基于信息熵的 RSA 硬件时间隐通道信息泄露量化研究

毛保磊[1),2)] 胡 伟[1),2)] 慕德俊[1)] 张慧翔[2)] 邰 瑜[2)] 洪 亮[2)]

[1)](西北工业大学深圳研究院 广东 深圳 518507)

[2)](西北工业大学自动化学院 西安 710072)

摘 要 RSA 密码算法作为主流的公钥加密和签名算法,其安全性被工业界和学术界广泛关注.RSA 算法的安全性主要包括算法自身的不易破解性和密钥的安全性两个方面.而通过能量和时间隐通道来攻击算法密钥往往比破解 RSA 算法更为有效.现有的研究大多关注 RSA 算法软件实现的安全性,并未深入探讨硬件 IP(Intellectual Property)核中的时间隐通道对安全性的影响;虽然有基于形式化验证的方法对时间隐通道进行检测和隔离,或者采用基于类型系统的方法从硬件设计语言的角度消除时间隐通道,但这些硬件方法都只能实现时间隐通道的定性分析,缺乏有效的模型对时间隐通道进行量化分析.文中针对上述两个基本问题(硬件 IP 核与时间隐通道)开展研究.首先介绍了 RSA 时间隐通道的研究背景和硬件实现的威胁模型.然后引入基于信息熵的研究方法,分别建立了基于信息熵的时间隐通道攻击模型和基于信息熵的时间隐通道量化分析模型.文中实验对 RSA 密码核进行基于信息熵的攻击和基于方差的攻击以评估信息熵攻击的效果.同时,针对同一密码核不同密钥信息泄露进行量化分析;针对多种不同的 RSA 硬件架构量化分析模幂优化算法对时间隐通道信息泄露的影响;针对时间隐通道抵抗措施评估其减少时间隐通道信息泄露的作用;并通过攻击相应 RSA 核密钥以验证信息熵量化分析的有效性.最后实验综合评估不同 RSA 架构对设计复杂度的影响.实验结果显示基于信息熵的攻击方法在猜测正确率确信度方面优于基于方差的攻击方法;信息熵量化分析方法能够有效的评估 RSA 密码核时间隐通道信息泄露,为 RSA 密码核时间隐通道的研究提供量化分析的理论依据和测试手段.实验结果同时表明信息熵指标能够辅助设计人员权衡时间隐通道安全性与性能、资源开销之间的关系,为硬件设计自动化提供潜在的时间隐通道硬件安全评价指标,实现对硬件设计特征更加精细和完善的描述.

关键词 硬件安全;隐通道分析;时间隐通道;信息泄露;RSA 算法;信息熵;量化分析

中图法分类号 TP309 **DOI 号** 10.11897/SP.J.1016.2018.00426

Quantitative Analysis of Information Leakage Through Hardware RSA Timing Channel Based on Entropy Theory

MAO Bao-Lei[1),2)] HU Wei[1),2)] MU De-Jun[1)] ZHANG Hui-Xiang[2)] TAI Yu[2)] HONG Liang[2)]

[1)](*Shenzhen Research Institute, Northwestern Polytechnical University, Shenzhen, Guangdong* 518507)

[2)](*School of Automation, Northwestern Polytechnical University, Xi'an* 710072)

Abstract The RSA algorithm is a widely deployed public key cipher for data encryption and digital signature, whose security has drawn attention from both academic and industry fields. Its security relies on both the computation complexity of breaking the algorithm itself and the security of the encryption key. Generally, it is much easier to recover the encryption key than break the RSA algorithm through power and timing side channel analysis. Previous work primarily focuses on timing side channels in software RSA implementations, without in-depth studying the effect of

收稿日期:2016-06-28;在线出版日期:2017-05-24.本课题得到国家自然科学基金(61672433)、深圳市科创委基础研究基金(201703063000517)、国家密码发展基金(MMJJ20170210)、中央高校基本科研业务费专项资金(3102017OQD094)资助.**毛保磊**,男,1987 年生,博士研究生,主要研究方向为隐通道分析、硬件设计信息流安全分析.E-mail:maobaolei@mail.nwpu.edu.cn.**胡 伟**(通信作者),男,1982 年生,博士,副教授,主要研究方向为硬件安全、可重构系统和嵌入式系统设计.E-mail:vinnie@mail.nwpu.edu.cn.**慕德俊**(通信作者),男,1963 年生,博士,教授,主要研究领域为信息安全、控制系统设计.E-mail:mudejun@nwpu.edu.cn.**张慧翔**,男,1981 年生,博士,副教授,中国计算机学会(CCF)会员,主要研究方向为信息安全、模式识别与应用.**邰 瑜**,男,1982 年生,博士研究生,主要研究方向为硬件安全、逻辑综合及优化.**洪 亮**,男,1979 年生,博士,副教授,主要研究方向为信息安全、嵌入式系统应用.

