基于ZUC加密的IPv6地址动态编码算法及应用方案

刘永清[1, 2]，曹玖新[1]
（1. 东南大学网络空间安全学院，江苏 南京 211102；
2. 国家计算机网络应急技术处理协调中心江苏分中心，江苏 南京 210019）

摘 要：IPv6 地址空间巨大，IPv6 单播地址可分为网络前缀和接口标识两部分，网络前缀由运营商（ISP, Internet service provider）分配，接口标识可以手工配置、随机生成或者通过 EUI-64 格式生成。手工配置或通过 EUI-64 格式生成的静态 IPv6 地址存在个人隐私泄露的网络安全风险；随机生成的 IPv6 地址不满足基于 IP 地址的网络访问控制需求。因此，提出了一种基于祖冲之（ZUC, ZU Chongzhi）加密的 IPv6 地址动态编码（ZBDA, ZUC-based dynamic addressing）算法，将网络终端的 MAC 地址通过 ZUC 算法加密生成动态的 IPv6 地址，在接收端解密即可获得终端的 MAC 地址，以此验证终端的访问权限。ZBDA 算法既解决了不当的 IPv6 地址编址带来的个人隐私泄露问题，又满足了基于 IP 地址的网络访问控制需求，且该算法的 IPv6 地址编码和地址验证速度快，具有实际应用价值。

关键词：IPv6；地址编码算法；接口标识；隐私泄露；EUI-64；祖冲之
中图分类号：TP393
文献标志码：A
doi：10.11959/j.issn.2096−3750.2024.00405

ZUC-based IPv6 dynamic addressing algorithm and application scheme

LIU Yongqing[1, 2], CAO Jiuxin[1]
1. School of Cyber Science and Engineering, Southeast University, Nanjing 211102, China
2. National Computer Network Emergency Technology Coordination Center Jiangsu Branch, Nanjing 210019, China

Abstract: IPv6 address space is very large. IPv6 unicast address is divided into two parts: network prefix and interface identifier. The network prefix is assigned by Internet service provider (ISP), and the interface identifier can be determined by manual configuration, random generation and the EUI-64 format. The IPv6 addresses determined by manual configured and statically generation in EUI-64 format bring a risk of personal privacy leakage. However, randomly generated IPv6 addresses sometimes do not meet the network access control requirements based on IP addresses. Therefore, a ZUC-based dynamic addressing (ZBDA) algorithm was proposed. The MAC address of a network host was encrypted using the ZUC stream cipher algorithm to generate a dynamic IPv6 address, which could be decrypted at the receiving server to obtain the MAC address of host, and it can be verified the host's access permissions from decrypted MAC address. The ZBDA algorithm not only solves the problem of personal privacy leakage caused by improper IPv6 addressing, but also meets the network access requirements based on IP address control. Moreover, IPv6 address generation speed and verification speed are fast. Therefore, the algorithm has the value of practical application.

Key words: IPv6, addressing algorithm, interface identifier, privacy leakage, EUI-64, ZUC

收稿日期：2024−01−02；**修回日期：**2024−06−18
通信作者：曹玖新，jx.cao@seu.edu.cn

0 引言

随着IPv4地址逐步耗尽，国家正在加快推进IPv6网络的规模部署与应用，要求各组织和单位加强IPv6关键核心技术研发[1]，落实IPv6网络地址编码规划方案，强化网络安全保障，维护国家网络安全[2]。国家IPv6发展监测平台数据显示，当前固定互联网IPv6流量占比21%，移动互联网IPv6流量占比63%[3]。Google统计的全球IPv6流量占比43%[4]。IPv6地址长度为128 bit[5]，拥有巨大的地址空间，每一个IPv6终端都可以分配到一段可路由的全球单播地址[6]，因而运营商不需再像IPv4一样，通过地址转换（NAT, network address translation）技术隐藏用户的实际IP地址，所有终端都以真实的IPv6地址在网络中通信。

IPv6地址分为网络前缀（network prefix）和接口标识（IID, interface identifier）[7]两部分，其中网络前缀由电信运营商分配，接口标识可以手工配置、随机产生或者采用IEEE定义的EUI-64格式[8]，根据终端MAC地址生成接口标识。不同的地址编码方式，带来了IP地址各异的时空特性[9]。EUI-64生成的接口标识唯一，即使运营商轮换网络前缀，这种地址编码的终端也很容易被追踪到。另一方面，根据EUI-64接口标识生成方法，可以容易地从EUI-64接口标识计算出终端的MAC地址。MAC地址的前24位代表组织唯一标识（OUI, organizationally unique identifier）[10]，后24位是制造商分配的唯一扩展标识，因此，从EUI-64接口标识即可推测出IPv6终端的品牌和类型，这就导致了个人隐私泄露的风险[11-14]，对于政府等重要部门人员的隐私泄露，可能会带来严重的威胁。

