基于内核指令检测技术的勒索病毒防护研究

靳　京

（奇安信网神信息技术（北京）股份有限公司，北京　100085）

摘　要：勒索病毒的核心和本质是对数据的加密操作，其在内核指令级的序列特征相对固定并有规律可循。对典型加密算法核心指令的基础特征进行了归纳和建模，形成基于特定 CPU 体系架构的典型加密算法汇编语言指令集。同时，采用基于 Trie 的递归行进算法对内存中指令代码序列进行动态解析分析，对运行中的加密算法指令及其序列特征进行匹配检测，可对典型加密算法核心操作实现指令级的实时监测和预警，从而提高对勒索病毒攻击过程中防护的准确性和有效性。实验证明，在某 ARM 架构平台中对使用特定加密算法指令的勒索病毒具有良好的检测效果。

关键词：指令检测；递归行进算法；勒索病毒防护；网络安全

中图分类号：TP309. 5　　**文献标识码：**A　　**DOI：**10. 19358/j. issn. 2097 - 1788. 2025. 08. 002

引用格式：靳京. 基于内核指令检测技术的勒索病毒防护研究［J］. 网络安全与数据治理，2025，44(8)：10 - 16.

Research on ransomware protection based on kernel instruction detection technology

Jin Jing

(Qi'anxin Wangshen Information Technology (Beijing) Co., Ltd., Beijing 100085, China)

Abstract: The core and essence of ransomware is the encryption operation of data. Its sequence characteristics at the kernel instruction level are relatively fixed and regular. The basic features of the core instructions of typical encryption algorithms are summarized and modeled to form a typical encryption algorithm assembly language instruction set based on a specific CPU architecture. At the same time, a Trie-based recursive marching algorithm is used to dynamically parse and analyze the instruction code sequence in memory, match and detect the running encryption algorithm instructions and their sequence characteristics. It can achieve real-time monitoring and early warning of the core operations of typical encryption algorithms at the instruction level, thereby improving the accuracy and effectiveness of protection against ransomware attacks. Experimental results have shown that it has a good detection effect on ransomware viruses using specific encryption algorithm instructions on a certain ARM architecture platform.

Key words: instruction detection; recursive marching algorithm; ransomware protection; cybersecurity

0　引言

近年来，勒索病毒的广泛传播逐步成为网络安全威胁的重要组成和发展趋势，且其攻击仍保持着强劲增长势头，同时已有从传统的加密勒索向数据泄露转变的迹象，对广大企业正常经营和社会稳定造成严峻挑战。

根据 NCC Group 的数据，2024 年共发生了不少于 5 263起成功攻击，成为勒索软件攻击数量最多的一年[1]。Chainalysis 的报告显示，数据泄露网站上的披露数量也不断上升[1]。世界财富 50 强企业、美国药品分销巨头 Cencora 甚至向“黑暗天使”（Dark Angels）勒索软件组织支付了创纪录的 7 500 万美元（约合人民币 5. 28 亿元）[1]。

相应地，越来越多网络安全企业投入到了反勒索病毒的技术和产品研发之中。然而，目前针对勒索病毒软件的防护思路主要集中在依靠监测勒索攻击的准备和投递环节，结合威胁情报，在病毒攻击的前期渗透阶段进行预警和阻断[2-3]，而对于攻击进行中的实时检测和分析机制相对较少，特别是针对勒索病毒的本质和核心技术[4]——加密行为的指令级监测和预警方法尚未见提及和应用。

根据对已知勒索病毒软件技术原理和攻击过程的研究，勒索攻击前期的准备和投递环节主要体现为渗透扩散、遍历检索、代码伪装等行为，病毒不断进行相应调整和改变生成新的变种，以应对主流的检测方法，但其核心加密算法却不会出现大的变化。因而对核心加密行