hardware architecture on timing channel security. Although there is work for detecting and isolating timing channel based on formal verification of timing information flow or building timing channel free hardware design by incorporating new type system into the hardware design language, they can only provide qualitative analysis of timing channel, lacking effective model to perform quantitative analysis of hardware timing channel security. In this work, we will concentrate on hardware RSA cores and provide a quantitative analysis model to evaluate such timing channel leakage. Firstly, we introduce hardware RSA timing channel and its threat model. We then employ the entropy theory to set up timing attack model and quantitative analysis model for RSA architecture timing channel. Besides, we attack RSA implementation based on entropy and variance analysis, respectively. In order to demonstrate the effectiveness of entropy in quantifying hardware RSA implementation leakage, we perform quantitative analysis of different key-pairs information leakage within the same RSA architecture, quantify information leakage for different RSA architecture implementations with timing channel algorithm optimization techniques, evaluate the effect of timing channel countermeasure on reducing information leakage; and also attack each RSA implementation to validate the effectiveness of our quantitative analysis model. Finally, we evaluate the effect of different algorithm optimizations, timing channel mitigation techniques and countermeasures on design complexity in terms of timing channel, performance and resource utilization. Experimental results show that entropy metric can be used to attack RSA timing channel and it can increase the success rate by combining variance analysis with entropy analysis. Entropy metric can be used to quantitatively analyze information leakage from timing channel in RSA hardware architectures effectively and efficiently, which provides an effective theoretical basis and test methodology to assess the severity of timing channel information leakage. In addition, entropy metric can help designers to tradeoff security requirements and design overheads such as performance and resource utilization, which provides a potential security metric to integrate timing channel security with traditional design metrics (e. g. area and performance) to characterize the hardware in more detail.

Keywords　hardware security; side channel analysis; timing channel; information leakage; RSA algorithm; entropy; quantitative analysis

1　引　言

现代密码算法设计通常基于困难问题求解，如 RSA 算法基于大数因式分解的极端困难性. 密码算法的安全实现不仅要求算法本身难以破解，还要求保障密钥安全性. 攻击算法密钥往往比攻击密码算法本身更为有效. 在密码算法的具体实现中，可能会存在时间或者能量隐通道. 例如，加解密程序指令根据不同的密钥值选择不同执行路径，产生不同时间延迟，攻击者根据时间延迟的特征使用统计分析等方法破解密钥[1-3]. 攻击者亦可通过测量、分析加解密实现的功耗获取密钥. 这些隐通道导致密钥泄露，严重影响密码算法安全.

Kocher 博士首次实现了对基本 RSA 模幂算法的时间隐通道的攻击[1]. Schindler 成功攻击了基于中国剩余定理的 RSA 实现并将其时序攻击方法扩展到其他高级模幂算法[2]. 斯坦福大学进行的实验进一步验证了广泛应用的 OpenSSL 库中的 RSA 模幂实现存在时间隐通道，导致其在嘈杂的网络环境下依然能被攻击，充分暴露 RSA 时间隐通道问题的严重性[3]. 同时基于共享机制的处理器硬件结构也容易遭受时间隐通道影响，被木马进程获取到机密信息[4]. 高速缓冲存储结构 Cache 是共享硬件泄露信息的重要来源，攻击者可以通过对 Cache 命中率分析提取 AES 密钥信息[5]. 这些研究集中于软件密码系统，或在 CPU 架构条件下衡量密钥信息泄露. 密码算法属于计算密集型应用，通常通过密码协处理器、IP 核等硬件加速. 硬件密码核被广泛地应用于国防、金融、通讯等社会重要领域的高速加解密计算中，其硬件体系结构(IP 核)的安全实现对保障国防、通讯系统的安全性和可靠性具有重要的意义. 密

码算法核的安全性问题日益突出.

硬件密码核实现区别于软件层次的实现主要在于:(1)攻击者可通过调用硬件密码核加解密来精确测量加解密所需的时间,测量结果是周期精确的,可以消除测量误差对时序分析的影响;(2)硬件架构实现的灵活性更高,设计描述方式和逻辑综合都对实现结果有更显著的影响,导致 RSA 架构的差异性较大,因此,RSA 密码核中时间隐通道的不确定性因素更高.

硬件时间隐通道问题受到了研究者的广泛关注,但是,尚无一种有效的方法来检测和消除隐通道. Ciet 等人利用并行的带有余数系统的 RSA 抵抗硬件时间隐通道的攻击[6],Zhang 等人设计硬件语言强化硬件时间隐通道的非干扰特性[7],Li 等人设计基于状态机和时间复用(Time Division Multiple Access,TDMA)的硬件语言迁移时间隐通道[8],但未提供有效的检测方法. Oberg 等人提出利用门级信息流追踪(Gate Level Information Flow Tracking,GLIFT)方法检测时间隐通道[9],能够验证不同模块间非干扰特性[10],却不能测量时间隐通道容量,无法量化分析硬件密码核时间隐通道脆弱性.

通信中使用的信息论方法能有效衡量一个通信信道的带宽容量,量化分析信道传输能力. 我们引入信息熵研究 RSA 硬件结构对时间隐通道的影响,即量化分析 RSA 硬件结构的时间隐通道的密钥信息泄露. 量化分析能够更加准确地对硬件实现的安全性进行评估. 例如,当硬件密码核只泄露密钥的某一位,其他密钥位不存在泄露时,虽然违反了非干扰特性,但是攻击者并不能破解完整的密钥,此时,硬件结构安全,无需降低性能、耗费资源进行改进和重新设计. 另外,量化分析能够有效地区分不同硬件密码结构的信息泄露程度,显露密钥是否容易被破解,即使攻击者目前尚未找到高效的破解手段. 量化分析亦可有效揭露硬件结构潜在的脆弱性,有利于设计者降低和消除硬件结构潜在的时间隐通道风险. 最终目的是为硬件设计者提供一种可量化分析和推理 RSA 硬件时间隐通道信息泄露的手段,并在资源、性能允许情况下,提高 RSA 密码核设计的安全性.

