

文本复制检测报告单(全文对照)

№:ADBD2018R_20180107222251201802231016081001863086756

检测时间:2018-02-23 10:16:08

检测文献: 4_李勃_面向移动设备的cache攻击关键技术研究

作者: 李勃

检测范围: 中国学术期刊网络出版总库

中国博士学位论文全文数据库/中国优秀硕士学位论文全文数据库

中国重要会议论文全文数据库

中国重要报纸全文数据库

中国专利全文数据库

互联网资源(包含贴吧等论坛资源)

英文数据库(涵盖期刊、博硕、会议的英文数据以及德国Springer、英国Taylor&Francis 期刊数据库等)

港澳台学术文献库

优先出版文献库

互联网文档资源

图书资源

CNKI大成编客-原创作品库

学术论文联合比对库

个人比对库

时间范围: 1900-01-01至2018-02-23

检测结果

总文字复制比: **1.8%**

跨语言检测结果: **0%**

去除引用文献复制比: **1.7%**

去除本人已发表文献复制比: **1.8%**

单篇最大文字复制比: **0.7%** (系统时间 - haima1998的专栏 - 博客频道 - CSDN.NET)

重复字数: [987]

总段落数: [7]

总字数: [53936]

疑似段落数: [4]

单篇最大重复字数: [390]

前部重合字数: [83]

疑似段落最大重合字数: [697]

后部重合字数: [904]

疑似段落最小重合字数: [72]



指标: ☐ 疑似剽窃观点 ☒ 疑似剽窃文字表述 ☐ 疑似自我剽窃 ☐ 疑似整体剽窃 ☐ 过度引用

表格: 0

脚注与尾注: 0

0% (0) 中英文摘要等 (总4277字)

2% (83) 第一章绪论 (总4093字)

0.6% (72) 第二章相关技术以及原理 (总11671字)

13% (697) 第三章计时方式及驱逐策略 (总5362字)

1.2% (135) 第四章攻击方案的设计 (总10853字)

0% (0) 第五章AES攻击验证 (总7564字)

0% (0) 第六章预防攻击措施 (总10116字)

(注释: 无问题部分 文字复制比部分 引用部分)

1. 中英文摘要等

总字数: 4277

相似文献列表 文字复制比: 0%(0) 疑似剽窃观点: (0)

2. 第一章绪论

总字数: 4093

相似文献列表 文字复制比: 2%(83) 疑似剽窃观点: (0)

1 城域网应用层P2P流量预测模型的研究

1.7% (70)

	董智超(导师：王东) - 《湖南大学硕士论文》 - 2010-05-10	是否引证：否
2	网络终端设备的双网物理隔离方法研究 许乃利(导师：徐超;谭守标) - 《安徽大学硕士论文》 - 2011-04-01	1.3% (53) 是否引证：否
3	基于粗集约简构造概念格的关联规则挖掘 崔立霞(导师：张健;王德兴) - 《上海海洋大学硕士论文》 - 2012-01-05	0.9% (37) 是否引证：否

原文内容		相似内容来源
1	<p>此处有 33 字相似</p> <p>异步攻击方式进行探索。</p> <p>第六章预防攻击措施。详细介绍了移动设备暴露出来的攻击漏洞，以及预防措施。</p> <p>总结与展望。首先对</p> <p>本文的工作作总结，并指出可能存在的不足，最后对后续工作的展望。</p>	<p>城域网应用层P2P流量预测模型的研究 董智超 - 《湖南大学硕士论文》 - 2010-05-10 (是否引证：否)</p> <p>1.到理想的预测效果，比其他的预测模型有着更高的预测精度，平均相对误差为4.04%。最后，对本文所做的研究工作进行了总结，并对未来的工作进行了展望。硕士学位论文第2章城域网P2P流量的特征分析 随着网络测量技术的发展，越来越多的网络研究人</p>

指 标

疑似剽窃文字表述

- 1.4 本文的组织结构
本文由五章组成，主要包含如下内容：
第一章绪论。包含Cache攻击的背景

3. 第二章相关技术以及原理

总字数：11671

相似文献列表 文字复制比：0.6%(72) 疑似剽窃观点：(0)

1	Wi-Fi无线局域网安全协议分析 张慧;-《湖北第二师范学院学报》- 2011-02-20	0.3% (35) 是否引证：否
2	基于可重构的密码算法的设计与实现 庄德坤(导师：时龙兴;虞建立) - 《东南大学硕士论文》 - 2016-06-23	0.3% (33) 是否引证：否

原文内容		相似内容来源
1	<p>此处有 36 字相似</p> <p>致较长的访问时间。</p> <p>2.3 AES加密</p> <p>AES加密技术是由美国国家标准研究所提出的，为了代替DES数据加密技术的高级加密技术。它是使用最广泛的对称加密算法，其加密和解密使用的是相同的密钥。</p> <p>其加密过程如下图所示：</p> <p>图 7 AES加密过程</p> <p>其中明文P代表需要加密的数据，密钥K则表示加密过程中使用的密钥。对于</p>	<p>Wi-Fi无线局域网安全协议分析 张慧;-《湖北第二师范学院学报》- 2011-02-20 (是否引证：否)</p> <p>1.然后接入点发出质询消息。移动设备响应质询,证明自己知道密钥,如果证据被接受,接入点发送成功消息。图1 WEP认证WEP的加密使用RC4的字节流加密算法,加密和解密使用相同的密钥。为了避免在加密中使用固定密钥,在WEP中引入初始化矢量(IV)。也就是说,将密钥和一个24位数组合在一起,代替固定密钥,</p>
2	<p>此处有 36 字相似</p> <p>S加密解密过程的流程图，待加密的数据组合为16字节的明文后，与扩展密钥进行轮密钥加之后进过总共10轮的变换，每轮变换包含</p> <p>字节代替、行移位、列混淆以及轮密钥加操作，其中最后一轮不包含列混淆变换，</p> <p>最后生成16字节的密文，也就是加密之后的数据。解密过程就是相应加密过程的逆操作，通过密钥和密文进行轮密钥加操作后同样执行</p>	<p>基于可重构的密码算法的设计与实现 庄德坤 - 《东南大学硕士论文》 - 2016-06-23 (是否引证：否)</p> <p>1.2 256Nk 4 6 8Nr 10 12 4加密运算的圈变换由四个不同的变换组成.它们分别是：字节替换、行移位、列混淆和轮密钥加。其中最后一轮圈变换不包含列混淆。AES算法的解密过程即加密运算的逆过程，解密运算的圈变换同样由四部分组成，它们分别是：逆向字节替换、逆向行移位、</p>

