级加密；安全协议；SM4；可信数据；数据隐私保护；身份加密

引言

云计算虽成功为数据处理提供算力支持，但智能终端设备爆发式增长及新兴应用场景涌现，对网络延迟等要求极高，上传云端处理数据弊端凸显，边缘计算应运而生。它通过部署计算节点就近处理数据，降低延迟与核心网络负载。然而，边缘计算独特架构带来新安全风险，如边缘节点易遭攻击、终端设备资源有限使传统安全协议不适用、拓扑结构动态要求协议快速建立和撤销安全关联、海量异构设备接入对协议可扩展性和互操作性要求高。这催生了“轻量级密码学”研究热潮，但现有方案存在不足。因此，本研究聚焦“面向边缘计算的轻量级加密协议设计”，在理论、技术和应用层面分别有深化威胁模型理解、提出协议框架、推动安全落地等重要意义。

1 边缘计算安全挑战与轻量级密码学基础

1.1 边缘计算环境下的安全威胁模型

在设计安全协议前，需明确其应抵御的威胁。针对边缘计算，我们定义包含以下三方的威胁模型：合法用户（LU）、边缘节点（EN）和恶意攻击者（Adv）。攻击者能力较强：可窃听、篡改、删除或重放LU与EN间的所有通信；可冒充任一方发起交互；甚至可能控制部分被攻破的边缘节点或终端设备[1]。在某些场景下，攻击者还可能物理接触设备以尝试提取密钥，尽管假设设备具备基本防篡改能力。基于此模型，协议须满足以下安全目标：（1）机密性：确保攻击者无法获知通信内容；（2）完整性：防止数据在传输中被篡改；（3）双向认证：LU与EN能相互验证对方身份；（4）前向保密性：即使长期私钥泄露，历史会话仍安全；（5）抗重放攻击：有效防御消息重放；（6）轻量级实现：所有安全机制须在计算、存储和通信开销上适配边缘设备的资源约束。（7）交互效率保障：数据交互全流程时延须严格控制在5秒内，涵盖从请求发起到获取完整响应的网络传输、数据处理与资源调度等全部环节。

1.2 轻量级密码学原语综述

1.2.1 轻量级对称密码

对称密码因其加解密速度快、资源消耗低，是轻量级协议的首选。主要包括：（1）轻量级分组密码：如PRESENT、SIMON、SPECK、CLEFIA和中国的国密标准SM4。它们通过简化S盒、减少轮数、使用轻量级线性变换等方式优化硬件和软件实现效率。其中，SM4因其在国内的广泛认可度和良好的安全性，成为本文协议的基础。（2）轻量级流密码：如Trivium、Grain-128a，通过生成伪随机密钥流与明文逐位异或来加密，非常适合处理连续数据流。（3）轻量级认证加密（AE）：如AES-GCM-SIV、ChaCha20-Poly1305的轻量化变种，能在一次操作中同时提供机密性和完整性保护，避免了“先加密后认证”或“先认证后加密”模式的潜在安全缺陷。

1.2.2 轻量级非对称密码

虽然非对称密码开销较大，但在实现密钥协商和数字签名方面不可或缺。椭圆曲线密码（ECC）因其在相同安全强度下所需的密钥长度远小于RSA，成为轻量级场景下的主流选择。例如，256位的ECC密钥提供的安全性约等于3072位的RSA密钥，但其运算速度更快，内存占用更少。本文将采用优化的ECC曲线（如NIST P-256或

Curve25519）来实现高效的密钥交换。

1.2.3 轻量级哈希函数

用于生成消息摘要、构建MAC（消息认证码）或作为密钥派生函数（KDF）。SHA-256虽然安全，但对于某些超低功耗设备仍显沉重。因此，SHA-3（Keccak）的轻量级配置或专门为IoT设计的哈希函数（如PHOTON）也是可选项。

