Camellia 访问驱动 Cache 计时攻击研究

赵新杰 王 韬 郑媛媛
(军械工程学院计算机工程系 石家庄 050003)

摘 要 Camellia 是 NESSIE 计划中 128 位分组密码的最终获胜者. 现有的针对 Camellia 的 Cache 计时攻击大多基于时序驱动模型,需百万计的样本在几十分钟内完成. 文中研究表明,由于频繁的查找表操作,Camellia 对访问驱动 Cache 计时攻击也是脆弱的,攻击所需样本量比时序驱动要小. 首先,基于访问驱动方式,给出了一种通用的针对对称密码 S 盒的分析模型,指出 Camellia 加密过程中的轮函数易泄露初始密钥和轮密钥的异或结果值,密钥扩展中的左移函数使得 Camellia 安全性大大降低. 然后,给出了多例针对 Camellia-128/192/256 的访问驱动 Cache 计时攻击,实验结果表明:500 和 900 个随机明文样本可恢复 Camellia-128、Camellia-192/256 密钥,文中的攻击可被扩展到针对已知密文条件下的解密过程或远程环境中进行实施, 3000 个随机明文可在局域网和校园网环境下恢复 Camellia-128/192/256 密钥. 最后,分析了 Camellia 易遭受 Cache 计时攻击的原因,并为密码设计者提出了防御该攻击的一些有效措施.

关键词 Camellia-128/192/256;分组密码;访问驱动;Cache 计时攻击;旁路攻击;远程攻击;F 函数;查找 S 盒;左移函数;密钥扩展;已知密文
中图法分类号 TP393 **DOI 号**: 10.3724/SP.J.1016.2010.01153

Research on Access Driven Cache Timing Attacks Against Camellia

ZHAO Xin-Jie WANG Tao ZHENG Yuan-Yuan
(*Department of Computer Engineering, Ordnance Engineering College, Shijiazhuang* 050003)

Abstract Camellia is the final winner of 128-bit block cipher in NESSIE. Most of the previous Cache timing attacks on Camellia are all based on timing driven model, our research shows that, due to its frequent S-box lookup operations, Camellia is also vulnerable to access driven Cache timing attacks. Firstly, this paper provides a general analysis model for symmetric ciphers using S-box based on access driven Cache timing attack model, points out that the F function of the Camellia can leak the result of encryption key XORed with expand-key, and the left circular rotating operation of the key schedule in Camellia has serious designing problem. Next, this paper presents several Cache timing attacks on Camellia-128/192/256. Experiment results demonstrate: 500 random plaintexts are enough to recover Camellia-128 key; 900 random plaintexts are enough to recover Camellia-192/256 key; also, the attacks can be expanded to known ciphertext conditions by attacking the Camellia decryption procedure; besides, the attacks are quite easy to be expanded to remote scenarios, 3000 random plaintexts are enough to recover Camellia-128/192/256 key in both local and campus networks. Finally, this paper discusses the reason why Camellia is weak in this type of attack, and provides some advices to cipher designers for hardening ciphers against Cache timing attacks.

收稿日期:2009-08-12;最终修改稿收到日期:2010-04-22. 本课题得到国家自然科学基金(60772082)、河北省自然科学基金数学研究专项(08M010)资助.**赵新杰**,男,1986 年生,博士研究生,主要研究方向为分组密码旁路分析和故障分析. E-mail: zhaoxinjieem@163.com. **王 韬**,男,1964 年生,博士,教授,博士生导师,主要研究领域为信息安全和密码旁路分析. **郑媛媛**,女,1981 年生,博士研究生,主要研究方向为分组密码安全和代数旁路分析.

Keywords Camellia-128/192/256; block cipher; access driven; Cache timing attack; side channel attack; remote attack; F function; S-box lookup; left circular rotating operation; key schedule; known ciphertext

1 引 言

1.1 相关工作

近来,随着旁路攻击的引入,各类密码算法均面临严峻威胁.传统来说,密码算法的安全性取决于数学函数 $E_K[P]\rightarrow C$,攻击者试图根据明文或密文信息,利用线性和差分数学分析方法去推测密钥 K.随着密钥长度和设计复杂度的增加,密码算法的安全性得到了很大提升,传统的数学分析已远远不能在有效时间和空间复杂度内完成攻击.然而,最近的研究表明在密码算法的执行过程中,可能会泄露出执行时间、能量消耗、电磁泄露、故障输出等物理效应信息,通常称之为旁路信息.实际上,确切的密码算法实现函数应该为 $E_K[P]\rightarrow(C,L)$,旁路信息 L 和加解密密钥 K 紧密相关,通过一定的分析方法,结合明文或密文,攻击者可快速推测出密钥 K.本文所描述攻击用到的是通过数据 Cache 部件单元泄露出来的时间旁路信息.

Cache 计时攻击是一种新的旁路攻击方法,其可行性首先由 Kocher[1] 和 Kelsey 等[2] 在 1996 年提出,其假定攻击者可以通过一个合法的用户来测量时间信息进而预测 Cache 访问信息,而这些条件在现实中往往是可行的.Page[3] 等在 2002 年仿真实现了针对 DES 的 Cache 计时攻击;随后,Tsunoo 等[4-5] 等提出了首例实际的针对 DES 和 Camellia 的 Cache 攻击;2004 年 Yoshitaka[6] 等对 Tsunoo 攻击进行了改进;2005 年 Bernstein[7] 与 Osvik,Shamir 和 Tromer[8-9] 两个小组在各自的研究中发现,AES 也易遭受 Cache 计时攻击,这极大地引发了密码界对 Cache 计时攻击研究的热情,后续的研究有对前人攻击验证[10-11]、攻击改进[12-15]、攻击防御[16-18].

目前的 Cache 计时攻击主要针对 DES 和 AES,防御方法也主要定位在密码算法实现过程中,只有 Tsunoo[5] 和 Yoshitaka[6] 对 Camellia-128 进行了 Cache 计时攻击研究.在 Tsunoo 攻击中,从 2^{28} 个随机明文中挑选出 2^{18} 个加密时间较短的样本,经 35 分钟左右分析后恢复 Camellia-128 密钥.Yoshitaka 在 Tsunoo 攻击基础上,使用 $2^{21.4}$ 个明文 22 分钟左右成功恢复 Camellia-128 密钥,上述两例攻击都是通过测量整个加密时间并分析密码某次查 S 盒运算中 Cache 访问碰撞带来的时间关系推断密钥,攻击属于时序驱动方式,一般需百万计的样本恢复密钥,而且在远程攻击条件下,即使是传输时延的抖动时间都要远大于整个加密时间,所以他们的攻击在真正远程环境中很难实现.本文提出了多例针对 Camellia-128/192/256 的访问驱动 Cache 计时攻击,使用了一个间谍程序来采集 Camellia 加密中访问的 Cache 组集合信息,经一定的分析来预测密钥,实验结果表明,无论是已知明文、已知密文、本地和远程条件下,针对 Camellia-128/192/256 的访问驱动 Cache 计时攻击都可在 3000 样本量以内经 1 秒钟的分析完成,攻击对象为 2009 年 7 月最新发布的 OPENSSL-1.0.0-beta3 中的 Camellia 算法实现.

1.2 本文的贡献

本文研究的主要贡献如下:

(1)给出了一种通用的针对使用 S 盒的对称密码的访问驱动 Cache 计时分析模型.

该模型可被应用到 AES、SMS4、Camellia、ARIA、HC-128、HC-256 等使用 S 盒的对称密码攻击中,并且可用于对这些算法抗 Cache 计时攻击能力进行评估测试.

(2)指出 Camellia 的 F 轮函数可能会泄露密钥相关信息.

Camellia 的 F 函数中进行了多次查表操作,查表索引值和初始密钥、扩展密钥紧密相关,经访问驱动 Cache 计时分析可获取这些相关密钥信息,具体来说,加密中的 F 函数可泄露初始密钥和扩展密钥的异或结果值,而密钥扩展中的 F 轮函数则可直接泄露初始密钥或扩展密钥值.

(3)Camellia 密钥扩展中使用的左移函数使得 Camellia 安全性大大降低.