本文研究硬件 RSA 时间隐通道的意义不仅在于从理论和实验中验证具体的密码核时间隐通道的信息泄露,还着重于建立一个基于信息论熵的计算模型评估不同 RSA 密码核时间隐通道的容量. 通过这种方法衡量硬件密码核时间隐通道的脆弱性,为 RSA 密码核的设计提供参考标准.

论文主要贡献:(1)提出基于信息熵的攻击方法攻击 RSA 密码核,并实验验证信息熵的攻击方法优于统计方差方法;(2)引入信息熵理论量化分析 RSA 密码核时间隐通道,评估 RSA 硬件设计的密钥信息泄露,量化分析和推理 RSA 密码核改进架构对时间隐通道安全性的提升,并辅助攻击实验验证其有效性;(3)综合分析 RSA 信息熵指标与性能、硬件资源利用情况,指出设计者可依据信息熵指标完善 RSA 硬件设计,权衡时间隐通道安全性与性能、资源利用;(4)利用信息熵理论量化分析粗粒度离散化抵抗措施对时间隐通道的迁移作用.

本文第 1、2 节介绍硬件 RSA 时间隐通道的研究背景和相关工作;第 3 节描述 RSA 密码核时间隐通道威胁模型;第 4 节详细阐述了基于信息熵的时间隐通道量化分析模型和攻击模型;第 5 节展开实验并进行讨论;第 6 节简明分析本文研究工作与相关工作的区别和改进,指出本文工作贡献和意义;第 7 节总结论文并讨论,指出未来研究工作的重点.

2 相关工作

2.1 面向信息论与系统设计的研究工作

Clark 等人使用信息论量化分析带有循环语句的"While"程序的信息泄露,测量私有变量对公共变量的影响[11];Heusser 和 Malacaria 引入信息论量化分析方法评估 C 代码,表明了在信息泄露方面与 Linux 内核相关的系统软件存在脆弱性[12];Newsome 等人利用信息通道容量测量 x86 二进制程序中输入对输出的影响[13];Doychev 等人测量主流密码算法(AES,eSTREAM 组合)和排序算法(冒泡排序,插入排序,选择排序)二进制可执行程序在底层 Cache 中的信息泄露[14];Standaert 等人使用信息论方法衡量 AES 密码实现的能量隐通道的信息泄露[15];Batina 等人利用互信息作为分类器实现对能量隐通道的攻击[16]. 上述研究大多关注软件层次编程语言、操作系统内核、密码算法设计的量化安全分析,或强调密码算法软硬件实现的能量隐通道信息泄露,并未对软硬件实现的时间隐通道问题进行分析,也缺乏对不同软硬件设计时间隐通道安全性强弱的研究.

2.2 面向软件时间隐通道分析的研究工作

Bernstein 对 AES 软件加密算法在 Cache 中的时间隐通道问题进行了研究,成功破解相关密钥①;

① Bernstein D. Cache-timing attacks on AES. http://cr.yp.to/antiforgery/cachetiming-20050414.pdf, 2005

Aciiçmez 等人利用条件分支预测实现对密码的破解[17-18]；Liu 等人对多核、多虚拟机环境下 RSA 密码算法(Square-and-Multiply 算法和滑动窗口算法)在共享 Cache 中的时间隐通道信息泄露问题进行了探讨[19]. 软件时间隐通道的问题多存在于系统的不确定性行为，如密码算法在 CPU 共享 Cache 中基于缓存命中/错失的时间差异，条件分支结构产生的指令跳转执行时间差异. Köpf 等人利用信息论方法评估 RSA 算法信息泄露的上限，提出利用“bucket”策略减少信息泄露[20-21]和平衡系统性能，对 AES 算法在 ARM 架构 Cache 中信息泄露进行量化分析[22]；Askarov 等人对 OpenSSL 中的 RSA 密码算法进行时间隐通道的迁移和量化分析[23]；Wang 等人利用分区机制抵抗时间隐通道信息泄露[24]. 这些研究中的量化分析方法和时间隐通道抵抗措施集中于 CPU 架构软件层次，并未深入探讨硬件结构设计变化对时间隐通道的影响，也未深入量化分析硬件时间隐通道抵抗措施的有效性.

2.3　面向硬件时间隐通道分析的研究工作

Lee 等人对不同 Cache 硬件结构时间隐通道特性进行量化研究，但其侧重 Cache 设计对 AES 密码算法的影响[5]；Chen 等人在微核层次架构下探测共享处理器硬件中的隐通道信息泄露，共享硬件包括总线、高速缓存和整数除法器等[4]；Zhang 等人则关注硬件设计语言的时间隐通道问题[7]；Oberg 和 Kastner 等人提出的门级信息流追踪安全框架虽然能够检测硬件时间隐通道并隔离 Cache、密码核、片上系统网络，但无法评估硬件设计时间隐通道容量的大小[9,25-26]，无法对不同的硬件设计脆弱性完成量化区分；Li 等人提出的基于 TDMA 的执行租赁(Execution-Lease)策略能够有效排除时间隐通道，但使用的非干扰特性过于严苛，会对系统性能造成严重干扰[8]. 因此，在保证系统时间隐通道安全性前提下，降低非干扰特性的严苛要求，不对系统性能造成过多约束和干扰，寻求一种有效检测和分析硬件密码核时间隐通道的方法是亟需解决的问题.