--	--	--

4. 第三章计时方式及驱逐策略	总字数：5362
-----------------	----------

相似文献列表 文字复制比：13%(697) 疑似剽窃观点：(0)

1	系统时间 - haima1998的专栏 - 博客频道 - CSDN.NET - 《网络 (http://blog.csdn.net) 》 - 2017	7.3% (390) 是否引证：否
2	clock - 《网络 (http://blog.csdn.net) 》 - 2017	6.1% (327) 是否引证：否
3	time,gettimeofday,clock_gettime, - 《网络 (http://blog.csdn.net) 》 - 2013	5.7% (306) 是否引证：否
4	time,gettimeofday,clock_gettime,_ftime (转载) - 乡禅 - 《网络 (http://blog.csdn.net) 》 - 2013	5.7% (306) 是否引证：否
5	时间转换相关函数 - zuokong的专栏 - 博客频道 - CSDN.NET - 《网络 (http://blog.csdn.net) 》 - 2013	5.7% (306) 是否引证：否
6	linux 上时间相关的函数_xiaoqi - 《网络 (http://blog.sina.com) 》 - 2013	5.7% (306) 是否引证：否
7	【转】time,gettimeofday,clock_gettime,_ftime_dawnstar1 - 《网络 (http://blog.sina.com) 》 - 2013	5.7% (306) 是否引证：否
8	time,gettimeofday,clock_gettime,_ftime,gmtime_一棵草Telen - 《网络 (http://blog.sina.com) 》 - 2014	5.7% (306) 是否引证：否
9	timerfd API使用总结 - 詆調 - CSDN博客 - 《网络 (http://blog.csdn.net) 》 - 2017	5.5% (297) 是否引证：否
10	201610231642111642_于茜_基于ARM架构的LBlock轻量级分组密码Cache计时分析研究 于茜 - 《学术论文联合比对库》 - 2016-10-23	5.5% (293) 是否引证：否
11	城市轨道交通车载二取二平台关键技术的研究 谢刚 - 《学术论文联合比对库》 - 2013-05-14	4.6% (245) 是否引证：否
12	201091303517288_杜江 杜江 - 《学术论文联合比对库》 - 2013-05-12	4.6% (245) 是否引证：否
13	如何统计程序占用cpu时间? - 博客频道 - CSDN.NET - 《网络 (http://blog.csdn.net) 》 - 2013	4.3% (229) 是否引证：否
14	linux 定时器 - whm2300的专栏 - 博客频道 - CSDN.NET - 《网络 (http://blog.csdn.net) 》 - 2013	4.3% (229) 是否引证：否
15	城市轨道交通车载二取二平台关键技术的研究 谢刚(导师：陈荣武) - 《西南交通大学硕士论文》 - 2013-05-01	3.9% (208) 是否引证：否
16	移动IPv6开源代码从用户空间到内核的移植 林胜洁(导师：刘彦明) - 《西安电子科技大学硕士论文》 - 2011-12-01	2.3% (121) 是否引证：否
17	空中中继网络链路仿真的设计与实现 陆金杏(导师：董燕) - 《华中科技大学硕士论文》 - 2015-05-01	1.9% (104) 是否引证：否
18	linux/unix - 程序狗的成长之路 - 博客频道 - CSDN.NET - 《网络 (http://blog.csdn.net) 》 - 2017	1.8% (98) 是否引证：否
19	2012年09月存档 - 博客频道 - CSDN.NET - 《网络 (http://blog.csdn.net) 》 - 2013	0.8% (43) 是否引证：否

	原文内容	相似内容来源
1	<p>此处有 52 字相似</p> <p>了通过读取寄存器获取CPU周期来度量时间外，还有其他3种可选方式来度量访存或访问cache的时间。包括：3.1.1</p> <p>clock_gettime系统调用</p> <p>"clock_gettime"是基于Linux C语言的时间函数，计时粒度能够精确到纳秒。使用时需要包含time.h头文件。其函数声明为"int clock_gettime(clock</p>	<p>系统时间 - haima1998的专栏 - 博客频道 - CSDN.NET - 《网络 (http://blog.csdn.net) 》 - (是否引证：否)</p> <p>1.1. clock_gettime 函数"clock_gettime"是基于Linux C语言的时间函数,他可以用于计算精度和纳秒。播放器中，一般是用CLOCK_MONOTONIC类型时间 函数介绍 1 2 e&lt;time.h</p>