攻击者如果找到对同一密钥相关变量左移前和左移后的一个异或碰撞,就很容易恢复该变量值,然后结合算法设计快速破解密钥.本文实验结果表明仅仅通过分析 Camellia-128 密钥扩展中一个初始密钥相关的左移 15 位函数就可将其搜索空间由 2^{128} 降低到 2.Camellia-192/256 的密钥扩展设计也存在缺陷,其第 3、4 轮和第 5、6 轮扩展密钥分别根据对 K_A 和 K_R 左移 15 位生成,通过对 Camellia 加密

前 6 轮进行 Cache 计时攻击，可直接获取密钥扩展第 5 个 F 函数的输入值 $K_A \oplus K_R$，然后得到密钥扩展第 6 个 F 函数的输出值 K_B，经进一步分析，可直接获取 Camellia-192/256 初始密钥.

(4) 提出并实现了多例针对 Camellia-128/192/256 的访问驱动 Cache 计时攻击.

本文设计实现了多例针对 Camellia-128/192/256 的访问驱动 Cache 计时攻击，实验结果表明，即使是使用了混淆层函数 FL/FL^{-1} 的 Camellia 算法，实现也易遭受访问驱动 Cache 计时攻击，攻击可被扩展至已知密文甚至是远程环境条件下，3000 个样本左右可在局域网或校园网环境下快速恢复 Camelia-128/192/256 密钥. 表 1 给出了本文攻击结果同前人攻击比较.

表 1　针对 Camellia 的 Cache 计时攻击结果比较

攻击类型	Camellia 算法	攻击轮数	是否使用 FL/FL^{-1}	样本量	时间
文献[5]给出的攻击	Camellia-128	6	×	2^{28}	35min
文献[6]给出的攻击	Camellia-128	6	×	$2^{21.4}$	22min
第 4 节给出的攻击	Camellia-128	4	×/√	$2^{8.97}$	1s
第 6 节给出的攻击	Camellia-128	4	×/√	$2^{8.97}$	1s
第 7 节给出的攻击	Camellia-128	4	×/√	$2^{11.55}$	1s
第 5 节给出的攻击	Camellia-192/256	6	×/√	$2^{9.81}$	1s
第 6 节给出的攻击	Camellia-192/256	6	×/√	$2^{9.81}$	1s
第 7 节给出的攻击	Camellia-192/256	6	×/√	$2^{11.55}$	1s

(5) 探讨了针对 Camellia 的访问驱动 Cache 计时攻击原因并给出了可能的防御措施.

针对 Camellia 的访问驱动 Cache 计时攻击根本原因在于 Camellia 算法自身，其 F 函数和密钥设计存在 Cache 计时信息泄露安全缺陷，并可用于快速恢复密钥. 因此，本文在第 8 节探讨了密码设计者如何提高 Camellia 抗 Cache 计时攻击的能力.

1.3　结构组织

本文第 2 节给出了针对 Camellia 的 Cache 计时攻击相关背景知识；第 3 节提出了基本的攻击模型，并阐述了如何将该模型应用到 Camellia 的 F 函数分析中；第 4 小节和第 5 小节分别给出了针对 Camellia-128 和 Camellia-192/256 的已知明文访问驱动 Cache 计时攻击；第 6 和第 7 小节分别给出了已知密文和远程条件下的 Camellia 攻击；第 8 小节对针对 Camellia 的访问驱动 Cache 计时攻击进行了讨论分析，并向密码设计者提出了防御该类攻击的一些可能防护措施；第 9 节为结束语.

2　相关知识

2.1　符号标记

$\oplus$：异或操作.

$\|$：两个变量的连接.

$\lll n$：左移 n 位.

S_1, S_2, S_3, S_4：OPENSSL-1.0.0-beta3 中 Camellia 实现中的 4 个 S 盒，分别对应其中的 SBOX1_1110，SBOX2_0222，SBOX3_3033，SBOX4_4404.

X_L, X_R：128 位变量 X 的左半部分和右半部分数据.

K, KE：Camellia 加解密初始密钥和扩展密钥.

s_i^r：第 r 轮 F 函数的第 i 个 32 位输入.

δ：一个 Cache 块中的 Cache 元素数量.

2.2　Cache 计时攻击

Cache 计时攻击主要利用从 Cache 中加载数据到 CPU 寄存器要比从 RAM 中要快的特性进行密码分析. 通过测量密码算法实现过程中由于 Cache 访问带来的时间差异信息，攻击者可以推测密码算法实现的内部状态信息. 下面，我们将阐述 Cache 成为计时旁路攻击隐通道的原理.

Cache 工作机制

现代微处理器大都使用高速缓存 Cache 来解决 CPU 与主存之间速度不匹配问题. 假设整个 Cache 包括 S 个 Cache 组，每个组有 W 个 Cache 行，每个行有 δ 个元素（B 字节），则整个 Cache 大小为 $S \times W \times B$ 字节.

主存地址和 Cache 之间的映射关系如下：

特征 1.　CPU 读取主存中一个字 A 时，首先将 A 地址放入主存地址寄存器，Cache 控制逻辑依据地址判断 A 当前是否在 Cache 中，如果是则地址变换成功，发生“Cache 命中”；否则发生“Cache 失效”，把包括 A 在内的一整块数据都从主存中读出来，装载到 Cache 中去.

特征 2.　在组相连 Cache 中，每一个主存块只能被映射到固定的 Cache 组中，具体来说，Cache 地址为 a 的内存块只能被映射到 Cache 组 $[a/B] \bmod S$ 中.

由特征 1 可知同样一条访问存储器的指令在目标数据不在 Cache 内需访问主存时就可能产生延迟,而这种延迟的表现就是程序较长的运行时间或较大的能量消耗,就典型处理器来说,“Cache 命中”时直接访问 Cache 需要 2～4 个时钟周期,而“Cache 失效”时访问主存则需要 12～100 时钟周期,因此 Cache 为密码实现提供了时间信息泄露源.

由特征 2 可知,不同进程在对自身数据进行内存访问时,这些数据可以被映射到同一 Cache 组中进而共享 Cache 存储空间,恶意进程可以通过对自身数据 Cache 访问时间或者能量消耗差异来监测其它进程的 Cache 组访问特征,故 Cache 也为密码实现提供了时间信息泄露隐通道.

据上可知,攻击者可能获取到密码算法实现过程中访问的 Cache 组集合信息,在本地攻击中,这些采样信息噪声比较小,而在远程攻击中,由于网络发包拆包本身要对 Cache 进行大量的访问,带来的系统噪声比较大,但通过多次采集或增加样本数量,仍可获取有效的 Cache 采样信息.需要注意的是,虽然 Cache 内容受存储器保护,攻击者无法直接获取,但其元数据会泄露 Cache 访问地址信息,而这些地址信息和分组密码查找 S 盒索引有密切关系,可被攻击者用来进行密码分析

攻击分类及可行性分析

根据计时攻击部件不同,可将 Cache 计时攻击分为针对数据 Cache、指令 Cache 两种.由于现代分组密码大多使用 S 盒查找表访问数据 Cache,所以目前利用数据 Cache 进行攻击的对象主要为分组密码;同样,现代公钥加密系统大都使用了大量的指令访问操作,其加解密过程中由于密钥位值不同所要进行的指令访问操作数目有很大的区别,此时会导致对指令 Cache 访问次数及时间存在很大区别,所以目前利用指令 Cache 的攻击对象主要是公钥加密系统.

根据所采集的时间信息不同,又可将 Cache 计时攻击分为时序驱动、访问驱动、踪迹驱动 3 种方式.时序驱动攻击采集的是密码进程整个加解密时间,采集方法简单,平台适用性强,但所需样本量大,一般都要百万计,离线分析方法比较复杂.更重要的是,在远程环境中,网络传输时延甚至是抖动时延都要远大于加密时间,采集到精确的加密时间显得极为困难,远程攻击适用性不强.访问驱动攻击主要利用间谍进程采集密码进程加解密中访问的 Cache 组集合信息,采集方法比时序驱动稍显复杂,但分析方法比较简单,在木马植入技术日趋成熟的今天,攻击本地和远程实现可行性比较强.踪迹驱动攻击比访问驱动攻击信息采集精度更高,攻击方需精确采集密码进程一次加解密过程中每次查表 Cache 访问的命中和失效信息,通过计时手段很难实现,一般都通过功耗检测手段进行,需物理接触密码设备,因此攻击不论在本地还是远程通过计时实现可行性都不强.