手工配置静态IPv6地址通常有一定的地址编码模式，因而很容易被别人探测到[15]而且配置静态IPv6的终端在访问互联网时，IPv6地址始终保持不变。以同一个IPv6地址分别开展网银交易、网上购物、邮件收发、网络搜索、网页浏览、即时通信、社交发帖等日常互联网活动时，互联网企业关联多方数据，可轻松追踪到个人的上网活动轨迹[16]。长期使用固定的IPv6地址，类似于在真实生活中，将个人的真实姓名、身份证号码、手机号码、家庭住址等个人所有信息写在衣服上，所到之处，人人皆知你是谁。固定IPv6地址，事实上等同于个人身份标识，严重威胁着个人的网络空间安全。因此，对于IPv6地址编码算法的研究具有一定的现实意义，符合国家的IPv6发展战略。

1 相关工作

IPv6网络地址编码研究一直受到学者的关注[17]，针对静态IPv6地址存在隐私泄露等问题，RFC 4941[18]建议使用隐私扩展模式，将EUI-64格式的IID串接历史值（初始为64位随机数）进行MD5散列运算，将摘要的高64位作为新IID，低64位存为历史值用于下一次IID生成，这样生成的IID可以随着时间而变化，降低被监听和被信息收集的风险。RFC 7217[19]建议对网络前缀Prefix、网络接口Net_Iface、网络标识Network_ID、地址冲突次数DAD_Counter以及密钥secret_key进行Hash运算，取摘要的低64位作为IID，该方法生成的IID虽然是随机的，但是在同一个网络内IID是保持固定不变的。Odero等[20]对RFC 7217的方法进行改进，将可选的Network_ID参数纳入Hash运算，并通过改变Network_ID参数，实现生成动态的IID。Micovic等[21]在被保护网络的边界部署了LISPP（lightweight stateless privacy protection）设备，利用保形加密算法（FPE, format-preserving encryption），对通过的网络报文的源IP地址中的主机部分（不包含网络前缀）和源端口进行组合加密，处理过程类似端口地址转换（PAT, port address translation），实现了动态地修改数据包的源IP地址和源端口，以此保护通信终端的隐私。

动态IP地址可以提高网络访问安全，但是互联网上很多信息系统都是以检查访问者的源IP地址来初步验证访问者的身份，需要访问者采用固定的IP地址，这就形成了矛盾。为此，本文提出了一种基于祖冲之（ZUC）序列密码加密的IPv6地址动态编码（ZBDA, ZUC-based dynamic addressing）算法，可以同时解决这两方面的问题，该算法以访问终端的MAC地址作为终端标识，并经ZUC算法加密后的数据作为IPv6地址的接口标识，与网络前缀结合生成动态的IPv6地址，降低了隐私泄露风险。在接收端，用ZUC算法解密源IPv6地址的接口标识，获取访问者的源MAC地址，若该MAC是授权的访问终端地址，则放行该通信数据包，否则拒绝该数据包的访问，实现了网络安全访问控制。

ZUC密码算法是我国发布的商用序列密码算法，具有较高的安全冗余，并且算法加解密速度快，适用于轻量级的瘦终端，可用于数据的机密性和完整性保护[22]，已于2011年成为了4G移动通信密码算法的国际标准。ZUC算法的安全性主要得益于线性反馈移位寄存器（LFSR, linear feedback shift register）、比特重组（BR, bit reorganization）和非线性函数F的设计。

ZUC算法密钥由128位的种子密钥和128位的初始向量组成，是算法安全的核心，需要严格保密。种子密钥和初始向量共同作用进行初始化，然后每次产生32位宽的数据加密密钥流，明文数据流与密钥流逐位异或运算即可实现数据的加密[23]，ZUC算法加密过程如图1所示。

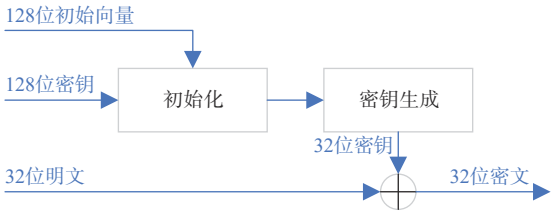

图1　ZUC算法加密过程

ZUC算法属于对称密码算法，解密密钥与加密钥相同，而且解密过程与加密过程也完全相同。加密端将种子密钥和初始向量通过保密途径发送至解密端，这样加解密双方就预共享了相同的种子密钥和初始向量，如此解密端就能生成和加密端完全一致的密钥流序列，密文流与密钥流逐位异或运算，即可解得明文流。

2　ZBDA算法

固定IP地址不仅存在泄露隐私的风险，而且不满足特定网络访问的需求，如端口扫描、网站爬取等。动态IP地址可以提高网络访问者的个人安全，但是却不满足基于IP地址的网络访问控制。本文提出的ZBDA算法，可以兼顾解决以上两个问题。

2.1　IPv6地址编码方法

IPv6地址由网络前缀和接口标识组成，网络前缀通常由运营商分配，本算法主要用于生成IPv6的接口标识，该算法的主要思想是用ZUC算法对MAC地址进行加密。加密策略为“一次一密，用完作废”，即在一定的时间范围内，同一局域网内不同终端接口标识的加密密钥不同，同一个终端在不同时期申请IPv6地址时的加密密钥也不相同。ZUC是序列密码算法，因此，加解密的密钥次序必须一致。

