计算机科学
COMPUTER SCIENCE

基于软件定义边界的零信任匿名访问方案

李惟贤, 张建辉, 曾俊杰, 贾洪勇, 门蕊蕊

引用本文

李惟贤, 张建辉, 曾俊杰, 贾洪勇, 门蕊蕊. 基于软件定义边界的零信任匿名访问方案[J]. 计算机科学, 2024, 51(12): 293-302.

LI Weixian, ZHANG Jianhui, ZENG Junjie, JIA Hongyong, MEN Ruirui. Zero Trust Anonymous Access Scheme Based on Software-defined Perimeters [J]. Computer Science, 2024, 51(12): 293-302.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.231000176

基于软件定义边界的零信任匿名访问方案

李惟贤[1] 张建辉[2] 曾俊杰[1] 贾洪勇[1] 门蕊蕊[1]
1 郑州大学网络空间安全学院 郑州 450000
2 嵩山实验室 郑州 450000
(lwx502@gs.zzu.edu.cn)

摘 要 软件定义边界作为一种具有良好可扩展性与安全性的零信任安全架构得到了广泛应用。标准的软件定义边界架构采用单包授权机制来实现对服务资源的隐藏与对访问者身份的验证,但现有的方案普遍采用集中式的方式存储与分发SPA密钥,且缺乏对访问者隐私信息的保护。针对以上问题,提出了一种软件定义边界架构下的零信任匿名访问方案,采用三方密钥协商实现SPA密钥的分发,并使用通用指定验证者签名实现了对访问者身份的匿名认证,且能够抵抗SPA密钥窃取、敲门放大攻击、身份假冒等网络攻击,与目前的软件定义边界方案相比具有更强的安全性。实验结果表明,所提方案降低了33%的通信开销,在多节点网络环境下降低了20%的平均认证时延。

关键词:零信任;软件定义边界;单包授权;三方密钥协商;通用指定验证者签名;匿名访问
中图分类号 TP309

Zero Trust Anonymous Access Scheme Based on Software-defined Perimeters

LI Weixian[1],ZHANG Jianhui[2],ZENG Junjie[1],JIA Hongyong[1] and MEN Ruirui[1]
1 School of Cyber Science and Engineering,Zhengzhou University,Zhengzhou 450000,China
2 Songshan Laboratory,Zhengzhou 450000,China

Abstract Software-defined perimeters,as a highly scalable and secure zero-trust security architecture,have gained widespread adoption. Conventional software-defined perimeter(SDP) architectures employ a single packet authorization mechanism to achieve resource hiding and visitor identity validation. However,existing solutions often store and distribute SDP keys in a centralized manner,and lack of robust protection for visitor privacy. In response to the aforementioned challenges,a zero-trust anonymous access scheme within the software-defined perimeter architecture is proposed. This scheme utilizes a three-party key agreement for SDP key distribution and employs generalized designated verifier signatures for anonymous visitor identity authentication. Moreover,it demonstrates resilience against network attacks such as SPA key theft,port knocking amplification attacks,and identity spoofing,thus exhibiting enhanced security compared to existing software-defined perimeter schemes. Experimental findings reveal a reduction of 33% in communication overhead and a 20% decrease in average authentication latency within multi-node network environments.

Keywords Zero trust,Software-defined perimeter,Single packet authorization,Three-party key agreement,Universal designated verifier signature,Anonymous access

1 引言

自“零信任”概念[1]在2010年被首次提出以来,零信任网络已逐渐从一种新兴网络安全理念走向工程实践与标准化。云安全联盟(Cloud Security Alliance,CSA)给出的软件定义边界(Software-Defined Perimeter,SDP)架构标准[2]是目前应用较为广泛的零信任实施方案。SDP架构采取服务资源隐藏的策略,从而有效减小了攻击面,并能够根据所保护的服务资源对访问控制策略进行动态调整;同时,SDP架构的部署方式也更加灵活,可以在不改变原有网络环境的前提下进行部署以提高网络安全性,避免了增添或更换网络基础设施带来的部署成本增加问题。SDP架构因具有良好的安全性与可扩展性,在物联网设备安全接入、电力网络、远程访问控制和云安全等场景得到了广泛的应用性研究[3-5]。

SDP架构的工作流程如图1所示,其组件包括SDP Controller(SDP控制器)、SDP Gateway(SDP网关)和SDP Client(SDP客户端)。在SDP架构中,SDP网关对服务端口进行隐藏以保护服务资源,访问者无论是处在互联网还是内部网络

到稿日期:2023-10-25 返修日期:2024-03-25
基金项目:河南省重大科技专项(221100210900-01);2021年中国高校产学研创新基金(2021ITA11021)
This work was supported by the Science and Technology Major Project of Henan Province(221100210900-01) and 2021 China University Industry-Academic-Research Innovation Fund(2021ITA11021).
通信作者:曾俊杰(zengjj_lab@163.com)

中都无法直接访问服务资源；SDP 控制器通过 TLS 安全连接对 SDP 网关进行认证；访问者通过 SDP 客户端与 SDP 控制器进行交互，SDP 控制器对访问者进行认证并向 SDP 客户端分发集中式生成与存储的口令或密钥；在 SDP 客户端使用口令或密钥与 SDP 网关完成验证后，SDP 网关对其开放相应服务端口；SDP 客户端与 SDP 网关进行 TLS 双向认证并建立加密信道以访问服务资源。

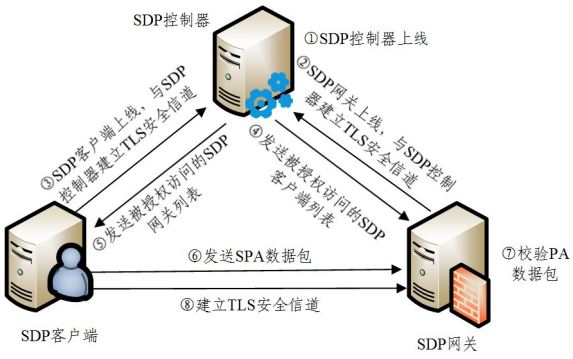

图 1 SDP 架构工作流程

Fig. 1 SDP architecture workflow

尽管基于 SDP 架构的零信任网络抵抗网络攻击的能力已得到了有力验证[6]，但如何采用密码学机制进一步增强 SDP 架构的安全性，拓展其安全应用场景和特殊安全需求，仍然是值得研究的问题。

SDP 架构的基础功能之一是服务隐藏，可有效防止端口扫描、拒绝服务攻击和中间人攻击等网络攻击，通常采用端口敲门(Port Knock，PK)技术[7]和单包授权认证(Single Packet Authorization，SPA)技术[8]实现。PK 技术采用在 TCP 握手报文中嵌入密码口令的方式来进行身份认证，安全强度较低，容易受到网络窃听和口令暴力破解等攻击[9]。SPA 技术采用对称密码算法和 HMAC 方式来保护 SPA 报文的机密性和认证性，其典型工程实现是已广泛应用的开源软件 Fwknop，但存在 SPA 密钥分发效率低的问题；SDP 架构下核心组件 SDP 控制器通常承担密钥管理中心的角色，集中式分发和存储 SPA 密钥，需要解决分发效率和存储安全问题，如 Fwknop 在 SDP 架构下核心组件 SDP 控制器需要手工分发和明文存储 SPA 密钥。SPA 报文通常以 UDP 报文形式进行传输，在 SPA 认证成功后通信双方通过 TLS 协议进行双向认证并建立加密信道，但在此传输模式下整个认证过程本质上是松耦合的身份认证协议，设计上存在敲门放大[10]的缺陷。此外，需要注意的是，SDP 控制器因高价值性势必成为网络攻击焦点，同时 SDP 架构各种基础组件本身可能因存在漏洞或后门而产生失陷风险。

SDP 架构在防护模式上从“以物理边界为中心”过渡到“以可信身份为中心”，在安全设计上强调基于单包授权机制和双向 TLS 认证加密机制实现零信任网络的认证性、机密性和完整性等安全需求，但缺乏对隐私保护需求的考虑。SDP2.0 协议中基于共享密钥的 SPA 认证过程要求用户身份标识不加密，且基于 TLS 协议的身份认证过程需要数字证书，这些环节均会暴露访问者的身份信息。