基于访问驱动 Cache 计时攻击的可行性:我们选择访问驱动作为攻击模型,Camellia 分组密码作为攻击对象,显然,攻击属于数据 Cache 计时攻击.

2.3 Camellia 算法

Camellia[19] 是由 NTT 和 Mitsubishi 电子公司在 2000 年联合提交的一个分组密码,在 2003 年被选择成为欧洲 NESSIE 计划的获胜者,2009 年3 月,Camellia 被集成到 OPENSSL-1.0.0-beta1[20] 密码库中,对 Camellia 算法的完整介绍可参考文献[19,21],下面仅对本文中用到的 Camellia 特性进行描述.

加密过程. Camellia 是一个迭代型分组密码,分组长度为 128 比特,支持 128、192 和 256 比特 3 种规模的密钥长度,采用 Feistel 整体结构.为提高安全性,在第 1 轮前和最后一轮后分别做前期和后期白化,即子密钥加,同时每隔 6 轮增加一个不规则轮,即 FL/FL^{-1} 混淆函数层.

设 L_r 和 R_r 为第 r 轮的输入,则轮变换可表示为

$$L_r=R_{r-1}\oplus F(L_{r-1},k_r),$$

$$R_r=L_{r-1},$$

其中 k_r 为第 r 轮的子密钥,$F=P\cdot S$ 为轮函数,S 和 P 定义如下:

S: $F_2^{64}\rightarrow F_2^{64}$

$$l_{1(8)}\|l_{2(8)}\|l_{3(8)}\|l_{4(8)}\|l_{5(8)}\|l_{6(8)}\|l_{7(8)}\|l_{8(8)}\rightarrow l'_{1(8)}\|l'_{2(8)}\|l'_{3(8)}\|l'_{4(8)}\|l'_{5(8)}\|l'_{6(8)}\|l'_{7(8)}\|l'_{8(8)}$$

$$l'_1=s_1(l_1),\ l'_2=s_2(l_2),\ l'_3=s_3(l_3),\ l'_4=s_4(l_4),$$

$$l'_5=s_2(l_5),\ l'_6=s_3(l_6),\ l'_7=s_4(l_7),\ l'_8=s_1(l_8).$$

P: $F_2^{64}\rightarrow F_2^{64}$

$$z_{1(8)}\|z_{2(8)}\|z_{3(8)}\|z_{4(8)}\|z_{5(8)}\|z_{6(8)}\|z_{7(8)}\|z_{8(8)}\rightarrow z'_{1(8)}\|z'_{2(8)}\|z'_{3(8)}\|z'_{4(8)}\|z'_{5(8)}\|z'_{6(8)}\|z'_{7(8)}\|z'_{8(8)}$$

$$z'_1=z_1\oplus z_3\oplus z_4\oplus z_6\oplus z_7\oplus z_8,$$

$$z'_2=z_1\oplus z_2\oplus z_4\oplus z_5\oplus z_7\oplus z_8,$$

$$z'_3=z_1\oplus z_2\oplus z_3\oplus z_5\oplus z_6\oplus z_8,$$

$$z'_4=z_2\oplus z_3\oplus z_4\oplus z_5\oplus z_6\oplus z_7,$$

$$z'_5=z_1\oplus z_2\oplus z_6\oplus z_7\oplus z_8,$$

$z_6'=z_2\oplus z_3\oplus z_5\oplus z_7\oplus z_8$，

$z_7'=z_3\oplus z_4\oplus z_5\oplus z_6\oplus z_8$，

$z_8'=z_1\oplus z_4\oplus z_5\oplus z_6\oplus z_7$.

密钥扩展算法. 首先由初始种子密钥和 64 位常量 $\Sigma_i(i=1,2,\cdots,6)$经过几个轮函数 F 生成 4 个 128 位变量 K_L、K_R、K_A 和 K_B，白化层子密钥 $kw_{i(64)}$、普通轮密钥 $k_{i(64)}$、FL/FL^{-1} 函数层子密钥 $kl^{i(64)}$ 均由 K_L、K_R、K_A和 K_B循环移位而成，详见文献[19].

3　攻击模型

3.1　通用攻击模型

现在软件实现的分组密码大都使用许多 S 盒查表访问 Cache 来提高效率，在 S 盒查表过程中，常满足

$$\alpha\odot\beta=\gamma \tag{1}$$

其中，α 表示明文、密文或中间状态的一部分，常为 1 个字节；β 表示同密钥和扩展密钥相关的密码参量；γ 表示查表结果或索引值；$\odot$ 表示加密查表 α 和 β 之间的逻辑操作，常为异或操作.

式(1)可转换为

$$\alpha\odot\gamma=\beta \tag{2}$$

由 2.2 节可知，攻击者通过访问驱动方式采集 Cache 计时信息，可获取到密码算法实现过程中访问过和未访问过的 Cache 组集合，并转换为查表索引或结果 γ，由于 α 通常已知，根据式(2)，不难预测 β 候选值，进而计算出密钥 K，分析过程通常采用下列策略.

策略 1. 分析查表中未访问过的 Cache 组集合.

通过分析 Cache 组集合和 S 盒索引值的映射关系，未访问过的 Cache 组集合可转化为不可能的查表索引或结果 γ，结合 α 和式(2)，可得到一组 β 的不可能候选值，多个样本排除分析后可获取正确的 β 值，然后推断出密钥 K.

策略 2. 分析查表中可能访问的 Cache 组集合.

首先为每个 β 候选值设置一个计数器，然后固定 α 值，产生一些随机样本 P 进行加密，得到加密访问的共同 Cache 组信息，转化为多个可能的 γ 候选值，根据式(2)得到 β 的多个候选值，并为每个 β 候选值对应计数器加 1，由于加密密钥是唯一的，正确的 β 值每次都被预测到，多次预测后，出现频率最高的 β 值即为正确的 β 值，然后经进一步分析推断出密钥 K.

由于每一个加密没有访问的 Cache 组关联 δ 个(通常 δ 等于 16)查表索引和结果值，使用策略 1 时每一个加密没有访问的 Cache 组可排除 δ 个 β 候选值，同策略 2 相比，分析效率要比较高，故下文中我们主要采用策略 1 对 Camellia 进行 Cache 计时分析.

3.2　针对 Camellia F 函数的攻击模型

根据 2.3 节，Camellia 每个 F 函数执行 8 次查找表操作，对 4 种 S 盒分别进行 2 次查表操作. 图 1 给出了 OPENSSL-1.0.0-beta3 中 Camellia F 函数的 C 语言实现代码.

```
# define Camellia_Feistel(_s0,_s1,_s2,_s3,_key) do {\
  1 register u32 _t0,_t1,_t2,_t3;\
  2 \
  3 _t0=_s0 ^(_key)[0];\
  4 _t3=S4[_t0&0xff];\
  5 _t1=_s1 ^(_key)[1];\
  6 _t3 ^=S3[(_t0≫8)&0xff];\
  7 _t2=S1[_t1&0xff];\
  8 _t3 ^=S2[(_t0≫16)&0xff];\
  9 _t2 ^=S4[(_t1≫8)&0xff];\
 10 _t3 ^=S1[(_t0≫24)];\
 11 _t2 ^=_t3;\
 12 _t3=RightRotate(_t3,8);\
 13 _t2 ^=S3[(_t1≫16)&0xff];\
 14 _s3 ^=_t3;\
 15 _t2 ^=S2[(_t1≫24)];\
 16 _s2 ^=_t2; \
 17 _s3 ^=_t2;\
 } while(0)
```

图 1　OPENSSL-1.0.0-beta3 中 Camellia F 函数 C 语言代码

假定输入值_s0，_s1，_s2，_s3 可表示为

$$\_s=\alpha\oplus\beta_1 \tag{3}$$

其中，$\_s$ 表示_s0，_s1，_s2 或_s3；α 表示明文或已知中间状态变量；β_1 和 β_2 分别表示初始密钥 K 和扩展密钥 KE 的一个字节. 根据代码第 4，6，7，8，9，10，13，15 行，可知查找 S_n 表的索引值为

$$\gamma=\varphi(\alpha\oplus\beta_1\oplus\beta_2,n) \tag{4}$$

其中，φ 函数表示返回 32 位值的第 n 个字节值. 由于 γ 的不可能候选值可通过访问驱动 Cache 计时攻击采集阶段获取到，α 和 n 已知，则 $\beta_1\oplus\beta_2$ 的不可能候选值可以被预测并排除掉，使用更多的样本进行分析，正确的$\beta_1\oplus\beta_2$可被获取到.