ZUC算法每次生成一个32位密钥，而接口标识是64位，因此，每次接口标识加密需要两个密钥，本文称为一对密钥，以密钥标识（KeyID）进行标记。ZUC算法初始化后，生成的第一对密钥的KeyID为0，第二对密钥的KeyID为1，依次类推。一对密钥中，先产生的密钥标记为Key0，后产生的密钥标记为Key1，接口标识IID生成方法可简述为：48位的MAC地址和密钥KeyID的低16位组成64位序列，该序列与ZUC算法生成的一对密钥异或运算，即生成IID，具体计算过程如式(1)所示。

$$\mathrm{IID}=\left(\mathrm{MAC}_{\mathrm{H32}}\oplus\mathrm{Key0}\right)\|\left(\left(\mathrm{MAC}_{\mathrm{L16}}\|\mathrm{KeyID}_{\mathrm{L16}}\right)\oplus\mathrm{Key1}\right) \tag{1}$$

其中，运算符意义见表1。

表1　运算符意义

运算符	运算规则
H32	取64位中的高32位
L32	取64位中的低32位
H16	取32位中的高16位
L16	取32位中的低16位
⊕	逐位异或运算
&	逐位与运算
‖	比特串连接符
>>k	右移k位

式(1)中，MAC地址高32位与Key0逐位异或运算，生成高32位数值；MAC地址低16位与密钥标识低16位进行串接后再与Key1逐位异或运算，生成低32位数值；两个运算结果串接生成一个64位数值，即接口标识。

生成的接口标识与网络前缀组合成IPv6地址，该IPv6地址可能与已经存在的地址冲突，虽然这个概率极低，但仍需经过重复地址检测（DAD, duplicate address detection）[24]才能确认IPv6地址可用，如果地址冲突，就需要再生成一对密钥，KeyID同时加1，然后按式(1)再次计算并生成新的IPv6地址，再次进行DAD检测，直到生成的IPv6地址没有冲突。利用ZBDA算法，可以定时更换终端IPv6地址，即使同一终端，在不同时刻对应的加密钥不同，KeyID也不同，因而生成的IID完全不同。

下面以种子密钥 k 和初始向量 **iv** 均为 128 位全 0 为例，介绍一下 IPv6 地址的计算过程。ZUC 算法计算密钥流的具体实现可参考文献[25]。ZUC 算法初始化后，生成的第一对密钥（KeyID = 0）[27BEDE74, 018082DA]，生成的第二对密钥（KeyID = 1）[87D4E5B6, 9F18BF66]。如果终端 1 的 MAC 地址为 0011-2233-4455，那么依据该算法生成的接口标识为(00112233⊕27BEDE74)‖((4455‖0000)⊕018082DA)=27AFFC4745D582DA。假设网络前缀为 2409:8A20:4A3:38D0::，则生成的终端 1 的 IPv6 地址为 2409:8A20:4A3:38D0:27AF:FC47:45D5:82DA。

如果另一台终端 2 紧接着申请 IPv6 地址，其 MAC 地址为 AABB-CCDD-EEFF，那么该终端生成的接口标识为(AABBCCDD⊕87D4E5B6)‖((EEFF‖0001)⊕9F18BF66)=2D6F296B71E7BF67，即生成的终端 2 的 IPv6 地址为 2409:8A20:4A3:38D0:2D6F:296B:71E7:BF67。

2.2　IPv6 地址验证方法

以 ZBDA 算法生成的 IPv6 地址为源地址的数据包发出后，在接收端为了验证访问终端的权限，需要从 IPv6 源地址的接口标识中解密出 MAC 地址。ZUC 是序列密码算法，接收到的密文序列必须与发送端一致才能正常解密，但是由于以下原因，会出现接收到的密文序列与发送端不一致的情况。

1) 生成 IPv6 地址后，终端未与接收端通信，或者通信数据出现丢包，IPv6 地址生存期超期后，该终端又重新获取新的 IPv6 地址。

2) 生成的 IPv6 地址冲突了，需要用新的密钥对重新生成新的 IPv6 地址，即存在少量的密钥对加密后的数据被丢弃。

3) 由于多路径传输导致数据包出现先发而后至。

以上情况不能完全按照发送端的密钥序列进行解密，解密密钥要在一定范围密钥库内搜索，地址验证方法可归纳为：接收端根据与发送端预共享的种子密钥和初始向量生成一定数量的解密密钥库，这些解密密钥根据生成的先后顺序依次排列，即按 KeyID 由小到大排列；接收到的 IPv6 地址与密钥库中的密钥从前往后依次逐一解密测试，解密成功则获取到源 MAC 地址，并需要对密钥库进行更新；若密钥库中所有密钥都解密失败，则说明接收的 IPv6 地址非 ZBDA 算法生成，直接丢弃。

IPv6 地址验证流程如图 2 所示，详细步骤可分为以下 4 步：

1) 初始化：接收端首先生成一个密钥库，如果发送端的所有终端接口的数量为 n，网络传输最大丢包率为 p，那么设置接收端的密钥库大小为 $K=\lceil n/(1-p)\rceil$。若丢包率等于 50%，则密钥库大小取