<div data-bbox="76 1081 100 1115" data-label="Text">2</div> <div data-bbox="150 38 802 790" data-label="Text"> <p>此处有 353 字相似</p> <p>ck_gettime"是基于Linux C语言的时间函数，计时粒度能够精确到纳秒。使用时需要啊包含time.h头文件。其函数声明为“int clock_gettime(clockid_t clk_id , struct timespec *tp)”，其中clk_id表示检索和设置的clk_id指定的时钟时间。其有多个可选值，其中</p> <p>CLOCK_REALTIME表示计时结果为系统实时时间，随系统实时时间改变而改变，即从UTC1970-1-1 0:0:0开始计时，中间时刻如果系统时间被用户改成其他，则对应的时间相应改变，CLOCK_MONOTONIC表示从系统启动这一刻起开始计时，不受系统时间被用户改变的影响，CLOCK_PROCESS_CPUTIME_ID表示本进程到当前代码系统CPU花费的时间，而</p> <p>CLOCK_THREAD_CPUTIME_ID表示本线程到当前代码系统CPU花费的时间。计时结果记录在timespec结构中，</p> <p>其中有两个变量，其中tv_sec表示计时的秒数，而tv_nsec表示计时结果的纳秒数。</p> <p>表1 POSIX接口获取系统时</p> </div>	<div data-bbox="839 38 1516 100" data-label="Text"> <p>系统时间 - haima1998的专栏 - 博客频道 - CSDN.NET - 《网络 (http://blog.csdn.net) 》 - (是否引证：否)</p> </div> <div data-bbox="860 100 1516 694" data-label="Text"> <p>1.计算精度和纳秒。播放器中，一般是用CLOCK_MONOTONIC类型时间 函数介绍 1 2</p> <pre>e<time.h>; intclock_gettime(clockid_tclk_id,structtimespec*tp); 参数 clk_id : 检索和设置的clk_id指定的时钟时间。 CLOCK_REALTIME:系统实时时间,随系统实时时间改变而改变,即从UTC1970-1-1 0:0:0开始计时,中间时刻如果系统时间被用户改成其他,则对应的时间相应改变 CLOCK_MONOTONIC:从系统启动这一时刻起开始计时,不受系统时间被用户改变的影响 CLOCK_PROCESS_CPUTIME_ID:本进程到当前代码系统CPU花费的时间 CLOCK_THREAD_CPUTIME_ID:本线程到当前代码系统CPU花费的时间 struct timespec { time_t tv_sec; /* 秒 */ long tv_nsec; /* 纳秒*/ }; 编译命令 gcc te</pre> </div> <div data-bbox="839 734 1516 768" data-label="Text"> <p>clock - 《网络 (http://blog.csdn.net) 》 - (是否引证：否)</p> </div> <div data-bbox="860 768 1516 1328" data-label="Text"> <p>1.这是一个很有用的函数，尤其在基于时间的程序的时候。 函数介绍 e<time.h>; int</p> <pre>clock_gettime(clockid_t clk_id,struct timespec *tp); clk_id 检索和设置的clk_id指定的时钟时间 CLOCK_REALTIME:系统实时时间,随系统实时时间改变而改变,即从UTC1970-1-1 0:0:0开始计时,中间时刻如果系统时间被用户改成其他,则对应的时间相应改变 CLOCK_MONOTONIC:从系统启动这一时刻起开始计时,不受系统时间被用户改变的影响 CLOCK_PROCESS_CPUTIME_ID:本进程到当前代码系统CPU花费的时间 CLOCK_THREAD_CPUTIME_ID:本线程到当前代码系统CPU花费的时间 例子： e<stdio.h>; e<time.h>; int main(int argc,char *argv[]) { s</pre> </div> <div data-bbox="839 1368 1516 1424" data-label="Text"> <p>time,gettimeofday,clock_gettime. - 《网络 (http://blog.csdn.net) 》 - (是否引证：否)</p> </div> <div data-bbox="860 1424 1516 2134" data-label="Text"> <p>1.me.h>; 2、编译&链接。在编译链接时需加上 -lrt ;因为在librt中实现了clock_gettime函数 3、函数原型 int</p> <pre>clock_gettime(clockid_t clk_id,struct timespect *tp); 参数说明： clockid_t clk_id 用于指定计时时钟的类型，有以下4种： CLOCK_REALTIME:系统</pre> <p>2.t timespect *tp); 参数说明： clockid_t clk_id 用于指定计时时钟的类型，有以下4种： CLOCK_REALTIME:系统实时时间,随系统实时时间改变而改变,即从UTC1970-1-1 0:0:0开始计时,中间时刻如果系统时间被用户该成其他,则对应的时间相应改变 CLOCK_MONOTONIC:从系统启动这一刻起开始计时,不受系统时间被用户改变的影响 CLOCK_PROCESS_CPUTIME_ID:本进程到当前代码系统CPU花费的时间</p> <p>CLOCK_THREAD_CPUTIME_ID:本线程到当前代码系统CPU花费的时间 struct timespect *tp用来存储当前的时间，其结构如下： struct timespec { time_t tv_sec; /* seco</p> </div>
--	---

	<div>time, gettimeofday, clock_gettime, ftime (转载) - 乡禅 - 《网络 (http://blog.csdn.net) 》 - (是否引证 : 否)</div> <div>1.me.h&gt; 2、编译&链接。在编译链接时需加上 -lrt ;因为在librt中实现了clock_gettime函数 3、函数原型 int clock_gettime(clockid_t clk_id,struct timespec *tp); 参数说明 : clockid_t clk_id 用于指定计时时钟的类型 , 有以下4种 : CLOCK_REALTIME:系统</div> <div>2.t timespec *tp); 参数说明 : clockid_t clk_id 用于指定计时时钟的类型 , 有以下4种 : CLOCK_REALTIME:系统实时时间,随系统实时时间改变而改变,即从UTC1970-1-1 0:0:0开始计时,中间时刻如果系统时间被用户该成其他,则对应的时间相应改变 CLOCK_MONOTONIC:从系统启动这一刻起开始计时,不受系统时间被用户改变的影响 CLOCK_PROCESS_CPUTIME_ID:本进程到当前代码系统CPU花费的时间 CLOCK_THREAD_CPUTIME_ID:本线程到当前代码系统CPU花费的时间 struct timespec *tp用来存储当前的时间 , 其结构如下 : struct timespec { time_t tv_sec; /* seco</div> <div>时间转换相关函数 - zuokong的专栏 - 博客频道 - CSDN.NET - 《网络 (http://blog.csdn.net) 》 - (是否引证 : 否)</div> <div>1. &lt;time.h&gt; 编译&链接。在编译链接时需加上 -lrt ;因为在librt中实现了clock_gettime函数 函数原型 int clock_gettime(clockid_t clk_id,structtimespect *tp); 参数说明 : clockid_t clk_id 用于指定计时时钟的类型 , 有以下4种 : CLOCK_REALTIME:系统</div> <div>2.ctime_t timespec *tp); 参数说明 : clockid_t clk_id 用于指定计时时钟的类型 , 有以下4种 : CLOCK_REALTIME:系统实时时间,随系统实时时间改变而改变,即从UTC1970-1-10:0:0开始计时,中间时刻如果系统时间被用户该成其他,则对应的时间相应改变 CLOCK_MONOTONIC:从系统启动这一刻起开始计时,不受系统时间被用户改变的影响 CLOCK_PROCESS_CPUTIME_ID:本进程到当前代码系统CPU花费的时间 CLOCK_THREAD_CPUTIME_ID:本线程到当前代码系统CPU花费的时间 struct timespec *tp用来存储当前的时间 , 其结构如下 : struct timespec { time_t tv_sec; /* seco</div> <div>linux&nbsp;上时间相关的函数_xiaoqi - 《网络 (http://blog.sina.com) 》 - (是否引证 : 否)</div> <div>1.me.h&gt; 2、编译&链接。在编译链接时需加上 -lrt ;因为在librt中实现了clock_gettime函数 3、函数原型 int clock_gettime(clockid_t clk_id,struct timespec *tp); 参数说明 : clockid_t clk_id 用于指定计时时钟的类型 , 有以下4种 : CLOCK_REALTIME:系统</div> <div>2.t timespec *tp); 参数说明 : clockid_t clk_id 用于指定计时时钟的类型 , 有以下4种 : CLOCK_REALTIME:系</div>
--	--

