计算机科学
COMPUTER SCIENCE

基于可验证凭证的软件定义边界匿名身份认证方案

司雪鸽, 贾洪勇, 李惟贤, 曾俊杰, 门蕊蕊

引用本文

司雪鸽, 贾洪勇, 李惟贤, 曾俊杰, 门蕊蕊. 基于可验证凭证的软件定义边界匿名身份认证方案[J]. 计算机科学, 2026, 53(1): 363-370.

SI Xuege, JIA Hongyong, LI Weixian, ZENG Junjie , MEN Ruirui. Software-defined Perimeter Anonymous Authentication Scheme Based on Verifiable Credentials [J]. Computer Science, 2026, 53(1): 363-370.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.250100080

基于可验证凭证的软件定义边界匿名身份认证方案

司雪鸽　贾洪勇　李惟贤　曾俊杰　门蕊蕊
郑州大学网络空间安全学院　郑州 450002
(s_xuege@gs.zzu.edu.cn)

摘　要　标准软件定义边界(SDP)架构采用基于访问者身份的认证与授权策略,实时监控与审计访问行为。但访问者需完全披露身份信息以获取访问权限,可能泄露与服务无关的敏感数据,从而带来隐私风险。针对当前软件定义边界架构下存在的用户隐私信息难以得到有效保护、访问记录容易遭受恶意关联等问题,提出一种适用于软件定义边界架构的基于可验证凭证的匿名认证方案。基于双线性映射和 CL 签名构建可验证凭证的验证算法,并将可验证凭证体系与标准软件定义边界架构相融合,在不改变原有基于单包授权与 TLS 安全连接认证模式的前提下实现用户匿名访问。理论分析表明,此方案能够抵抗敲门放大攻击、身份仿冒攻击等常见的网络攻击。实验结果表明,此方案在多节点网络环境进行身份认证所产生的时延更短。

关键词:软件定义边界;可验证凭证;匿名认证;CL 签名;隐私保护
中图分类号　TP309

Software-defined Perimeter Anonymous Authentication Scheme Based on Verifiable Credentials

SI Xuege,JIA Hongyong,LI Weixian,ZENG Junjie and MEN Ruirui
College of Cyber Science and Engineering,Zhengzhou University,Zhengzhou 450002,China

Abstract　The standard SDP architecture employs identity-based authentication and authorization strategies to monitor and audit access activities in real time. However,users must fully disclose their identity information to obtain access,potentially exposing sensitive data unrelated to the requested service and introducing privacy risks. To address challenges such as ineffective user privacy protection and vulnerability of access records to malicious linkage in the current SDP architecture,this paper proposes an anonymous authentication scheme based on verifiable credentials(VCs) for SDP. The scheme constructs a VC verification algorithm using bilinear pairing and CL-signature,integrating the VC system with the standard SDP architecture to enable anonymous user access without altering the original single-packet authorization and TLS secure connection authentication model. Theoretical analysis demonstrates that the proposed scheme resists common network attacks,including knock amplification and identity impersonation. Experimental results show that it achieves shorter authentication latency in multi-node network environments.

Keywords　Software defined perimeter,Verifiable credentials,Anonymous authentication,CL-signature,Privacy preservation

1　引言

软件定义边界(Software-Defined Perimeter,SDP)架构[1]作为零信任安全理念中的一种典型实现架构,以良好的可扩展性与安全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中广泛应用[2-4]。云安全联盟于 2022 年发布了 SDP 架构 2.0 标准[1]。如图 1 所示,SDP 架构基于单包授权机制[5]实现,其组成部分包括 SDP 控制器、SDP 网关和 SDP 客户端。

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”,其安全设计基于单包授权机制和双向传输层安全协议(Transport Layer Security,TLS)认证加密机制,满足了零信任网络的认证性、机密性和完整性等安全需求,在抵抗横向攻击、提升网络安全性和强化身份验证方面取得实际效果[6],但在隐私保护机制上仍存在缺失。云安全联盟发布的 SDP 架构 2.0 标准中,基于共享密钥的单包授权(Single Packet Authorization,SPA)未对用户身份标识加密,且用户与 SDP 网关之间的 TLS 身份认证依赖数字证书[7],这些环节均会导致访问者身份信息和访问记录泄露。因此,如何在身份认证过程中增强用户隐私信息的安全保护,是当前 SDP 架构亟待解决的问题[8]。

为实现 SDP 架构身份认证环节的匿名性与安全性,在软

到稿日期:2025-01-13　返修日期:2025-05-10
基金项目:河南省重点研发专项(231111211900);河南省重大科技专项(221100210900)
This work was supported by the Key Research and Development Projects of Henan Province(231111211900) and the Management of Major Science and Technology of Henan Province(221100210900).
通信作者:贾洪勇(jiahy_pla@126.com)

件定义边界架构下提出了一种基于属性的软件定义边界身份隐私保护访问方案。具体研究内容如下：

1)通过将可验证凭证(Verifiable Credentials，VC)引入软件定义边界架构，将其原有的基于身份的访问控制策略转变为基于属性的访问控制策略。

2)基于双线性映射与 CL(Camenisch-Lysyanskaya)签名[9]构建可验证凭证的生成、聚合与验证算法，在有效认证访问者身份的前提下，实现身份隐私信息的最小披露。