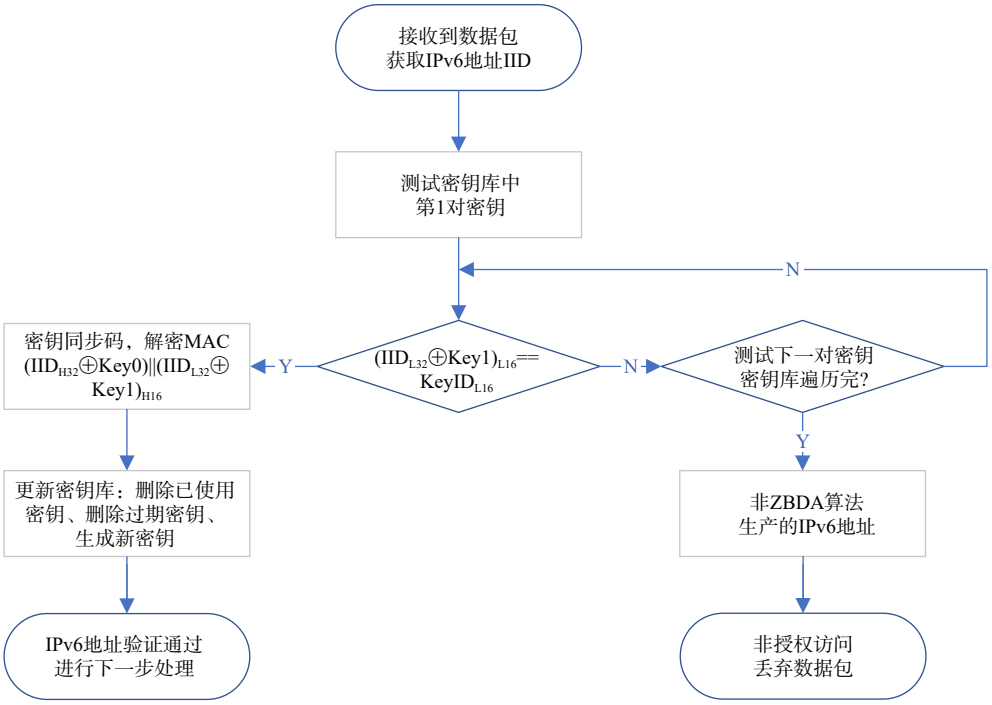

图 2　IPv6 地址验证流程

2n对密钥。

2) 解密测试1：接收到一个数据包后，将源IPv6地址中的接口标识IID的低32位与密钥库中第一对密钥中Key1逐位异或运算并取低16位，如果结果与该对密钥标识KeyID的低16位相等，说明解密密钥正确，即可按式(2)进一步解密MAC地址，并进行步骤4)密钥库更新，地址验证通过。

$$\mathrm{MAC}=\left(\mathrm{IID}_{\mathrm{H32}}\oplus\mathrm{Key0}\right)\|\left(\mathrm{IID}_{\mathrm{L32}}\oplus\mathrm{Key1}\right)_{\mathrm{H16}} \quad (2)$$

3) 解密测试2：若异或运算结果与该密钥标识KeyID的低16位不等，则说明密钥不正确，进行下一对密钥的匹配测试，重复步骤2)至步骤3)，直到找到正确的密钥为止。如果遍历整个密钥库，都未找到正确的密钥，说明该IPv6地址并非ZBDA算法生成，即非授权终端访问，将其直接丢弃。

4) 密钥库更新：由于“一次一密，用完作废”，已成功解密的密钥需从密钥库中删除，同时还需检查密钥库，并删除过期密钥，即与最大密钥标识KeyID的差值超过K+n的密钥，这些过期的密钥通常是发送端加密后的IPv6地址未到达接收端。密钥库更新时，删除几对密钥，就需要在密钥库末尾生成相同数量的新密钥对，保持密钥库的大小不变。

继续第2.1节示例，假设终端2的数据包先于终端1到达，接收端的验证过程如下：

1) 初始化：设终端数n = 2，丢包率p = 0.5，则解密密钥库的大小K = 4。接收端的密钥k和初始向量**iv**必须与发送端相同，仍以128位全0为例。此时，接收端生成的密钥库为{[27BEDE74, 018082DA], [87D4E5B6, 9F18BF66], [32070E0F, 39B7B692], [B4673EDC, 3184A48E]}，对应的密钥标识KeyID为{0, 1, 2, 3}。

2) 接收到终端2的IPv6数据包后，获得终端2的接口标识IID为2D6F:296B:71E7:BF67，先用密钥库中的第1对密钥测试，即用KeyID = 0的密钥对计算得(71E7BF67⊕018082DA) & FFFF=3DBD≠0，解密获取的KeyID与当前KeyID不一致，解密密钥不正确。

3) 测试第2对密钥，即用KeyID = 1的密钥对计算得(71E7BF67⊕9F18BF66) & FFFF=0001，解密获取的KeyID与当前KeyID一致，解密密钥成功找到，因此可以解密MAC地址，计算为(2D6F296B⊕87D4E5B6)‖(((71E7BF67⊕9F18BF66)>>16) & FFFF)=AABBCCDDEEFF，即解得终端2的MAC地址为AABB-CCDD-EEFF。

4) 密钥库更新：从密钥库中删除当前已成功解密的密钥对，并生成新的密钥对，此时的密钥库为{[27BEDE74, 018082DA], [32070E0F, 39B7B692], [B4673EDC, 3184A48E], [27636F44, 14510D62]}，对应的密钥标识KeyID为{0, 2, 3, 4}。此时密钥库中最小KeyID为0，最大KeyID为4，差值为4，小于K+n，即密钥库中没有过期密钥。