3　RSA 时间隐通道威胁模型

在片上系统(System-on-Chip，SoC)中，专用硬件模块通常被集成到系统中作加速使用，通过原语被嵌入式系统核访问和调用，例如 RSA、AES 核作为 ARM 核的加解密协处理器. 虽然独立的函数模块可有效加速嵌入式系统运算，但也带来了潜在的隐通道信息泄露问题，例如时间隐通道，攻击者通过测量硬件函数模块执行时间获取硬件函数的重要信息. 例如图 1 所示，以机密性规则为例，高安全级别 IP 核和低安全级别 IP 核都可调用 RSA 密码核进行加解密，并观测到执行时间，当高安全级 IP 核调用密码核加解密时，低安全级 IP 核可通过请求调用密码核的延迟时钟周期数观测到高安全级 IP 核加解密原语的运行时间，进而推断高安全级别 IP 核的密钥信息.

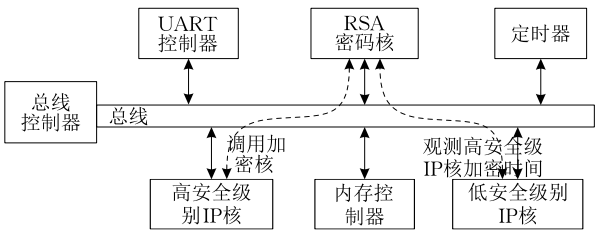

图 1　SoC 威胁模型示意图

本文主要以 RSA 密码核为例研究时间隐通道密钥信息泄露问题，但研究方法可推广应用到其它受系统时钟驱动、与运行时间相关的硬件函数模块. RSA 算法是公钥算法，其加密过程中的密钥是公开的，需要加密的信息是机密信息；而解密过程中的私钥是机密信息，需要解密的密文可以是公开的，可被攻击者获取. 本文研究建立在 RSA 解密过程基础上，密钥是机密信息，使用非选择性密文攻击，密文数据随机输入.

本文假设攻击者能够确定硬件函数模块的执行时间. 硬件函数模块一般基于系统时钟驱动，其硬件接口中包含时序信号端口，攻击者依据时序信号可确定硬件函数操作的起止时钟，如依据图 2 中的开始和结束等时序信号. 同时，假设攻击者可以访问硬件函数模块的输入和输出，例如，可以为图 2 中 RSA 密码核提供不同的密文、密钥等输入并获得结果输出. 因此攻击者能够对给定输入数据的硬件函数模块的执行时间进行测量.

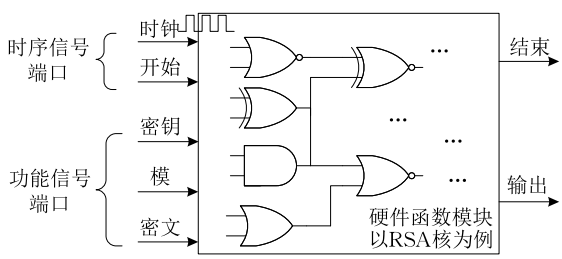

图 2　硬件函数模块时序测量模型

本文实验建立在 FPGA(Field Programming Gate Array)平台基础上，对不同的 RSA 密码核实现进

行评估,通过相应的辅助逻辑电路对 RSA 密码核时序信息处理并收集执行时间,例如图 2 所示,根据 RSA 密码核的开始、结束信号,截取开始结束信号之间的时钟周期的个数表示解密时间长短,对不同密文输入的执行时间进行周期精确的记录和分析.

4 RSA 时间隐通道分析与攻击方法

4.1 RSA 模幂算法和时间隐通道

RSA 算法中的模乘可采用不同算法实现,如移位加算法(Blakley 算法)等. 模幂可采用 Square-and-Multiply 算法实现. 模幂运算中的 IF-ELSE 条件分支语句存在延时差异是造成时间隐通道的根本原因,如算法 1 所示,由于语句 3～4 的模乘运算和语句 5～6 的赋值运算存在明显的时间差异导致不同的密钥会产生不同的解密运算时间. 另外密文、密钥和模都参与模乘运算,对语句 4 和语句 7 中的模乘时间亦造成影响. 上述两种因素是造成 RSA 密码算法时间隐通道信息泄露的重要原因.

算法 1. RSA 模幂算法.

输入:m,e,n(m 为密文,e 为私钥,n 为模)

输出:$c=m^e \bmod n$

1. LET $s_0=m$, $c_0=1$
2. FOR $i=0$ TO $w-1$:
3. IF (bit i OF e) IS 1 THEN
4. LET $c_{i+1}=c_i \times s_i \bmod n$
5. ELSE
6. LET $c_{i+1}=c_i$
7. LET $s_{i+1}=s_i^2 \bmod n$
8. ENDFOR
9. RETURN c_w

4.2 信息熵

本文引入香农信息熵(Shannon Entropy)[27]理论进行时间隐通道的研究. 香农将信息定义为离散随机事件出现的概率. 假设 x 是采样空间 $\boldsymbol{X}$ 中的一个随机变量,p 代表 $\boldsymbol{X}$ 空间中变量的概率分布密度函数,香农信息熵的定义为

$$H(\boldsymbol{X})=-\sum_{x\in\boldsymbol{X}}p(x)\log p(x) \tag{1}$$

信息熵衡量系统不稳定性.