为的识别和拦截才是对病毒攻击中期的防护关键。

1 主流反勒索病毒软件技术现状

根据奇安信天眼威胁监测系统对勒索病毒常用攻击手法的分析可知，当前主流勒索病毒的攻击手法主要包括边界突破、病毒投放、加密勒索和横向感染等环节，如图 1 所示。

常见的防护方法一般采取威胁拦截、病毒查杀、威胁监测以及联动处置等方式加以应对，每种应对方式又包含多项细分技术和手段。具体攻击流程和防护手段如图 2 所示。

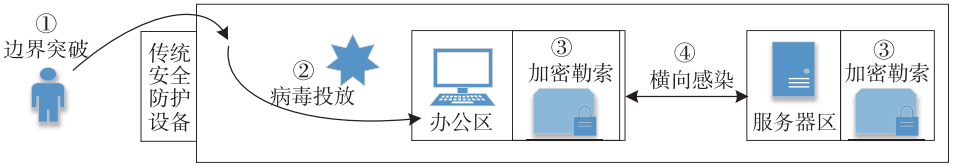

图 1 当前勒索病毒主要攻击过程示意图

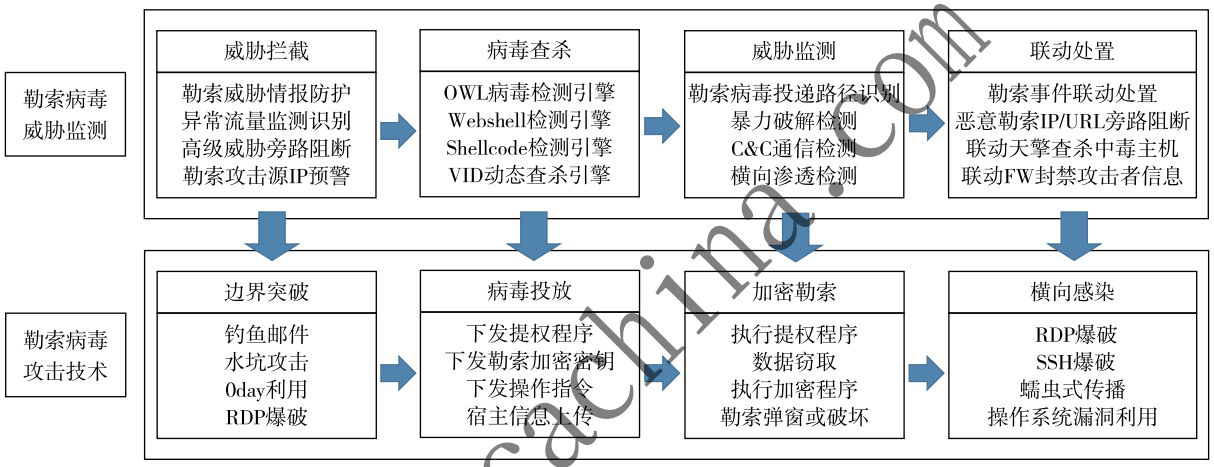

图 2 当前主流勒索病毒攻击手段及相应防护方式示意图

然而以上防护方式大都局限于把勒索病毒参照常规病毒攻击特征进行防护[5]，缺少对于勒索病毒本质攻击行为的监测和拦截思路。因而当病毒变种对在边界突破、病毒投放及横向感染环节进行调整升级后，相应的防护效果将大打折扣。

不论勒索病毒软件在传播、感染、渗透时采用多少千变万化的技巧和方式，其最终目的及本质核心操作是通过执行对数据进行加密，且绝大多数在完成加密时均使用常见固定的加密算法，并越来越多地采用对称加密与非对称加密算法相结合的复合加密方式[6]。

例如三六零安全科技股份有限公司对 2024 年仍在活跃传播且具有代表性的勒索病毒软件家族进行了深入分析，并统计了各家所采用的编程语言、加密算法及非对称密钥生成方式，具体情况如表 1 所示。

目前，信息安全企业逐渐认识到指令级防护的重要性，并已开始着手弥补勒索病毒防护拼图体系的此项空白。奇安信天狗漏洞攻击防护系统基于指令层的攻击检测，采用常规软件智能识别和权限管控技术，对受保护文件或数据进行权限算法匹配和管控，实现对勒索病毒攻击中期的拦截[7]。

表 1 主流勒索病毒加密算法分析（部分）

家族名称	编译语言	加密算法
RansomEXX	Rust	RSA4096 + AES256
Makop	C ++	RSA1024 + AES256
Buran	Delphi	RSA2048/512 + AES256
Loki	C#	RSA2048 + AES256
BeijingCrypt	C ++	RSA1024 + AES256
Stop	C ++	RSA1024 + Salsa20

然而，天狗系统虽然创新性地深入到了指令级的分析检测，但本质上还只是对特定指令的权限与合法性进行检测分析，而并非针对勒索病毒的核心特征——加密算法指令序列进行检测和防护，难免会影响检测的准确性和效率。