2 EdgeLightSec 协议设计

2.1 设计目标与原则

基于前述分析，EdgeLightSec协议的设计遵循以下核心原则：（1）混合架构：结合对称密码的高效率与非对称密码的密钥管理优势。使用ECC进行初始的、一次性的密钥协商以获得会话密钥，后续通信则完全依赖高速的对称加密（SM4）。（2）最小化公钥操作：将计算密集型的ECC运算限制在协议初始化阶段，并尽可能减少其调用次数[2]。（3）利用预共享信息：假设每个合法设备在出厂时已预置了EN的公钥（Pub_EN）和自身的长期私钥（Priv_LU）及证书（Cert_LU），以简化认证流程。（4）简洁的交互流程：采用两轮或三轮握手，减少通信延迟。

2.2 协议符号与假设

LU: 合法用户（边缘设备）

EN: 边缘节点

Priv_LU, Pub_LU: LU的长期ECC私钥和公钥

Priv_EN, Pub_EN: EN的长期ECC私钥和公钥

Cert_LU = Sign(Priv_CA, Pub_LU ‖ ID_LU): 由可信CA签发的LU证书

K_session: 本次会话的临时对称会话密钥

TS: 当前时间戳

Nonce: 随机数

SM4_Enc/Dec(K, M): 使用密钥K对消息M进行SM4加/解密

ECC_SharedSecret(a, B): 基于私钥a和公钥B计算ECC共享密钥

H(): 安全的哈希函数（如SM3）

KDF(): 密钥派生函数

假设：LU和EN都已安全地获取了对方的长期公钥。EN拥有一个由可信CA签发的证书，其公钥Pub_EN被硬编码在所有合法LU的固件中。LU的证书Cert_LU在首次注册时由EN验证并存储。

2.3 协议详细流程

2.3.1 安全关联建立阶段

在安全关联建立阶段，协议通过一个四步握手流程完成双向认证和会话密钥的协商。首先，LU向EN发起连接请求，发送其身份ID、证书、一个随机构造的Nonce_LU以及当前时间戳TS_LU。EN收到此请求后，会立即验证时间戳TS_LU的有效性以防范重放攻击，并利用CA的公钥验证LU证书Cert_LU的真实性。验证通过后，EN生成自己的随机数Nonce_EN和时间戳TS_EN，并用自己的私钥对（Nonce_LU, Nonce_EN, TS_EN）的组合进行签名，将签名结果Sig_EN连同Nonce_EN和TS_EN一并返回给LU。LU收到EN的响应后，同样会验证时间戳TS_EN的新鲜性，并使用预置的Pub_EN验证签名Sig_EN的有效性。一旦双向认证成功，LU和EN便各自独立地计算本次会话的共享密钥。它们首先利用各自的长期私钥和对方的长期公钥计算出一个ECC共享秘密，然后将此秘密与双方交换的随机数Nonce_LU和Nonce_EN一同输入密钥派生函数KDF，最终生成唯一的会话密钥K_session。随后，LU使用K_session加密其首次业务请求，并附上基于K_session和时间戳计算的消息认证码（MAC），将加密后的请求发送给EN。EN使用相同的K_session解密并验证MAC，至此，一个安全、可信的通信通道正式建立。

2.3.2 安全数据传输阶段

进入安全数据传输阶段后，所有后续的通信数据都将使用K_session进行SM4加密，并附带相应的MAC和时间戳，以持续保证通信的机密性、完整性和时效性。会话密钥K_session的有效期可根据具体安全策略动态设定，过期后需重新执行完整的四步握手流程以建立新的安全关联[3]。这种设计确保了即使会话密钥被短暂破解，其影响范围也被严格限制在单次会话之内，从而增强了系统的整体韧性。

2.4 关键安全机制解析

2.4.1 双向认证与前向保密性

协议通过交换和验证数字证书与签名实现了严格的双向认证，确保了通信双方身份的真实性，从根本上杜绝了伪装攻击。其前向保密性则由会话密钥K_session的独特生成方式所保障：K_session不仅依赖于双方的长期私钥，更关键地融合了本次会话独有的、一次性的随机数Nonce_LU和Nonce_EN。这意味着，即便攻击者在未来某个时刻成功窃取了某一方的长期私钥，由于无法获知历史会话中使用的随机数，他依然无法重构出过去的会话密钥，从而保护了历史通信的机密性。