方差是衡量数据集合中变量值偏离期望的程度的参数. 当方差较小时,表示数据分布集中于均值附近,当方差较大时,则数据分布分散. Kocher 在文献[1]中使用方差攻击时间隐通道. 为了与本文中使用的信息熵作对比,在此给出方差的定义:x_1,x_2,…,x_n是变量 $\boldsymbol{X}$ 的采样值,可得方差计算公式:

$$VAR(\boldsymbol{X})=\sum_{i=1}^{n}(x_i-\bar{x})^2 \tag{2}$$

$\bar{x}$ 表示变量 $\boldsymbol{X}$ 的期望值. 下面本文将描述信息熵衡量信息泄露的模型和相应的攻击模型,并在 5.2 节中通过实验比较分析基于信息熵和基于方差的攻击方法.

4.3 时间隐通道量化分析模型

针对某个具体 RSA 核,不同密文、密钥对解密消耗的系统时钟数不同,产生时间隐通道,泄露相关密钥信息;针对不同架构 RSA 核实现时,可能因为应用不同运算方法使得一个密码核的模乘运算比另一个核的模乘运算消耗时钟少,造成不同的密码核对同一个解密测试(相同的密文、密钥对)产生不同运算时间,导致不同密码核时间隐通道存在差异. 基于观测到的事实我们引入定义 1 的描述.

定义 1. RSA 硬件体系结构在密文、密钥对(密钥和模)等输入数据的作用下产生时间信息流.

依据第 3 节中威胁模型和定义 1 建立信息泄露模型. 以机密性为安全规则来定义相关信息:高安全级别的信息包括密钥 K;低安全级别的信息包括 RSA 核的解密时间 T,因为解密时间 T 是攻击者可以观测的. RSA 硬件密码核时间隐通道信息泄露的特征函数用 F 表示,密码核时间隐通道会导致密钥 K 的信息流入到低安全域中. 其特征反映相应 RSA 密码核的时间隐通道的特性.

依据定义 1 以数学形式抽象完整描述 RSA 信息泄露如式(3)所示,表示密钥 K,密文 C 和模 M 的信息通过函数 F 流入到时间域 T 中:

$$T\leftarrow F(K;C;M) \tag{3}$$

RSA 是公钥算法,模 M 的值是公开的,并与私钥 K 成对出现,可将模型简化为

$$T\leftarrow F(K;C) \tag{4}$$

由于密文和密钥分布是离散的,本文使用离散概率密度函数来刻画 RSA 密码核关于密钥 K 的时间隐通道. 假设当密文和密钥确定时($K=k$,$C=c$),$T=t$ 是 RSA 核关于密钥信息泄露的一次采样,即一条密文模幂时间是攻击者对密钥的一次时间隐通道的采样. 那么,刻画时间隐通道特征函数 F 的泄露函数 $L(K,C)$可定义为

$$L(K,C)=\underset{c\in C,k\in K}{p}\{T=t|C,K\} \tag{5}$$

若能够采集所有的实验数据(密文、密钥、时间 T 能够遍历其空间中所有值),理论上可以构造完整精确

的泄露函数 L. 但实际应用中只能选择适当数目的测试向量(一定数目的密文和密钥)来近似估计函数 L. 本文将此估计函数表示为 $\tilde{L}$.

基于估计函数 $\tilde{L}$ 的假设,若密文数目固定(例如指定 5000 条密文),随着采样样本数目的增加,解密时间的概率密度函数将会随着采样样本的变化而变化,并显现密钥的特征. 本文中将其抽象为时间隐通道,并利用信息熵的概念量化通道容量. 将式(5)代入式(1),获得相应信息熵计算公式:

$$H(T)=-\sum_{k\in K,c\in C} p(T=t)\log p(T=t) \quad (6)$$

4.4　时间隐通道攻击模型

Kocher 在建立 RSA 时间隐通道攻击方案时假设每个密钥位的处理时间相互独立[1]. 基于我们对文献[1]中描述的方差攻击过程的观测,当密钥位数较大时,每个密钥位攻击成功率之间的影响微弱. 例如,一个 RSA 密码核密钥是 W 位,前 s 位($s\ll W$)密钥位攻击成功率对后面密钥位攻击成功率影响很小. 本文攻击模型也遵从同样的前提假设.

假设 1.　当密钥位数较大时处理每个密钥位的时间是相互独立的.

本文以 Kocher 的攻击方案为基础,描述基于信息熵的攻击方法. 当攻击者从最低位 LSB(Least Significant Bit)开始攻击 RSA 核时,会逐渐猜测出各个密钥位的值,直到密钥最高位 MSB(Most Significant Bit)被破解.

假设正确的密钥的概率分布密度获得的信息熵为 H,由式(6)计算获得;由已猜测密钥位及其处理时间产生的概率分布密度获得的信息熵表示为 Hs,$\hat{T}_s$ 表示前 s 位已猜测密钥位的处理时间,则

$$Hs(T)=-\sum p(T=\hat{T}_s)\log p(T=\hat{T}_s) \quad (7)$$

考虑到单个密钥位处理时间可近似相互独立,将未猜测密钥位的值及其处理时间也描述为一个概率分布密度函数并以熵 ΔH 的形式来表述其特征,$\Delta\hat{T}$ 表示余下未猜测密钥位的运算时间,则

$$\Delta H(T)=-\sum p(T=\Delta\hat{T})\log p(T=\Delta\hat{T}) \quad (8)$$

随着密钥位被逐渐破解,熵 ΔH 逐渐减小,如果所有密钥比特位都猜测正确,ΔH 最终减少至 0. 但如果存在密钥比特位猜测错误,ΔH 不能减少至 0. 基于此事实分析,我们提出定理 1.