针对以上问题，本文提出了一种 SDP 架构下的零信任匿名访问方案，主要工作包括：

1)基于三方密钥协商协议实现 SDP 客户端和 SDP 网关共享 SPA 密钥，且 SDP 控制器无需存储 SPA 密钥；

2)基于通用指定验证者签名实现 SDP 客户端与 SDP 网关间的一次性匿名认证；

3)采用随机身份标识实现 SPA 认证和 TLS 认证中用户匿名身份的一致性，解决 UDP 模式下 SPA 认证存在的“敲门放大”问题；

4)对本文方案进行了安全性分析与性能评估，验证了本文方案的正确性、安全性、可行性和运行效率。

本文第 2 章介绍了相关工作；第 3 章给出了预备知识；第 4 章详细阐述了本文所提方案；第 5 章对本文方案的安全性进行了分析；第 6 章介绍了在实验环境下对本文方案进行测试与对比的情况；最后对全文进行总结。

2 相关工作

目前，已有众多学者在增强 SDP 架构安全性与匿名性方面开展了大量研究工作。

在增强 SDP 架构安全性方面，目前的研究主要从对 SPA 认证机制的改进入手。文献[11]提出了一种轻量级端口敲门方案以抵抗 TCP 重放攻击和端口扫描。文献[12]提出了一种能够支持自定义访问策略的 OpenSPA 方案。文献[13-14]采用时间同步的 HTOP 机制来改进单包授权认证技术以增强 SDP 架构的安全性。SDP 架构下 SDP 客户端和 SDP 网关间通信首包为 SPA 认证包，但出于对抗 DDOS 攻击的考虑，文献[2]提出应采用对称密码技术构建轻量级单包授权机制，而不宜采用公钥密码体制，因此 SPA 密钥的分发与存储安全至关重要。文献[15]提出一种“Honeykeys”的欺骗机制来降低集中式存储 SPA 密钥带来的密钥泄露风险，但没有考虑 SPA 密钥分发安全性。文献[16]采用基于密钥派生函数和 HMAC 的 HKDF 方案实现对称密钥的分发。文献[17]基于多重秘密共享方案实现了多个参与方安全共享多个密钥。文献[18]基于安全多方计算实现了在多个参与方之间进行密钥协商。此外，文献[14] 提出基于内生安全技术来增强零信任网络架构安全性，以避免未知漏洞攻击导致零信任网络失效。

在实现 SDP 架构匿名性方面，目前的研究主要通过对 SPA 密钥分发机制、SDP 客户端与 SDP 网关间的认证机制两方面进行改进以实现匿名性。文献[19]提出了一种基于匿名身份标识的密钥协商协议，实现了 SPA 密钥分发的匿名性与不可追踪性。在密钥协商模式下，由于 SDP 架构要求双方通信首包必须为 SPA 报文，因此 SPA 秘钥协商需要 SDP 控制器、SDP 客户端和 SDP 网关三方执行多方秘钥协商协议来实现。文献[20-22]在 Diffie-Hellman 密钥协商协议的基础上采用双线性映射实现了一种三方密钥交换方案，同时对传统的三方认证密钥协商协议进行了安全增强，能够在不安全信道中完成密钥协商。该方案与 SDP 架构的契合度较高，能够有效解决 SPA 密钥分发环节的匿名性问题。一些研究也为如何在 SDP 客户端与 SDP 网关间进行匿名认证提供了可供参考的解决方案。文献[23]基于其提出的具有可追溯性的通用指定验证者签名方案，并结合 Kerberos 协议实现了一种零信任网络匿名认证方案，同时指出该方案可用于 SDP 架构单包授权机制。文献[24]提出了一种基于群签名的匿名认证

方案。文献[3]提出了一种基于累加器的物联网设备匿名接入方案,实现了设备认证与接入阶段的隐私保护。文献[25]基于双线性映射实现了一种匿名凭证系统以保证凭证持有者的匿名性。相较于匿名凭证系统,通用指定验证者签名[26-28]能够以更加轻量级的方式完成对访问者身份的认证,同时实现了对访问者隐私信息的保护。

3 预备知识

3.1 双线性映射

设 G 是 p 阶加法循环群,G_T 是 p 阶乘法循环群。若一个映射 $e:G\times G\rightarrow G_T$ 满足以下 3 条性质,则称映射 e 为双线性映射。

1)双线性:对于 $\forall a,b\in Z_p$ 和 $\forall g_1,g_2\in G$,总有 $e(ag_1,bg_2)=e(g_1,g_2)^{ab}$。

2)非退化性:$\forall g_1,g_2\in G, e(g_1,g_2)\neq 1$。

3)可计算性:对于 $\forall g_1,g_2\in G$,存在一个多项式时间算法来计算 $e(g_1,g_2)$。

3.2 困难性假设

3.2.1 DL 困难假设

给定 G 的生成元 Q 及 $A=aQ$,在仅获取公开参数(G,Q,A)的情况下,攻击者 $\mathcal{A}$ 计算出 a 的优势 $Adv_{\mathcal{A},\mathrm{DLP}}=Pr[A(Q,\mathcal{A})=a]$是可以忽略的。

3.2.2 CDH 困难假设

给定 G 的生成元 Q 及 $aQ,bQ\in G$,在仅能获取(G,Q,aQ,bQ)的情况下,攻击者 $\mathcal{A}$ 成功计算出 abQ 的优势$Adv_{\mathcal{A},\mathrm{CDH}}=Pr[\mathcal{A}(Q,aQ,bQ)=abQ]$是可以忽略的。

3.2.3 BDH 困难假设

给定 G 的生成元 Q 及 $aQ,bQ,cQ\in G$,在仅能获取(G,Q,aQ,bQ,cQ)的情况下,攻击者 $\mathcal{A}$ 计算出 $e(Q,Q)^{abc}$ 的优势 $Adv_{\mathcal{A},\mathrm{BDH}}$是可以忽略的。其中$Adv_{\mathcal{A},\mathrm{BDH}}=Pr[\mathcal{A}(Q,aQ,bQ,cQ)=e(Q,Q)^{abc}]$。

3.3 三方密钥协商协议

本文在文献[20]与文献[22]的基础上对三方密钥协商协议进行简化,以实现 SPA 密钥的分发。在本文所使用的三方密钥协商协议中,对 KGC 所承担的功能进行了简化,其只需要承担发布公共参数的功能。协议包括 KGC 和 3 个参与方 A,B,C。

3.3.1 协议流程

KGC 执行如下操作:

KGC 选择 p 阶加法循环群 G 和 p 阶乘法循环群 G_T(p 是大素数),G 的生成元 Q 及双线性映射 $e:G\times G\rightarrow G_T$。KGC 将$\langle G,G_T,Q,e\rangle$作为公共参数公布。

参与方 A,B,C 执行如下操作:

1)A 随机选择 $a\in Z_p^*$ 将 $G_A=aQ$ 发送给 B 和 C;B 随机选择 $b\in Z_p^*$ 将 $G_B=bQ$ 发送给 A 和 C;C 随机选择 $c\in Z_p^*$ 将 $G_C=cQ$ 发送给 A 和 B;

2)A 计算密钥 $k=e(G_B,G_C)^a=e(Q,Q)^{abc}$;

3)B 计算密钥 $k=e(G_A,G_C)^b=e(Q,Q)^{abc}$;

4)C 计算密钥 $k=e(G_A,G_B)^c=e(Q,Q)^{abc}$。

经过一轮通信后,3 个协议参与方协商得到共同的会话密钥 $k=e(Q,Q)^{abc}$。

3.3.2 协议安全性

Joux 首次基于椭圆曲线离散对数提出了一个有效的三方一轮密钥协商协议[20],并在文献中对密钥的保密性进行了阐述。Shim[22]则对三方密钥协商协议进行了拓展,证明了三方密钥协商协议具有以下安全属性:

1)密钥保密性:攻击者无法在仅获得公开参数的情况下计算出会话密钥。

2)已知密钥安全:攻击者获得了除当前会话之外的会话密钥也不会影响当前会话的安全。

3)无密钥控制:会话密钥由全体参与方共同参与生成,单个参与方无法控制密钥生成。

3.4 通用指定验证者签名方案

本文在文献[26]提出的通用指定验证者签名的基础上构建了适用于 SDP 架构的通用指定签名者签名方案,可以有效解决 SDP 客户端访问 SDP 网关时认证环境的隐私保护问题。

3.4.1 方案流程

方案共有 3 个参与方:签名方、签名持有方、指定验证方,包括以下 6 个算法:

初始化算法 $Setup(\lambda)\rightarrow(s,S,Q)$:签名方选择 p 阶加法循环群 G 和 p 阶乘法循环群 G_T(p 是大素数),G 的生成元 Q 及双线性映射 $e:G\times G\rightarrow G_T$;选择哈希函数 $H:\{0,1\}^*\rightarrow G$;随机选择签名私钥 $s\in Z_p^*$ 并计算签名公钥 $S=sQ$。签名方发布系统公共参数$\langle G,G_T,Q,H,S\rangle$。

密钥生成算法 $KeyGen(G,Q)\rightarrow(v,V)$:指定验证方随机选择签名私钥 $v\in G$,计算并公布指定签名公钥 $V=vQ$。

签名算法 $Sign(m,s)\rightarrow\sigma$:签名持有方将信息 m 发给签名方,后者利用签名私钥 s 对签名持有方需要签名的信息 m 计算 $\sigma=s\cdot H(m)$。将(m,σ)发送给签名持有方。

签名验证算法 $SignVer(m,\sigma)\rightarrow\{0,1\}$:签名持有方收到$(m,\sigma)$后,计算验证 $e(Q,\sigma)=e(S,H(m))$。如果相等则接受(m,σ)为来自签名方的签名凭证。

指定签名算法 $Designation(V,\sigma)\rightarrow\delta$:签名持有方计算指定签名凭证 $\delta=e(V,\sigma)$,将(m,δ)发送给指定验证方。

指定签名验证算法 $DesignationVer(m,\delta)\rightarrow\{0,1\}$:指定验证方计算验证 $\delta=e(vS,H(m))$。如果相等则通过该指定签名的验证,事实上相信签名持有方持有签名方颁发的某个凭证(m,σ)。

3.4.2 方案安全性

Steinfeld 等[26]在指定验证者签名方案基础上实现了通用指定验证者签名方案,并给出了安全性证明,证明了方案具有以下安全属性:

1)指定验证性:由签名持有方生成的指定验证签名只能被指定的验证者验证。

2)不可伪造性:攻击者在仅获得公开参数的情况下无法伪造合法的指定签名。

3)不可转移性:任何指定验证方都能生成与签名方生成的签名不可区分的签名,因而恶意指定验证方在收到指定签名后无法向第三方证明指定签名是由签名方还是指定验证方生成的。

4 方案设计

本文提出的软件定义边界架构下匿名访问方案主要包括

系统建立、注册和匿名访问 3 个阶段。在系统建立阶段，SDP 控制器生成并发布系统公开参数；在注册阶段，SDP 网关和 SDP 客户端向 SDP 控制器提供身份标识和数字证书等信息完成注册，SDP 网关需要向 SDP 控制器提供网络信息和指定签名公钥；在匿名访问阶段，SDP 客户端和 SDP 网关上线并与 SDP 控制器完成 SPA 秘钥协商，SDP 客户端作为签名持有方获得 SDP 控制器的凭证，最后利用该 SPA 密钥完成与 SDP 网关的单包授权认证，利用凭证完成与 SDP 网关的匿名认证，并建立二者间 TLS 双向认证加密信道进行服务资源访问。整体方案符合 SDP 架构标准并可保护访问者的身份和访问记录等隐私信息。

本文假设 SDP 客户端、SDP 网关和 SDP 控制器均持有数字证书，前两者注册与上线时需通过数字证书与 SDP 控制器建立 TLS 安全信道。SDP 客户端是访问者的安全代理，其数字证书是通过该 SDP 客户端访问服务资源的访问者所持有的数字证书。

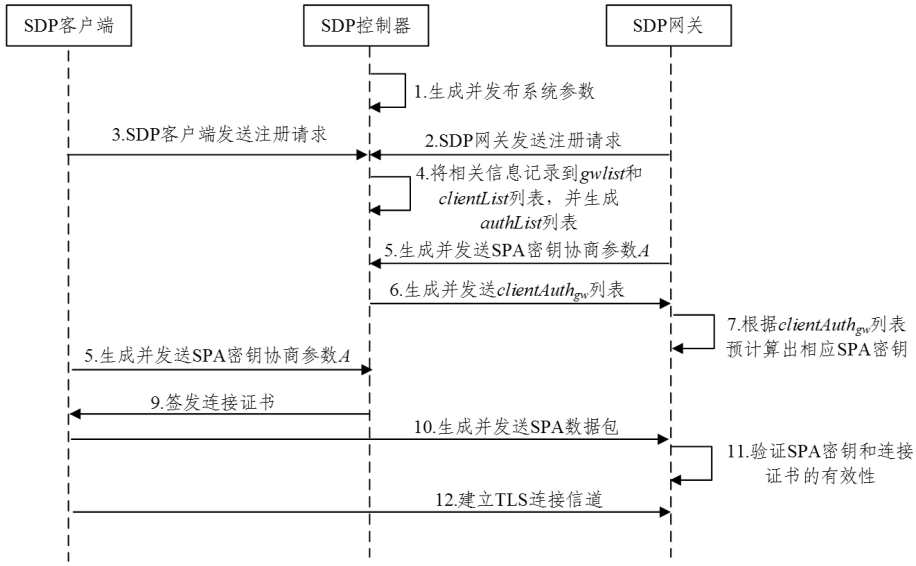

图 2 方案整体流程

Fig. 2 Overall program process

4.1 系统建立阶段

SDP 控制器选择 p 阶加法循环群 G 和 p 阶乘法循环群 G_T（p 是大素数），并选择 G 的生成元 Q；给定双线性映射 e：$G\times G\rightarrow G_T$、哈希函数 H_1：$\{0,1\}^*\rightarrow G$ 和哈希函数 H_2：$\{0,1\}^*\rightarrow Z_p^*$；随机选择签名私钥 $s\in Z_p^*$ 并计算 $S=sQ$；随机选择 $t\in Z_p^*$ 计算 SPA 密钥参数 $T=tQ$，公布系统参数 $\langle G,G_T,e,Q,S,T,H_1,H_2\rangle$。

4.2 注册阶段

SDP 控制器负责管理整个零信任网络中各类服务资源的访问权限，SDP 网关和 SDP 客户端均需要与其交互以完成注册。SDP 控制器维护 SDP 网关列表 *gwList*、SDP 客户端列表 *clientList* 和访问权限列表 *authList*，并在 SDP 网关和 SDP 客户端注册时对列表中的相应内容进行更新。注册流程如图 3 所示。

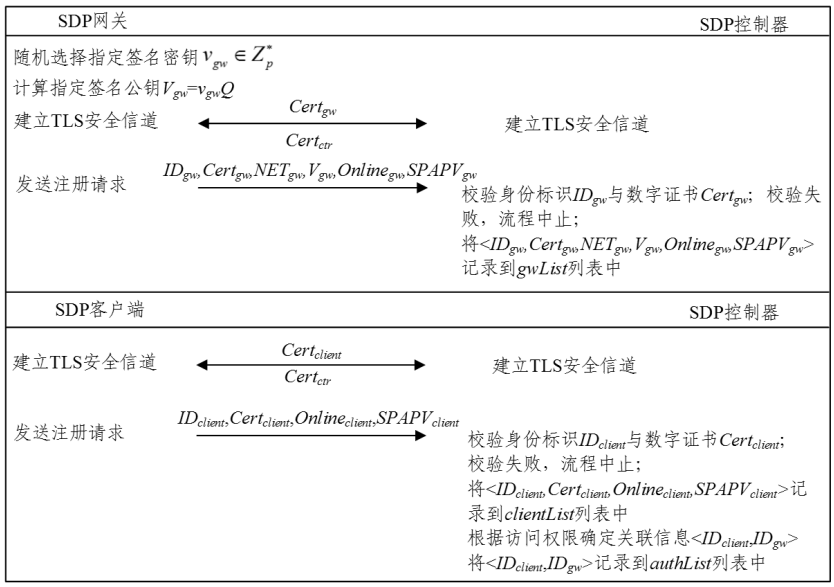

图 3 注册流程

Fig. 3 Registration process

4.2.1 SDP 网关注册

SDP 网关执行以下步骤向 SDP 控制器进行注册:

步骤 1 执行密钥生成算法以生成指定签名私钥 v_{gw},并计算指定签名公钥 $V_{gw}=v_{gw}Q$。

步骤 2 通过数字证书与 SDP 控制器完成双向认证后建立 TLS 安全信道。

步骤 3 向 SDP 控制器发送注册请求。注册请求包括身份标识 ID_{gw}、数字证书 $Cert_{gw}$、访问该 SDP 网关所需的网络信息 Net_{gw}、指定签名公钥 V_{gw}、在线标志 $Online_{gw}$、SPA 密钥协商参数 $SPAPV_{gw}$ 等信息。

步骤 4 SDP 控制器对 SDP 网关的身份标识 ID_{gw} 与数字证书 $Cert_{gw}$ 进行校验,校验通过后将〈ID_{gw},$Cert_{gw}$,Net_{gw},V_{gw},$Online_{gw}$,$SPAPV_{gw}$〉记录到 $gwList$ 列表中。在 SDP 网关未上线时,在线标志 $Online_{gw}$ 为离线状态,SPA 密钥协商参数 $SPAPV_{gw}$ 为空值。

4.2.2 SDP 客户端注册

SDP 客户端执行以下步骤向 SDP 控制器进行注册:

步骤 1 通过数字证书与 SDP 控制器完成双向认证后建立 TLS 安全信道。

步骤 2 向 SDP 控制器发送注册请求。注册请求中包括身份标识 ID_{client}、数字证书 $Cert_{client}$、在线标志 $Online_{client}$、SPA 密钥协商参数 $SPAPV_{client}$ 等信息。

步骤 3 SDP 控制器对 SDP 客户端的身份标识 ID_{client} 与数字证书 $Cert_{client}$ 进行校验;校验通过后将〈ID_{client},$Cert_{client}$,$Online_{client}$,$SPAPV_{client}$〉记录到 $clientList$ 列表中。

步骤 4 SDP 控制器将 SDP 客户端及其被授权访问的 SDP 网关的关联信息〈ID_{client},ID_{gw}〉记录到 $authList$ 列表中。在 SDP 客户端未上线时,在线标志 $Online_{client}$ 为离线状态,SPA 密钥协商参数 $SPAPV_{client}$ 为空值。

4.3 匿名访问阶段

4.3.1 SDP 网关上线

SDP 网关执行以下步骤与 SDP 控制器进行交互(上线流程如图 4 所示),以完成上线:

步骤 1 随机选择私钥 $a\in Z_p^*$,计算 SPA 密钥协商参数 $SPAPV_{gw}=A=aQ$。

步骤 2 将 A 发送给 SDP 控制器。

步骤 3 SDP 控制器查询 $clientList$ 列表和 $authList$ 列表,生成一个包含所有拥有访问权限的 SDP 客户端的 SPA 密钥协商参数的列表 $clientAuth_{gw}$,并发送给 SDP 网关。

步骤 4 对 $clientAuth_{gw}$ 中所有参数预计算出 SPA 密钥 $SPAKEY=H_1(e(T,B)^a)$,并建立索引表 $SPAKEYList_{gw}=\langle B,SPAKEY\rangle$。

4.3.2 SDP 客户端上线

SDP 客户端执行以下步骤与 SDP 控制器进行交互,以完成上线:

步骤 1 随机选择私钥 $b\in Z_p^*$,计算 SPA 密钥协商参数 $SPAPV_{client}=B=b\cdot Q$。

步骤 2 将 B 发送给 SDP 控制器。

步骤 3 SDP 控制器通过 $clientList$ 列表和 $authList$ 列表生成 SDP 客户端可访问的 SDP 网关信息〈ID_{gw},Net_{gw},V_{gw},$SPAPV_{gw}$〉,生成授权访问列表 $gwAuth_{client}$。

步骤 4 SDP 控制器生成连接证书信息 $m=B\parallel Lifetime\parallel ID_{controller}$,并根据 m 和 s 执行签名算法,生成凭证 $\sigma=s\cdot H_2(m)$。

步骤 5 SDP 控制器将 (m,σ)和 $gwAuth_{client}$ 列表发送给 SDP 客户端。

步骤 6 执行签名验证算法对凭证 σ 的有效性进行验证。

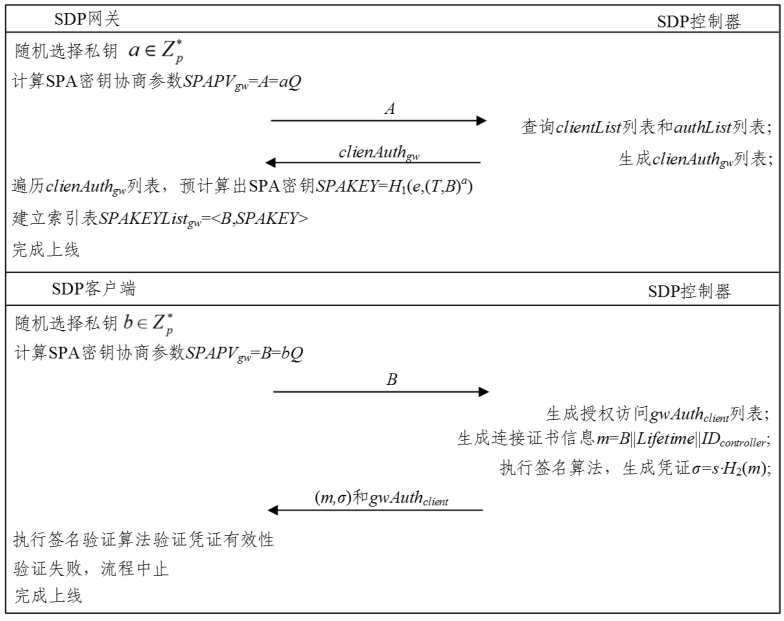

图 4 上线流程

Fig. 4 Online process

4.3.3 匿名单包授权认证

SDP 客户端在访问 SDP 网关时需要根据相应的 SPA 密钥协商参数计算出 SPA 密钥。SDP 客户端和 SDP 网关分别从各自的列表中查询得到相应的 SPA 密钥协商参数,从而

计算出 SPA 密钥。在这一过程中，SDP 客户端和 SDP 网关无需与其他参与方进行交互，SDP 客户端的身份信息也没有向外暴露，从而实现了匿名单包授权认证，认证流程如图 5 所示。

步骤 1 SDP 客户端从 $gwAuth_{client}$ 列表中查询 SDP 网关及其 SPA 密钥协商参数 A。

步骤 2 计算 SPA 密钥 $SPAKEY=H_1(e(T,A)^b)$。

步骤 3 SDP 客户端使用 SPA 密钥 $SPAKEY$ 对 SPA 数据包进行加密和 HMAC 认证后发送给 SDP 网关。

步骤 4 SDP 网关收到 SPA 数据包后，根据字段值 B 检索其是否在 $SPAKEYList_{gw}$ 中，利用查询出的 $SPAKEY$ 解密 SPA 数据包并进行 HMAC 校验。

步骤 5 校验通过后，SDP 网关确认 SDP 客户端的访问权限，在 $clientAuth_{gw}$ 列表中为对应的参数 B 添加验证标志 $SPAauth$，并开放相应端口，准备与 SDP 客户端建立 TLS 安全信道。