		<p>统实时时间,随系统实时时间改变而改变,即从UTC1970-1-1 0:0:0开始计时,中间时刻如果系统时间被用户该成其他,则对应的时间相应改变 CLOCK_MONOTONIC:从系统启动这一刻起开始计时,不受系统时间被用户改变的影响 CLOCK_PROCESS_CPUTIME_ID:本进程到当前代码系统CPU花费的时间</p> <p>CLOCK_THREAD_CPUTIME_ID:本线程到当前代码系统CPU花费的时间 struct timespec *tp用来存储当前的时间,其结构如下: struct timespec { time_t tv_sec; long tv_nsec;</p>
		<p>【转】time, gettimeofday, clock_gettime, ftime, dawnstar1 - 《网络 (http://blog.sina.com) 》 - (是否引证: 否)</p> <p>1.1、头文件 2、编译&链接。在编译链接时需加上 -lrt ;因为在librt中实现了clock_gettime函数 3、函数原型 int clock_gettime(clockid_t clk_id, struct timespec *tp); 参数说明: clockid_t clk_id 用于指定计时时钟的类型,有以下4种: CLOCK_REALTIME:系统</p> <p>2. t timespec *tp); 参数说明: clockid_t clk_id 用于指定计时时钟的类型,有以下4种: CLOCK_REALTIME:系统实时时间,随系统实时时间改变而改变,即从UTC1970-1-1 0:0:0开始计时,中间时刻如果系统时间被用户该成其他,则对应的时间相应改变 CLOCK_MONOTONIC:从系统启动这一刻起开始计时,不受系统时间被用户改变的影响 CLOCK_PROCESS_CPUTIME_ID:本进程到当前代码系统CPU花费的时间</p> <p>CLOCK_THREAD_CPUTIME_ID:本线程到当前代码系统CPU花费的时间 struct timespec *tp用来存储当前的时间,其结构如下: struct timespec { time_t tv_sec; long tv_nsec;</p>
		<p>time, gettimeofday, clock_gettime, ftime, gmtime 一棵草 Telen - 《网络 (http://blog.sina.com) 》 - (是否引证: 否)</p> <p>1.1、头文件 2、编译&链接。在编译链接时需加上 -lrt ;因为在librt中实现了clock_gettime函数 3、函数原型 int clock_gettime(clockid_t clk_id, struct timespec *tp); 参数说明: clockid_t clk_id 用于指定计时时钟的类型,有以下4种: CLOCK_REALTIME:系统</p> <p>2. t timespec *tp); 参数说明: clockid_t clk_id 用于指定计时时钟的类型,有以下4种: CLOCK_REALTIME:系统实时时间,随系统实时时间改变而改变,即从UTC1970-1-1 0:0:0开始计时,中间时刻如果系统时间被用户该成其他,则对应的时间相应改变 CLOCK_MONOTONIC:从系统启动这一刻起开始计时,不受系统时间被用户改变的影响 CLOCK_PROCESS_CPUTIME_ID:本进程到当前代码系统CPU花费的时间</p> <p>CLOCK_THREAD_CPUTIME_ID:本线程到当前代码系统CPU花费的时间 struct timespec *tp用来存储当前的时间,其结构如下: struct timespec { time_t tv_sec; long tv_nsec;</p>
		<p>timerfd API使用总结 - 詆調 - CSDN博客 - 《网络</p>

	<p>(http://blog.csdn.net) 》 - (是否引证：否)</p> <p>1.11的这个事件入口，对产生的事件进行响应，从而构成了一个事件轮训程序。timerfd 相关函数 e &lt;time.h>; int clock_gettime(clockid_t clockid,struct timespec *tp); clock_gettime函数主要用于获取系统时间，精确到纳秒级别。在编译时需要添加-lrt库，clockid_t c</p> <p>2.何种模式获取时间，struct timespec *tp用于存储获取到的时间。其中clockid主要有如下常用的参数： CLOCK_REALTIME:系统实时时间,随系统实时时间改变而改变,即从UTC1970-1-1 0:0:0开始计时,中间时刻如果系统时间被用户改成其他,则对应的时间相应改变 CLOCK_MONOTONIC:从系统启动这一时刻起开始计时,不受系统时间被用户改变的影响 CLOCK_PROCESS_CPUTIME_ID:本进程到当前代码系统CPU花费的时间 CLOCK_THREAD_CPUTIME_ID:本线程到当前代码系统CPU花费的时间 e &lt;sys/timerfd.h>; int timerfd_create(int clockid,int flags); int timerfd_se</p>
	<p>201091303517288 杜江 杜江 - 《学术论文联合比对库》 - 2013-05-12 (是否引证：否)</p> <p>1.spect *tp);参数说明：clockid_t clk_id 用于指定计时时钟的类型，有以下4种：CLOCK_REALTIME：系统实时时间,随系统实时时间改变而改变,即从UTC1970-1-1 0:0:0开始计时,中间时刻如果系统时间被用户该成其他，则对应的时间相应改变；CLOCK_MONOTONIC：从系统启动这一时刻起开始计时,不受系统时间被用户改变的影响；CLOCK_PROCESS_CPUTIME_ID：本进程到当前代码系统CPU花费的时间 ；CLOCK_THREAD_CPUTIME_ID：本线程到当前代码系统CPU花费的时间。Struct timespec *tp用来存储当前的时间，其结构如下：Struct timespec{time_tv_sec；/*</p>
	<p>城市轨道交通车载二取二平台关键技术的研究 谢刚 - 《学术论文联合比对库》 - 2013-05-14 (是否引证：否)</p> <p>1.spect *tp);参数说明：clockid_t clk_id 用于指定计时时钟的类型，有以下4种：CLOCK_REALTIME：系统实时时间,随系统实时时间改变而改变,即从UTC1970-1-1 0:0:0开始计时,中间时刻如果系统时间被用户该成其他，则对应的时间相应改变；CLOCK_MONOTONIC：从系统启动这一时刻起开始计时,不受系统时间被用户改变的影响；CLOCK_PROCESS_CPUTIME_ID：本进程到当前代码系统CPU花费的时间 ；CLOCK_THREAD_CPUTIME_ID：本线程到当前代码系统CPU花费的时间。Struct timespec *tp用来存储当前的时间，其结构如下：Struct timespec{time_tv_sec；/*</p>
	<p>如何统计程序占用cpu时间？ - 博客频道 - CSDN.NET - 《网络 (http://blog.csdn.net) 》 - (是否引证：否)</p> <p>1.clockid_t which_clock,struct timespec *tp); which_clock参数解释 CLOCK_REALTIME:系统实时时</p>