5) 依此方法，重复步骤2)至步骤4)，即可以用KeyID = 0的密钥对解密终端1的MAC地址为0011-2233-4455。

2.3　算法安全性分析

ZBDA算法的安全性依赖于ZUC算法，128位的种子密钥和初始向量是算法的安全所在，需要严格保密，决不能泄露。ZUC算法穷尽搜索复杂度为$O(2^{128})$[25]，在实际应用中不能使用本文举例的简单密钥，建议通过随机数直接或间接生成。种子密钥和初始向量只在算法初始化时使用到，初始化后不再使用，因而可以把种子密钥和初始向量加密保存在U-key中，算法初始化后拔出U-key保存在密码柜中。ZUC是对称密码算法，种子密钥和初始向量需要从加密端传递到解密端，密钥的分发必须通过保密途径传递，可以人工分发或者加密在线分发。为了避免遭到暴力破解，种子密钥和初始向量需要定期销毁后更换新的密钥。由ZUC算法生成的加密密钥流临时储存在内存中，加密完一次后就不再使用，从内存中删除，因而可以对抗重放攻击。即使第三方截取到一段加密密钥，也无法据此推算出下一次的加密密钥，因此加密密钥无须特殊保护。解密时密钥库的密钥亦是如此，“一次一密，用完作废”。

2.4　实验分析

为了验证ZBDA算法的性能，实验在一台12核Intel i5-1240P、1.70 GHz的CPU、16 GB内存、Windows 11操作系统、并安装了Python 3.10.11的笔记本计算机上进行，编写Python程序对比分析了EUI-64[8]、RFC4941[18]、RFC7217[19]和ZBDA算法的性能，其中EUI-64、RFC4941和ZBDA算法都是基于固定的00-11-22-33-44-55的MAC地址进行IPv6地址编码，而RFC7217算法固定取值Prefix = FE80::，Network_ID = “inet”, DAD_Counter = 0、secret_key = 0

（128 位），每次变换 Net_Iface，由 0 逐次加 1。实验主要开展了以下 4 方面的分析。

1) 算法功能性验证：实验程序实现了 ZBDA 算法的 IPv6 地址编码和地址解码功能，验证了 ZBDA 算法功能的可行性。

2) 地址冲突率分析：4 种算法各自连续生成 10 000 个 IPv6 地址，未出现一例 IPv6 地址冲突的情况，验证了这 4 种算法都具有较低的地址冲突率。

3) 地址随机性测试：为了检测生成地址的随机性，通过计算生成接口标识 IID 的熵值进行比较，熵值的计算是以 16 进制表示的半字节（0, 1, 2, …, F）在每个 IID 对应位置出现的频率 $P(x_i)$，按式(3)算得相应 i 位的熵值，然后按式(4)计算 IID 中所有 16 位半字节熵值的均值[26]。根据式(3)定义可知，熵值的范围为[0, 1]，0 表示 IID 各位完全相同，1 表示 IID 各位完全随机。由于 EUI-64 算法对同一 MAC 地址生成的 IID 是固定不变的，即熵值为 0，没有对比意义，所以实验将 EUI-64 替换为随机生成 IID。

$$H(X_i)=-\sum_{x_i=0}^{f}P(x_i)\log_{16}P(x_i) \tag{3}$$

$$H(\mathrm{IID})=\frac{1}{16}\sum_{i=1}^{16}H(X_i) \tag{4}$$

实验以 4 种算法分别生成不同数量的 IID，接口标识 IID 的熵值分布如图 3 所示，可见 ZBDA 和 RFC7217 生成的 IID 的熵值最大，当生成地址的数量超过 1 000 时，IID 接近于均匀分布的随机数，因而安全性最高。而 python 的 randint 函数生成的 IID 的随机分布相对不均匀。

4) 地址编码时效比较：4 种算法分别进行了 10、100、1 000、10 000 次循环的 IPv6 地址编码，本次测试未进行重复地址冲突检测。实验记录了这 4 种算法生成不同数量 IPv6 地址的运行时间，IPv6 地址编码时间如图 4 所示，从图 4 中可以看出，算法越安全复杂，生成 IPv6 地址所需时间就越长。