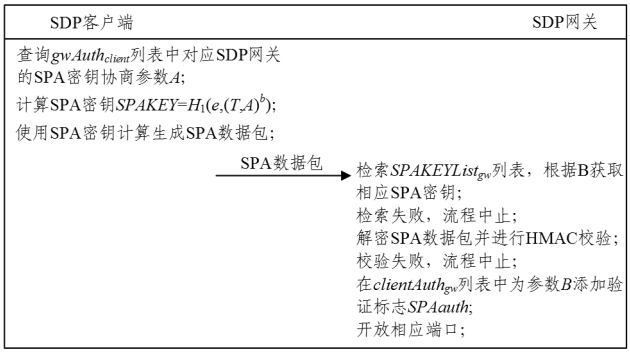

图 5 匿名单包授权认证流程

Fig. 5 Anonymous single packet authorization process

4.3.4 匿名认证访问

在完成匿名单包授权认证后，SDP 网关对 SDP 客户端的连接证书进行校验，SDP 客户端对 SDP 网关的数字证书进行校验。SDP 客户端所出示的连接证书是由 SDP 控制器所签发的，SDP 网关无需与 SDP 控制器或第三方证书颁发机构进行交互，只需要通过指定签名验证算法完成对连接证书的校验。在完成双向认证后，双方建立 TLS 安全信道以访问服务资源。访问流程如图 6 所示。

步骤 1 SDP 客户端执行指定签名算法生成指定签名凭证 δ，向 SDP 网关发送 (m,δ)。

步骤 2 SDP 网关根据连接证书 m 中的关键字 B 查询 $clientAuth_{gw}$ 列表来验证 SDP 客户端访问权限的有效性，并执行指定签名验证算法验证 SDP 客户端所持有的连接证书 m 的正确性，同时根据连接证书 m 的生命周期 $Lifetime$ 来判断连接证书是否过期。

步骤 3 SDP 客户端利用 SDP 网关所持有的数字证书完成对 SDP 网关的身份认证。

步骤 4 SDP 网关根据 $clientAuth_{gw}$ 列表核验 SDP 客户端发送的证书中的公钥 B 对应的验证标志 $SPAauth$ 的有效性，在核验通过后与 SDP 客户端建立 TLS 安全信道。

步骤 5 SDP 客户端通过 TLS 安全信道对服务资源进行访问。

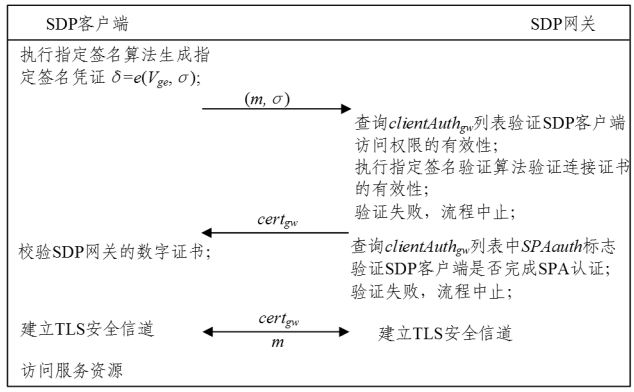

图 6 匿名认证访问流程

Fig. 6 Anonymous authentication access process

5 安全性分析

本文方案相较于标准 SDP 架构实现了更多安全属性，对各类攻击均具有良好的防御效果，同时有效地保护了访问服务资源过程中访问者的隐私信息。

5.1 攻击模型

标准的 SDP 架构通过单包授权机制与 TLS 认证加密机制实现了较为有效的身份认证，并能够保证通信信道的安全性[2]。因此，攻击者无法通过重放攻击、窃听攻击、中间人攻击等常见的网络攻击手段对 SDP 架构产生威胁。但攻击者仍能够通过以下几种攻击手段对 SDP 架构实施网络攻击。

1)SPA 密钥窃取攻击。攻击者通过窃取 SPA 密钥来加密生成有效的 SPA 数据包以完成 SPA 认证。

2)敲门放大攻击。合法 SDP 客户端完成单包授权认证后，由于建立 TLS 安全信道时 SDP 网关对数字证书的校验没有与 SPA 认证机制有效结合，因此同一 NAT 下的攻击者冒用其网络信息抢先与 SDP 网关建立连接，实施敲门放大攻击。

3)隐私攻击。恶意 SDP 网关通过 SPA 密钥与数字证书对 SDP 客户端的身份进行追踪，以获得访问记录等隐私数据。

4)身份假冒攻击。攻击者伪造出经过 SDP 控制器签名的连接证书，假冒合法的 SDP 客户端与 SDP 网关建立连接。

5)SDP 网关共谋。多个恶意的 SDP 网关通过 SDP 客户端与其建立 TLS 连接过程中暴露的身份标识与公钥等信息对 SDP 客户端的访问记录进行恶意分析。

5.2 安全性分析与证明

5.2.1 抗 SPA 密钥窃取

定理 1 假设哈希函数 H_1 是一个随机预言机，攻击者 $\mathcal{A}$ 可在多项式时间内进行 q_{H_1} 次随机预言查询。如果 BDH 问题是困难的，那么攻击者 $\mathcal{A}$ 通过公开参数计算出 SPA 密钥的优势为 $Adv_{\mathcal{A}}\leqslant Adv_{\mathcal{A},\mathrm{BDH}}$，即攻击者 $\mathcal{A}$ 实施 SPA 密钥窃取是困难的。

证明：假设存在攻击者 $\mathcal{A}$ 能够以概率 ε 构造一个模拟器 $\mathcal{S}$ 解决 BDH 困难问题。给定参数 (Q,aQ,bQ,cQ) 作为输入。

随机预言查询：模拟器 $\mathcal{S}$ 维护列表 $List$。攻击者 $\mathcal{A}$ 输入

$\hat{c}_i$，$i\in[1,q_{H_1}]$，其中 q_{H_1} 为随机预言机的查询数量。模拟器 $\mathcal{S}$ 回复 $\hat{C}_i=\hat{c}_iQ$ 给攻击者，并将元组($\hat{c}_i$，$\hat{C}_i$)记录到列表 *List* 中。

挑战：攻击者 $\mathcal{A}$ 输入 $\hat{c}_i$。模拟器对列表 *List* 进行查询，如果 $\hat{C}_i\neq G_C$，则挑战中止；反之将($\hat{c}_i$，$\hat{C}_i$)回复给攻击者 $\mathcal{A}$。

密钥伪造输出：攻击者 $\mathcal{A}$ 输出伪造的密钥对($\hat{c}_i$，$\hat{C}_i$)，可以满足 $e(Q,Q)^{ab\hat{c}_i}=e(Q,Q)^{abc}$，从而得到了 BDH 困难问题的解。

模拟器 $\mathcal{S}$ 的成功概率取决于攻击者 $\mathcal{A}$ 的查询能否满足 $\hat{C}_i=G_C$，因此模拟器 $\mathcal{S}$ 解决 BDH 困难问题的优势为 $Adv_{\mathcal{A}}=\varepsilon/q_{H_1}\leqslant Adv_{\mathcal{A},\mathrm{BDH}}$。故攻击者无法实施 *SPA* 密钥窃取。

5.2.2 抗敲门放大攻击

定理 2 假设哈希函数 H_1 是一个随机预言机，攻击者 $\mathcal{A}$ 可在多项式时间内进行 q_{H_1} 次随机预言查询。如果 DL 问题是困难的，那么攻击者 $\mathcal{A}$ 通过 SPA 密钥协商参数 B 计算出 SDP 客户端生成的私钥 b 的优势为 $Adv_{\mathcal{A}}\leqslant Adv_{\mathcal{A},\mathrm{DL}}$，即攻击者 $\mathcal{A}$ 实施敲门放大攻击是困难的。

