DOI:10.16836/j.cnki.jcuit.2011.02.003
第26卷第2期 成都信息工程学院学报 Vol.26 No.2
2011年4月 JOURNAL OF CHENGDU UNIVERSITY OF INFORMATION TECHNOLOGY Apr. 2011

文章编号:1671-1742(2011)02-0127-05

功耗分析平台中混合编程的应用研究

孙敦灿, 陈 运, 万武南, 索 望
(成都信息工程学院信息安全研究所,四川 成都 610225)

摘要:功耗分析平台是有效实施功耗分析攻击的工具,需要服务器与示波器等多种外设进行通信,同时还需要处理大量高维数组并且要求有良好的可操作性,如果采用单一编程语言来开发,则存在功耗分析攻击效率较低的问题。针对该问题,提出了一种新的解决方法,将Java、C++和Matlab这3种编程语言结合,在功耗波形采集、波形特征处理等方面采用混合编程技术进行开发,充分发挥这3种编程语言各自的优势。实验结果表明,在保证功耗分析攻击准确率的基础上,攻击的效率比采用单一语言编程时提高了30%以上。

关 键 词:功耗分析攻击;攻击效率;混合编程;Java;C++;Matlab
中图分类号:TN918.1 文献标识码:A

当前,单纯从数学上来破解某一成熟的密码体制已经很难。具有易实现、易探测、易分析等特点的功耗分析攻击[1]成为密钥破解的强有力手段之一。而设计和实现功耗分析平台的目的就是要简化功耗攻击的流程、方便操作、在保证功耗分析攻击准确率的基础上提高攻击的效率。

通常功耗分析平台采用单一编程语言进行开发。国外荷兰RISCURE公司等开发的Inspector,是采用Java开发的功耗分析平台,虽然利用了Java的多线程并发操作,但是在频繁处理功耗分析攻击中的高维数组时,速度很慢,攻击的效率也很低。国内的西安电子科技大学[2]等科研结构采用VB作为开发语言,开发出的功耗分析平台功能较简单,没有实现多线程并发操作,而且运行时占用内存很大,导致运行速度慢,攻击的效率低。

文中将功耗分析攻击中功耗波形的采集、波形特征的处理、波形文件的显示等功能,采用混合编程[3]的技术实现。把攻击的的流程集成化,减少了冗余的操作,达到了在保证攻击准确率的基础上提高攻击效率的目的。

1 混合编程的思想概述

在科学研究计算中,数据的获取和处理可能会使用多种手段。比如通过示波器采集数据,利用Matlab进行高维数组的处理等。如果将数据的获取和处理都集成在一起,从得到原始数据到中间过程处理再到最后结果的分析进行一体化操作,那么就会提高数据处理的效率。由于每种编程语言都只是在某一方面具有优势,这样一来,只使用单一的某种语言来做上述一体化操作就不能满足数据集成处理的要求。

如果把不同部分的算法和数据交给在这方面具有优势的编程语言来处理,这些不同的语言之间通过混合编程的技术进行数据交互,开发人员就不需要为一些操作和算法而专门编写大量的代码。和使用单一语言进行开发相比,这样可以实现编程语言的优势互补,既提高了数据处理的效率又缩短了开发的时间。

2 功耗分析平台

基于混合编程来实现的功耗分析平台,其功能就是希望通过一系列高效的数据处理,获取智能卡中的密钥信息[4]。

与以往采用单一的编程语言开发的功耗分析平台不同,文中将Java、Matlab和C++这3种主流的开发语言利用混合编程的技术进行结合,系统的主架构采用Java作为开发语言,利用JavaSwing技术实现界面的显示和各

收稿日期:2011-03-30
基金项目:国家自然科学基金资助项目(60873216);四川省科技支撑计划重点项目(2011GZ0170)

个功能部分之间的衔接；采用 Matlab 作为大部分攻击算法的实现语言；采用 C++作为系统示波器控制部分的开发语言等，这样就可以提高系统的集成度，弥补单一语言开发的系统攻击效率低的不足。