3)将可验证凭证与软件定义边界架构原有的 SPA 认证、TLS 安全连接等安全机制进行有效耦合，确保访问者身份一致性，以抵御敲门放大攻击。

4)通过安全性与性能分析实验，证明了本文方案对多种网络攻击具有较强抵御能力，且具备高可用性和高效性。

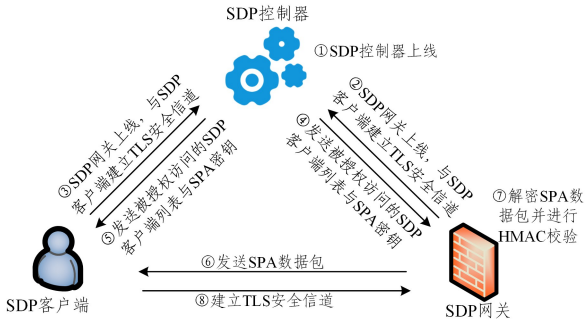


图 1 SDP 架构工作流程

Fig. 1 Workflow of SDP architecture

2 相关工作

当前已有部分研究者对可应用于常见零信任架构下的隐私保护措施进行了研究。文献[10]提出了一种适用于物联网环境的基于无证书策略的轻量级身份认证方式，对访问方的身份信息进行了隐藏，但验证方仍能对访问方的访问行为进行关联。文献[11]提出了一种跨链交易协议以实现零信任物联网环境下身份验证的安全性与匿名性，但所适用的应用场景受到较大限制。文献[12]基于通用指定验证者签名实现了零信任架构的隐私保护认证方案，但计算开销较大，难以应用到资源受限的物联网设备上。文献[13]通过区块链智能合约来实现凭证的生成与认证，有效保护了用户的隐私信息，但当迁移到具有较多参与方的网络环境中时，系统性能会受到较大影响。文献[14]基于区块链与数字签名构建了零信任物联网设备身份验证模块 BasIoT 以实现设备身份隐私验证，但集中式的身份管理模块容易成为网络攻击的目标，导致隐私数据泄露。文献[15]结合区块链与交换式零知识证明来实现物联网环境下的匿名身份验证与隐私保护，但在认证过程中带来了大量的计算开销与存储开销，无法应用到性能较差的物联网终端设备上。文献[16]采用差分隐私算法实现对用户隐私信息的保护，并能适应零信任架构下不断变化的信任关系，但需要引入过高的隐私预算，导致无法准确识别用户身份。可验证凭证[17]作为一种新兴认证工具，正被越来越多地应用到匿名认证与身份管理中[18-20]，有效地规避了身份认证过程中用户的隐私数据泄露与访问行为被恶意关联。

3 预备知识

3.1 双线性映射

设$\mathbb{G}$，$\mathbb{G}_T$是素数阶群，若一个映射 $e:\mathbb{G}\times\mathbb{G}\rightarrow G_T$ 满足以下 3 条性质，则称映射 e 为双线性映射[9]：

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

2)非退化性：$\exists g_1,g_2\in\mathbb{G}$，$e(g_1,g_2)\neq 1$。

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

3.2 可验证凭证

可验证凭证[17]是一种用于表示实体属性的数字证书，可以在提高身份验证过程的安全性、互操作性与可靠性的同时，有效保护访问者隐私信息。可验证凭证通过数字签名机制使其具备不可伪造性与不可篡改性，相较于传统的物理凭证，可验证凭证具有更高的可信度。

有效的可验证凭证包括 3 个组成部分：关于实体属性信息的属性声明(Claims)、由发证方签名生成用于验证可验证凭证真实性和完整性的数字签名(Proof)、包括但不限于发证方信息与有效期等内容的元数据(Metadata)。基于可验证凭证的认证流程如图 2 所示。当用户需要进行认证时，首先请求发证方为用户签发包含属性信息的 VC。VC 中包含了发证方的数字签名以证明 VC 的真实性和可信度，用户获得 VC 的所有权后成为持证方。

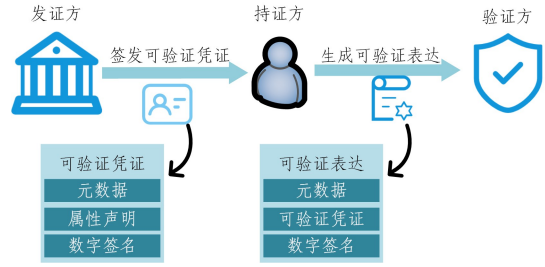


图 2 可验证凭证工作流程

Fig. 2 Workflow of verifiable credentials

在进行验证前，持证方根据需要选择性地从其 VC 集合中选择一个或多个 VC，并创建一个可验证表达(Verifiable Presentation，VP)发送给验证方。验证方对 VP 中的每个 VC 的真实性与有效性进行验证，并检查 VP 中的数字签名、发证方信息、有效期等内容，以确保 VP 的合法性。验证成功后，验证方可以信任 VP 中包含的一个或多个属性信息，而不需要直接和发证方通信。证明可验证凭证体系可以由用户控制其属性信息的披露范围，使用户的隐私和安全得到有效的保护。

3.3 CL 签名

CL 签名[9]是 2004 年提出的一种基于属性的签名方案，可在需要保护用户隐私的场景中提供灵活和高效的数字签名功能。CL 签名的核心优势在于其不仅仅对整个消息或文档进行签名，更支持对用户的特定属性进行签名，使得 CL 签名非常适用于实现可验证凭证和基于属性的凭证系统。CL 签名方案基于椭圆曲线密码学、双线性配对和强 RSA 假设等密

码学理论实现,其基本过程包括密钥生成、签名生成和签名验证 3 个阶段。

本文所用的 CL 签名算法基于双线性映射和 LRSW[21]假设构建,签名过程的具体细节描述如下。

1)密钥生成阶段:给定素数 p 阶乘法循环群$\mathbb{G}$和$\mathcal{G}_T$,$\mathbb{G}$的生成元 g 以及双线性映射 $e:\mathbb{G}\times\mathbb{G}\rightarrow\mathcal{G}_T$ 作为系统公开参数。签名者随机选择签名私钥对 $x,y\in\mathbb{Z}_p$ 与属性私钥 $z_i\in\mathbb{Z}_p$,$i\in[1,L]$,L 为签名属性数量,并计算签名公钥对 $X=g^x$,$Y=g^y$,$Z_i=g^{z_i}$,$i\in[1,L]$。签名者保存私钥$(x,y,z_1,z_2,\cdots,z_L)$,并公布公钥$(X,Y,Z_1,Z_2,\cdots,Z_L)$。

2)签名阶段:对于给定消息$(m_0,m_1,\cdots,m_L)$、签名者私钥$(x,y,z_1,z_2,\cdots,z_L)$和签名者公钥$(X,Y,Z_1,Z_2,\cdots,Z_L)$,签名者随机选取 $a\in\mathbb{G}$,计算 $A_i=a^{z_i}$,$c=a^{x+xym_0}\prod_{i=1}^{L}A_i^{xym_i}$,$b=a^y$,$B_i=(A_i)^y$,$i\in[1,L]$。接下来输出消息$(m_0,m_1,\cdots,m_L)$的每个对应的签名内容$(a,\{A_i\},b,\{B_i\},c)$,$i\in[1,L]$。

3)验证阶段:对于消息$(m_0,m_1,\cdots,m_L)$、签名者公钥$(X,Y,Z_1,Z_2,\cdots,Z_L)$和签名$(a,\{A_i\},b,\{B_i\},c)$,$i\in[1,L]$,验证者执行如下计算对签名进行验证:

$$e(a,Z_i)=e(g,A_i) \tag{1}$$

$$e(a,Y)=e(g,b) \tag{2}$$

$$e(A_i,Y)=e(g,B_i) \tag{3}$$

$$e(X,a)\cdot e(X,b)^{m_0}\cdot\prod_{i=1}^{L}e(X,B_i)^{m_i}=e(g,c),i\in[1,L] \tag{4}$$