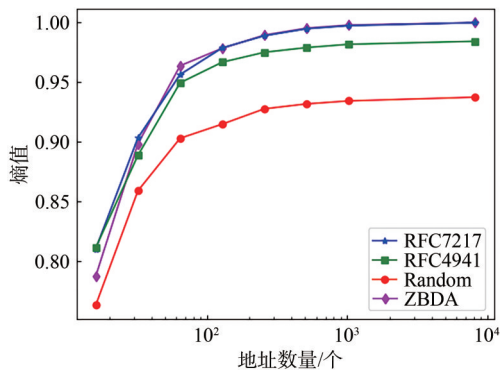

图 3　接口标识 IID 的熵值分布

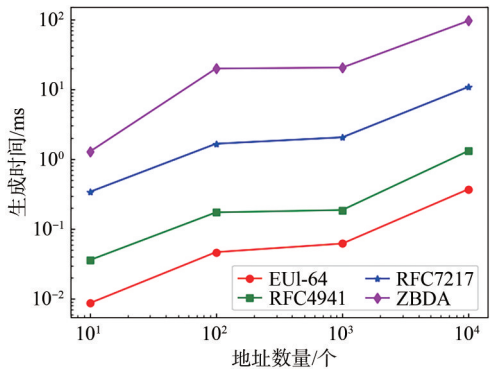

图 4　IPv6 地址编码时间

4 种算法中，虽然 ZBDA 算法运行时间相对较长，但是生成 10 000 个 IPv6 地址所需要的时间只需 95.74 ms，这个时间远低于重复地址检测所需要的时间，因而在实际应用中是完全可以接受的。另一方面，只有 ZBDA 算法能够从 IPv6 地址中解密出 MAC 地址，既保留了动态 IPv6 地址带来的隐私保护，又达到了基于 IP 地址的网络访问控制需求。

3　应用方案

ZBDA 算法对保护个人隐私，提高 IPv6 网络安全具有一定的实际应用价值，ZBDA 算法应用场景如图 5 所示。站点 A 有多个 IPv6 网络终端一直在互联网上从事网站内容爬取等工作，为了躲避网站检测，这些终端需要不停地变换其 IPv6 地址；另一方面，站点 A 的网络终端还要将爬取结果上传至站点 B 的应用服务器，或者从应用服务器下载特征数据，而站点 B 中的服务器都不向互联网公众开放，仅允许特定的授权主机访问。在这种应用场景下，站点 A 为客户端站点，所有终端采用 ZBDA 算法进行 IPv6 地址动态编码；而站点 B 为服务器站点，服务器采用常规的固定 IPv6 地址。

站点 A 和站点 B 采用预共享种子密钥和初始向量方式协同完成终端 MAC 地址的加密与解密，站点 A 将网内终端的 MAC 地址事先上报给站点 B。站点 B 接收到 IPv6 报文后，解密出源 MAC 地址，如果是事先上报的授权终端，那么就对其处理响应，否则丢弃接收到的数据包。

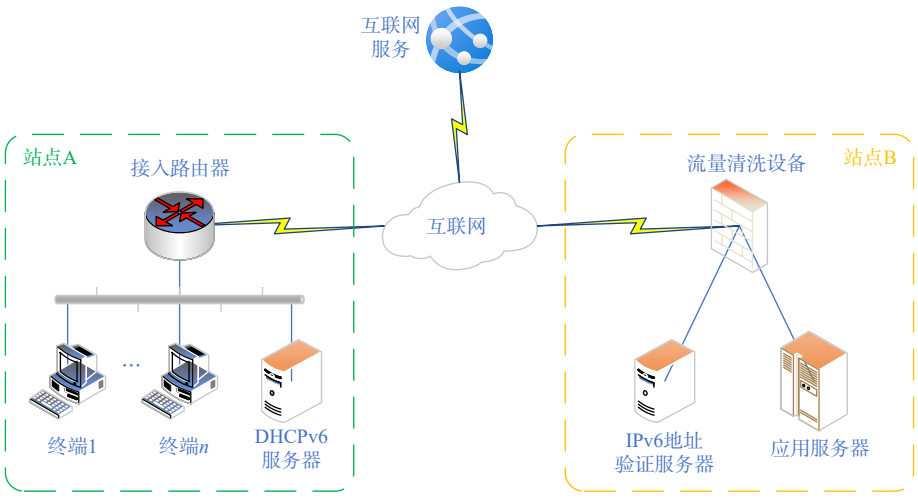

图5　ZBDA算法应用场景

3.1　动态地址编址方案

站点A的网络设备和终端均运行IPv6邻居发现协议（ND, neighbor discovery for IP version 6）[24]，网络管理员配置一定的规则，接入路由器周期性地向本地链路范围的所有节点多播地址FF02::1发送路由器通告报文（RA, router advertisement）[27]，RA报文中的M标志（managed address configuration）和O标志（other stateful configuration）置位，即通知本地链路范围内的所有终端采用有状态地址自动配置机制[28]。于是各终端向DHCPv6服务器申请全球单播地址，DHCPv6服务器从终端的DHCP唯一标识（DUID, DHCP unique identifier）中提取其MAC地址，利用本文提出的ZBDA算法生成接口标识IID，并与网络前缀组合生成IPv6单播地址，DHCPv6服务器经过重复地址检测，若发现地址冲突则重新生成IPv6。另外DHCPv6服务器还可设置适当的首选生存期（preferred lifetime）和有效生存期（valid lifetime）等参数，以满足相应的业务需求。DHCPv6服务器将生成的动态IPv6地址、DNS服务器及其他网络参数推送给各终端，这样站点A内所有终端都获得了包含加密MAC地址信息的动态IPv6地址，并定时向DHCPv6服务器申请更新IPv6地址。

3.2　流量检测识别方案

站点B在其网络边界部署了流量清洗设备，该设备与IPv6地址验证服务器建立BGP（border gateway protocol）邻居，IPv6地址验证服务器通过BGP FlowSpec路由[29-30]将已经通过验证的IPv6源地址通告给流量清洗设备，流量清洗设备将BGP FlowSpec路由转换为转发层面的流量控制策略，对于已经通过验证的IPv6源地址的流量通过路由直接转发给应用服务器处理，而对于没有通过验证的IPv6源地址的流量则重定向给IPv6地址验证服务器。