功耗分析平台系统框架图如图 1 所示：

基于混合编程技术设计的功耗分析平台，通过服务器给密码芯片和示波器发送指令来获取密码芯片工作时密码算法的功耗，最后在服务器上进行数据处理进而获取密钥信息。

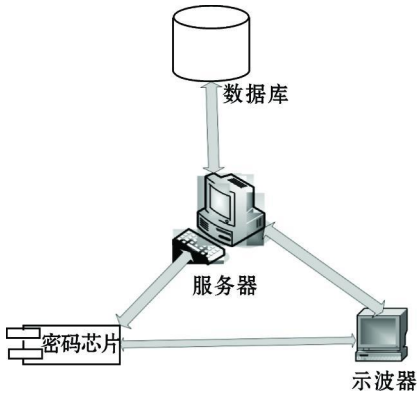

图 1 功耗分析平台系统框架图

2.1 功耗分析平台中 **Java** 和 **C++** 混合编程的应用

功耗分析攻击的一个重要方面就是采集功耗信号，这是进一步进行数据分析和处理的前提。

在本平台中，对功耗信号的采集使用某型号的示波器操作，但是示波器的生产厂商提供的开发文档中不支持 Java 作为示波器和工作站的传输语言介质，因此选取 C++作为控制示波器的编程语言，通过 Java 与 C++的混合编程，从而使 Java 也能控制示波器进行一定的操作。

Java 和 C++的混合编程，有多种方式可以实现，文中采用可靠性系数更高的 JNI[5] (Java Native Interface)技术来实现。具体的流程如下：

(1)在 Java 工程中编写要实现的本地化方法的接口函数，以便对示波器进行控制；

(2)编译该 Java 源文件，得到对应的.class 文件，假设名字为 A.class；

(3)在 JDK 安装路径下的 bin 目录中调用 javah 命令，执行生成.h 头文件的命令，图 2 是在命令提示符中执行该命令的截图。

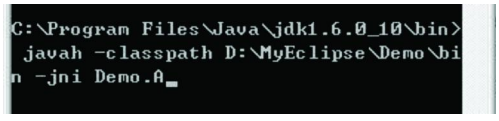

图 2 jni 命令执行图

该命令执行完后，就会在 JDK 的 bin 目录下生成.h 头文件，名字和上述第二步中的 A.class 文件相对应；

(4)在 VC 开发环境中新建一个空白的 DLL 工程，在此工程中对上述生成的.h 头文件中定义的接口函数进行实现，这部分是真正对示波器起作用的代码；

(5)编译运行本 DLL 工程，生成 dll 文件，将其放到 Java 工程的类路径目录下；

(6)在 Java 主程序的代码中，加载编译好的 dll 文件，这样就可以在 Java 程序中对示波器进行控制。

同时 Java 和 C++的混合编程技术还应用到了本平台中的单例运行模式。

2.2 功耗分析平台中 **Java** 和 **Matlab** 混合编程的应用

对功耗数据的分析需要大量比较复杂的数学运算，Java 并不擅长做数学运算，而 Matlab 的优势恰恰就是可以很方便地对数据进行数学处理。把功耗分析攻击中需要进行的复杂数学运算利用 Matlab 处理，在 Java 中调用 Matlab，就可以充分发挥 Matlab 和 Java 的优势，提高功耗分析攻击的效率。

Java 和 Matlab 混合编程[6] 的流程如下：

(1)在 Matlab 中编写提取波形特征的.m 文件；

(2)以 Matlab7.11.0 版本为例，在主面板的“文件”菜单下有名为“New”的子菜单，在“New”菜单中选取“Deployment Project”菜单选项，之后选取合适的保存路径和要编译成 Java 类的类名，将写好的.m 文件添加到要打包的类中编译即可。编译好的类就在 jar 包中被封装好了；

(3)用 Matlab 打包成的 jar 包可以像其他普通的 jar 包一样在 Java 文件中被调用，前提是将 Matlab 提供的 javabuilder.jar 包也添加到同样的路径下；

(4)在 Java 代码中调用上述生成的 jar 包中的类，如此便在 Java 中完成提取功耗波形特征的功能。

3　性能比较

采用混合编程的技术来开发功耗分析平台和以往不采用混合编程开发的功耗分析平台相比,在功能的实现上和攻击的效率上差别会很大。下面以波形源数据的采集、功耗波形在服务器上的显示和功耗波形特征的提取为例来具体说明(实验数据均是在实验 500 次以后得到的平均值)。