当且仅当以上等式成立时,签名验证有效。

4 协议设计

4.1 整体架构

本文提出的基于可验证凭证的软件定义边界架构匿名认证方案主要包括系统初始化、注册、可验证凭证聚合和身份认证 4 个阶段。方案整体架构如图 3 所示。

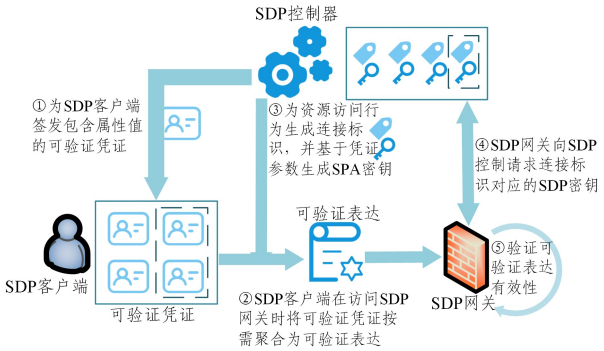


图 3 方案整体架构

Fig. 3 Overall structure of the proposed solution

在系统初始化阶段,由 SDP 控制器生成并发布系统公开参数;在注册阶段,SDP 控制器为 SDP 客户端的属性信息签发 VC,SDP 客户端对 VC 的有效性与合法性进行校验;在 VC 聚合阶段,SDP 客户端根据所访问的 SDP 网关的属性要求按需对 VC 进行聚合以生成 VP;在身份认证阶段,SDP 客户端向 SDP 控制器请求生成连接标识以及对应的 SPA 密钥,通过 SPA 密钥将 VP 加密为 SPA 数据包发送给 SDP 网关。SDP 网关根据连接标识向 SDP 控制器请求相应的 SPA 密钥,对收到的 SPA 数据包进行解密与可验证表达校验。校验通过后,开放相应端口以提供服务资源。

4.2 初始化阶段

SDP 控制器选择素数 p'并生成安全大素数 $p=2p'+1$,选择 p 阶乘法循环群$\mathbb{G}$和G_T,并选择$\mathbb{G}$的生成元 g;给定双线性映射 $e:\mathbb{G}\times\mathbb{G}\rightarrow\mathcal{G}_T$ 和哈希函数 $H:\{0,1\}^*\rightarrow\mathbb{Z}_p^*$;随机选择 $u\in\mathbb{Z}_p^*$ 与 $v\in\mathbb{Z}_p^*$ 并计算 $U=g^u$ 与 $V=g^v$。SDP 控制器公布公开参数$\langle\mathbb{G},\mathcal{G}_T,g,H,U,V,e,p\rangle$。

4.3 注册阶段

4.3.1 SDP 客户端注册

在注册阶段,SDP 客户端需要与 SDP 控制器建立 TLS 安全信道,并从 SDP 控制器获得为其生成的可验证凭证以及与可验证凭证关联的随机数。

Step1 SDP 客户端随机选择私钥 $sk_{CL}\in\mathbb{Z}_p^*$,并计算公钥 $pk_{CL}=g^{sk_{CL}}$。

Step2 SDP 客户端通过数字证书与 SDP 控制器进行双向认证,在 SDP 控制器对 SDP 客户端的数字证书校验通过后,SDP 控制器与 SDP 客户端建立 TLS 安全信道。

Step3 SDP 客户端向 SDP 控制器发送注册请求,提供包括 SDP 客户端的公钥 pk_{CL}等信息。

Step4 SDP 控制器确定 SDP 客户端的属性信息集合 $\{Attr_i\}_{i=1}^{L}$,并为 SDP 客户端执行可验证凭证生成算法,如算法 1 所示。

算法 1 可验证凭证生成算法

输入:属性集合$\{\mathrm{Attr_i}\}_{i=1}^{L}$,u,v,$\mathbb{G}$,e

输出:可验证凭证集合$\{\mathrm{VC_i}\}_{i=1}^{L}$,$\{\mathrm{R_i}\}_{i=1}^{L}$

1. / * 初始化随机数列表 * /
2. $\{\mathrm{R_i}\}_{i=1}^{L}=\{\ \}$
3. w=rand($\mathbb{G}$)
4. / * 为属性集合中的每个属性生成可验证凭证 * /
5. for each i in L
6. / * 为可验证凭证生成随机数 * /
7. $\mathrm{r_i}=\mathrm{rand}(\mathbb{Z}_p^*)$,$\mathrm{R_i}=\mathrm{g^{r_i}}$,$\{\mathrm{R_i}\}_{i=1}^{L}$.add($\mathrm{R_i}$)
10. $\mathrm{W_i}=\mathrm{w^{r_i}}$,$\mathrm{t}=\mathrm{w^v}$,$\mathrm{T_i}=\mathrm{W_i^v}$,$\mathrm{s_i}=\mathrm{W_i^{uvAttr_i}}$
11. / * 为可验证凭证生成签名 * /
12. $\sigma_i=(\mathrm{w},\mathrm{W_i},\mathrm{t},\mathrm{T_i},\mathrm{s_i})$
13. / * 将属性和签名写入可验证凭证 * /
14. $\mathrm{VC_i}$.claims=$\mathrm{Attr_i}$
15. $\mathrm{VC_i}$.proof=σ_i
16. end for

Step5 SDP 控制器将$\{VC_i\}_i^L$ 和$\{R_i\}_{i=1}^{L}$发送给 SDP 客户端。

Step6 SDP 客户端通过计算判断以下等式是否成立:

$$e(w,R_i)=e(g,W_i) \tag{5}$$

$$e(w,V)=e(g,t) \tag{6}$$

$$e(W_i,V)=e(g,T_i) \tag{7}$$

$$e(g,s_i)=e(U,T_i^{Attr_i}),i\in[1,L] \tag{8}$$

当上述等式成立时,SDP 客户端可以相信从 SDP 控制器

收到的可验证凭证的正确性，完成注册流程。

4.3.2 SDP 网关注册

在注册阶段，SDP 网关需要与 SDP 控制器建立 TLS 安全信道，并向其提供访问控制策略。

Step1 SDP 网关随机选择私钥 $k \in \mathbb{Z}_p^*$ 并计算公钥 $K=g^k$，同时生成访问相应服务资源所需满足的访问控制策略 $\{\mathcal{A}_j\}_{j=1}^n$，$n$ 为访问控制策略相关的属性数量。