IPv6地址验证服务器根据第2.2节描述的IPv6地址验证方法，对接收到流量的源IPv6地址进行验证，如果发现是非授权终端的访问，那么就直接丢弃该非法流量。若是授权终端访问，则IPv6地址验证服务器一方面要将该合法访问流量回注给流量清洗设备，由流量清洗设备通过路由转发给应用服务器处理，另一方面，将验证合法的IPv6地址通过BGP FlowSpec路由通告给流量清洗设备，这样该终端在下一次访问时，就不会再重定向给地址验证服务器，而是由流量清洗设备直接路由转发给应用服务器处理，提高站点B的流量识别验证速度。

3.3　密钥同步方案

ZUC算法要求收发端的密文密序列完全一致，如第2.2节所述，ZDBA算法通过在接收端设置解密密钥库，一定程度上解决了数据丢失及密文序列错乱的问题。但是当网络出现中断时，站点A有大量的密文无法发送到站点B，网络恢复后，站点B接收到密文，其加密密钥KeyID已远远超出密钥库的范围，网络恢复后的后续所有密文都将解密失败。要解决此问题，就需要在站点A和站点B之间进行密钥同步，方法是站点A的加密设备（DHCPv6服务器）定期将当前的加密密钥KeyID发送给站点B的解密设备（IPv6地址验证服务器），解密设备检查接收到的KeyID是否在解密密钥库范围之内，如果超出，则清空当前密钥库，并根据接收的KeyID为起始向后重新生成K个解密密钥对，这样就可以保

证站点A和站点B的加解密密钥再次同步。在不掌握种子密钥和初始向量的情况下，即使第三方截取到传送的KeyID，也无法生成对应的解密密钥，因此，站点A向站点B定期传送KeyID时，无须加密保护，明文传送即可。

3.4 多方通信方案

图5是一种简化的应用场景，实际应用中，站点B为服务站点，不但要与站点A通信，还需要与其他多个站点通信，假设存在另一个与站点A对等的站点C，那么站点C的所有终端也采用ZBDA算法生成动态的IPv6地址，站点C和站点B预共享另一套种子密钥和初始向量，站点B在接收到流量后，根据网络前缀区分出流量是发自站点A还是站点C，然后用对应的密钥库进行解密，从而实现了站点B分别与站点A和站点C通信的目的。此时，站点A和站点C的所有终端都是动态IPv6地址，因而这两者之间是无法直接通信的，可以通过站点B的协助进行通信。

4 结束语

互联网正在由IPv4向IPv6逐步迁移中，IPv6地址长度为128位，分为网络前缀和接口标识两部分，网络前缀由电信运营商分配，接口标识可以手工配置、随机生成或者通过EUI-64格式生成。研究发现，手工配置或者通过EUI-64格式生成的静态IPv6地址存在个人隐私泄露的网络安全风险，而随机生成的IPv6地址虽然提高了安全性，但是却不满足某些场景下基于IP地址进行网络访问控制的应用需求。为此，本文提出了一种基于ZUC加密的IPv6地址动态编码（ZBDA）算法，可以兼顾解决这两方面的问题。ZBDA算法基于ZUC加解密，算法安全性高，IPv6地址编码和地址验证速度快，在物联网及其他网络安全系统中具有一定的实际应用价值。

参考文献：

[1] 中共中央办公厅，国务院办公厅. 推进互联网协议第六版（IPv6）规模部署行动计划[EB]. 2017.
General Office of the Communist Party of China Central Committee, General Office of the State Council. Action plan for promoting the scale deployment of internet protocol version 6 (IPv6) [EB]. 2017.

[2] 中央网信办，国家发改委，工信部. 关于加快推进互联网协议第六版（IPv6）规模部署和应用工作的通知[EB]. 2021.
Office of the Central Cyberspace Affairs Commission, National Development and Reform Commission, Ministry of industry and information Technology of the People's Republic of China. Notice on accelerating the scale deployment and application of internet protocol version 6 (IPv6) [EB]. 2021.

[3] 中国信息通信研究院. 国家IPv6发展监测平台[EB]. 2024.
China Academy of Information and Communications Technology. China IPv6 development monitoring platform[EB]. 2024.

[4] Google. IPv6 adoption statistics[EB]. 2024.

[5] IETF. RFC 8200: Internet protocol, version 6 (IPv6) specification[S]. 2017.

[6] 杭州华三通信技术有限公司. IPv6技术[M]. 北京:清华大学出版社, 2010: 12-16.
H3C. IPv6 technology[M]. Beijing: Tsinghua University Press, 2010: 12-16.

[7] IETF. RFC 7136: Significance of IPv6 interface identifiers[S]. 2014.

[8] IETF. RFC 4291: IP version 6 addressing architecture[S]. 2006.

[9] PLONKA D, BERGER A. Temporal and spatial classification of active IPv6 addresses[C]//Proceedings of the 2015 Internet Measurement Conference. New York: ACM Press, 2015: 509-522.

[10] Organizationally unique identifier[EB]. 2024.