	<p>间,随系统实时时间改变而改变,即从UTC1970-1-1 0:0:0开始计时,中间时刻如果系统时间被用户该成其他,则对应的时间相应改变</p> <p>CLOCK_MONOTONIC:从系统启动这一时刻起开始计时,不受系统时间被用户改变的影响</p> <p>CLOCK_PROCESS_CPUTIME_ID:本进程到当前代码系统CPU花费的时间</p> <p>CLOCK_THREAD_CPUTIME_ID:本线程到当前代码系统CPU花费的时间</p> <pre>//test9.c #include <stdio.h> #include <time.h> int main() { struct timespe</pre>
	<p>linux 定时器 - whm2300的专栏 - 博客频道 - CSDN.NET - 《网络 (http://blog.csdn.net) 》 - (是否引证 : 否)</p> <p>1.情都不会发生,直到定时器被启动,如下一节所示。</p> <p>clockid_t clk_id 用于指定计时时钟的类型,有以下4种:</p> <p>: CLOCK_REALTIME:系统实时时间,随系统实时时间改变而改变,即从UTC1970-1-1 0:0:0开始计时,中间时刻如果系统时间被用户该成其他,则对应的时间相应改变</p> <p>CLOCK_MONOTONIC:从系统启动这一时刻起开始计时,不受系统时间被用户改变的影响</p> <p>CLOCK_PROCESS_CPUTIME_ID:本进程到当前代码系统CPU花费的时间</p> <p>CLOCK_THREAD_CPUTIME_ID:本线程到当前代码系统CPU花费的时间</p> <p>下面的范例会创建一个新的定时器CLOCK_PROCESS_CPUTIME_ID的POSIX时钟,以将定时器的标识符存入t</p>
	<p>城市轨道交通车载二取二平台关键技术的研究 谢刚 - 《西南交通大学硕士论文》 - 2013-05-01 (是否引证 : 否)</p> <p>1.说明: clockidt lkid用于指定计时时钟的类型_c_,有以下4种:</p> <p>CLOCKREALTIME时时间, 随系统实时时间改变而改变,即从UTC1970-1-1_ : 系统实0:0:0开始计时,中间时刻如果系统时间被用户该成其他,贝!J对应的时间相应改变;</p> <p>CLOCKMONOTONIC_: 从系统启动这一时刻起开始计时,不受系统时间被用户改变的影响;</p> <p>CLOCKPROCESSCPUTIME__JD : 本进程到当前代码系统CPU花费的时间</p> <p>; CLOCKTHREADCPUTIMEID本线程到当前代码系统CPU花费的时间___; 。</p> <p>Struct timespect *tp用来存储当前的时间,其结构如下: Struct timespec</p>
	<p>移动IPv6开源代码从用户空间到内核的移植 林胜洁 - 《西安电子科技大学硕士论文》 - 2011-12-01 (是否引证 : 否)</p> <p>1.OCKREALTIME 时时间,随系统实时时间改变而改变,即从格林_: 系统实尼治时间1970年1月1日0时0分0秒开始计时,中间时刻如果系统时间被用户该成其他,则对应的时间相应改变。</p> <p>CLOCKMONOTONIC间被用户—:从系统启动这一时刻起开始计时,不受系统时改变的影响。</p> <p>CLOCKPROCESSCPUTIME__JD : 本进程到当前代码系统CPU花费的时第三章移动IPv6开源代码的具体移植 ^间。</p> <p>CLOCKTHREADCPUTIMEID本线程到当前</p>
	<p>空天中继网络链路仿真的设计与实现 陆金杏 - 《华中科技大学硕士论文》 - 2015-05-01 (是否引证 : 否)</p> <p>1.,其功能也是返回当前的系统时间,其函数原型及参数说明如下所示: #include <time.h>;int</p>