定理 1.　猜测正确的密钥位将使 ΔH 减小,并逐渐减少至 0.

依据定理 1,当 Hs 趋近于 H 时,ΔH 趋近于 0,可得

$$\text{当 } Hs\to H \text{ 时},\Delta H\to 0$$
$$\Delta H\propto H-Hs \quad (9)$$

本文采用的攻击步骤算法流程如下:

过程 1.　信息熵攻击过程.

1. 假设 $\boldsymbol{T}$ 表示使用正确密钥处理 n 条密文消耗时间的向量,T_j 表示使用正确密钥处理第 j 条密文的时间,则

$$\boldsymbol{T}=\{T_1,T_2,\cdots,T_n\} \quad (10)$$

2. 假设前 $s-1$ 位密钥位已被猜出,需要猜测第 s 位密钥位时:处理第 j 条密文,当密钥第 s 位为 0 时,密钥前 s 位处理时间是 $\hat{T}_{sj}(k_s=0)$,当密钥第 s 位是 1 时,其前 s 位处理时间是 $\hat{T}_{sj}(k_s=1)$. 可计算:

$$\Delta\hat{T}_{0j}=T_j-\hat{T}_{sj}(k_s=0) \quad (11)$$

$$\Delta\hat{T}_{1j}=T_j-\hat{T}_{sj}(k_s=1) \quad (12)$$

考虑到共有 n 条密文,则

$$\Delta\hat{\boldsymbol{T}}_0=\{\Delta\hat{T}_{01},\Delta\hat{T}_{02},\cdots,\Delta\hat{T}_{0n}\} \quad (13)$$

$$\Delta\hat{\boldsymbol{T}}_1=\{\Delta\hat{T}_{11},\Delta\hat{T}_{12},\cdots,\Delta\hat{T}_{1n}\} \quad (14)$$

3. 依据定理 1 和式(9),ΔH 越小,则该密钥位越可能被猜测正确. 若 $\Delta\hat{\boldsymbol{T}}_0$ 的熵 $\Delta H(\Delta\hat{\boldsymbol{T}}_0)$ 比 $\Delta\hat{\boldsymbol{T}}_1$ 的熵 $\Delta H(\Delta\hat{\boldsymbol{T}}_1)$ 小,则密钥 s 位的值为 0;反之,密钥 s 位值为 1.

4. 重复步骤 2、3,直至攻击至密钥最高位.

另外本文直接测量系统时钟周期数,已经内在排除了由于时间测量误差引起的噪声. 影响密钥破解的因素集中于未知密钥位引起的时间变化,这也是如果所有密钥位都猜测正确,熵 ΔH 会减少至 0 的原因. 本文将在实验部分详细描述信息熵在 RSA 时间隐通道攻击方面较方差具备更好的特性.

5　实验结果分析和讨论

5.1 节简要介绍实验设计框架. 为检验硬件 RSA 密码核时间隐通道安全性和攻击模型的有效性,5.2 节利用信息熵攻击 RSA 密码核,并实验验证基于信息熵的方法优于基于方差的攻击方法. 为验证量化分析模型的有效性,5.3 节首先利用信息熵量化分析同一密码核不同密钥的信息泄露;然后量化不同 RSA 架构的信息泄露,并辅助以攻击实验进行佐证;最后综合评估 RSA 密码核设计复杂度. 5.4 节利用信息熵量化分析时间隐通道抵抗措施对 RSA 密码核时间隐通道信息泄露的迁移作用. 实验结果表明了信息熵量化分析对 RSA 密码核安全设计工作的改进和意义. 设计者可依据信息熵指标完善 RSA 硬件安全设计.

5.1 实验设计

为了对 RSA 密码核中时间隐通道进行评估，本文使用 Xilinx ML605 评估板构建硬件测试平台. 整个实验环境由两部分组成：FPGA 端和主机端测试程序. 具体框架如图 3 所示.

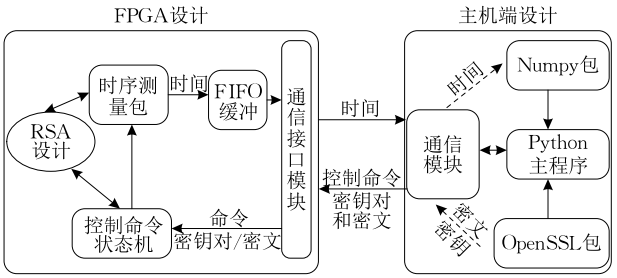

图 3 RSA 时间隐通道测试框架

FPGA 端程序包括可综合的 RSA 设计和相应的控制逻辑，主要完成 RSA 计算时间的测量、存储和传输. RSA 核选用两种不同方案产生. 第一种方案依据 4.1 节中 RSA 密码算法的时间隐通道特征，设计产生五种不同的 RSA 架构，分别检验模乘器时间差异和条件分支语句时间差异对时间隐通道特征的影响；第二种方案采取时间离散化抵抗措施产生五种不同粒度的 RSA 时间隐通道的迁移设计.

主机测试程序主要完成两项功能：第一、向 FPGA 发送测试向量(密文、密钥、模)，从 FPGA 读取所测量的 RSA 运算时间；第二、对相应 RSA 核进行攻击和量化分析. 主机端程序模拟片上系统中可以访问密码核的处理器运算.