[11] ZOHAIB A, HOUMANSADR A. Automated detection of IPv6 privacy leakage in home networks[J]. Free and Open Communications on the Internet (FOCI), 2023(1): 30-34.

[12] SAIDI S J, GASSER O, SMARAGDAKIS G. One bad apple can spoil your IPv6 privacy[J]. ACM SIGCOMM Computer Communication Review, 2022, 52(2): 10-19.

[13] DUNLOP M, GROAT S, MARCHANY R, et al. IPv6: Now you see me, now you don't[C]//Proceedings of the Tenth International Conference on Networks (ICN). Wilmington: IARIA Press, 2011: 18-23.

[14] GROAT S, DUNLOP M, MARCHANY R, et al. The privacy implications of stateless IPv6 addressing[C]//Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research. New York: ACM, 2010: 1-4.

[15] ULLRICH J, KIESEBERG P, KROMBHOLZ K, et al. On reconnaissance with IPv6: a pattern-based scanning approach[C]//Proceedings of the 2015 10th International Conference on Availability, Reliability and Security. Piscataway: IEEE Press, 2015: 186-192.

[16] HOANG N P, NIAKI A A, GILL P, et al. Domain name encryption is not enough: privacy leakage via IP-based website fingerprinting[J]. arXiv preprint arXiv:2102.08332, 2021(4): 1-21.

[17] 张千里, 姜彩萍, 王继龙, 等. IPv6地址结构标准化研究综述[J]. 计算机学报, 2019, 42(6): 1384-1405.
ZHANG Q L, JIANG C P, WANG J L, et al. A survey on IPv6 address structure standardization researches[J]. Chinese Journal of Computers, 2019, 42(6): 1384-1405.

[18] NARTEN T, DRAVES R, KRISHNAN S, Privacy extensions for stateless address autoconfiguration in IPv6[S], RFC 4941, IETF,

2007: 1-23, https://www.rfc-editor.org/rfc/pdfrfc/rfc4941.txt.pdf.

[19] GONT F. A method for generating semantically opaque interface identifiers with IPv6 stateless address autoconfiguration (SLAAC)[S]. RFC 7217, IETF, 2014: 1-20, https://www.rfc-editor.org/rfc/pdfrfc/rfc7217.txt.pdf.

[20] ODERO S, DARGAHI T, TAKRURI H. Privacy enhanced interface identifiers in IPv6[C]//Proceedings of the 2020 12th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP). Piscataway: IEEE Press, 2020: 1-6.

[21] MIĆOVIĆ M, RADENKOVIĆ U, VULETIĆ P. Network layer privacy protection using format-preserving encryption[J]. Electronics, 2023, 12(23): 4800.

[22] 霍炜，郭启全，马原 . 商用密码应用与安全性评估[M]. 北京：电子工业出版社, 2020: 29-40.
HUO W, GUO Q Q, MA Y. Commercial cryptography application and security evaluation[M]. Beijing: Publishing House of Electronics Industry, 2020: 29-40.

[23] 中华人民共和国国家质量监督检查检疫总局，中国国家标准化管理委员会 . 信息安全技术 祖冲之序列密码算法 第1部分：算法描述GB/T 33133.1-2016[S]. 2017: 1-12.
General Administration of Quality Supervision, Inspection and Quarantine of the People's Republic of China, Standardization Administration of China. Information security technology-ZUC stream cipher algorithm-Part 1: Algorithm description, GB/T 33133.1-2016[S]. 2017: 1-12.

[24] IETF. RFC 4861: Neighbor discovery for IP version 6 (IPv6)[S]. 2007.

[25] 李子臣 . 商用密码算法原理与C语言实现[M]. 北京:电子工业出版社, 2020: 8-27.
LI Z C. Commercial cryptography algorithm theory and C language implementation[M]. Beijing: Publishing House of Electronics Industry, 2020: 8-27.

[26] FOREMSKI P, PLONKA D, BERGER A. Entropy/IP: uncovering structure in IPv6 addresses[C]//Proceedings of the Proceedings of the 2016 Internet Measurement Conference. New York: ACM, 2016: 167-181.

[27] IETF. RFC 4862: THOMSON S, NARTEN T, JINMEI T. IPv6 stateless address autoconfiguration[S]. 2007.

[28] IETF. RFC 8415: Dynamic host configuration protocol for IPv6 (DHCPv6)[S]. 2018.

[29] UTTARO J, HAAS J, TEXIER M, et al. BGP Flow-spec redirect to IP action draft-ietf-idr-flowspec-redirect-ip-01.txt[J]. IDR Working Group, Internet-Draft, Intended Status: Standards Track, 2014: 1-8.

[30] LITKOWSKI S, SIMPSON A, PATEL K, HAAS J. Applying BGP flowspec rules on a specific interface set draft-ietf-idr-flowspec-interfaceset-00.txt[J]. IDR Working Group, Internet-Draft, Intended Status: Standards Track, 2015: 1-8.

[作者简介]

刘永清（1979– ），男，东南大学网络空间安全学院博士生，国家计算机网络应急技术处理协调中心江苏分中心高级工程师，主要研究方向为下一代互联网、网络空间测绘。

曹玖新（1967– ），男，博士，东南大学网络空间安全学院教授，博士生导师，主要研究方向为社会计算、计算机网络、复杂网络。