因此，本文在对典型加密算法流程及工作机制进行深入研究的基础上，对其核心指令汇编指令集的基础特

征进行归纳和分析，设计出了一种基于 Trie 递归行进算法的加密操作核心指令序列的检测技术。该项技术以典型加密算法核心指令为例建立了基于 ARM 架构精简指令集的汇编指令和对应的机器码，配合特定指令序列分析和匹配检测方法，可有效对加密算法操作实施实时监测和预警，大大提高对勒索病毒攻击中期防护的有效性和准确率。

2 基于加密算法核心指令的检测技术

基于加密算法核心指令序列的检测技术包含典型加密算法汇编语言指令集设计、基于 Trie 的指令检测递归行进算法，以及特定加密算法指令及其序列特征的匹配检测三个主要部分。

2.1 典型加密算法汇编语言指令集

当前国内外主流加密算法的标准和机制是固定的，虽然不同加密软件的代码实现方式各有不同，但其核心机制均有其独有特征。

通过分析采用复合加密过程的勒索软件，大多数加密方法是首先使用 AES 等对称加密算法对目标文件进行加密，然后从控制服务器获取 RSA 等非对称加密算法的公钥，对每个加密文件进行加密并附加对称加密算法密钥[8-9]。因而，加强对于勒索病毒攻击过程中的对称加密算法核心指令的检测是重中之重，也会提高在攻击发生的早期的检测成功率。本文以目前最有代表性的 SM4 和 AES 为例，详细介绍对于对称加密算法的核心指令序列检测技术。

2.1.1 SM4 加密算法特征指令分析

对典型国密算法 SM4 软件的加密算法进行分析可发现，在某个时段中会出现以下特定指令：

（1）连续进行 32 轮的四个字节的错位异或操作；

（2）含有字节代换指令，且按照已知 SM4 的 S 盒表进行 S 盒的代换；

（3）进行了反序变换。

通过对某开源 SM4 加密源程序进行如下逆向分析，可以分别得到这三方面指令序列特征：

（1）迭代错位异或操作分析

在轮密钥迭代伪代码中找到了如图 3 所示重复指令，把四个字节的数据转化为一个字长（32 位 CPU）的数据。

该指令共循环了 8 次，可知是一组连续存储的数据，访问了一个字节单元之后将 r9d 中的数据逻辑左移八位，与 eax 中的数据进行或运算存储到 eax 中，再进行零拓展。

```
shl     r9d, 8
or      r9d, eax
movzx   eax, byte ptr [rdx+3]
```

图 3 SM4 轮密钥算法字节错位拓展汇编指令截图

分析可知，轮密钥扩展参数的个数是 4 个，每个是一个字长的大小，也就是四个字节，那么异或操作就需要原始密钥的每个扩展参数也是一个字长的大小，而这里的操作是左位移八位再或运算，0 与任何数或都是任何数，也就是把四个字节的数字转化为一个字长的数字，再存到数组里与系统参数进行异或。

此运算共循环 32 次，对应 SM4 算法中连续 32 轮的四个字节的错位异或操作。

（2）非线性变换特征分析

SM4 算法中轮迭代算法和加密算法的显著特征是均存在 T 置换，而 T 置换中一定会重复用到非线性变换也就是 S 盒。如图 4 所示，在其指令中能找到在轮迭代和加密算法中均出现的轮动操作重复地址，查看其内容，可确定为 S 盒数据地址。

SM4 算法中 S 盒的系统参数表有固定取值，找到其 S 盒地址后，可确定其机器码，作为 SM4 算法加密指令的匹配基准。

（3）反序代码特征分析

在 SM4 加密算法伪代码中能发现如图 4 中指令代码。

```
mov     ecx, dword ptr [rsp+0C8h+var_48+0Ch
mov     eax, ecx
shr     eax, 18h
mov     [r8], al
mov     eax, ecx
shr     eax, 10h
mov     [r8+1], al
movzx   eax, byte ptr [rsp+0C8h+var_48+0Ch]
mov     [r8+3], al
shr     ecx, 8
mov     [r8+2], cl
mov     ecx, dword ptr [rsp+0C8h+var_48+8]
mov     eax, ecx
shr     eax, 18h
mov     [r8+4], al
mov     eax, ecx
shr     eax, 10h
mov     [r8+5], al
movzx   eax, byte ptr [rsp+0C8h+var_48+8]
mov     [r8+7], al
shr     ecx, 8
mov     [r8+6], cl
```