		<p>clock_gettime(clockid_t clk_id, struct timespec *tp);(1) clk_id 参数选择有以下几种情况 : (a) CLOCK_REALTIME : 系统实时时间 , 随着系统实时时间的改变而改变 ;</p> <p>2. struct timespec *tp);(1) clk_id 参数选择有以下几种情况 : (a) CLOCK_REALTIME : 系统实时时间 , 随着系统实时时间的改变而改变 ; (b) CLOCK_MONOTONIC : 由系统启动的时刻起开始计时 , 系统时间不受用户改变而影响。由以上</p> <p>linux/unix - 程序狗的成长之路 - 博客频道 - CSDN.NET - 《网络 (http://blog.csdn.net) 》 - (是否引证 : 否)</p> <p>1.clockid_t which_clock,struct timespec *tp); which_clock参数解释 CLOCK_REALTIME:系统实时时间,随系统实时时间改变而改变,即从UTC1970-1-1 0:0:0开始计时,中间时刻如果系统时间被用户该成其他,则对应的时间相应改变 CLOCK_M... 2016-03-18 09:31 阅读(201) 评论(0) exit()与_exit()的区别 从图中可以看出 , _ex</p> <p>2012年09月存档 - - 博客频道 - CSDN.NET - 《网络 (http://blog.csdn.net) 》 - (是否引证 : 否)</p> <p>1.clockid_t which_clock,struct timespec *tp); which_clock参数解释 CLOCK_REALTIME:系统实时时间,随系统实时时间改变而改变,即从U... 2012-09-27 17:49 阅读(138) 评论(0) linux一些系统调用的再实现 由于linux的系统函数存</p>
3	<p>此处有 230 字相似</p> <p>获取统计结果。</p> <p>表2 Perf获取系统时间</p> <pre> 1. void perf_init(int* fd) 2. { 3. static struct perf_event_attr attr; 4.attr.type = PERF_TYPE_HARDWARE; 5.attr.config = PERF_COUNT_HW_CPU_CYCLES; 6.attr.size = sizeof(attr); 7.attr.exclude_kernel = 1; 8.attr.exclude_hv = 1; 9.attr.exclude_callchain_kernel = 1; 11.fd = syscall(__NR_perf_event_open, &attr, 0, - 1, </pre>	<p>201610231642111642_于茜_基于ARM架构的LBlock轻量级分组密码Cache计时分析研究 于茜 - 《学术论文联合比</p> <p>对库》 - 2016-10-23 (是否引证 : 否)</p> <p>1.S性能计数器返回一个精确的时钟周期 , 构建一个结构体perf_event_attr attr来进行存储 , 具体定义如下 : static struct perf_event_attr attr;attr.type = PERF_TYPE_HARDWARE;attr.config = PERF_COUNT_HW_CPU_CYCLES;attr.size = sizeof(attr);attr.exclude_kernel = 1;attr.exclude_hv = 1;attr.exclude_callchain_kernel = 1;系统调用方法如下 : session->perf.fd = syscall(__NR_perf_event_op</p>
4	<p>此处有 62 字相似</p> <pre> r.exclude_hv = 1; 9.attr.exclude_callchain_kernel = 1; 11. fd = syscall(__NR_perf_event_open, &attr, 0, -1, - 1, 0); </pre>	<p>201610231642111642_于茜_基于ARM架构的LBlock轻量级分组密码Cache计时分析研究 于茜 - 《学术论文联合比</p> <p>对库》 - 2016-10-23 (是否引证 : 否)</p> <p>1.ttr.exclude_callchain_kernel = 1;系统调用方法如下 : session->perf.fd = syscall(__NR_perf_event_open, &attr, 0, -1, -1, 0);assert(session->perf.fd &gt;= 0);5.2.3计时信息分析模块计时信息分析模块的主要功能</p>

	13. return true; 14. } 15. int64 perf_timing(int* fd) 16.	
--	--	--

指 标

疑似剽窃文字表述

1. clock_gettime系统调用
- "clock_gettime"是基于Linux C语言的时间函数，
2. 其有多个可选值，其中CLOCK_REALTIME表示计时结果为系统实时时间，随系统实时时间改变而改变，即从UTC1970-1-1 0:0:0开始计时，中间时刻如果系统时间被用户改成其他，则对应的时间相应改变，CLOCK_MONOTONIC表示从系统启动这一刻起开始计时，不受系统时间被用户改变的影响，CLOCK_PROCESS_CPUTIME_ID表示本进程到当前代码系统CPU花费的时间，而CLOCK_THREAD_CPUTIME_ID表示本线程到当前代码系统CPU花费的时间。计时结果记录在timespec结构中，

5. 第四章攻击方案的设计

总字数：10853

相似文献列表 文字复制比：1.2%(135) 疑似剽窃观点：(0)

1	民航飞行员执照理论考试现状分析研究 刘银(导师：李卫东) - 《中国民用航空飞行学院硕士论文》 - 2011-04-22	0.6% (65) 是否引证：否
2	基于多网络的视频会议身份认证技术研究 周志坚; - 《电脑知识与技术》 - 2017-03-07 1	0.5% (55) 是否引证：否
3	面向3G/4G移动网络保密终端安全通信技术研究 周磊(导师：蒋睿) - 《东南大学硕士论文》 - 2016-05-20	0.5% (55) 是否引证：否
4	AES加解密算法详解 - yxtxiaotian的专栏 - CSDN博客 - 《网络 (http://blog.csdn.net) 》 - 2017	0.5% (55) 是否引证：否
5	密码算法详解——AES - 知其所以然 - CSDN博客 - 《网络 (http://blog.csdn.net) 》 - 2017	0.5% (55) 是否引证：否
6	密码算法详解——AES(高级加密算法) - Aeroleo的博客 - CSDN博客 - 《网络 (http://blog.csdn.net) 》 - 2017	0.5% (55) 是否引证：否
7	密码学笔记4 对称密钥算法 - 谢厂节的博客 - CSDN博客 - 《网络 (http://blog.csdn.net) 》 - 2017	0.5% (55) 是否引证：否
8	密码算法详解——AES - 幕色夜行 - CSDN博客 - 《网络 (http://blog.csdn.net) 》 - 2017	0.5% (53) 是否引证：否
9	现代学习理论指导下的高中数学作业改革的实验研究 谭武昌(导师：张一民) - 《云南师范大学硕士论文》 - 2006-06-06	0.3% (29) 是否引证：否
10	相对独立事件同时发生的概率PPT课件 - 《网络 (http://www.glzy8.com) 》 - 2012	0.3% (29) 是否引证：否
11	内江市2005—2006学年度高中二年级第二学期期末检测 数学(文科)-百度文库 - 《互联网文档资源 (http://wenku.baidu.c) 》 - 2012	0.3% (29) 是否引证：否
12	陕西咸阳2010高考第三次模拟考试试卷--数学(理)-百度文库 - 《互联网文档资源 (http://wenku.baidu.c) 》 - 2012	0.3% (29) 是否引证：否
13	陕西省咸阳市2011届高三第三次高考模拟题(文科数学)-百度文库 - 《互联网文档资源 (http://wenku.baidu.c) 》 - 2012	0.3% (29) 是否引证：否
14	2012年全国各省北京、上海、天津、江苏、浙江、安徽、湖北南、广东等高考理科数学试题及答案-新课标-百度文库 - 《互联网文档资源 (http://wenku.baidu.c) 》 - 2012	0.3% (29) 是否引证：否