证明：在本文方案中，SDP 网关在收到 SPA 数据包后对其中的 SPA 密钥协商参数 B 进行记录，同时在与 SDP 客户端建立 TLS 安全信道时对连接证书中的 B 进行校验。假设攻击者 $\mathcal{A}$ 在 SDP 客户端完成匿名单包授权认证后，窃取到 SDP 客户端的连接证书：

$$m=B\parallel Lifetime\parallel ID_{\mathrm{controller}}$$

攻击者必须通过 SPA 密钥协商参数 B 计算出 SDP 客户端私钥 b 才能与 SDP 网关建立 TLS 连接。假设存在攻击者 $\mathcal{A}$ 能够以概率 ε 构造一个模拟器 $\mathcal{S}$ 解决 DL 困难问题，从而计算出 SDP 客户端私钥 b。给定公开参数 B 作为输入。

随机预言查询：模拟器 $\mathcal{S}$ 维护列表 *List*。攻击者 $\mathcal{A}$ 输入 $\hat{b}_i$，$i\in[1,q_{H_1}]$，其中 q_{H_1} 为随机预言机的查询数量。模拟器 $\mathcal{S}$ 回复 $\hat{B}_i=\hat{b}_iQ$ 给攻击者，并将元组($\hat{b}_i$，$\hat{B}_i$)记录到列表 *List* 中。

挑战：模拟器 $\mathcal{S}$ 对列表 *List* 进行查询，如果 $\hat{B}_i\neq B$，则挑战中止；反之将($\hat{b}_i$，$\hat{B}_i$)回复给攻击者 $\mathcal{A}$。

密钥伪造输出：攻击者 $\mathcal{A}$ 输出伪造的 SDP 客户端私钥 $\hat{b}_i$，可以满足 $\hat{b}_iQ=B$，从而得到了 DL 困难问题的解。

模拟器 $\mathcal{S}$ 的成功概率取决于攻击者 $\mathcal{A}$ 的查询能否满足 $\hat{B}_i=B$，因此模拟器 $\mathcal{S}$ 解决 DL 困难问题的优势为 $Adv_{\mathcal{A}}=\varepsilon/q_{H_1}\leqslant Adv_{\mathcal{A},\mathrm{DL}}$。由于在本文方案中实施敲门放大攻击取决于攻击者 $\mathcal{A}$ 能否伪造 SDP 客户端私钥，因此可以认为攻击者 $\mathcal{A}$ 实施敲门放大攻击是困难的。

5.2.3 访问者身份隐私保护

在本文方案中，SDP 客户端采用上线时生成的 SPA 密钥协商参数 B 作为临时身份进行匿名单包授权认证，SDP 网关无法根据 B 关联到 SDP 客户端的身份信息。在匿名认证访问阶段，SDP 控制器所签发的凭证仅包含 SPA 密钥协商参数 B，SDP 网关可以对指定签名凭证执行验证算法来验证 SDP 客户端身份的有效性，但连接凭证中不包含与 SDP 客户端身份相关联的信息，无法将连接凭证与 SDP 客户端的身份建立关联；同时，根据通用指定验证者签名方案的指定验证性，其他 SDP 网关无法验证该指定签名凭证，也无法将连接凭证与 SDP 客户端进行关联。因此，在 SDP 控制器可信的情况下，SDP 网关无法对 SDP 客户端真实身份进行识别或关联，从而说明本文方案能够有效保护访问者的身份信息。

5.2.4 抗身份假冒

定理 3 假设哈希函数 H_1 是一个随机预言机，攻击者 $\mathcal{A}$ 可在多项式时间内进行 q_{H_1} 次随机预言查询。如果 CDH 问题是困难的，攻击者 $\mathcal{A}$ 伪造出经签名验证算法验证有效的连接证书的优势为 $Adv_{\mathcal{A}}\leqslant Adv_{\mathcal{A},\mathrm{CDH}}$，即攻击者 $\mathcal{A}$ 假冒合法的 SDP 客户端以实施身份假冒攻击是困难的。

证明：假设存在攻击者 $\mathcal{A}$ 能够以概率 ε 在未获取签名方私钥 s、指定验证方私钥 v 以及签名方生成的签名 σ 的情况下为信息 m 生成有效的指定签名 δ，即

$$Pr[DesignationVer(V,m,\delta)=1]=1$$

给定参数(V,S,Q)作为输入。攻击者 $\mathcal{A}$ 构造一个模拟器，由模拟器 $\mathcal{S}$ 随机选择 $x,y\in Z_p$，将签名公钥设置为 $S=\hat{s}Q$，其中 $\hat{s}$ 与 s 是等价的。

随机预言查询：挑战者 $\mathcal{C}$ 选择 $i'\in[1,q_{H_1}]$，并维护初始为空的列表 *List*。对于 m_i 上的查询，当 m_i 在列表 *List* 中时，模拟器 $\mathcal{S}$ 向攻击者 $\mathcal{A}$ 回复此次查询；反之随机选择 w_i，设置 $H_1(m_i)$ 为：

$$H_1(m_i)=\begin{cases}(v+w_i)Q, & i=i'\\ w_iQ, & \text{other}\end{cases}$$

模拟器 $\mathcal{S}$ 向攻击者 $\mathcal{A}$ 回复 $H(m_i)$，并将$(i,m_i,w_i,H(m_i))$添加到列表 *List* 中。

挑战：攻击者 $\mathcal{A}$ 对 m_i 进行查询。当 $i\neq i'$时，有 $H_1(m_i)=w_iQ$。模拟器 $\mathcal{S}$ 计算 $\delta_{m_i}=\hat{s}w_iQ$。

签名伪造：攻击者 $\mathcal{A}$ 在信息 $\hat{m}$ 上生成未经查询的伪造签名 $\delta_{\hat{m}}$。当 $\hat{m}$ 是列表 *List* 上第 i' 个查询时，$H(\hat{m})=(v+w_{\hat{i}})Q$。根据指定签名的定义，有 $\delta_{\hat{m}}=\hat{s}(v+w_{\hat{i}})Q$。模拟器 $\mathcal{S}$ 可计算出 $\delta_{\hat{m}}-s w_{\hat{i}}Q=\hat{s}(v+w_{\hat{i}})Q-\hat{s}w_{\hat{i}}Q=\hat{s}vQ$ 作为 CDH 困难问题的解。

进行查询时所生成的查询回复可以形成列表

$$\{s,w_1,\cdots,w_{\hat{i}-1},v+w_{\hat{i}},w_{\hat{i}+1},\cdots,w_{q_{H_1}}\}$$

假设攻击者 $\mathcal{A}$ 在进行 q_{H_1} 次查询后，成功伪造出了指定签名，则可以认为攻击者 $\mathcal{A}$ 以优势 $Adv_{\mathcal{A}}=\varepsilon/q_{H_1}\leqslant Adv_{\mathcal{A},\mathrm{CDH}}$ 解决了 CDH 困难问题。因此本文方案是抗身份假冒的。

5.2.5 抗 SDP 网关共谋

在标准的 SDP 架构下，SDP 网关能够通过数字证书、SPA 密钥等识别出 SDP 客户端的身份，此时多个恶意 SDP 网关可共谋以大量获取 SDP 客户端的访问记录并挖掘其隐私信息。在本文方案中，SDP 客户端不再使用数字证书与 SDP 网关建立连接，而是使用仅对本次连接有效的连接证书。当与其他 SDP 网关建立连接时，SDP 客户端所持有的连接证书将被更新，因此多个 SDP 网关无法判别两次访问行为之间的关联关系，即两次访问行为是否由同一个 SDP 客户端

进行。因此,即使在多个 SDP 网关共谋的情况下本文方案仍可保证 SDP 客户端的匿名性。

6 性能分析

本文主要从通信效率和计算效率两个方面对本文方案进行实验分析,并基于 Network Simulator 3 平台进行了仿真实验,验证了本文方案在多节点网络环境下的有效性。实验环境参数如表 1 所列。

表 1 实验环境参数

Table 1 Experimental environment parameters

环境参数	描述
操作系统	CentOS 7
CPU	Intel(R) Core(TM) i5-10500 CPU @ 3.10GHz
内存	16GB
开发平台	Visual Studio Code 1.8, Network Simulator 3
开发语言	C++, Python