图 4 SM4 算法反序代码汇编指令截图

这里能够发现［rsp + 0C8h + var_48 + 0Ch］之后的数据变为了 8，结合数据的移动，可知该段指令是对数据进行反序操作。

以上指令与 SM4 的加密指令序列特征完全吻合，可作为 SM4 加密算法预设指令集。

若正在运行的某程序通过检测并识别其指令流，与上述指令集高度吻合，且在固定时段按照特定序列持续出现，即可初步判别为含有 SM4 加密算法。

2.1.2 AES 算法指令特征分析

在对 AES 加密算法的 CPU 指令进行分析后可发现，其指令序列中包含以下四种特定指令：

（1）按照 AES 算法的 S 盒表进行字节代换操作；

（2）按照第 0 行左移 0 字节，第 1 行左移 1 字节，第 2 行左移 2 字节，第 3 行左移 3 字节对状态矩阵进行行位移操作；

（3）行位移后所形成的状态矩阵与式（1）中的固定矩阵进行列相乘混合操作：

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \tag{1}$$

（4）对列混合后的矩阵进行密钥逐位异或操作。

由此可知，如果在某程序的指令序列中连续出现以上所述四种特征指令，则可初步判定包含 AES 加密算法。同时，还可根据变换操作进行的轮数，判断该算法为 AES128、AES192 还是 AES256[10-11]。

类似地，可对 RSA 及 ECC 等非对称加密算法特征指令及其序列进行分析和匹配。

综上，通过把识别到的 CPU 特定指令及其序列特征与预设加密算法指令集相对比，可准确判断出某程序是否包含加密行为，以及对应的算法种类。

2.2 基于 Trie 的递归行进算法

在需要对加密程序进行动态分析时，首先解析内存中的指令代码，在指令解析方法中引入 Trie 树结构。同时，解析过程使用从所选指令到向前反向解析方法，能够快速提取可能的指令序列，有效地利用 ARM 架构中高速缓存（cache）和指令集优化的特点，显著提高代码指令比较分析的准确性[12]。

Trie 树结构突出特点是其每个逻辑分支由其中的每个字符来确定。因此，可使每个节点保存一条程序指令，然后由每个节点到根节点的路径叠加即可形成相应的指令流，从而快速找出所有可能的指令序列。

在执行指令解析之前，必须首先选择 RET 指令等关键指令作为 Trie 树的根节点。然后设定从关键指令进行反向解析的字节长度，该字节长度既要符合指令序列的长度需求，又要有一定的限制范围，比如 16 个字节。

具体的解析步骤如下：

（1）分析待测程序的逻辑结构，找到需要解析的代码段，以便快速定位到可执行的加密指令序列。

（2）对于定位到的机器码要将代码段从头到尾逐字节读取，并与预置关键指令序列集中相应的机器码进行对比，判断是否为关键指令。

（3）如果是非关键指令，则循环执行第（2）步。

（4）如果是关键指令，则以该指令作为 Trie 树的根节点，然后向前按设置的字节长度读取指令机器码，与预置的跳转指令集进行匹配，判断是否匹配成功。如果未匹配成功，则依次增加读取的字节数重复判断；如果匹配成功，则需继续判断该指令能否对程序的执行流程产生影响。如影响，则继续增加读取的字节数重复判断；如果不影响，则在所创建 Trie 树上新增一个以该指令作为内容的节点，并在此节点上继续创建子树，以此类推。具体 Trie 树构造流程如图 5 所示。