4　Camellia-128 攻击

4.1　前 4 轮攻击

假定 X 为明文 P 和 128 位加密密钥 K(也可表

示为 KE_0,KE_1,KE_2,KE_3)的异或结果,在 Camellia 加密第 1 轮的 F 函数中,式(3)和(4)中的 β_1 可表示为 KE_0,KE_1,则可得到第 1 轮 8 次查表索引

$$\gamma_1=P_0\oplus KE_{0,0}\oplus KE_{4,0},\ \gamma_1=P_7\oplus KE_{1,3}\oplus KE_{5,3},$$
$$\gamma_2=P_1\oplus KE_{0,1}\oplus KE_{4,1},\ \gamma_2=P_4\oplus KE_{1,0}\oplus KE_{5,0},$$
$$\gamma_3=P_2\oplus KE_{0,2}\oplus KE_{4,2},\ \gamma_3=P_5\oplus KE_{1,1}\oplus KE_{5,1},$$
$$\gamma_4=P_3\oplus KE_{0,3}\oplus KE_{4,3},\ \gamma_4=P_6\oplus KE_{1,2}\oplus KE_{5,2} \tag{5}$$

其中,γ_n 表示查找 S_n 的查表索引值,其不可能候选值可通过攻击信息采集阶段获取到,由于明文 P 已知,根据式(5)可得到 $KE_0\oplus KE_4$ 和 $KE_1\oplus KE_5$ 的不可能候选值,多次分析得到正确的 $KE_0\oplus KE_4$ 和 $KE_1\oplus KE_5$,并恢复第 1 轮 F 函数的 8 次查找 S 盒的输入索引值和结果值.则第 1 轮右半部分的输出值 s_2^1,s_3^1 可表示为其输入值同多个查表结果值的异或结果,如式(6)所示,并可用于第 2 轮分析.

$$s_2^1=$$
$$s_2^1\hat{}S_2[P_4\oplus KE_{1,0}\oplus KE_{5,0}]\hat{}S_3[P_5\oplus KE_{1,1}\oplus KE_{5,1}]\hat{}$$
$$S_4[P_6\oplus KE_{1,2}\oplus KE_{5,2}]\hat{}S_1[P_7\oplus KE_{1,3}\oplus KE_{5,3}]\hat{}$$
$$S_1[P_0\oplus KE_{0,0}\oplus KE_{4,0}]\hat{}S_2[P_1\oplus KE_{0,1}\oplus KE_{4,1}]\hat{}$$
$$S_3[P_2\oplus KE_{0,2}\oplus KE_{4,2}]\hat{}S_4[P_3\oplus KE_{0,3}\oplus KE_{4,3}],$$
$$s_3^1=$$
$$s_3^1\hat{}S_2[P_4\oplus KE_{1,0}\oplus KE_{5,0}]\hat{}S_3[P_5\oplus KE_{1,1}\oplus KE_{5,1}]\hat{}$$
$$S_4[P_6\oplus KE_{1,2}\oplus KE_{5,2}]\hat{}S_1[P_7\oplus KE_{1,3}\oplus KE_{5,3}]\hat{}$$
$$S_1[P_0\oplus KE_{0,0}\oplus KE_{4,0}]\hat{}S_2[P_1\oplus KE_{0,1}\oplus KE_{4,1}]\hat{}$$
$$S_3[P_2\oplus KE_{0,2}\oplus KE_{4,2}]\hat{}S_4[P_3\oplus KE_{0,3}\oplus KE_{4,3}]\hat{}$$
$$RightRotate(S_1[P_0\oplus KE_{0,0}\oplus KE_{4,0}]\hat{}$$
$$S_2[P_1\oplus KE_{0,1}\oplus KE_{4,1}]\hat{}S_3[P_2\oplus KE_{0,2}\oplus KE_{4,2}]\hat{}$$
$$S_4[P_3\oplus KE_{0,3}\oplus KE_{4,3}],8) \tag{6}$$

同样,根据 Camellia 加密第 2 轮 F 函数,可得到其中第 2 轮 8 次查表索引

$$\gamma_1=P_8\oplus KE_{2,0}\oplus KE_{6,0},\ \gamma_1=P_{15}\oplus KE_{3,3}\oplus KE_{7,3}$$
$$\gamma_2=P_9\oplus KE_{2,1}\oplus KE_{6,1},\ \gamma_2=P_{12}\oplus KE_{3,0}\oplus KE_{7,0}$$
$$\gamma_3=P_{10}\oplus KE_{2,2}\oplus KE_{6,2},\ \gamma_3=P_{13}\oplus KE_{3,1}\oplus KE_{7,1}$$
$$\gamma_4=P_{11}\oplus KE_{2,3}\oplus KE_{6,3},\ \gamma_4=P_{14}\oplus KE_{3,2}\oplus KE_{7,2} \tag{7}$$

应用 3.2 节 F 函数分析模型,正确的 $KE_2\oplus KE_6$ 和 $KE_3\oplus KE_7$ 可被预测出,第 2 轮的右半部分输出同样可表示为第 2 轮的右半部分输入同多个查表结果值的异或结果

$$s_2^2=$$
$$s_2^2\hat{}S_2[P_{12}\oplus KE_{3,0}\oplus KE_{7,0}]\hat{}S_3[P_{13}\oplus KE_{3,1}\oplus KE_{7,1}]\hat{}$$
$$S_4[P_{14}\oplus KE_{3,2}\oplus KE_{7,2}]\hat{}S_1[P_{15}\oplus KE_{3,3}\oplus KE_{7,3}]\hat{}$$
$$S_1[P_8\oplus KE_{2,0}\oplus KE_{6,0}]\hat{}S_2[P_9\oplus KE_{2,1}\oplus KE_{6,1}]\hat{}$$
$$S_3[P_{10}\oplus KE_{2,2}\oplus KE_{5,2}]\hat{}S_4[P_{11}\oplus KE_{2,3}\oplus KE_{6,3}],$$
$$s_3^2=$$
$$s_3^2\hat{}S_2[P_{12}\oplus KE_{3,0}\oplus KE_{7,0}]\hat{}S_3[P_{13}\oplus KE_{3,1}\oplus KE_{7,1}]\hat{}$$
$$S_4[P_{14}\oplus KE_{3,2}\oplus KE_{7,2}]\hat{}S_1[P_{15}\oplus KE_{3,3}\oplus KE_{7,3}]\hat{}$$
$$S_1[P_8\oplus KE_{2,0}\oplus KE_{6,0}]\hat{}S_2[P_9\oplus KE_{2,1}\oplus KE_{6,1}]\hat{}$$
$$S_3[P_{10}\oplus KE_{2,2}\oplus KE_{5,2}]\hat{}S_4[P_{11}\oplus KE_{2,3}\oplus KE_{6,3}]\hat{}$$
$$RightRotate(S_1[P_8\oplus KE_{2,0}\oplus KE_{6,0}]\hat{}$$
$$S_2[P_9\oplus KE_{2,1}\oplus KE_{6,1}]\hat{}S_3[P_{10}\oplus KE_{2,2}\oplus KE_{5,2}]\hat{}$$
$$S_4[P_{11}\oplus KE_{2,3}\oplus KE_{6,3}],8) \tag{8}$$

成功分析密钥 K_L 值需对 Camellia-128 进行至少 4 轮攻击,应用 3.2 节攻击模型,结合前两轮攻击结果,可进一步得到第 3、4 轮的攻击结果,这样前 4 轮攻击结果如表 2 所示.