3.1　波形源数据采集部分的对比

在采样率固定为 1M 的前提下,变化采样点和采样组数的数值,分别在使用混合编程技术和不使用混合编程开发的功耗分析平台上进行采样,平均采样时间对比如表 1 所示。

表 1　平均采样时间对比表

采样点数	采样组数	混合编程平均所需时间	非混合编程平均所需时间	混合编程/非混合编程
100K	1	7 秒	7 秒	100%
100K	10	49 秒	59 秒	83.1%
100K	50	3 分 54 秒	4 分 45 秒	82.1%
100K	100	7 分 30 秒	9 分 23 秒	79.95
1M	1	10 秒	13 秒	76.0%
1M	10	1 分 28 秒	2 分 1 秒	72.75
1M	50	6 分 50 秒	9 分 29 秒	72.0%
1M	100	13 分 27 秒	17 分 19 秒	68.0%
10M	1	45 秒	1 分 1 秒	73.7%
10M	10	9 分 30 秒	13 分 23 秒	70.9%
10M	50	47 分 3 秒	1 时 8 分 7 秒	69.1%
10M	100	1 小时 14 分 45 秒	1 时 51 分 6 秒	67.2%

由表 1 可见,在采样率、采样点和采样组数全都一样的情况下,使用混合编程的技术开发的功耗分析平台,可以使得在采集大量数据时,完成采样的平均时间提高近 30%。并且当采样率/采样点的值越大,样本条数也越多的时候,这种效率的提高会更加明显。

3.2　功耗波形的显示时间对比

由于 Java 并不擅长做大规模数据的处理,因此采用 Java 和 Matlab 混合编程的技术进行数据处理,在处理的效率上肯定会有所提高。表 2 是在不同平台上打开相同大小的数据文件所耗费的时间对比。

表 2　数据文件波形显示所耗费的平均时间对比表

波形文件大小	混合编程平台平均所需时间	非混合编程平台平均所需时间	混合编程/非混合编程
30K	1 秒	3 秒	33.3%
300K	1 秒	5 秒	20.0%
3M	2 秒	8 秒	25.0%
30M	3 秒	50 秒	6.0%
300M	14 秒	3 分 58 秒	5.8%

由表 2 可以看到,在波形文件较小时,混合编程平台上显示波形的速度要比非混合编程开发的平台快 70%以上,当波形文件比较大的时候,比如 30M,甚至是 300M 时,混合编程平台上波形显示的速率就比在非混合编程平台上显示同样波形文件的速率提高 90%以上。波形文件越大,这种效率的提高就越明显。

图 3 是采用 Java 和 Matlab 混合编程技术后对某一数据文件的波形显示;图 4 是不采用混合编程技术,对同一数据文件的波形显示。

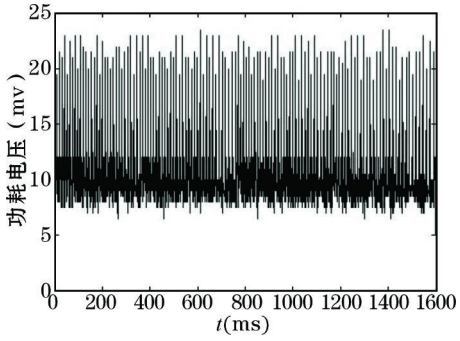

图 3 采用混合编程技术所显示的波形文件

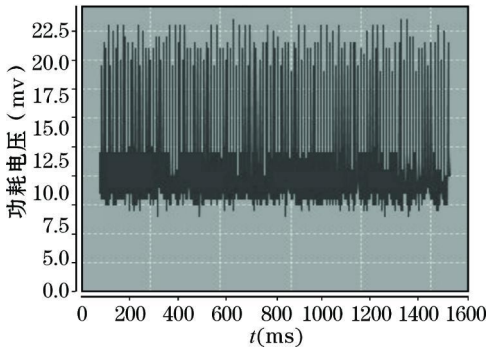

图 4 不采用混合编程技术所显示的波形文件

通过对图 3、图 4 的比较可知，采用混合编程技术开发的功耗分析平台中，功耗波形显示的精度也要比不采用混合编程技术时高。

3.3 功耗波形的特征提取时间对比

表 3 是当单条样本大小固定为 30M，在对样本进行特征处理时，采用混合编程技术开发的功耗分析平台和不采用混合编程开发的功耗分析平台在处理样本时间上的对比。

表 3 波形特征处理所耗平均时间对比表

样本条数	混合编程平均所需时间	非混合编程平均所需时间	混合编程/ 非混合编程
2	25 秒	29 秒	86.2%
10	42 秒	1 分 6 秒	63.6%
20	1 分 3 秒	2 分 17 秒	45.9%
50	1 分 45 秒	4 分 53 秒	35.8%

可见，混合编程开发的功耗分析平台在对波形数据进行特征处理时，比在非混合编程平台上处理相同的数据用的时间短，而且随着数据量的增加，时间上的差距越大。

3.4 完整攻击过程的效率和准确率的对比

以对某厂商生产的智能卡（使用 RSA 密码算法）的 SPA 攻击为例，分别针对样本条数为 2 条、10 条、20 条、50 条、100 条的情况，不同平台下完整攻击过程的攻击时间和攻击准确率的曲线图如图 5 所示。