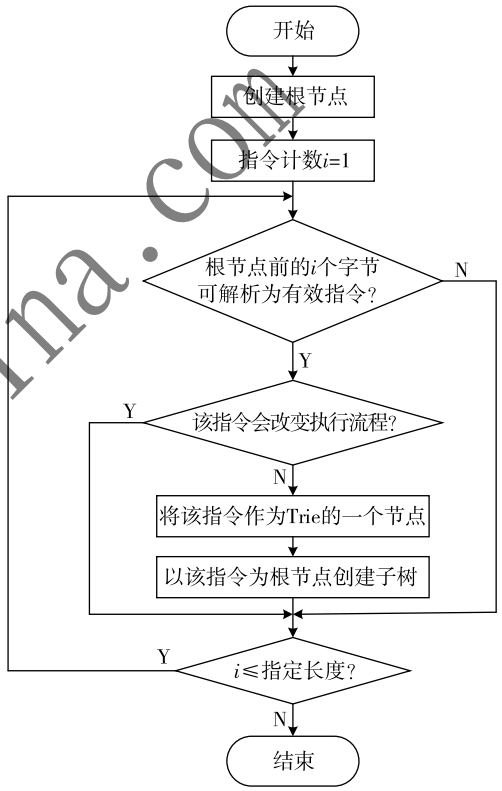

图 5 Trie 树构造流程图

由此，可以准确有效地将内存中加密算法指令序列进行解析，作为后续匹配分析的基础。

2.3 特定加密算法指令特征的匹配检测

结合前述递归行进算法，加密算法检测系统中设计了动态指令检测模块，并利用调试软件中的动态插桩工具 PIN 对指令进行分析。

首先，使用 PIN 实时收集程序运行关键位置的信息。PIN 提供的基于事件的插装功能可以在程序操作之前和期间收集与堆栈和线程对应的程序指令地址空间信息详细情况，以实现后续的指令匹配检测功能[13]。

通过收集到的详细信息，监测过程将调用动态指令检测匹配模块，并根据分析到的加密算法指令特征来设

计典型 ret、jmp 和 call 指令的检测逻辑。在程序预处理阶段，利用 PIN 动态加载二进制程序。程序信息加载到回调函数后，程序的主跳转指令信息将被拦截。进而根据指令类别可进入相应的指令逻辑分析阶段。如果存在上述三种类型的跳转指令信息，则将进入与之对应的检测模块进行指令检测，并在对应模块中调用基于 Trie 树的递归行进指令解析模块执行加密算法的指令解析。如果未含有上述三种跳转指令信息，程序将会直接执行加密算法的指令解析。具体流程如图 6 所示。

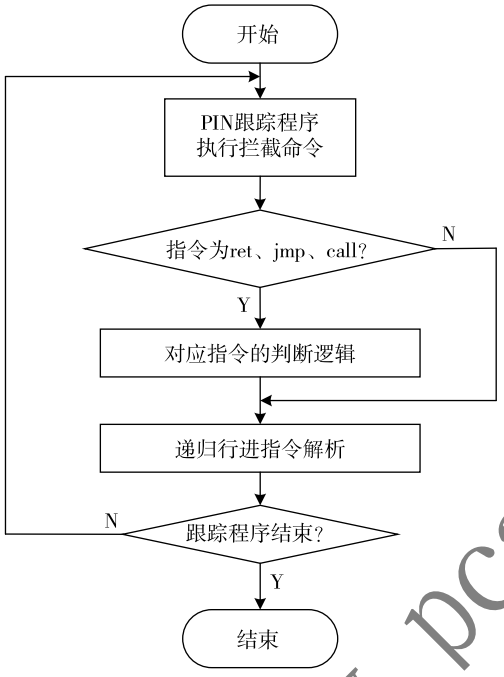

图 6　动态指令检测流程图

加密指令特征序列检测系统按照以上流程可实现所有代码分支的动态指令检测，并与预置的加密算法核心特征汇编指令集的机器码相匹配，即可实现对加密指令特征序列的精准检测。

2.4　检测方法与流程

根据前述关键技术，针对每个需要实时检测勒索病毒的用户实际情况和具体应用场景，基于内核指令特征的勒索病毒检测可按照以下方法和流程实施。

（1）载入主流加密算法指令特征库

作为一个预置插件或者工具库，用户需要首先安装或载入已知主流加密算法的指令特征库，该特征库是基于对不同加密算法的流程设计要求及关键参数设置深入分析而生成的标准库，这是所有实际检测所必备的基础。该特征库可根据主流加密算法的演变而持续更新完善。

（2）分析检测用户基础环境

因为不同的基础环境会生成不同的指令机器码，因而需要对用户软硬件基础环境进行自动或手动分析检测，明确具体的 CPU、操作系统指令集类型以及内存管理机制，便于生成相应的指令栈首地址及其他参数设置。