6.1 通信效率

本文方案的通信开销主要来自于 SDP 控制器向 SDP 客户端和 SDP 网关分发 SPA 密钥协商参数以及 SDP 客户端与 SDP 网关进行单包授权认证。本文方案使用的哈希算法为 SM3,输出长度为 32 Bit;使用的椭圆曲线公钥密码算法为 SM2,密钥长度为 16 Bit;使用的对称加密算法为 SM4,对称密钥长度为 16 Bit;身份标识长度为 8Bit,加法群成员数量长度为 32 Bit。本文使用的通信假设如表 2 所列。

表 2 通信假设

Table 2 Communication hypothesis

符号	描述	长度/Bit
L_{SPA}	SPA 数据包长度	256
L_H	哈希算法输出长度	32
L_{Cert}	TLS 数字证书长度	512
L_{KEY}	SPA 密钥长度	16
L_{Para}	SPA 密钥协商参数长度	8
L_{DC}	指定签名凭证长度	44
L_G	加法群成员数量长度	32
L_{ID}	身份标识长度	8

本节对本文方案、文献[5]所提方案以及 Waverley 实验室推出的开源 SDP 架构[29](以下简称 WaverleySDP)的通信开销进行了分析。SDP 客户端完成一次身份认证所需的通信开销如表 3 所列。相较于 WaverleySDP,文献[5]所提方案对标准 SDP 架构进行了安全增强,但由于缺乏相对高效的 SPA 密钥分发方案,因此其通信开销仍显著高于本文方案。不同于 WaverleySDP 在每次 SDP 客户端访问 SDP 网关时双方都需要与 SDP 控制器建立 TLS 安全信道以获取 SPA 密钥的做法,本文方案在 SDP 客户端或 SDP 网关上线时一次性完成 SPA 密钥协商参数的分发,因而有效减少了通信开销。相较于 WaverleySDP,SDP 客户端完成一次认证的通信开销降低了 35%。

表 3 身份认证通信开销

Table 3 Communication overhead for authentication

	通信开销	通信量/bit
WaverleySDP	$2L_{Cert}+L_{SPA}+2L_{KEY}+L_{ID}$	1320
文献[5]	$2L_{Cert}+L_{SPA}+4L_H+2L_{KEY}+2L_{ID}$	1456
本文方案	$L_{Cert}+L_{SPA}+2L_{Para}+2L_{DC}+2L_G+2L_H$	982

本文基于 Network Simulator 3 平台在仿真环境下对本文方案、文献[5]所提方案和 WaverleySDP 执行一次完整认证流程各参与方的所需通信开销进行了测试,仿真参数基于表 2 中的通信假设进行设置。SDP 客户端、SDP 控制器和 SDP 网关节点之间采用 Network Simulator 3 平台内置的标准网络通信协议模块进行交互,以模拟真实网络环境并提高实验数据的准确性。实验结果(见图 7)表明:本文方案减少了建立 TLS 安全信道的次数,可较大幅度地优化 SDP 客户端与 SDP 网关的通信开销;本文方案对 SPA 密钥的分发过程的优化,可小幅度降低 SDP 控制器的通信开销;相较于 WaverleySDP,本文方案执行一次完整认证流程所需通信开销降低了 33%。

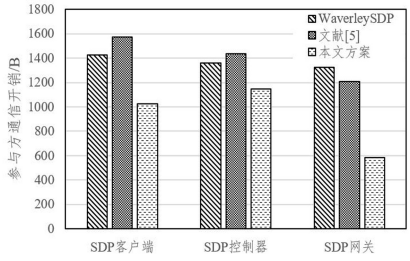

图 7 通信开销对比

Fig. 7 Comparison of communication overhead

6.2 计算效率

本文方案的计算开销主要来自 SPA 密钥的计算和指定签名凭证的生成与验证。本节对本文方案、文献[5]所提方案以及 WaverleySDP 的计算开销进行对比。为评估方案的性能,基于 OpenSSL 对双线性映射、对称密钥加解密、非对称密钥加解密、哈希函数计算、数字签名等密码学操作进行计算。具体计算开销如表 4 所列。

表 4 计算开销

Table 4 Computational overhead

符号	描述	计算开销/ms
T_{bp}	双线性映射	6.479
T_{enc}	非对称加密算法	4.337
T_{dec}	非对称解密算法	5.102
T_s	签名算法	6.539
T_v	签名验证算法	6.945
T_{ed}	对称加解密算法	0.207
T_h	哈希算法	0.037

相较于文献[6]所提方案以及 WaverleySDP,由于本文方案采用三方密钥协商协议来分发 SPA 密钥,因此需要进行较多的双线性映射计算。同时,SDP 网关上线需要对 $clientAuth_{gw}$ 列表中所有参数预计算出 SPA 密钥,因此会在初始阶段产生一定的计算开销。完成一次认证所需要的计算开销对比如表 5 所列。本文方案的计算开销相较于 WaverleySDP 增加了 25%,与文献[5]所提方案计算开销总体持平。尽管本文方案计算开销略大于 WaverleySDP,但本文方案能够抵抗敲门放大攻击等 SDP 架构下的常见攻击方式,且实现了 SDP 客户端与 SDP 网关的匿名认证,因此可以认为本文方案在没有显著增加计算开销的前提下提高了 SDP 架构的整体安全性能。

表 5 身份认证所需计算开销

Table 5 Computational overhead for authentication

	计算开销		计算耗时/ms	计算总耗时/ms
WaverleySDP	SDP 控制器	$2T_{enc}+2T_{dec}+2T_s+2T_v$	45.846	103.788
	SDP 网关	$2T_{enc}+2T_{dec}+16T_{ed}+T_h+T_v$	29.174	
	SDP 客户端	$2T_{enc}+2T_{dec}+16T_{ed}+T_h+T_s$	28.768	
文献[5]	SDP 控制器	$2T_{enc}+2T_{dec}+T_s+T_v+8T_{ed}+2T_h$	34.093	122.741
	SDP 网关	$2T_{enc}+2T_{dec}+T_s+T_v+2T_h+16T_{ed}$	35.75	
	SDP 客户端	$2T_{enc}+2T_{dec}+2T_s+2T_v+2T_h+16T_{ed}$	52.898	
本文方案	SDP 控制器	$3T_{bp}+T_{enc}+2T_{dec}+T_h$	47.905	130.306
	SDP 网关	$2T_{bp}+2T_{enc}+T_{dec}+8T_{ed}+2T_h$	42.355	
	SDP 客户端	$3T_{bp}+2T_{enc}+2T_{dec}+8T_{ed}+2T_h$	40.046	

为分析本文方案在多节点网络环节中的性能表现，本文以 SDP 客户端发起认证请求到完成匿名认证这一过程所需的平均认证时延作为性能指标，将 6.1 节中所给出的通信假设与本节中所给出的密码学操作的计算开销作为基础参数，基于 Network Simulator 3 平台在仿真环境下对本文方案、文献[5]所提方案和 WaverleySDP 进行了测试。仿真参数包括网络节点数量、访问频率、延迟和数据传输速率等，具体的仿真参数值如表 6 所列。

表 6 仿真参数

Table 6 Simulation parameters

仿真参数	参数值
SDP 客户端数量	50，100，150，200，250，300
SDP 网关数量	10，20，30，40，50，60
访问频率	0～4 次每秒
延迟	2 ms
数据传输速率	5 Mbps

实验预设 SDP 控制器已经完成了初始化设置且 SDP 客户端与 SDP 网关均完成了注册；在实验过程中，SDP 客户端随机选择某个 SDP 网关进行访问，以模拟真实网络环境下的访问行为；SDP 客户端的访问频率按照所设定的访问频率随机设置；共进行了 SDP 客户端数量为 50，100，150，200，250，300 的 6 组实验；为模拟真实网络环境中 SDP 客户端访问行为的随机性并提高实验数据的准确性，将 SDP 网关的数量设置为 SDP 客户端数量的 1/5。