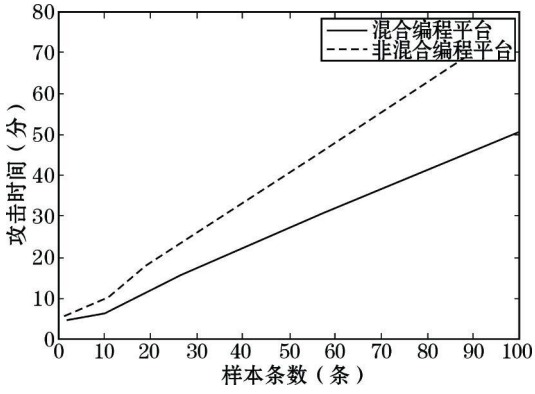

图 5 波形特征处理所耗时间对比表

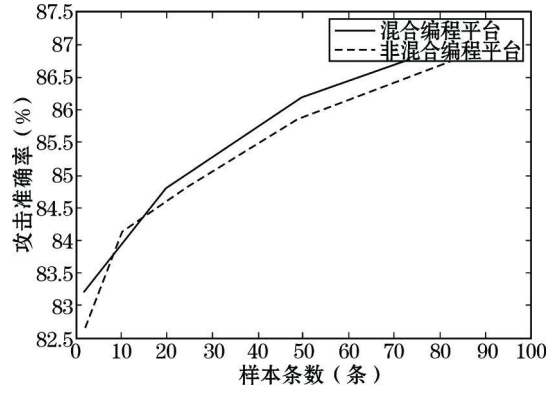

图 6 样本条数一样时的攻击准确率的对比

由图 5、图 6 对比可知，样本条数和样本大小一定时，采用混合编程技术和不采用混合编程技术时，在攻击的准确率方面相差不大，但是在攻击的时间上，采用混合编程技术要明显优于非混合编程，整体攻击的效率要提高 30%以上。

4　结束语

针对单纯用 Java 开发功耗分析平台存在功能较简单、效率不高的问题,提出了基于混合编程的功耗分析平台的设计。并且针对波形采样、数据显示等功能,分别在混合编程开发的平台上和非混合编程开发的平台上做了测试。通过对比,发现基于混合编程实现的功耗分析平台充分利用了各种语言的优势,软件本身具有很高的质量,可以在保证攻击准确率的同时使得攻击的效率提升 30%以上,有很强的自动性,便于对智能卡的安全性进行更深入的分析。这对于功耗分析攻击的研究具有重要的意义,同时也对其他需要采用混合编程技术的平台具有借鉴意义。

参考文献:

[1] Kocher P, Jaffe J, Jun B. Differential Power Analysis[C]. Proc. of Crypto' 99. Berlin, Germany: Springer-Verlag. 1999.
[2] 孙春辉. 边信道攻击设备的研究与实现[D]. 西安:西安电子科技大学, 2009.
[3] 任文杰. Matlab 和 Java 的混合编程研究与实现[J]. 测控技术, 2009, 28(1):77－82.
[4] 周丽莎, 陈运. 真实环境下对幂剩余指数的 SDPA 攻击[J]. 计算机工程, 2010, 36(7):156－158.
[5] 马海民. 基于 Java 的分布式异构资源监测模型研究[J]. 计算机工程与设计, 2009, 30(8):1863－1868.
[6] 廖云伢. 基于 Java 与 Matlab 集成的虚拟实验平台的设计与实现[J]. 计算机应用, 2007, 27(2):394－396.

Approach on Hybrid Programming Technique Applied in Power Analysis Attack System

SUN Dun-can,　CHEN Yun,　WAN Wu-nan,　SUO Wang
(Information Security Institute, Chengdu University of Information Technology, Chengdu 610225, China)

Abstract: Power analysis system is the right tool for power analysis attack. To accomplish the attacks properly, it needs the PC server communicates with different types of peripheral devices such oscilloscope. And it requires to disposal of mass of high dimension arrays with good operation-ability. Programming with unique programming language would lead to deficiency of power analysis attack. A new method was proposed by hybrid programming of Java, C++ and Matlab, which took the advantages of all these languages. Experiments showed that more than 30% attack efficiency increased and with no accuracy deficiency by using our method.

Key words: power analysis attack; attack efficiency; hybrid programming; Java; C++; Matlab