（3）生成加密算法的汇编指令集及对应机器码

基于用户实际硬件环境生成特征库中所有加密算法关键特征指令的汇编指令集，进而建立具体的机器码，作为特征指令序列检测的匹配基准，以提高检测准确性及匹配效率。

（4）对内存指令进行序列分析

调用加密算法特征指令分析程序对内存中运行的疑似加密进程进行指令分析，并通过基于 Trie 的递归行进算法还原相应的指令序列。

（5）指令特征匹配检测

将每条分析还原出的指令生成对应的机器码，与特征库中的已知加密算法的核心指令机器码相匹配，判断是否为疑似勒索病毒，进而采取相应的处置措施。

其主要检测流程如图 7 所示。

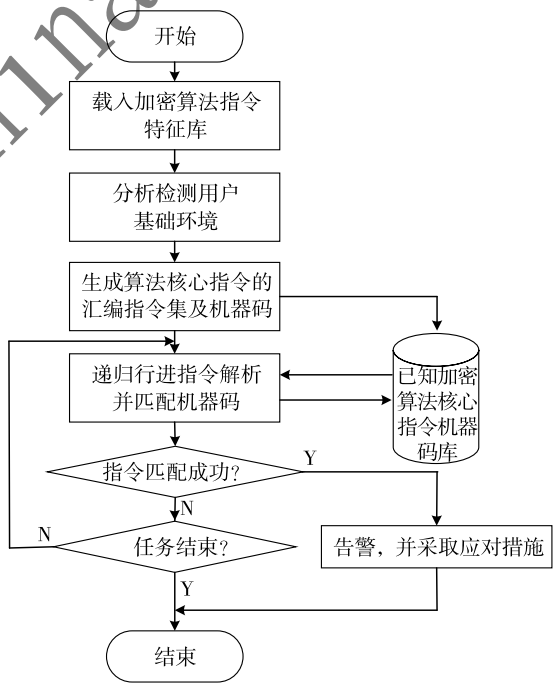

图 7　基于内核指令特征的勒索病毒检测流程图

以上为指令检测的核心流程，实际实施过程中可根据具体场景需求补充更为丰富和必要的辅助环节。

3　实验与测试

为了验证本加密算法内核指令检测技术的效果，本文选用了 MiniArm Pro 的精简指令集、可移植 μC/OS-II 操作系统，并重新配置了内存地址分配，搭建了专用实验环境。

3.1　CPU 指令系统

MiniARM Pro 是基于 MiniARM[14] 仿制的一个 5 级流

水线结构的 32 位 CPU，可实现独立的指令和数据缓存，并实现了 ARM7TDMI 指令集[15]的一个子集，能够支持 USR、SVC、IRQ、SYS 4 种运行模式。

MiniARM Pro 实现的指令系统是根据 μC/OS-II 内核代码分析后得到的精简指令集，该指令集支持 μC/OS-II 内核运行所需的必要指令，并支持 ARM7 的所有寻址方式[16]。为了提高指令检测效率，本文在 MiniARM 架构的基础上又对指令集进行了精简和完善。具体指令集如表 2 所示。

表 2　MiniARM Pro 指令系统

指令类型	条数	指令名称
数据传送指令	2	mov，movxz
数据处理指令	2	and，sub
转移指令	2	bl，bx
加载/存储指令	6	ldr，ldrb，ldrh 等
状态寄存器指令	2	mrs，msr
空指令	1	nop
跳转指令	3	ret，jmp，call
操作数位移操作指令	4	shl，lsl，lsr，asr

3.2　内存地址分配设置

与 MiniARM 平台一样，MiniARM Pro 也不含地址转换器，根据指令检测需要重新设置了 μC/OS-II 的内存分配设置，使 MiniARM Pro 的内存接口保持一致。在内核 main.inc 里对 ROM、RAM 和堆栈的大小以及起始地址等设置如下：

```
ROM_Base EQU 0x00000000;                    //ROM
ROM_Size EQU 0x00002000;                    //RAM
RAM_Base EQU 0x00002000;        //全局变量空间
Globe_Variable_Size EQU 10 * 1024;
STACK_SIZE EQU 1024 * 1;
RAM_Source_Base EQU RAM_Base + Globe_Variable_Size + STACK_SIZE;
STACK_LOCATION EQU RAM_Source_Base;
```