Step2 SDP 网关与 SDP 控制器进行双向认证。在 SDP 控制器对 SDP 网关的数字证书校验通过后，与 SDP 网关建立 TLS 安全信道。

Step3 SDP 网关将公钥 K、访问该 SDP 网关所需网络信息 Net_{gw} 和访问相应服务资源所需满足的访问控制策略 $\{\mathcal{A}_j\}_{j=1}^n$ 通过 TLS 安全信道发送给 SDP 控制器。

Step4 SDP 控制器收到并确认 SDP 网关发送的信息，SDP 网关完成注册。

4.4 可验证凭证聚合阶段

在 SDP 客户端访问 SDP 网关前，需要按照相应的访问控制策略对所持有的可验证凭证进行聚合以生成可验证表达，通过可验证表达向 SDP 网关展示其身份。

Step1 SDP 客户端访问 SDP 网关前，首先向 SDP 控制器发送包含 SDP 网关标识符 ID_{GW} 的访问请求。

Step2 SDP 控制器根据 SDP 网关的身份标识，检索相应的访问控制策略 $\{\mathcal{A}_j\}_{j=1}^n$，并生成 SDP 客户端进行本次认证的连接标识 c、SPA 密钥 $SPAKEY$ 与仅用于本次访问的连接证书 $\mathcal{M}=pk_{CL} \| c \| Lifetime \| ID_{Ctr}$。

Step3 SDP 控制器计算 $\delta_1=H(SPAKEY)^u$，$\delta_2=H(c)^u$ 和 $\delta_3=H(\mathcal{M})$，完成对连接证书的签名。

Step4 SDP 控制器向 SDP 客户端发送 $(\delta_1,\delta_2,\delta_3,w^{u+uvc},c,SPAKEY,\mathcal{M},\{\mathcal{A}_j\}_{j=1}^n)$，并建立访问连接列表 $ConnectList=\langle c,ID_{GW},SPAKEY\rangle$。

Step5 SDP 客户端根据 $\{\mathcal{A}_j\}_{j=1}^n$ 选择对应的可验证凭证集合 $\{VC_j\}_{j=1}^n$ 执行可验证凭证聚合算法以生成可验证表达，算法描述如算法 2 所示。

算法 2 可验证凭证聚合算法

输入：可验证凭证集合 $\{VC_i\}_{i=1}^L$，w^{u+uvc}，c，S，δ_1，δ_2，δ_3，e，K，V，W_i

输出：可验证表达 VP

```
1. /* 初始化属性列表与可验证凭证列表 */
2. AttrList={},VCList={}
3. /* 选择满足访问控制策略的可验证凭证 */
4. for each VCi in {VCi}^L_{i=1}
5.     if VCi. claims in {Aj}^n_{j=1}
6.         AttrList. add(VCi. claims)
7.         VCList. add(VCi)
8. /* 判断是否满足访问控制策略 */
9.     if AttrList={Aj}^n_{j=1}
10.        coutinue
11.    else
12. return
13. /* 判断 SDP 控制器提供的连接标识与 SPA 密钥的有效性 */
14. if e(g,δ1)=e(V,H(c)) and e(g,δ2)=e(V,H(SPAKEY))
15.    δ̂1=e(K,δ1)
16.    δ̂2=e(K,δ2)
17.    end if
18. /* 判断经过 SDP 控制器签名的连接证书的有效性 */
19.    if e(g,δ3)=e(V,H(M))
20.        δ̂3=e(K,δ3)
21.    end if
22. /* 对属性进行聚合 */
23.    S=w^{u+uvc} ∏_{j=1}^n W_j^{uvAttrj}
24. /* 将可验证凭证、签名和连接证书写入可验证表达 */
25.    VP. verifiableCredentials=VCList
26.    VP. proof=(c,S,δ̂1,δ̂2,δ̂3)
27.    VP. matedata=(c,M)
```

4.5 身份认证阶段

SDP 客户端在访问 SDP 网关前将可验证表达附加到 SPA 数据包中。SDP 网关对连接标识校验完成后，向 SDP 控制器请求 SDP 密钥以对 SPA 数据包以及可验证表达进行校验。校验完成后向 SDP 客户端提供服务资源。SDP 客户端执行以下流程以完成身份认证与服务资源访问。

Step1 SDP 客户端在生成可验证表达 VP 后，使用 SPA 密钥 SPAKEY 加密生成 UDP 格式的 SPA 数据包，并将连接标识 c 置于数据报头部。

Step2 SDP 客户端将加密生成的 SPA 数据包发送给 SDP 网关。

Step3 SDP 网关收到 SPA 数据包后，对连接标识 c 计算 $\hat{c}=e(U,H(c)^k)$。

Step4 SDP 网关向 SDP 控制器发送认证请求消息，其中包括接收到的连接标识 $(c,\hat{c})$，以获取到相应的 SPA 密钥。

Step5 SDP 控制器对连接标识 $\hat{c}$ 进行验证，对 SDP 网关发送的连接标识的正确性进行验证，当且仅当 $\hat{c}=e(K^u,H(c))$ 时验证通过。

Step6 SDP 控制器基于 SDP 网关注册阶段建立的 TLS 安全信道向 SDP 网关传输相应的 SPA 密钥 SPAKEY。

Step7 SDP 网关对 SPA 数据包进行解密，对 SPA 数据包中的可验证表达进行验证。当且仅当以下等式成立时，SDP 网关认为 SDP 客户端具有访问权限。

$$\hat{\delta}_1=e(U^k,H(c)) \tag{9}$$

$$\hat{\delta}_2=e(U^k,H(SPAKEY)) \tag{10}$$

$$\hat{\delta}_3=e(U^k,H(\mathcal{M})) \tag{11}$$

$$e(U,w)\cdot e(U,t)^c \prod_{j=1}^n e(U,T_j)^{Attr_j}=e(g,S) \tag{12}$$

校验完成后，SDP 网关打开相应端口准备与 SDP 客户端建立 TLS 安全连接以提供服务资源。

Step8 SDP 客户端对 SDP 网关的数字证书进行校验，校验完成后基于连接证书 $\mathcal{M}$ 与 SDP 网关建立 TLS 连接，访问相关服务资源。