表 2 Camellia-128 前 4 轮攻击结果

加密轮	攻击结果
1	$KE_0\oplus KE_4\|KE_1\oplus KE_5((K_L\lll 0)_L\oplus(K_A\lll 0)_L)$
2	$KE_2\oplus KE_6\|KE_3\oplus KE_7((K_L\lll 0)_R\oplus(K_A\lll 0)_R)$
3	$KE_0\oplus KE_8\|KE_1\oplus KE_9((K_L\lll 0)_L\oplus(K_L\lll 15)_L)$
4	$KE_2\oplus KE_{10}\|KE_3\oplus KE_{11}((K_L\lll 0)_R\oplus(K_L\lll 15)_R)$

4.2 密钥扩展分析

在对 Camellia-128 进行 4 轮攻击后,可获取到 128 位变量 $C=(K_L\lll 0)\oplus(K_L\lll 15)$,经下面算法分析可获取 K_L 值:

K_L 搜索算法. Searching $K_L(S_K,C)$.

```
unsigned char K_P[128], cTemp
S_K ← ∅
For each i from 0x00 to 0x01
{
  K_P[0] ← i
  For each j from 0 to 127
  {
    cTemp ← (K_P[(15*j)%128] ^ C[(15*j)%128])
            &0x01
    If (j!=127)
      K_P[(15*(j+1))%128] ← cTemp;
    Else if (j==127)
    {
      If (cTemp==K_P[0])
      Add K_P to S_K
    }
  }
}
```

应用上面算法,最多可得到 2 个 K_L 候选值,在某些情况下甚至可直接得到唯一的 K_L 值.

4.3 攻击实验

我们在 64 位 AMD 处理器上对 OPENSSL-1.0.0-beta3 中的 Camellia 实现进行了访问驱动 Cache 计时攻击实验，其中计时功能是通过调用 RDTSC 指令获取系统自上电以来的时间戳值实现的，加密功能是通过黑盒方式调用 OPENSSL 库中函数来进行的，图 2 给出了一个 $KE_0 \oplus KE_4$ 密码字节密钥搜索空间和样本量的关系，可见 400 个样本左右即可恢复一个 $KE_0 \oplus KE_4$ 密码字节.

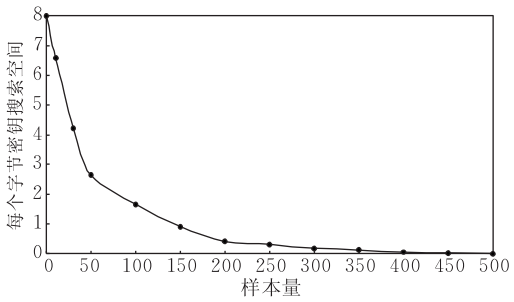

图 2　$KE_0 \oplus KE_4$ 每个字节密钥搜索空间和样本量(N)的关系

通过对 Camellia 前 4 轮进行攻击，利用 4.2 节方法对密钥扩展进行分析，可进一步得到 128 位初始密钥，实验结果表示大约 500 个样本即可快速恢复完整的 Camellia-128 密钥.

5 Camellia-192/256 攻击

Camellia-192/256 加密过程同 Camellia-128 基本相似，不同的就是比 Camellia-128 多调用了 6 次轮函数和 1 次 FL/FL^{-1} 函数. Camellia-192/256 的密钥扩展算法比 Camellia-128 要复杂，对 Camellia-128 密钥扩展的密钥分析方法并不适用于 Camellia-192/256，但这并不意味着 Camellia-192/256 要比 Camellia-128 安全. 下面给出针对 Camellia-192/256 的分析和实验过程.

5.1 前 6 轮攻击

应用 3.2 节 Camellia F 函数访问驱动 Cache 计时攻击模型，Camellia-192/256 前 6 轮攻击结果如表 3 所示.

表 3　Camellia-192/256 前 6 轮攻击结果

加密轮	攻击结果
1	$KE_0 \oplus KE_4 \parallel KE_1 \oplus KE_5 ((K_L \lll 0)_L \oplus (K_B \lll 0)_L)$
2	$KE_2 \oplus KE_6 \parallel KE_3 \oplus KE_7 ((K_L \lll 0)_R \oplus (K_B \lll 0)_R)$
3	$KE_0 \oplus KE_8 \parallel KE_1 \oplus KE_9 ((K_L \lll 0)_L \oplus (K_R \lll 15)_L)$
4	$KE_2 \oplus KE_{10} \parallel KE_3 \oplus KE_{11} ((K_L \lll 0)_R \oplus (K_R \lll 15)_R)$
5	$KE_0 \oplus KE_{12} \parallel KE_1 \oplus KE_{13} ((K_L \lll 0)_L \oplus (K_A \lll 15)_L)$
6	$KE_2 \oplus KE_{14} \parallel KE_3 \oplus KE_{15} ((K_L \lll 0)_R \oplus (K_A \lll 15)_R)$

5.2 密钥扩展分析

通过对 Camellia-192/256 前 6 轮攻击，分析第 3、4、5、6 轮攻击结果，可获取密钥扩展中第 5 轮输入状态值 $(K_A \lll 15) \oplus (K_R \lll 15)$，由于 Σ_5 为已知常量，根据密钥扩展算法，第 6 个 F 函数的输出值即 K_B (KE_4, KE_5, KE_6, KE_7) 可得到，根据表 3，K_L 和 K_R 可被恢复出来，最终得到 Camellia-192/256 完整密钥.

5.3 攻击实验

图 3 给出了 Camellia-192/256 一个 $KE_0 \oplus KE_4$ 字节密钥搜索空间同样本量的关系，可见 600 个样本可分析获取 $KE_0 \oplus KE_4$ 密钥字节，比 Camellia-128 所需的 400 个样本要大些，这主要是由于 Camellia-192/256 查表所用的 192 次要大于 Camellia-128 的 144 次，攻击采集到的加密没有访问到的 Cache 组数量要少些造成的.

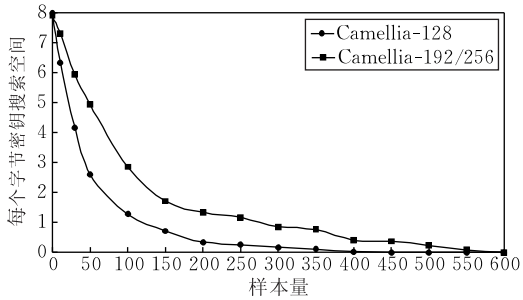

图 3　Camellia 128 and 192/256 中 $KE_0 \oplus KE_4$ 每个字节密钥搜索空间和样本量的关系

通过对 Camellia 前 6 轮进行攻击，利用 5.2 节方法对密钥扩展进行分析，实验结果表示大约 800～900 个样本即可快速恢复完整 Camellia-192/256 初始密钥.

6 已知密文攻击

第 4、5 节给出了针对 Camellia 加密过程的已知明文 Cache 计时攻击，根据 Camellia 的对称密码特性，攻击者也可针对 Camellia 解密过程进行已知密文 Cache 计时攻击.

6.1 Camellia-128 攻击

攻击者首先采集到 Camellia-128 解密过程中的 Cache 组集合信息，根据式(4)，通过对前 4 轮解密过程攻击，恢复相关密钥结果如表 4 所示. 通过分析第 3 轮和第 4 轮攻击结果 $(K_A \lll 111) \oplus (K_A \lll 94)$，参考 4.2 节密钥恢复方法，可恢复 1～2 个 K_A 候选值，然后进一步得到初始密钥 K_L.

表 4 Camellia-128 前 4 轮解密攻击结果

解密轮	攻击结果
1	$KE_{48}\oplus KE_{44}\|KE_{49}\oplus KE_{45}((K_A<<<111)_L\oplus(K_L<<<111)_L)$
2	$KE_{50}\oplus KE_{46}\|KE_{51}\oplus KE_{47}((K_A<<<111)_R\oplus(K_L<<<111)_R)$
3	$KE_{48}\oplus KE_{40}\|KE_{49}\oplus KE_{41}((K_A<<<111)_L\oplus(K_A<<<94)_L)$
4	$KE_{50}\oplus KE_{42}\|KE_{51}\oplus KE_{43}((K_A<<<111)_R\oplus(K_A<<<94)_R)$

6.2 Camellia-192/256 攻击

攻击者首先采集到 Camellia-192/256 解密过程中的 Cache 组集合信息，根据式(4)，通过对前 6 轮解密过程攻击，恢复相关密钥结果如表 5 所示. 通过分析第 3、4、5、6 轮攻击结果，可恢复 $K_A\oplus K_R$ 值，参考 5.2 节密钥恢复方法得到 K_B，K_L，K_R，进而得到初始密钥 K.