原文内容		相似内容来源
1	此处有 30 字相似	民航飞行员执照理论考试现状分析研究 刘银 - 《中国民用航空飞行学院硕士论文》 - 2011-04-22 (是否引证：否)
	法第二轮访问的情况，依此进行攻击，获取部分或全部的密钥值。 4.2 KS检验 4.2.1 KS检验概述	1.系数的显著性。如此重复直到方程和每个回归系数都通过了显著性检验为止。2.2.2 假设检验 假设检验是用来判断样本与样本，样本与总体的差异是由抽样误差引

	<p>KS检验是假设检验中的一种，它可以被用来判断随机抽样中样本与样本或总体</p> <p>与总体之间的关系是由于误差引起的还是由本质差别引起的。假设检验的原理是首先对总体做出某种假设，然后通过抽样的方法来判断是</p>	<p>起还是本质差异造成的统计推断方法。假设检验根据不同的使用条件可以分为参数检验和非参数检</p>
2	<p>此处有 50 字相似</p> <p>若产生不合理现象则拒绝原假设，反之没有足够的理由拒绝原假设，及接受原假设。与反证法不同的是，假设检验中不合理现象指的是小</p> <p>概率事件发生了。小概率事件指的是在一次实验中几乎不可能发生的时间，如果在单次实验中小概率事件发生了，</p> <p>则表示产生了矛盾，通常将发生概率小于0.05的时间称为小概率事件，视情况也能修改该值。在KS检验中将小概率事件发生的概率</p>	<p>民航飞行员执照理论考试现状分析研究 刘银 - 《中国民用航空飞行学院硕士论文》 - 2011-04-22 (是否引证：否)</p> <p>1.实际运用中比参数检验有更广阔的运用空间。 2.2.2.1 假设检验的原理 假设检验的基本思想是应用小概率原理，即小概率事件在一次实验中是几乎不可能发生的。如果小概率事件在一次实验中发生了，就有充分的理由怀疑原假设为真，即拒绝原假设。究竟多大的概率为小小概率事件？在一个问</p> <p>相对独立事件同时发生的概率PPT课件 - 《网络 (http://www.glzy8.com) 》 - (是否引证：否)</p> <p>1.命中的概率；(3)恰好有2次命中的概率；(4)恰好有3次命中的概率；(5)恰好有4次命中的概率；=1 如果在1次实验中某事件发生的概率为P，那么在n次独立重复实验中这个事件恰好发生 k次的概率为 注：可以看作是 展开式中的第k+1项 例5、某气象站天气预报的准确率为80%，计算 (结果保留两位有效数字</p> <p>现代学习理论指导下的高中数学作业改革的实验研究 谭武昌 - 《云南师范大学硕士论文》 - 2006-06-06 (是否引证：否)</p> <p>1. 如果事件A、B相互独立，那么 $P(A \cap B) = P(A) \cdot P(B)$;如果事件A在一次实验中发生的概率是P，那么n次独立重复实验中恰好发生k次的概率 $P(k) = C_n^k P^k (1-P)^{n-k}$ 一选择题:本</p> <p>陕西咸阳2010高考第三次模拟考试试卷--数学(理)-百度文库 - 《互联网文档资源 (http://wenku.baidu.c) 》 - 2012-11-11 18:59:04 (是否引证：否)</p> <p>1.P(B). 如果事件 A、B 相互独立，那么 $P(A \cap B) = P(A) \cdot P(B)$. 如果事件 A 在一次实验中发生的概率是 P，那么 n 次独立重复试验中事件 A 恰好发生 k 次的概率 $1 [(x_1 \cdot x)^2 + (x_2 \cdot x)^2 + \dots + (x_n \cdot x)$</p> <p>陕西省咸阳市2011届高三第三次高考模拟题(文科数学)-百度文库 - 《互联网文档资源 (http://wenku.baidu.c) 》 - 2012-11-11 19:08:40 (是否引证：否)</p> <p>1. 相互独立，那么 $P(A \cap B) = P(A) \cdot P(B)$ 其中 R 表示球的半径 $P(A \cap B) = P(A) \cdot P(B)$ 如果事件 A 在一次实验中发生的概率是 P，那么 n 次独立重复试验中事件 A 恰好发生 k 次的概率 $p_n(k) = C_n^k p^k (1-p)^{n-k}$ ($k=0, 1, 2, \dots, n$</p> <p>内江市2005—2006学年度高中二年级第二学期期末检测 数学(文科)-百度文库 - 《互联网文档资源 (http://wenku.baidu.c) 》 - 2012-12-24 15:16:36 (是否引证：否)</p> <p>1.? B ? ; 如果事件 A ,B 相互独立，那么 $P(A \cap B) = P(A) \cdot P(B)$ 如果事件 A 在一次实验中发生的概率为 P，那么 n 次独立重复实验中恰好发生 k 次的概率</p>