5 正确性及安全性分析

5.1 正确性分析

式(13)－式(16)的正确性推导验证如下:

$$\begin{aligned}
&e(U,w)\cdot e(U,t)^{c}\prod_{j=1}^{n}e(U,T_j)^{Attr_j}\\
&=e(g^{u},w)\cdot e(g^{u\cdot},w^{v})^{c}\prod_{j=1}^{n}e(g^{u},W_j^{v})^{Attr_j}\\
&=e(g^{u},w)\cdot e(g^{u\cdot},w^{v})^{c}\prod_{j=1}^{n}e(g^{u},W_j^{v})^{Attr_j}\\
&=e(g,w)^{u}\cdot e(g,w)^{uvc}\prod_{j=1}^{n}e(g,W_j^{v})^{uAttr_j}\\
&=e(g,w)^{u+uvc}\prod_{j=1}^{n}e(g,W_j^{uvAttr_j})\\
&=e(g,w^{u+uvc})e(g,\prod_{j=1}^{n}W_j^{uvAttr_j})\\
&=e(g,w^{u+uvc}\prod_{j=1}^{n}W_j^{uvAttr_j})\\
&=e(g,S)
\end{aligned}\tag{13}$$

$$\begin{aligned}
e(w,R_i)&=e(w,g^{r_i})=e(g,w^{r_i})\\
&=e(W,g_i)=e(g,W_i)
\end{aligned}\tag{14}$$

$$\begin{aligned}
e(w,V)&=e(w,g^{v})=e(w^{v},g)\\
&=e(t,g)=e(g,t)
\end{aligned}\tag{15}$$

$$\begin{aligned}
e(W_i,V)&=e(w^{r_i},g^{v})=e(w^{vr_i},g)\\
&=e(T_i,g)=e(g,T_i)
\end{aligned}\tag{16}$$

$$\begin{aligned}
e(g,s_i)&=e(g,W_i^{uvAttr_i})=e(g,w^{r_iuvAttr_i})=e(g^{u},w^{r_ivAttr_i})\\
&=e(U,W_i^{vAttr_i})=e(U,T_i^{Attr_i})
\end{aligned}\tag{17}$$

5.2 安全性分析

5.2.1 形式化安全模型

本文基于适应性选择消息攻击模型(EU-CMA)定义方案的安全性,攻击者可访问以下预言机。

1)签名预言机(O_{Sign}):攻击者提交属性集合 A,预言机返回对应的可验证凭证 VC_A。

2)验证预言机(O_{Verify}):攻击者提交可验证表达 VP,预言机返回验证结果。

若攻击者在多项式时间内以不可忽略的概率伪造有效可验证凭证或篡改凭证属性,则称方案不安全。

5.2.2 可验证凭证抗篡改攻击

定理 1 若存在概率多项式时间(PPT),攻击者 A 能够以不可忽略优势 ε 篡改可验证凭证中的属性值,则存在挑战者 $\mathcal{B}$ 以 $\varepsilon\geqslant\varepsilon-negl(\lambda)$ 优势解决 DBDH 问题,其中 $negl(\lambda)$ 为可忽略函数。

证明 假设挑战者 $\mathcal{B}$ 接受问题实例 (g,g^{a}),能够突破 DBDH 问题求解 $a\in\mathbb{Z}_p^{*}$。$\mathcal{B}$ 为模仿方案环境,先生成系统参数 $(p,,e,g,X=g^{x},Y=g^{y},Z=g^{z},w=rand(\mathbb{G}))$,公开公钥参数 $PK=(g,X,Y,Z)$,保存私钥参数 $SK=(x,y,z)$,并维护列表 L_{VC} 来记录已签发的可验证凭证。

攻击者 $\mathcal{A}$ 可查询 O_{Sign} 获取任意属性集合 A_i 的 VC_{A_i}。若能成功输出篡改后的凭证 $VC_{A'}^{*}=(A',\sigma')$,且 $A'\not\subset A_i$ 通过验证,才能实现篡改攻击。

设篡改后的 $VC_{A'}^{*}$ 签名为 $\sigma'=(w',W',t',T',s')$,要求能够满足 $W=w^{r}$,$t=w^{x}$,$T=W^{x}$,$s=W^{xyA}$。如果攻击者 $\mathcal{A}$ 成功篡改,则挑战者 $\mathcal{B}$ 可以提取其中的 $W^{xy}=w^{rxy}$,即从 X,Y 中推导出 g^{xy}。

挑战者 $\mathcal{B}$ 即使能够获取属性相关随机数 r,还要能够解决 DBDH 问题,才能将攻击者的优势扩大。这与 DBDH 问题困难相矛盾,可得出方案能抗篡改攻击,满足 EU-CMA 安全性。

5.2.3 可验证表达抗伪造攻击

定理 2 若存在 PPT,攻击者 $\mathcal{A}$ 以不可忽略优势 ε 伪造有效可验证表达,则存在挑战者 $\mathcal{B}$ 以优势 $\varepsilon'\geqslant\varepsilon-negl(\lambda)$ 解决 DBDH 问题,其中 q_{Sign} 为签名查询次数。

证明 同定理 1,挑战者 $\mathcal{B}$ 进行方案环境模拟。首先由攻击者 $\mathcal{A}$ 进行想要伪造的可验证表达包含的属性 A 中包含 q_{Sign} 个属性的签名查询,获取 VC_{A_i},$i\in[1,q_{\text{Sign}}]$。然后攻击者 $\mathcal{A}$ 根据 $(\delta_1,\delta_2,\delta_3,w^{x+xyc},c,\{\mathcal{A}_j\}_{j=1}^{q_{\text{Sign}}})$ 输出伪造可验证表达 VP_A^{*}。

伪造的 VP_A^{*} 签名要求能够满足 $VP_A^{*}.proof=(c,S,\overset{\wedge}{\delta}_1,\overset{\wedge}{\delta}_2,\overset{\wedge}{\delta}_3)$,其中:

$$\overset{\wedge}{\delta}_1=e(Z,\delta_1)\tag{17}$$

$$\overset{\wedge}{\delta}_2=e(Z,\delta_2)\tag{18}$$

$$\overset{\wedge}{\delta}_3=e(Z,\delta_3)\tag{19}$$

$$S=w^{x+xyc}\prod_{j=1}^{q_{\text{Sign}}}W_j^{xyA_j}\tag{20}$$

如果攻击者 $\mathcal{A}$ 想要成功伪造可验证表达,则需要首先伪造其中的聚合属性元素 W^{xyA_i},即从 X,Y 中推导出 g^{xy}。同定理 1,规约到 DBDH 问题,可得出方案能抗伪造攻击,满足 EU-CMA 安全性。