表 5 Camellia-192/256 前 6 轮解密攻击结果

解密轮	攻击结果
1	$KE_{64}\oplus KE_{60}\|KE_{65}\oplus KE_{61}((K_B<<<111)_L\oplus(K_L<<<111)_L)$
2	$KE_{66}\oplus KE_{62}\|KE_{67}\oplus KE_{63}((K_B<<<111)_R\oplus(K_L<<<111)_R)$
3	$KE_{64}\oplus KE_{56}\|KE_{65}\oplus KE_{57}((K_B<<<111)_L\oplus(K_A<<<94)_L)$
4	$KE_{66}\oplus KE_{58}\|KE_{67}\oplus KE_{59}((K_B<<<111)_R\oplus(K_A<<<94)_R)$
5	$KE_{64}\oplus KE_{52}\|KE_{65}\oplus KE_{53}((K_B<<<111)_L\oplus(K_R<<<94)_L)$
6	$KE_{66}\oplus KE_{54}\|KE_{67}\oplus KE_{55}((K_B<<<111)_L\oplus(K_R<<<94)_L)$

6.3 攻击实验

在 4.3 节环境下，我们对 Camellia-128/192/256 进行了 Cache 计时攻击实验，使用 6.1 节和 6.2 节分析方法，500 和 900 个左右样本 1s 左右可分别恢复 Camellia-128 和 Camellia-192/256 密钥.

7 远程攻击

在本地攻击实验中，我们在程序中通过黑盒方式调用 OPENSSL 密码库来进行攻击实验，攻击程序和加密服务之间并没有直接进行交互. 为了表明本文攻击对远程环境下的适应性，我们将攻击程序和解密程序分开，在局域网和校园网环境下进行了远程攻击实验，实验结果良好.

远程攻击实验使用了 3 个程序，攻击程序(AP)、Camellia 服务程序(CSP)、间谍程序(SP). SP 和 CSP 被部署在同一电脑上，SP 通过观测访问自身数据命中、失效信息采集访问驱动 Cache 计时信息，攻击步骤如下：

1. AP 通知 SP 清空 CSP 所在电脑数据 Cache，初始化 Cache 为一个固定状态；

2. AP 向 CSP 发送加密请求，CSP 加密后将密文反馈给 AP；

3. AP 收到密文后，通知 SP 再次访问 Cache，根据命中失效信息采集 CSP 加密过程中访问过和没有访问过的 Cache 组集合信息. 之后，SP 将所采集信息发送给 AP.

4. 利用相应的分析方法，AP 对采集的访问驱动 Cache 计时信息对密钥 K 进行离线分析.

远程攻击实验中我们发现，和本地攻击相比，远程攻击由于信息采集过程中网络发包拆包需多次访问 Cache，带来一定的噪声，攻击样本量比本地大. 本地和局域网环境下 AES 访问驱动 Cache 计时攻击一个样本中间谍程序对所有 Cache 组采样分别如图 4、图 5 所示，其中横坐标表示 Cache 组序号，纵坐标表示 Cache 组访问时钟周期，易见，Camellia 的 4 个 S 盒分别对应的起始 Cache 组为 261，277，293，309，本地攻击中由系统进程和其它用户进程对

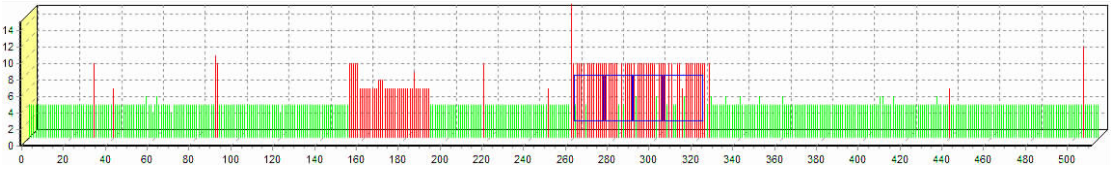

图 4 本地攻击中一个样本 Cache 组访问时钟周期图

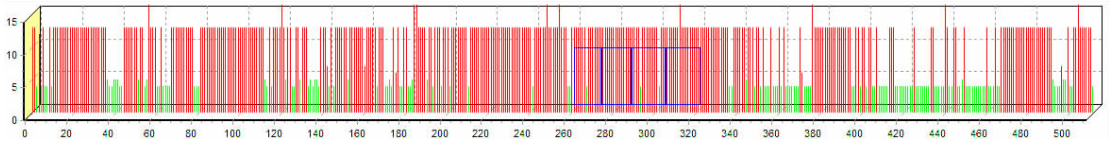

图 5 远程攻击中一个样本 Cache 组访问时钟周期图

Cache 访问带来的噪声比较小，对攻击影响不大，而远程攻击中网络发包拆包确实给攻击带来很大影响，但 AES 加密过程中查找多个表访问 Cache 组信息（方框选择区域中访问时钟周期较小的 Cache 组）仍可被采集到，攻击仍可成功实现，只是需要更多的样本量而已. 实验结果表明 3000 个样本左右可恢复 Camellia 128/192/256 完整密钥.

8　攻击分析与防护措施

8.1　攻击分析

8.1.1　和其它分组密码访问驱动 Cache 计时攻击的比较

我们对典型的几个分组密码进行了访问驱动 Cache 计时攻击实验，如 AES、SMS4、Camellia，实验结果良好，证明文中 3.1 节攻击模型十分有效.

(1) AES[22]. 对 AES 进行访问驱动 Cache 计时攻击只需获取任何一轮扩展密钥即可. 在 AES 加密第 1 轮中，根据采集的加密访问 Cache 组集合信息及其转化的查表索引值，结合明文经分析可直接获取初始密钥；在 AES 最后一轮中，根据加密访问的 Cache 组集合信息及其转化的查表结果值，结合密文分析可恢复第 10 轮扩展密钥，根据 AES 可逆的密钥扩展结构，可直接恢复初始密钥.

(2) SMS4[23]. 对 SMS4 进行攻击需要获取前 4 轮或最后 4 轮的扩展密钥. 在 SMS4 第 1 轮攻击中，查表索引值等于 3 个 32 位明文字和第 1 轮扩展密钥 rk_0 异或值，根据加密访问的 Cache 组集合信息及其转化的查表索引值，结合明文可恢复 rk_0，进一步可计算出第 2 轮的输入值. 用同样方法恢复第 2、3、4 轮扩展密钥 rk_1，rk_2，rk_3，结合密钥扩展算法恢复 128 位 SMS4 初始密钥. 同样，通过对 SMS4 最后 4 轮进行分析，可恢复 rk_{28}，rk_{29}，rk_{30}，rk_{31}，结合密钥扩展算法恢复 128 位 SMS4 初始密钥.

(3) Camellia. 对 Camellia 进行访问驱动 Cache 计时攻击要比 AES 和 SMS4 复杂一些，因为 Camellia查表索引值并不是简单的和初始密钥相关，而是同初始密钥和轮密钥的异或结果值相关，并且 Camellia 的密钥扩展过程也要比 AES 和 SMS4 复杂，但是由于其密钥扩展过程中使用了大量的左移函数，为 Camellia 密钥分析提供了极大的便利.

8.1.2　攻击成功原因分析

Camellia 的 F 函数可泄露加解密密钥和扩展密钥异或结果，Camellia S 盒在 Cache 中的不对齐分布特性极大地加速了初始密钥和扩展密钥异或结果值的恢复，Camellia 密钥扩展中的左移函数严重影响了其安全性.