重新设置 Ram 的起始地址：

```
ER_data_plus_bss 0x00002000; …
```

3.3　生成汇编指令集及机器码

根据专用实验环境硬件型号，生成了 SM4 加密指令汇编指令集及对应的机器码，部分指令及机器码如下，作为测试匹配的基准：

汇编指令：mov r2d，#0　　→机器码：6c a3 00；

汇编指令：cmp r2d，#31　　→机器码：55 c8 1b；

汇编指令：shl r9d，8　　→机器码：e4 25 7b；

汇编指令：or r9d，eax　　→机器码：51 32 84；

汇编指令：movzx eax，byte ptr［rdx + 3］→机器码：e7 16 c3 90；

汇编指令：mov ecx，dword ptr［rsp + 0C8h + var_48 + 0ch］　　→机器码：f0 73 cc 28；

汇编指令：mov ecx，dword ptr［rsp + 0C8h + var_48 + 8］　　→机器码：0a 3e 64 b5；

S 盒系统参数机器码（16 × 16）：f6 92 d7 41 cf 3e 9b 1d 5b a0 c6 2b 56 73 fa 61 31 a7 33 19 4f ca 59 24 b6 f8 ad c3 8e 27 d5 11 3c 91 45 f2 1d 85 b4 30 41 72 c5 ab f3 16 34 d1 87 5e 33 79 09 8d 62 23 5c cb 26 92 dc 4a 15 33 74 d0 6c 48 af 1b f5 69 44 02 00 3f 6e a8 9d b2 1d db fa 88 c0 cd 20 65 47 5c 27 d7 53 46 81 09 6f 5b 10 b8 90 5f b6 44 54 df 59 eb 22 36 f7 93 e9 9a 50 1c a7 81 0e 46 8a 5d c2 26 43 fa 19 4b 3b e7 ce 30 b7 3c 7a 95 ba 8b a2 52 b5 55 e1 be ef 08 73 64 58 d3 c7 34 05 99 62 a6 a8 35 87 9e a1 e3 90 67 42 b3 07 6b 24 00 bf ae a0 1c 13 88 5a dc b7 db d2 47 0d 21 8b 5e 4b 0b 15 b5 c6 d6 27 a6 ff 30 f2 e8 df ed 54 73 bd 5d 26 ca fc de 34 b0 45 33 a7 89 fe 19 4f 05 24 c9 99 60 68 94 62 31 44 aa 3c 40 eb a2 67 a9 76 b2 3e 1b c8 5f 03 23 39 da 55 a1 43 b9 4a d5 ea 0e 84 93 bc 7d；

……

3.4　测试过程及结果

将原开源 SM4 加密源代码作为样本 1，然后将其分别进行“在主程序代码段中增加干扰指令（如打印“hello world!”）”“将迭代循环改为 4 轮”，以及“将 AES 加密算法代码中的逐位异或指令删除”等编辑修改操作，并将多个测试程序分别编译为可执行文件后作为测试样本 2 ~ 4。同时，从 GitHub 开源共享资源中得到了已将关键恶意代码去除的某 Wannacry 勒索病毒样本（防止被二次利用），通过分析得知该样本使用了一种 Rijndael S-Box 的 AES 加密算法对目标文件进行加密，而后用 RSA2 加密算法对 Rijndael S-Box 的对称加密 Key 进行加密。因此通过 IDA 反编译出其对加密 Key 进行加密的 RSA2 算法源码，作为测试样本 5。调用加密算法检测系统进行对比扩展检测。

由于真实勒索病毒样本代码中有大量的过程攻击流程和步骤，与单纯加密算法样本在反编译及检测环节的复杂度和计算量等方面均有不同，因而在对比实验设置时对于非加密类指令进行了优化分析设计，使单纯加密算法样本检测用时略有增加，但大大提高了真实勒索病毒样本的检测效率。

该系统检测结果分别截图如图 8 所示。

经对两种加密程序反复测试，其检测结果如表 3 所示。

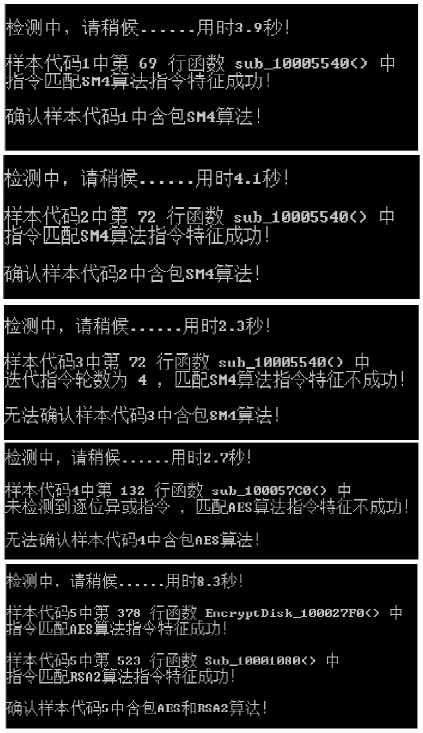

图 8　特征指令检测结果截图

表 3　加密指令特征序列检测系统测试结果

待测程序	测试结果	命中特征	耗时/s
样本 1	确认包含 SM4 算法指令	算法规则匹配成功	3.9
样本 2	确认包含 SM4 算法指令	算法规则匹配成功	4.1
样本 3	确认包含 SM4 算法指令	轮迭代指令不匹配	2.3
样本 4	未确认包含 AES 算法指令	密钥逐位异或操作不匹配	2.7
样本 5	确认包含 AES 及 RSA2 算法指令	算法规则匹配成功	8.3