5.2.4 访问者身份隐私保护

本文方案能够有效保护访问者的身份信息。在本文方案中,SDP 客户端在身份认证与服务资源访问阶段向 SDP 网关出示的可验证表达不包含 SDP 客户端的真实身份信息,仅使用 SDP 控制器生成的连接标识 c 作为本次连接的标识符。同时,可验证表达内嵌入的可验证凭证也不包含与访问者存在关联的信息,仅包含单一的属性信息与 SDP 控制器对其生成的数字签名,因而访问者的身份信息没有向外暴露。同时,在标准软件定义边界架构下,SDP 客户端需要使用数字证书与 SDP 网关建立连接,而在本文方案中,数字证书仅在 SDP 客户端注册阶段与 SDP 控制器交互时使用,与 SDP 网关建立 TLS 连接时所使用的证书是 SDP 控制器为本次连接所签发的连接证书,而连接证书不包含任何与访问者身份或属性相关的信息。因此,本文方案可以有效保护访问者的身份信息。

5.2.5 抗身份仿冒攻击

在本文方案中,攻击者 $\mathcal{A}$ 无法假冒合法的 SDP 客户端来访问服务资源。假设攻击者 $\mathcal{A}$ 试图伪造可验证表达进行身份仿冒,根据 LRSW 假设,攻击者 $\mathcal{A}$ 无法伪造出正确的可验证凭证,同时攻击者 $\mathcal{A}$ 无法生成合法的可验证表达,因此无法冒用合法 SDP 客户端的身份进行 SPA 认证。同时,假设攻击者 $\mathcal{A}$ 窃取到了一份可验证表达,并获取到了连接证书 $\mathcal{M}$,由于 DLP 困难性假设,攻击者 $\mathcal{A}$ 无法通过 pk_{CL} 计算出 sk_{CL},因而无法与 SDP 网关进行协商以生成 TLS 安全信道的通信密

钥，无法假冒合法的 SDP 客户端来与 SDP 网关建立 TLS 安全信道。因此本文方案能够有效抵抗身份仿冒攻击。

5.2.6 抗敲门放大攻击

敲门放大攻击是标准软件定义边界架构的常见攻击方式。标准软件定义边界架构下，在合法 SDP 客户端完成 SPA 认证后，同一 NAT 下的其他主机都可以访问 SDP 网关。攻击者$\mathcal{A}$在合法 SDP 客户端完成 SPA 认证后实施敲门放大攻击，冒用其网络信息抢先与 SDP 网关建立连接。本文方案能够有效抵抗敲门放大攻击。假设攻击者$\mathcal{A}$尝试伪造出一份合法的连接证书，由于攻击者$\mathcal{A}$无法获取到 SDP 控制器对连接证书的签名，因此无法向 SDP 网关证明连接证书的有效性。假设攻击者$\mathcal{A}$在 SDP 客户端完成 SPA 认证后，窃取到 SDP 客户端的连接证书$\mathcal{M}=pk_{\mathrm{CL}}\parallel c\parallel Lifetime\parallel ID_{\mathrm{Ctr}}$并尝试冒用其网络信息访问 SDP 网关，由于连接证书中的公钥基于 DLP 困难性假设生成，因此攻击者$\mathcal{A}$无法获取到 SDP 客户端的私钥，从而与 SDP 网关进行密钥协商以与其建立 TLS 安全信道。

6 性能对比

本章中将通信及计算开销作为方案性能评估的主要指标，对本文方案进行了实验分析。实验环境为 CentOS 7，硬件配置为 Intel(R) Core(TM) i5-10500 CPU @ 3.10 GHz，内存为 16 GB，软件平台采用 VS Code 1.8，开发语言为 C++，网络仿真平台为 Network Simulator 3。

6.1 通信开销

本文方案的主要通信开销涉及两个方面。首先，SDP 控制器在 SDP 客户端注册阶段，向 SDP 客户端发送可验证凭证。其次，SDP 客户端与 SDP 网关进行 SPA 认证过程中，所传输的 SPA 数据包与 SDP 客户端生成的可验证表达。算法的通信假设如表 1 所列。

表 1 通信假设

Table 1 Communication assumptions

符号	描述	长度/B
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_{SM9}	国密算法签名长度	358
L_{VC}	可验证凭证长度	420
L_{ID}	身份标识长度	8

在访问者仅拥有一个属性信息的情况下，将本文方案与 WaverleySDP[22] 和 BlockchainSDP[23] 的通信开销进行对比分析。SDP 客户端完成一次身份认证所需的通信开销如表 2 所列。

表 2 身份认证通信开销

Table 2 Communication overhead of identity authentication

方案	通信开销	通信量/B
Waverley SDP	$2L_{\mathrm{Cert}}+L_{\mathrm{SPA}}+2L_{\mathrm{KEY}}+L_{\mathrm{ID}}$	1320
SDP 电力物联网	$3L_{\mathrm{SM9}}+L_{\mathrm{Cert}}+L_{\mathrm{ID}}+2L_{\mathrm{H}}$	1658
本文方案	$L_{\mathrm{Cert}}+L_{\mathrm{SPA}}+L_{\mathrm{VC}}+L_{\mathrm{H}}+L_{\mathrm{ID}}$	1228

本文方案减少了 SDP 客户端建立 TLS 连接的次数，通信开销总体上与 WaverleySDP 持平。与 SDP 电力物联网方案相比，减少了认证所需通信轮次。接下来，使用 Network Simulator 3 仿真平台对本文方案和 WaverleySDP 在执行一次完整认证流程时，各参与方所需通信开销进行了测试。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。实验结果如图 4 所示。

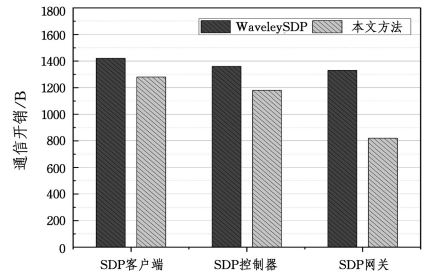


图 4 通信开销对比

Fig. 4 Comparison of communication overhead

由于本文方案减少了 SDP 控制器与 SDP 网关间建立 TLS 安全信道的次数，较大幅度地优化了 SDP 网关的通信开销；同时，SDP 网关不再需要定时向 SDP 控制器请求认证信息与访问控制列表，SDP 客户端也不再需要在每次访问时向 SDP 控制器请求访问信息，因此整体通信效率得到了一定程度的优化。相较于 WaverleySDP，本文方案的整体通信开销降低了 19%。