5.1.1 基于 Xilinx ML605 的板级实现

依据第 3 节中时序测量模型的描述，我们在 FPGA 端设计并实现对 RSA 解密时间的记录、保存和传输. RSA 密码核的控制逻辑包括控制命令状态机模块，RSA 时序测量包模块和通信接口模块. 控制命令状态机完成通信接口模块提供的命令和数据的编解码，为 RSA 密码核提供密文、密钥和模值信息. 当控制命令状态机完成对密文、密钥和模的解析转化后，若遇到主机端来的开始解密命令，则会通知 RSA 时序测量包启动 RSA 核开始进行第一条密文的解密. 由于量化信息泄露的过程和密钥攻击过程都需要大量的密文，实验中利用块存储器来存储数据，控制命令状态机同时维护块存储器的数据存取.

RSA 时序测量包模块主要完成每个 RSA 解密操作的时间记录和控制，例如控制 RSA 解密操作启动和停止的时间，需要的解密操作个数(即解密多少条密文)以及每个解密操作的时钟周期数的记录存储等.

通信接口模块完成 FPGA 端 RSA 时间数据的发送和主机端发来的命令和数据的接收. 这些命令和数据经过控制命令状态机进行编解码，其具体的命令和数据格式定义如表 1 所示. 由于解密速度可能快于通信传输速度，为防止时间数据被覆盖而导致数据丢失，使用 Xilinx 的 FIFO(First-In-First-Out) IP 核以缓存数据.

表 1 控制命令和数据命令格式定义

命令类型	命令编码	数据类型	数据名称
开始命令	0xAA		
结束命令	0x00		
数据命令	0x05	0xA5	密文
数据命令	0x05	0x55	密钥
数据命令	0x05	0x5A	模

5.1.2 主机端实现

本文使用 Python 脚本实现主机端主程序设计，包括利用 OpenSSL0.9.8h 程序产生测试向量(RSA 密钥和模)，利用 Python 的内置随机函数库产生随机密文数据，使用 Python 的科学计算包 Numpy 进行统计方差和信息熵的计算，利用 Python 的 PySerial 构建通信接口模块.

主机端程序主要完成两项计算任务：第一，给定密钥情况下，采集多条密文的完整解密时间，进而利用 4.3 节信息熵量化方法测量 RSA 密码核的信息泄露；第二，持续向 FPGA 发送测试向量，收集相应的时间信息进行计算，采用 4.4 节中的攻击步骤(过程 1)完成攻击(Kocher 采用基于统计方差的方法[1]，本文使用信息熵的攻击方法).

5.2 利用信息熵攻击 RSA 密码核

在此实验中，本节对 Blakley 算法实现的密码核进行攻击，使用 5000 个密文测试 50 个不同密钥对. 如图 4 所示为使用方差、信息熵攻击 RSA 密码核的成功率.

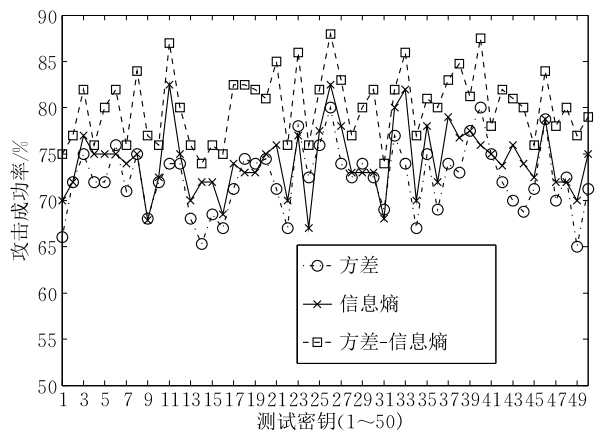

图 4 使用方差、信息熵攻击 RSA 结果

基于对攻击过程的观察，随着猜测密钥位的逐渐增加，ΔH 呈现逐渐减小的趋势，假设猜测第一位之后，ΔH 值为 ΔH_1，猜测第二位之后 ΔH 值为 ΔH_2，…，可得

$$\nabla H_i = \Delta H_i - \Delta H_{i+1} \tag{15}$$

当 ∇H_i 值较大时，第 i 位密钥位猜测正确的可能性就很高，而当 ∇H_i 很小时，通常意味着第 i 位密钥位猜测错误. 为衡量对每个密钥位猜测正确性的确信度，本文首先引入方差和熵的猜测信心域参数 $\%V$ 和 $\%H$：

$$\%V = \frac{1}{2} \times \frac{\nabla Var_i}{\max(\nabla Var_i)} \times 100\% + 50\% \tag{16}$$

$$\%H = \frac{1}{2} \times \frac{\nabla H_i}{\max(\nabla H_i)} \times 100\% + 50\% \tag{17}$$

将信心域参数值平均分成 10 个区间，即[50%，55%]，[55%，60%]，…，[95%，100%]. 统计对应区间下，一共存在的密钥位个数 N，以及猜测正确密钥位的个数 Nr. 获取确信度的计算公式：

$$D = \frac{Nr}{N} \times 100\% \tag{18}$$

绘制相关曲线如图 5 所示，例如，[55%，60%]区间中的 Entropy-Local 的节点表示由 ∇H_i 的值获得的 $\%H$ 值在区间[55%，60%]中，则攻击者有 81%的确信度认为此密钥位的猜测是正确的. 而 Variance-Local 在[55%，60%]区间中的确信度几乎为 0. 当 Entropy-Global 信息熵的信心域参数达到[65%，70%]时，其猜测正确率确信度达到 100%，而方差 Variance-Global 的信心域参数需达到[75%，80%]，其猜测正确率确信度达到 100%. 局部(Local)的 $\max(\nabla H_i)$ 中的 ∇H_i 本文使用测量的单个指定密钥的所有密钥位 ∇H_i；全局(Global)的 $\max(\nabla H_i)$ 中的 ∇H_i 文中使用测量的 100 个密钥所有密钥位的 ∇H_i. 方差分析亦然.