(1) Camellia S 盒在 Cache 中的不对齐特征

假设 O 代表 Camellia 第一个查找表第一个字节在 Cache 组中的起始位置（0～15），根据 $O=0$ 和 $O\neq 0$ 两种情况进行如下分析：

① $O_l=0$

如果查找表第一个字节在 Cache 组中是对齐的，即其恰好对应 Cache 组的第一个字节，那么利用所采集的样本加密没有访问的每个 Cache 组将对应着 16 个高 4 位相同、低 4 位连续的索引字节，每次将排除掉 16 个高 4 位相同、低 4 位连续的密钥字节不可能值，而正确的密钥字节值不可能被排除；同样，同正确的密钥字节高 4 位相同、低 4 位不同的另外 15 个密钥字节也是不可能被排除掉的，这种情况下每个相关密钥字节候选值最多可从 256 降低到 16.

② $O_l\neq 0$

那么如果查找表第一个字节在 Cache 组中不是对齐的，那么加密没有访问的每个 Cache 组将对应着两个查表行，第 1 个查表行对应 16-O 个索引，其高 4 位相同、低 4 位连续，第 2 个查表行对应 O 个索引，其高 4 位相同、低 4 位连续. 这样每次排除的 16 个密钥和索引一样，均对应两个密钥行，第 1 个行对应 16-O 个值，高 4 位相同、低 4 位连续；第 2 个行对应 O 个值，高 4 位相同、低 4 位连续，易见，这两个行的数量越接近，即 $O=7$、8 时，排除效果越好，攻击所需样本量也小.

以上可知，当 $O_l=0$ 时，很难直接恢复 Camellia 初始密钥和扩展密钥的异或结果值，但是大部分情况下，Camellia 查找表在 Cache 中是不对齐的，该特征极大地加速了 Camellia 的密钥恢复效率.

(2) 密钥扩展设计问题

很多分组密码在密钥扩展中使用了左移和右移函数来产生扩展密钥，如 Camellia、ARIA[24].

① Camellia. Camellia-128 的扩展密钥通过左移 K_L 和 K_A 产生，而 Camellia-192/256 的扩展密钥则通过左移 K_L，K_R，K_A 和 K_B 产生. 我们称获取到某一个 128 位密钥相关状态值和其移位后的状态的异或结果为一个碰撞，那么一旦攻击者找到这样一个碰撞，应用 4.2 节分析方法，可快速恢复该 128 位密钥相关状态值. 对于 Camellia-128 来说，第 3、4 轮攻击结果可找到 K_L 的一个碰撞，初始密钥 K_L 密钥

搜索空间可由 2^{128} 降低到 2；而对于 Camellia-192/256 来说，通过对第 3、4 轮分析可得到 $(K_L \lll 0) \oplus (K_R \lll 15)$，第 5、6 轮分析可得到 $(K_L \lll 0) \oplus (K_A \lll 15)$，进一步分析可直接得到 $K_A \oplus K_R$，凑巧的是 $K_A \oplus K_R$ 也是 Camellia-192/256 第 5 个 F 函数的输入值，进而可快速恢复 K_B、K_L 和 K_R.

② ARIA. ARIA 算法是韩国国家分组密码加密标准，常被应用到轻量级的密码设备中，分组长度为 128 位，支持 128、192、256 位 3 种密钥长度. 在其加密过程中，查表索引值等于一个已知的中间状态值和轮密钥 rk_r 的异或结果，通过访问驱动 Cache 计时攻击，可获取 ARIA 每轮的扩展密钥 rk_r，并不能直接获取初始密钥 K. 但在 ARIA 密钥扩展中，13 轮扩展密钥均通过对 4 个 128 位变量 W_0，W_1，W_2，W_3 的经简单移位操作而得，通过分析前 4 轮的攻击结果，攻击者可找到 W_0 的一个碰撞，然后直接获取初始密钥 W_0.

因此，密钥设计者在密钥扩展设计中使用移位函数时一定要谨慎.

8.1.3 同前人远程 *Cache* 计时攻击比较

在远程攻击方面，目前国内外已公布针对 AES 的 Cache 远程计时攻击只有 Bernstein[7] 和 Acıiçmez[14] 两例，攻击均属时序驱动 Cache 计时攻击.

(1) Bernstein 远程攻击. 攻击端和加密服务端虽部署在不同的 PC 机上，但负责采集从加密开始到完成时间信息并回传给攻击端的却是远程加密端，绝对消除了网络传输时延，使用 $2^{27.5}$ 个样本恢复出 128 位 AES 密钥，这在实际攻击中是不现实的，攻击实际上仍属本地计时攻击.

(2) Acıiçmez 远程攻击. 攻击端和加密服务端部署在同一台 PC 机上，攻击端负责采集从加密开始到加密完成时间，也是在消除了网络传输时延的条件下，使用 $2^{26.66}$ 个样本才恢复出完整密钥，验证了理想条件下进行时序驱动远程计时攻击的可行性. 在真实远程条件下，网络传输时延本身甚至其抖动时延一般远大于加密时间，时序驱动 Cache 远程计时攻击信息的精确采集十分困难.

本文实现的访问驱动 Cache 远程计时攻击，在很大程度上避开了时序驱动远程计时攻击中网络传输时延干扰比较大的问题，在局域网甚至不同教学楼之间的远程环境下将攻击端和加密服务端分别部署在不同的 PC 机上，同时将间谍程序植入到加密服务器上，受远程攻击端操控执行正常的数据访问操作采集 Camellia 加密过程中的 Cache 旁路信息，结合明文/密文对信息开展密钥分析，在有限样本下快速恢复完整 Camellia 密钥，攻击样本量和所需时间相对比较小.

8.2 防护措施

本节只讨论从算法设计角度对 Camellia 访问驱动 Cache 计时攻击的防护措施.

(1) 使用多个密钥相关变量(至少 3 个)作为查找 S 盒的输入索引值.

对于 AES 和 SMS4 分组密码，查表操作仅仅使用了一种密钥相关变量(初始密钥或扩展密钥)，经过访问驱动 Cache 计时分析，攻击者可直接恢复初始密钥或扩展密钥. 但 Camellia 查找表使用了两个密钥相关变量(初始密钥和扩展密钥)，如果不对 Camellia 密钥扩展进行分析，将不能直接获取初始密钥，如果在 Camellia 加密查找 S 盒时，输入索引和 3 个以上密钥变量相关，则会大大增加攻击者分析的复杂度.

(2) 在相邻的两个或多个 F 函数之间插入 FL/FL^{-1} 混淆层.

在 Camellia 原有设计中，通过在每 6 轮插入由与、或、异或简单逻辑操作组成的 FL/FL^{-1} 混淆层可大幅度提高非线性度和算法安全性. 值得提出的是，由于 Camellia 中函数 FL/FL^{-1} 混淆层存在，攻击者很难对第 6 轮后的其它轮进行成功攻击，但不幸的是攻击前 6 轮便足以恢复 Camellia-128/192/256 完整密钥. 如果攻击者增加 FL/FL^{-1} 混淆层的使用频率，如每隔两轮就插入 FL/FL^{-1} 混淆层，则会大大增加攻击复杂度.

(3) 在扩展密钥生成过程中除了移位操作以外使用更多的其它线性逻辑操作，如异或、列混淆等.

通过移位操作来产生轮密钥实现效率较高，但安全隐患比较大，一旦攻击者找到了对同一个密钥变量的一个异或碰撞，将很容易获取该密钥变量，经进一步分析获取初始密钥. 在目前的 Camellia 实现中，仅使用了移位一种操作来生成扩展密钥，如果能在扩展密钥生成过程中增加更多的简单逻辑操作如与、或、异或等操作，也会在某种程序上大大增加攻击复杂度. 另外，在 Camellia-192/256 密钥扩展算法设计中，我们也建议其在对 K_A 和 K_R 进行的移位操作位数不要相同.

9 结束语

本文就当前最安全的 Camellia 分组密码算法

访问驱动 Cache 计时攻击进行了一些相关研究，研究结果表明此类针对 Camellia 的计时攻击手段对信息安全将带来突出威胁：首先，Camellia 是目前欧洲和日本占主导地位的 128 位分组加密算法，因此，攻击产生的影响将是广泛而深远的；其次，实施攻击并不需要物理地获得密码执行部件以测量泄漏信息，在网络环境下也可成功获取远程加密服务器密钥；还有，此类攻击能够作用于一切实现于“Cache-Memory”层次存储结构计算机设备上软件形式的 AES 算法，从而危害到服务器、桌面以及嵌入式等各种主流的计算机系统. 因此，应对这类攻击予以充分的关注.