6.2 计算开销

本文方案的计算开销主要来自可验证凭证生成算法、可验证凭证聚合算法和可验证表达验证过程中的双线性映射计算。为评估方案性能，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。密码学操作的预设参数和具体计算开销如表 3 所列。

表 3 计算假设

Table 3 Computational assumptions

符号	描述	参数规模	耗时/ms
T_{bp}	双线性映射	2048 位	6.479
T_{enc}	非对称加密算法	128 位	4.337
T_{dec}	非对称解密算法	128 位	5.102
T_{ed}	对称加解密算法	128 位	0.207
T_h	哈希算法	256 位	0.037

根据表 3 中给出的各类计算开销假设，假设属性数量为 n，可以分析出完成一次认证时各参与方的整体计算开销，具体如表 4 所列。

表 4 身份认证计算开销

Table 4 Computational overhead of identity authentication

参与方	计算开销	计算耗时/ms
SDP 客户端	$(2n+8)T_{\mathrm{bp}}+2T_{\mathrm{enc}}+2T_{\mathrm{dec}}+3T_{ed}+3T_{\mathrm{h}}$	$71.422+n12.958$
SDP 控制器	$(2n+6)T_{\mathrm{bp}}+2T_{\mathrm{enc}}+2T_{\mathrm{dec}}+6T_{\mathrm{ed}}+3T_{\mathrm{h}}$	$58.565+n12.958$
SDP 网关	$(2n+6)T_{\mathrm{bp}}+2T_{\mathrm{enc}}+2T_{\mathrm{dec}}+6T_{\mathrm{ed}}+3T_{\mathrm{h}}$	$58.565+n12.958$

在身份认证签名方案中，VC 相关的步骤的计算开销与属性的数值的增长呈现强相关变化。对 VC 生成、VC 聚合与 VP 验证算法在属性值个数为 1，3，5，7，9 的情况下执行 50 次，并求得其计算开销的平均值。算法的计算开销随属性个数增多而线性增长的整体变化趋势如图 5 所示。

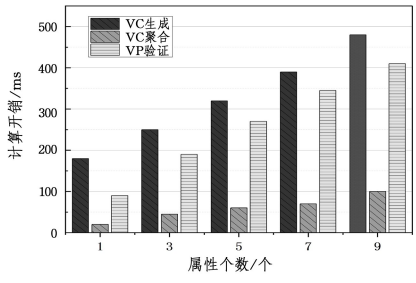


图 5　各类算法计算开销

Fig. 5　Computational overhead of algorithms

在可用性方面，对 Waverley SDP、SDP 电力物联网[24]和本文方案在多节点网络下的平均认证时延进行了仿真分析，将 SDP 网关的访问控制策略上限设置为 5，并在 SDP 客户端每次访问时随机给出访问控制策略。共进行了 SDP 客户端数量为 25，50，75，100，125，150，175，200，225，250，275 的 12 组实验，SDP 客户端每次访问时随机选择要访问的 SDP 网关，访问频率设置为 0－5 次/秒。平均认证时延对比如图 6 所示。

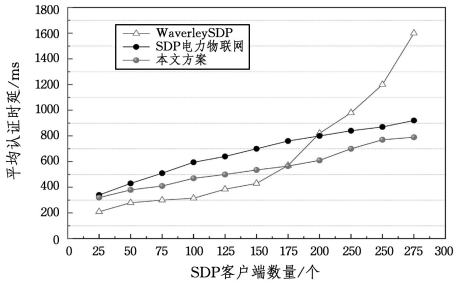


图 6　平均认证时延对比

Fig. 6　Comparison of average authentication delays

在认证成功率方面，将 WaverleySDP、BlockchainSDP 和本文方案在不同丢包率场景下进行对比。使用 Network Simulator 3 的 ErrorModel 模块设置不同丢包率，范围为 0%～30%，每次递增 5%；RateErrorModel 模拟丢包行为，测试数据包传输时的丢失与重传。丢包率与认证成功的关系如图 7 所示。

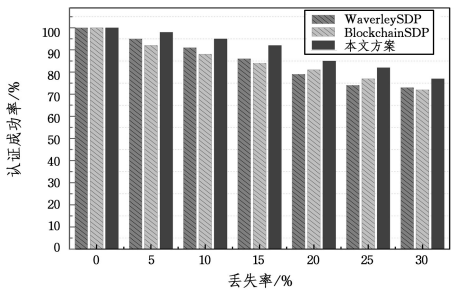
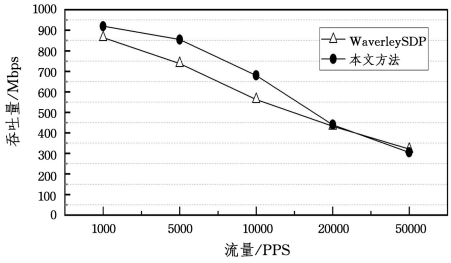


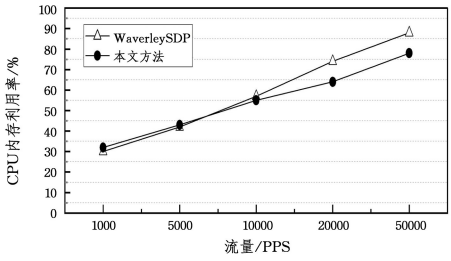
图 7　丢包率与认证成功率

Fig. 7　Packet loss rate and authentication success rate

在安全性方面，通过仿真模拟验证了在敲门放大的攻击下，本文方案和 Waverley SDP 的安全认证效果。Iperf 工具可实时显示这两种情况下的网络吞吐量。hping3 可模拟虚拟 TCP 数据包发送到 SDP 网关，模拟目标端口的多次“敲门”，伪装为正常的 SPA 认证请求。在不同强度下逐步增加伪造流量（PPS），分别为 1 000，5 000，10 000，20 000，50 000。网关负载和网络吞吐量对比如图 8 所示。



(a) Waverley SDP 方法与本文方法的网络吞吐量对比图



(a) Waverley SDP 方法与本文方法的网络对比图

图 8　网关负载与网络吞吐量对比

Fig. 8　Comparison of gateway load and network throughput

在高丢包率和流量受限的情况下，本文方案展现出更高的认证成功率和更大的吞吐量，以及更低的网关负载。虽然在高流量情况下两者趋同，但在常规的流量场景，本文方案能够提供更好的性能。