		<p>为：$P_n \cdot k \cdot C_n P \cdot 1 \cdot P \cdot k \cdot n \cdot k$；2 球的表面积</p> <p>2012年全国各省北京、上海、天津、江苏、浙江、安徽、湖北南、广东等高考理科数学试题及答案-新课标-百度文库 - 《互联网文档资源 (http://wenku.baidu.c)》 - 2012-9-9 9:50:10 (是否引证：否)</p> <p>1.表示柱体的底面积，h 表示柱体的高 锥体的体积公式 $P(A \cdot B) \cdot P(A) \cdot P(B)$ 如果事件 A 在一次实验中发生的概率是 p，那么 n 次独立重复试验中事件 A 恰好发生 k 次的概率 $V \cdot 1 \cdot 3 \cdot S \cdot h$ 其中 S 表示锥体的底面积，h 表示柱体的高 球的表面积公式 P_n</p>
3	<p>此处有 55 字相似</p> <p>击程序访存情况的误判。</p> <p>此外，由于AES加密过程中的查表操作涉及到多种操作：字节替代、行移位、列混淆以及轮密钥加操作。</p> <p>加解密中每轮的密钥分别由初始密钥扩展得到。且加密算法中16字节的明文、密文和轮密钥都通过一个4*4的矩阵表示。</p> <p>在对AES的T-table实现进行密钥攻击时，需要熟悉攻击过程以及每个过程中对那个table进行查找，并挖掘出可利用的内</p>	<p>密码算法详解——AES - 幕色夜行 - CSDN博客 - 《网络 (http://blog.csdn.net)》 - (是否引证：否)</p> <p>1.AddRoundKey)。解密过程分别为对应的逆操作。由于每一步操作都是可逆的，按照相反的顺序进行解密即可恢复明文。加解密中每轮的密钥分别由初始密钥扩展得到。算法中16字节的明文、密文和轮密钥都以一个4x4的矩阵表示。接下来分别对上述5种操作进行介绍。1.1 字节代替 字节代替的主要功能是通过S盒完成一个字节到另外一个字节的映射。S</p> <p>AES加解密算法详解 - yxtxiaotian的专栏 - CSDN博客 - 《网络 (http://blog.csdn.net)》 - (是否引证：否)</p> <p>1. (AddRoundKey)。解密过程分别为对应的逆操作。由于每一步操作都是可逆的，按照相反的顺序进行解密即可恢复明文。加解密中每轮的密钥分别由初始密钥扩展得到。算法中16字节的明文、密文和轮密钥都以一个4x4的矩阵表示。AES算法中，密钥扩展也是理解AES算法的关键，接下来分别对上述5种操作进行介绍。1.0 预备知识：有限域GF(2ⁿ)</p> <p>密码算法详解——AES - 知其所以然 - CSDN博客 - 《网络 (http://blog.csdn.net)》 - (是否引证：否)</p> <p>1.密过程的每一步分别对应操作的逆操作，2) 加解密所有操作的顺序正好是相反的。正是由于这两点保证了解密能够正确地恢复明文。加解密中每轮的密钥分别由初始密钥扩展得到。算法中16字节的明文、密文和轮密钥都以一个4x4的矩阵表示。接下来分别对上述5种操作进行介绍。1.1 字节代替 字节代替的主要功能是通过S盒完成一个字节到另外一个字节的映射。S</p> <p>密码算法详解——AES(高级加密算法) - Aeroleo的博客 - CSDN博客 - 《网络 (http://blog.csdn.net)》 - (是否引证：否)</p> <p>1. (AddRoundKey)。解密过程分别为对应的逆操作。由于每一步操作都是可逆的，按照相反的顺序进行解密即可恢复明文。加解密中每轮的密钥分别由初始密钥扩展得到。算法中16字节的明文、密文和轮密钥都以一个4x4的矩阵表示。接下来分别对上述5种操作进行介绍。1.1 字节代替 字节代替的主要功能是通过S盒完成一个字节到另外一个字节的映射。S</p> <p>密码学笔记4 对称密钥算法 - 谢厂节的博客 - CSDN博客 - 《网络 (http://blog.csdn.net)》 - (是否引证：否)</p> <p>1. (AddRoundKey)。解密过程分别为对应的逆操作。</p>

	<p>由于每一步操作都是可逆的，按照相反的顺序进行解密即可恢复明文。加解密中每轮的密钥分别由初始密钥扩展得到。算法中16字节的明文、密文和轮密钥都以一个4x4的矩阵表示。 ECB模式（电子密码本模式：Electronic codebook）ECB是最简单的块密码加密模式，加密前根据加密块</p> <p>面向3G/4G移动网络保密终端安全通信技术研究 周磊 - 《东南大学硕士论文》 - 2016-05-20（是否引证：否）</p> <p>1.) 口61。解密过程分别为对应的逆操作。由于每一步操作都是可逆的，按照相反的顺序进行解密即可恢复明文。加解密中每轮的密钥分别由初始密钥扩展得到。算法中16字节的明文、密文和轮密钥都W一个4x4的矩阵表示。2.4.3分组加巧的加巧模式ECB模式：如图2-3所示是对分组密码的直接使用，给定一个64比特的明文分组</p> <p>基于多网络的视频会议身份认证技术研究 周志坚; - 《电脑知识与技术》 - 2017-03-07 1（是否引证：否）</p> <p>1.dd Round Key)。解密过程分别为对应的逆操作。由于每一步操作都是可逆的,按照相反的顺序进行解密即可恢复明文。加解密中每轮的密钥分别由初始密钥扩展得到。算法中16字节的明文、密文和轮密钥都以一个4x4的矩阵表示。图5表示AES算法的加密和解密过程。图5 AES算法的加密和解密流程图2设计实现2.1系统结构本文研究与实现的基于多网络</p>
--	--

指 标
疑似剽窃文字表述
1. 概率事件发生了。小概率事件指的是在一次实验中几乎不可能发生的时间，如果在单次实验中小概率事件发生了，
2. 加解密中每轮的密钥分别由初始密钥扩展得到。且加密算法中16字节的明文、密文和轮密钥都通过一个4*4的矩阵表示。

6. 第五章AES攻击验证	总字数：7564
相似文献列表 文字复制比：0%(0) 疑似剽窃观点：(0)	

7. 第六章预防攻击措施	总字数：10116
相似文献列表 文字复制比：0%(0) 疑似剽窃观点：(0)	

- 说明：1.总文字复制比：被检测论文总重合字数在总字数中所占的比例
- 2.去除引用文献复制比：去除系统识别为引用的文献后，计算出来的重合字数在总字数中所占的比例
- 3.去除本人已发表文献复制比：去除作者本人已发表文献后，计算出来的重合字数在总字数中所占的比例
- 4.单篇最大文字复制比：被检测文献与所有相似文献比对后，重合字数占总字数的比例最大的那一篇文献的文字复制比
- 5.指标是由系统根据《学术论文不端行为的界定标准》自动生成的
- 6.红色文字表示文字复制部分;绿色文字表示引用部分
- 7.本报告单仅对您所选择比对资源范围内检测结果负责



✉ amlc@cnki.net

🌐 <http://check.cnki.net/>

👤 <http://e.weibo.com/u/3194559873>