相较于 WaverleySDP 和文献[5]所提方案，本文方案在网络内节点数量较多时平均认证时延较低。由于本文方案由计算能力更强的 SDP 网关来预计算 SPA 密钥，避免了 SDP 客户端访问频率增加时 SDP 网关频繁地向 SDP 控制器请求 SPA 密钥，因而有效降低了认证时延。当节点数量增多时，本文方案的整体性能优于 WaverleySDP 和文献[5]所提方案。实验对比如图 8 所示。

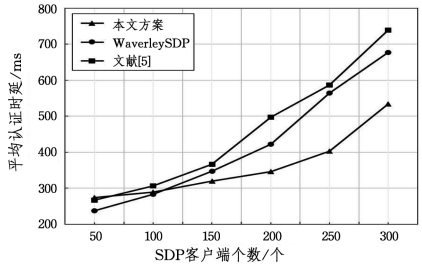

图 8 平均认证时延对比

Fig. 8 Comparison of average authentication delays

6.3 安全功能对比

本文方案和文献[5]所提方案以及 WaverleySDP 的功能对比如表 7 所列。本文方案通过三方密钥协商协议与通用指定验证者签名实现了匿名访问方案，有效保护了访问者的隐私信息，并能够抵抗 SDP 架构下常见的如敲门放大攻击等攻击方式。相较于文献[5]所提方案和 WaverleySDP，本文方案具有更好的安全性能。

表 7 安全功能对比

Table 7 Comparison of security functions

	WaverleySDP	文献[5]	本文方案
抗 SPA 密钥窃取	×	√	√
抗敲门放大攻击	×	×	√
访问者隐私保护	×	×	√
抗身份假冒	√	√	√
抗 SDP 网关共谋	×	×	√

结束语 本文基于三方密钥协商与通用指定验证者签名实现了一种软件定义边界架构下的零信任匿名访问方案，提高了软件定义边界架构的安全性，并有效保护了访问者的隐私信息。实验表明，本文方案降低了通信开销，并实现了更多的安全功能。下一步将尝试对本文方案进行改进，采用更加轻量级的密码算法以提高架构整体计算效率。

参 考 文 献

[1] EVAN G，DOUG B. Zero Trust Networks：Building Secure Systems in Untrusted Networks [M]. Beijing：People's Posts and Telecommunications Publishing House，2019.

[2] GARBIS J，KOILPILLAI J. Software-Defined Perimeter(SDP) Specification v2. 0[J/OL]. https://cloudsecurityalliance. org/artifacts/sdp-architecture-guide-v2.

[3] CHEN B，XU H，XI J F，et al. Power Internet of Things device access management based on cryptographic accumulators [J]. Computer Science，2022，49(S2)：750-755.

[4] WANG F，LI G，WANG Y，et al. Privacy-aware traffic flow prediction based on multi-party sensor data with zero trust in smart city[J]. ACM Transactions on Internet Technology，2023，23(3)：1-19.

[5] WU K H，CHENG R，JIANG X C，et al. Security protection scheme for power Internet of Things based on SDP [J]. Information Network Security，2022，22(2)：32-38.

[6] SAKO K. Topics in Cryptology-CT-RSA 2016[C]// The Cryptographers' Track at the RSA Conference 2016. 2016.

[7] MAJOR W，BUCHANAN W J，AHMAD J. An authentication

protocol based on chaos and zero knowledge proof[J]. Nonlinear Dynamics, 2020, 99: 3065-3087.
[8] RASH M. Single packet authorization with fwknop [J]. The USENIX Magazine, 2006, 31(1): 63-69.
[9] READING D. Fwknop- Port Knocking Tool with Single Packet Authorization[C]// Cyber Warfare and Digital Forensic(CyberSec). IEEE, 2023: 247-252.
[10] ROSSOW C. Amplification Hell: Revisiting Network Protocols for DDoS Abuse[C]// NDSS. 2014: 1-15.
[11] ALI F H M, YUNOS R, ALIAS M A M. Simple port knocking method: Against TCP replay attack and port scanning[C]// 2012 International Conference on Cyber Security, Cyber Warfare and Digital Forensic(CyberSec). IEEE, 2012: 247-252.
[12] KRMELJ G R, PANČUR M, GROHAR M, et al. Openspa-an open and extensible protocol for single packet authorization [C]// Proceedings of the Central European Cybersecurity Conference 2018. 2018: 1-6.
[13] JIANG K, XIAO Y, YUAN S, et al. Implementing Continuous Authentication in Network Connection Based on Improved SPA [C]// 2022 IEEE 22nd International Conference on Communication Technology(ICCT). IEEE, 2022: 1318-1322.
[14] XU M, GUO J, YUAN H, et al. Zero-Trust Security Authentication Based on SPA and Endogenous Security Architecture[J]. Electronics, 2023, 12(4): 782.
[15] BUTAKOV S, ZAVARSKY P, MIRHEYDARI S. Honeykeys: deception mechanisms in single packet authorization[C]// Proceedings of the 14th Pre-ICIS Workshop on Information Security and Privacy. 2019: 1-8.
[16] KRAWCZYK H. Cryptographic extraction and key derivation: The HKDF scheme[C]// Annual Cryptology Conference. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010: 631-648.
[17] BLUNDO C, DE SANTIS A, DI CRESCENZO G, et al. Multi-secret sharing schemes[C]// Annual International Cryptology Conference. Springer Berlin Heidelberg, 1994: 150-163.
[18] ALEXOPOULOS N, KIAYIAS A, TALVISTE R, et al. {MCMix}: Anonymous Messaging via Secure Multiparty Computation[C]// 26th USENIX Security Symposium(USENIX Security 17). 2017: 1217-1234.
[19] MAHMOOD K, ARSHAD J, CHAUDHRY S A, et al. An enhanced anonymous identity-based key agreement protocol for smart grid advanced metering infrastructure[J]. International Journal of Communication Systems, 2019, 32(16): e4137.
[20] JOUX A. A one round protocol for tripartite Diffie-Hellman [C]// International algorithmic Number Theory Symposium. Berlin, Heidelberg: Springer, 2000: 385-393.
[21] ISLAM S K H, BASU S. PB-3PAKA: Password-based three-party authenticated key agreement protocol for mobile devices in post-quantum environments[J]. Journal of Information Security and Applications, 2021, 63: 103026.
[22] SHIM K. Efficient ID-based authenticated key agreement protocol based on Weil pairing[J]. Electronics Letters, 2003, 39(8): 653-654.
[23] TANG F, MA C, CHENG K. Privacy-preserving authentication scheme based on zero trust architecture[J/OL]. https://doi.org/10.1016/j.dcan.2023.01.021.
[24] ZHANG L, LI C, LI Y, et al. Group signature based privacy protection algorithm for mobile ad hoc network[C]// 2017 IEEE International Conference on Information and Automation (ICIA). IEEE, 2017: 947-952.
[25] CAMENISCH J, LYSYANSKAYA A. Signature schemes and anonymous credentials from bilinear maps[C]// Annual International Cryptology Conference. Berlin, Heidelberg: Springer, 2004: 56-72.
[26] STEINFELD R, BULL L, WANG H, et al. Universal designated-verifier signatures [C] // Advances in Cryptology-ASIACRYPT 2003: 9th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, November 30 — December 4, 2003. Proceedings 9. Springer Berlin Heidelberg, 2003: 523-542.
[27] DE ALMEIDA M P, DE SOUSA JÚNIOR R T, GARCIA VILLALBA L J, et al. New dos defense method based on strong designated verifier signatures[J]. Sensors, 2018, 18(9): 2813.
[28] RASTEGARI P, BERENJKOUB M, DAKHILALIAN M, et al. Universal designated verifier signature scheme with non-delegatability in the standard model[J]. Information Sciences, 2019, 479: 321-334.
[29] KOILPILLAI J. Software defined perimeter(SDP) a primer for cios [J/OL]. https://waverleylabs.com/wp-content/uploads/2017/10/waverleylabs-sdp-white-paper.pdf.

LI Weixian, born in 1999, postgraduate. His main research interests include zero-trust networks and privacy protection.

ZENG Junjie, born in 1977, master, lecturer. His main research interests include cryptography and zero-trust networks.

（责任编辑：何杨）