致　谢　感谢评审老师们的辛勤工作、郑天明硕士在 Camellia 密钥扩展分析中的精彩讨论、国家现代通信实验室罗岚副研究员对本文提出的宝贵建议以及美国麻省理工学院计算机科学与人工智能实验室的 Tromer Eran 博士对攻击分析和远程实验的建议！

参　考　文　献

[1] Kocher Paul C. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems//Koblltz N ed. Proceedings of Advances in Cryptology—CRYPTO 96. Santa Barbara, California, USA, 1996: 104-113

[2] Kelsey John, Schneier Bruce, Wagner David, Hall Chris. Side channel cryptanalysis of product ciphers//Lecture Notes in Computer Science 1485. Springer, 1998: 97-110

[3] Page Dan. Theoretical use of Cache memory as a cryptanalytic side-channel. Department of Computer Science, University of Bristol: Technical Report CSTR-02-003, 2002: 1-23

[4] Tsunoo Yukiyasu, Saito Teruo, Suzaki Tomoyasu, Shigeri Maki, Miyauchi Hiroshi. Cryptanalysis of DES implemented on computers with Cache//Walter C D et al eds. Proceedings of the Cryptographic Hardware and Embedded Systems — CHES 2003. Cologne, German, 2003: 62-76

[5] Tsunoo Y, Suzaki T, Saito T, Kawabata T, Miyauchi H. Timing attack on camellia using Cache delay in S-Boxes (in Japanese)//Proceedings of the 2003 Symposium on Cryptography and Information Security — SCIS2003. Hamamatsu, Japan, 2003: 179-184

[6] Yoshitaka Ikeda, Toshinobu Kaneko. A study on the effect of Cache structure to the Cache timing attack for a block cipher(in Japanese). IEIC Technical Report WBS2003 174-190, 2004, 103(714): 37-42

[7] Bernstein Daniel J. Cache-timing attacks on AES, 2004. Available online at http: //cr. yp. to/papers. html\ # cachetiming

[8] Osvik Dag Arne, Shamir Adi, Tromer Eran. Cache attacks and countermeasures: The case of AES//Pointcheval D ed. Proceedings of the Topics in Cryptology — CT-RSA 2006. San Jose, CA, USA, 2006: 1-20

[9] Tromer Eran, Osvik Dag Arne, Shamir Adi. Efficient Cache attacks on AES, and countermeasures. Journal of Cryptology, 2010, 23(1): 37-71

[10] Neve Michael, Seifert J-P. Advances on access-driven Cache attacks on AES//Proceedings of the Selected Areas in Cryptography—SAC2006. Montreal, Canada, 2007: 147-162

[11] Bertoni G, Zaccaria V, Breveglieri L, Monchiero M, Palermo G. AES power attack based on induced Cache miss and countermeasure//Proceedings of the International Symposium on Information Technology: Coding and Computing — ITCC 2005. Las Vegas, NV, USA, 2005, 1: 586-591

[12] Percival C. Cache missing for fun and profit//Proceedings of the Technical BSD Conference 2005. Ottawa, 2005: 1-13

[13] Bonneau J, Mironov I. Cache-collision timing attacks against AES//Goubin L, Matsui M eds. Proceedings of the Cryptographic Hardware and Embedded Systems—CHES2006. Yokohama, Japan, 2006: 201-215

[14] Acıiçmez O, Schindler W, Koç Ç K. Cache-based remote timing attack on the AES//Abe M ed. Proceedings of the Topics in Cryptology—CT-RSA 2007. San Francisco, CA, USA, 2007: 271-286

[15] Zhao Xinjie, Wang Tao, Mi Dong. Robust first two rounds access driven Cache timing attack on AES//Proceedings of the International Conference on Computer Science and Software Engineering (CSSE 2008). Wuhan, China, 2008, 3: 785-788

[16] Brickell E, Graunke E, Neve M, Seifert S. Software mitigations to hedge AES against Cache-based software side-channel vulnerabilities. Cryptology ePrint Archive, 2006. http: //eprint. iacr. org/2006/052. pdf

[17] Blömer Johannes, Krummel Volker. Analysis of countermeasures against access driven Cache attacks on AES//Adams C, Miri A, Wiener M eds. Proceedings of the Selected Areas in Cryptography—SAC 2007. Ottawa, Ontario, Canada, 2007: 96-109

[18] Wang Z, Lee R. New Cache designs for thwarting software Cache-based side channel attacks//Proceedings of the ISCA 2007. San Diego, CA, USA, 2007: 494-505

[19] Aoki Kazumaro, Ichikawa Tetsuya, Kanda Masayuki, Matsui Mitsuru, Moriai Shiho, Nakajima Junko, Tokita Toshio. Camellia: A 128-bit block cipher suitable for multiple platforms design and analysis//Proceedings of the Selected Areas in Cryptography — SAC2000. Waterloo, Ontario, Canada, 2001: 39-56

[20] OpenSSL the open-source toolkit for SSL /TLS [EB/OL], 2005. Available online at http: //www. openssl. org/

[21] Aoki K, Ichikawa T, Kansa M, Matsui M, Moriai S, Nakajima, Tokita T. Specification of Camellia-a 128-bit Block Cipher. 2000. http://www.cosic.esat.kuleuven.be/nessie/workshop/submissions

[22] Hu Xiang-Dong, Wei Qin-Fang. Application Cryptology. Beijing: Publishing House of Electronics Industry, 2006: 83-105(in Chinese)
(胡向东，魏琴芳. 应用密码学. 北京：电子工业出版社，2006：83-105)

[23] Office of state commercial cipher administration. Block cipher for WLAN products—SMS4. http://www.oscca.gov.cn/UpFile/200622026423297990.pdf

[24] Kwon D, Kim J, Park S et al. New block cipher: ARIA// Proceedings of the Information Security and Cryptology—ICISC'03. Seoul, Korea, 2003: 432-445

ZHAO Xin-Jie, born in 1986, Ph. D. candidate. His main research interests include block cipher side channel analysis, block cipher fault analysis.

WANG Tao, born in 1964, Ph. D., professor, Ph. D. supervisor. His main research interests include information security analysis, side channel analysis.

ZHENG Yuan-Yuan, born in 1981, Ph. D. candidate. Her main research interests include block cipher security analysis, algebraic side channel analysis.

Background

This work is supported by the National Natural Science Foundation of China under grant No. 60772082 and the Natural Science Foundation of Hebei Province under grant No. 08M010. They aim to find out how to analyze the key of the cipher systems with the leakage of the timing information during their implementations by the micro-architectural units, such as data Cache, instruction Cache, and branch Prediction unit etc, make research on the timing information measuring and key analyzing methods of different encryption algorithms, finally realize the timing attack under real remote environment such as local network, campus network even the internet, find out a common side channel analysis and attack framework for different cipher systems, make some breakthroughs in both methods and theories of side channel attacks so as to built stable foundations for the further innovations.

In these two programs, the RSA, AES, SMS4, Camellia, ARIA, HC128, HC256 etc ciphers were analyzed with Cache timing attacks under both local and remote environments and the experiments showed a good result. Besides, a formal security model for cipher under side channel attack is analyzed, it's hopeful to evaluate the security of cipher under side channel attack quantitatively and propose corresponding countermeasures so as to prevent this type of attack. This paper is an important content of Camellia block cipher Cache timing attack research, the experiment results show that Camellia is vulnerable to access driven Cache timing attacks, both the F function and the left circular rotating operation of the key schedule has serious designing problem. Experiment results demonstrate: 500 and 900 random plaintexts are enough to recover Camellia-128 and Camellia-192/256 key, and the attacks can be easily expanded to known ciphertext condition by attacking the Camellia decryption procedure and even remote scenarios, 3000 random plaintexts are enough to recover Camellia-128/192/256 key in both local and campus networks. Finally, the authors analyze the reason why Camellia is vulnerable to this type of attack, and provide some advices to cipher designers for hardening ciphers against Cache timing attacks.