在单 SDP 控制器的情况下，SDP 控制器成为了网络瓶颈，难以承受多节点高并发的访问请求。而在本文方案中，SDP 控制器将可验证凭证签发给 SDP 客户端，由 SDP 客户端自主生成可验证表达以进行访问，有效减少了 SDP 控制器的计算负载，SDP 控制器仅在 SDP 客户端注册时完成一次可验证凭证签发工作，之后只需要完成 SPA 密钥分发工作。同时，SDP 客户端避免了定期向 SDP 网关和 SDP 客户端发送访问控制列表带来的计算开销与通信时延。因此本文方案在多节点网络中表现出了较好的性能。

结束语　本文提出了一种基于属性的软件定义边界身份隐私保护访问方案，通过可验证凭证实现了访问过程中访问者身份信息的最小披露。方案通过可验证凭证将软件定义边界架构基于身份的访问控制转变为基于属性的访问控制，SDP 客户端访问 SDP 网关时可以按需将可验证凭证聚合为可验证表达，有效保护了访问者身份隐私信息，同时也能够避免访问记录被恶意关联。理论分析表明，本文方案能够抵御可验证凭证篡改和敲门放大攻击等网络攻击。实验分析表明，本文方案在多节点网络下具有更好的性能。

然而，基于双线性映射与 CL 签名的算法在实现身份保护和安全认证时，计算复杂度较高，可能导致物联网设备或低配置终端性能下降，增加存储负担。此外，本文方案尚未充分

考虑不同操作系统、硬件平台和网络环境的兼容性，可能面临与现有系统集成困难和跨域认证的问题。在未来，将尝试使用更加轻量级的算法以降低计算开销，并将方案扩展到具有多个 SDP 控制器的跨域认证场景下。

参 考 文 献

[1] GARBIS J, KOILPILLAI J. Software-Defined Perimeter(SDP) Specification v2.0[M]. Working Group: SDP and Zero Trust, 2022.

[2] YAN J, LI S X, LI G Z, et al. Security Protection Method of Oower Internet of Things Based on Software Defined Perimeter [J]. Techniques of Automation and Applications, 2025, 44(3): 93-95, 114.

[3] 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.

[4] CHIY P, LIU J H, LIANG J M. Design of SDP Trust Evaluation Model Based on Federated Learning [J]. Journal of Information Security Research, 2024, 10(10): 903-911.

[5] RASH M. Single packet authorization[J]. Linux Journal, 2007, 2007(156): 1.

[6] ZHANG L, GE J, WU Y, et al. On Improved Efficiency and Forward Security of 0-RTT Key Exchange for SDP[C]// 2024 33rd International Conference on Computer Communications and Networks(ICCCN). IEEE, 2024: 1-9.

[7] LEE H, KIM D, KWON Y. TLS 1.3 in practice: How TLS 1.3 contributes to the internet[C]// Proceedings of the Web Conference 2021. 2021: 70-79.

[8] SINGH J, BELLO Y, REFAEY A, et al. Five-Layers SDP-Based Hierarchical Security Paradigm for Multi-access Edge Computing[J]. arXiv: 2007.01246, 2020.

[9] CAMENISCH J, LYSYANSKAYA A. Signature schemes and anonymous credentials from bilinear maps[C]// Annual international cryptology conference. Berlin: Springer, 2004: 56-72.

[10] FENGJ Y, YU T T, WANG Z Y, et al. An Edge Zero-Trust Model Against Compromised Terminals Threats in Power IoT Environments [J]. Computer Research and Development, 2022, 59(5): 1120-1132.

[11] YANG Y, BAI F, YU Z, et al. An anonymous and supervisory cross-chain privacy protection protocol for zero-trust IoT application[J]. ACM Transactions on Sensor Networks, 2024, 20(2): 1-20.

[12] TANG F, MA C, CHENG K. Privacy-preserving authentication scheme based on zero trust architecture[J]. Digital Communications and Networks, 2024, 10(5): 1211-1220.

[13] SONGZ M, YU Y M, WANG G W, et al. Zero-knowledge authentication and management architecture for digital identity verifiable credentials based on blockchain smart contracts [J]. Journal of Information Security, 2023, 8(1): 55-77.

[14] LI S, IQBAL M, SAXENA N. Future industry internet of things with zero-trust security[J]. Information Systems Frontiers, 2024, 26: 1653-1666.

[15] RASHEED A, MAHAPATRA R N, VAROL C, et al. Exploiting zero knowledge proof and blockchains towards the enforcement of anonymity, data integrity and privacy(ADIP) in the iot [J]. IEEE Transactions on Emerging Topics in Computing, 2021, 10(3): 1476-1491.

[16] SONG Y, DING L, LIU X, et al. Differential Privacy Protection Algorithm Based on ZeroTrust Architecture for Industrial Internet[C]// 2022 IEEE 4th International Conference on Power, Intelligent Computing and Systems(ICPICS). IEEE, 2022: 917-920.

[17] VerifiableCredentials Data Model v2.0 [EB/OL]. (2024-12-19) [2024-12-30]. https://www.w3.org/TR/vc-data-model-2.0/.

[18] SEDLMEIR J, SMETHURST R, RIEGER A, et al. Digital identities and verifiable credentials[J]. Business & Information Systems Engineering, 2021, 63(5): 603-613.

[19] ALAM S. A blockchain-based framework for secure educational credentials[J]. Turkish Journal of Computer and Mathematics Education, 2021, 12(10): 5157-5167.

[20] MUKTA R, MARTENS J, PAIK H, et al. Blockchain-based verifiable credential sharing with selective disclosure[C]// 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications(TrustCom). IEEE, 2020: 959-966.

[21] SHIM K A. A strong designated verifier signature scheme tightly related to the LRSW assumption[J]. International Journal of Computer Mathematics, 2013, 90(2): 163-171.

[22] KOILPILLAI J. Software defined perimeter(SDP) a primer for cios[J]. Waverley Labs LLC, 2017, 267: 56-62.

[23] YAN J, YANG B, SU L, et al. Blockchain based Software Defined Perimeter(SDP) in Support of Authentication and Authorization[C]// 2022 International Conference on Blockchain Technology and Information Security(ICBCTIS). 2022: 40-42.

[24] WU K H, CHENG R, JIANG X C, et al. Security Protection Scheme of Power IoT Based on SDP[J]. Netinfo Security, 2022, 22(2): 32-38.

SI Xuege, born in 1999, postgraduate. Her main research interests include cryptography and zero trust security.

JIA Hongyong, born in 1975, Ph.D, lecturer. His main research interests include cloud computing security and zero trust security of the IoT system.

（责任编辑：柯颖）