2.4.2 抗重放攻击与效率优化

为了抵御重放攻击，协议采用了双重防护策略，即时间戳（TS）与随机数（Nonce）相结合。时间戳用于快速过滤掉那些明显超出有效时间窗口的过期消息，而

Nonce则确保了即使在有效时间窗口内，每一条消息的内容也是独一无二的，任何重复或延迟的消息都会因Nonce或时间戳的不匹配而被接收方拒绝。最后，在效率方面，协议通过将公钥运算（一次ECC共享密钥计算和一次ECC签名验证）集中于初始化阶段，并在后续通信中完全依赖高效的SM4对称加密，成功地在提供强大安全保障的同时，将整体资源消耗控制在了边缘设备可接受的范围内。

3 安全性与性能分析

3.1 形式化安全性分析

为严谨地证明EdgeLightSec协议满足其预设的安全目标，我们采用了BAN逻辑（Burrows-Abadi-Needham Logic）对其进行形式化验证。该分析建立在几个合理的基本前提之上：LU相信其预置的Pub_EN确实属于EN；EN通过验证CA签发的证书而相信Pub_LU属于LU；并且双方生成的Nonce_LU和Nonce_EN都是新鲜的、不可预测的。基于这些前提，BAN逻辑的推导链条清晰地展示了协议如何逐步建立起双方的信任。EN通过验证LU证书确认了其身份，并通过验证自身签名确认了Nonce_LU的新鲜性；LU则通过验证EN的签名确认了其身份和Nonce_EN的新鲜性。最终，双方都确信他们共同拥有一个新鲜的、安全的会话密钥K_session，从而完成了安全的密钥交换和双向认证。

3.2 性能与资源消耗分析

我们将EdgeLightSec与TinySec（纯对称，无前向保密）、LEAP+（基于预共享密钥的多层安全）以及标准的ECDH+ECDSA组合进行对比。

表1 性能与资源消耗对比

指标	TinySec	LEAP+	ECDH+ECDSA	EdgeLightSec(本文)
双向认证	否	是（特定层）	是	是
前向保密	否	否	是	是
公钥运算次数	0	0	2(ECDH)+2(Sign/Verify)	1(ECDH)+1(Verify)
对称加密轮次	多	多	多	多
通信轮次	1-2	可变	2-3	3
适用性	静态网络	分层网络	通用但重	动态边缘网络

从表中可见，EdgeLightSec在保留了前向保密和强认证等关键安全属性的同时，将公钥运算的次数减半，显著优于标准的ECDH+ECDSA方案。相较于TinySec和LEAP+，它提供了更全面的安全保障。在资源消耗方面，SM4的软件实现非常高效，在ARM Cortex-M系列MCU上，加密吞吐量可达数百Mbps，而一次256位ECC点乘运算通常在几十毫秒量级。对于非高频通信的边缘设备，这样的开销是完全可以接受的。

4 结语

本文针对边缘计算环境安全需求，提出 EdgeLightSec 轻量级加密协议。它创新融合国密 SM4 对称加密算法与优化椭圆曲线密码，经四步握手流程，高效达成双向认证等核心安全目标。未来研究将从多方向推进：一是探索后量子安全性，把格密码等新型密码原语融入协议框架，应对未来威胁；二是扩展协议支持群组通信，设计高效群组密钥管理与分发机制；三是研究安全协议与人工智能深度融合，用轻量级机器学习模型实时检测异常，构建主动智能防御体系；四是推动协议标准化，研究 SM4 和 ECC 运算专用硬件加速器设计，通过软硬协同提升整体安全效能。五是推动应用落地，面向政务、金融、公安、通信运营商等对数据安全与传输可靠性要求极高的关键行业，深化轻量级加密协议在敏感信息传输、身份认证、设备接入等核心场景的集成应用，构建可复用、可推广的安全防护范式，助力边缘计算环境下的可信数据流转与智能服务部署。

参考文献

[1]邓喆,吴亚洁.边缘计算场景下无线传感器网络信息轻量级加密算法[J].信息技术与信息化,2024,(01):157-160.

[2]衣孚民.基于流加密的轻量级Modbus TCP协议安全研究[D].齐鲁工业大学,2023.(05):200-202.

[3]程小辉,丁黄婧,邓昀,等.基于边缘计算的多授权属性加密方案[J].计算机工程与设计,2023,44(08):2272-2279.