根据测试结果可知，加密指令特征序列检测系统能够对 SM4、AES 及 RSA 算法加密指令特征进行有效检测，测试成功。

4　结论

通过对典型加密算法的原理及指令特征分析，结合基于 Trie 的递归行进算法进行指令检测与匹配，设计了一种基于内核指令检测技术的勒索病毒预警防护方法，并以 SM4 加密算法为例，在某 ARM 架构 CPU 精简指令集及 μC/OS-II 系统下实现了对加密算法核心指令集序列及对应机器码的有效检测。

该方法探索了从勒索病毒攻击本质入手进行精准检测的新思路，对只在欺骗、感染等外围手段方面更新升级而核心加密算法相同的各种勒索病毒变种的防护效果尤其显著。随着对更多架构 CPU 指令集及主流加密算法核心指令特征的深入研究，特别是与 AI 大模型乃至量子计算技术的进一步结合，该方法将会在我国自主安全环境下的诸多行业反勒索病毒领域中得到更广泛的应用。

参考文献

[1] 网络空间安全军民融合创新中心. 2024 年受害企业支付了约 60 亿元勒索软件赎金，但更多企业选择不妥协 [EB/OL]. [2025 - 04 - 01]. https://www. secrss. com/articles/75357.

[2] 刘川琦. 2023 年国内企业勒索病毒攻击态势分析 [J]. 中国信息安全，2024 (8)：44 - 47.

[3] 姜彬，居晓琴，施志刚. 勒索病毒的攻击原理与防范措施探究 [J]. 电脑知识与技术，2023，19 (31)：95 - 97，106.

[4] 刘波，张晓荣，祖婷. 勒索病毒技术研究综述 [J]. 电脑知识与技术，2024，20 (22)：79 - 81.

[5] 赵佩. 勒索病毒攻击事件漏洞分析及应对防护策略 [J]. 电子技术与软件工程，2019 (4)：201.

[6] 360 安全能力中心反病毒部. 2024 年勒索软件流行态势报告 [R]. 2025.

[7] 奇安信网神信息技术 (北京) 股份有限公司. 奇安信天狗漏洞攻击防护系统产品白皮书 [Z]. 2024.

[8] 董昱宏，宋广佳. 勒索病毒技术发展研究综述 [J]. 计算机应用与软件，2023，40 (1)：331 - 343.

[9] 薛丹丹，王媛媛，邵一潇，等. 勒索病毒的原理及防御措施 [J]. 网络安全技术与应用，2023 (2)：10 - 12.

[10] 靳京，孙烁、蒋红宇. 一种病毒防御方法、装置以及云端浏览器：中国，116127455B [P]. 2024 - 03 - 15.

[11] 靳京，孙烁、蒋红宇. 一种病毒防御方法、装置以及云端浏览器：中国，CN202211737917. 0 [P]. 2022 - 12 - 31.

[12] 韩卓，冉晓旻，陈迎春，等. 一种基于 Trie 的内存指令反向解析方法 [C]//2010 Asia-Pacific Conference on Information Theory，2010：366 - 369.

[13] 蒋廉. 一种基于程序运行期间指令特征的 ROP 攻击检测方法 [D]. 成都：电子科技大学，2021.

[14] 李山山，汤志忠，周继群. 基于 FPGA 的开放式教学 CPU 的设计与测试系统 [J]. 计算机工程与应用，2005 (14)：98 - 100.

[15] 杜春雷. ARM 体系结构与编程 [M]. 北京：清华大学出版社，2003.

[16] 李佳. ARM 系列处理器应用技术完全手册 [M]. 北京：人民邮电出版社，2006.

（收稿日期：2025 - 05 - 16）

作者简介：

靳京 (1975 -)，男，博士，工程师，主要研究方向：密码学、APT 网络攻击与防御技术、网络态势感知与评估、隐蔽信道检测等。

版权声明

凡《网络安全与数据治理》录用的文章，如作者没有关于汇编权、翻译权、印刷权及电子版的复制权、信息网络传播权与发行权等版权的特殊声明，即视作该文章署名作者同意将该文章的汇编权、翻译权、印刷权及电子版的复制权、信息网络传播权与发行权授予本刊，本刊有权授权本刊合作数据库、合作媒体等合作伙伴使用。同时，本刊支付的稿酬已包含上述使用的费用，特此声明。

《网络安全与数据治理》编辑部