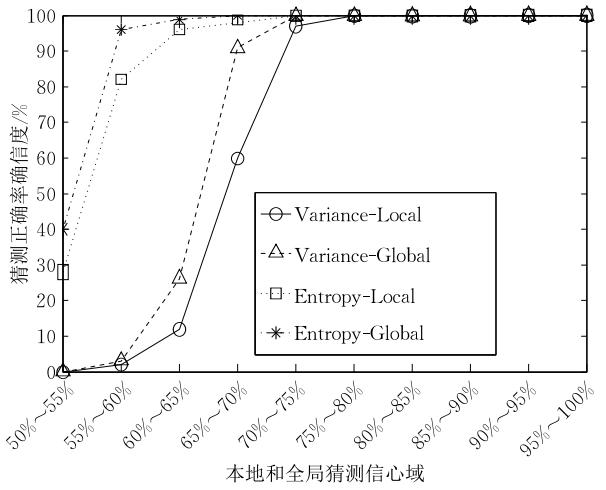

图 5　方差、信息熵攻击成功率信心参数

由图 4 实验结果，基于熵的攻击方法与基于方差的攻击方法具备相当的密钥破解能力，图 5 实验结果表明基于熵的攻击方法在全局、局部猜测信心域参数和猜测正确率确信度方面较方差具备更好的性能. 图 4 结果的方差-信息熵曲线表明同时利用信息熵和方差攻击相应密钥，能显著提高密钥的破解成功率.

5.3　信息熵量化 RSA 密码核信息泄露应用分析

5.3.1　信息熵量化 RSA 密码核不同密钥信息泄露

本节实验使用 Blakley 算法 RSA 核，随机测试 30 个不同的密钥. 为更好佐证密钥信息泄露，实验对相应的密钥进行实际攻击. 使用 4.3 节中的信息熵量化分析方法衡量密钥信息的泄露，并利用 4.4 节中的攻击方法(过程 1)攻击对应的 30 个密钥，结果如图 6 所示.

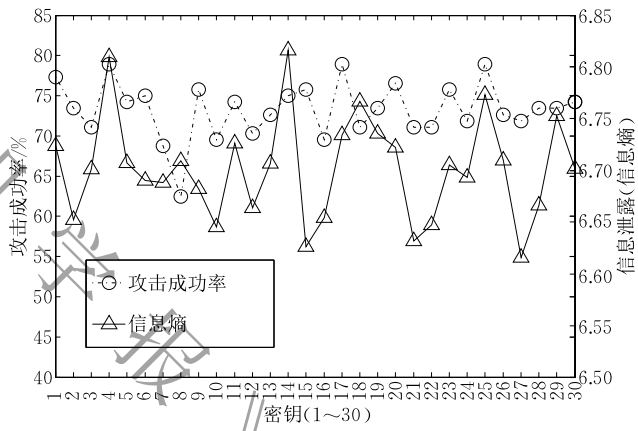

图 6　信息熵量化 RSA 信息泄露和攻击结果

实验结果表明，由信息熵量化的信息泄露能够较好的反映不同密钥导致的时间隐通道泄露. 当信息熵较大时，攻击成功率明显增加；而信息熵较小时攻击成功率则呈现减小趋势.

5.3.2　信息熵量化不同 RSA 密码核信息泄露

本节实验使用五个不同的 RSA 密码核，其中三个采用不同模乘算法，分别是 Blakley 算法[28]、Barrett 算法[29]、Montgomery 算法[30]. 另外一种设计采用 Power-Ladder 模幂算法[31]，是基本模幂算法的一种改进设计用以平衡条件分支语句执行时间. 第五种设计是一个改进的并行 RSA 密码核①(两个模乘运算器并行执行，后文以 Parallel 设计表示). 由于前四个密码核结构中的模乘和模平方操作

① http://opencores.org/project,basicrsa

使用同一个模乘器，模乘和模平方操作在时序上交替进行. 而 Parallel 设计利用两个模乘器，模乘和模平方可以并行执行.

同样利用 4.3 节中的信息熵量化密钥信息的泄露，利用 4.4 节中的攻击方法（过程 1）攻击相应 RSA 密码核，结果如图 7 所示. Blakley 算法密码核是五种设计中攻击成功率最高的，其信息熵也是五种设计中最大的. Barrett 算法设计的攻击成功率和信息熵比 Blakley 设计都要减小. Power-Ladder 设计与 Barrett 设计攻击成功率和信息熵相近，Power-Ladder 设计信息熵和攻击成功率较 Blakley 设计显著减小. 而 Montgomery 设计和 Parallel 设计则是五种设计中时间隐通道安全性最高的，二者的攻击成功率相同，但是 Parallel 设计的信息熵较 Montgomery 设计的信息熵减小.

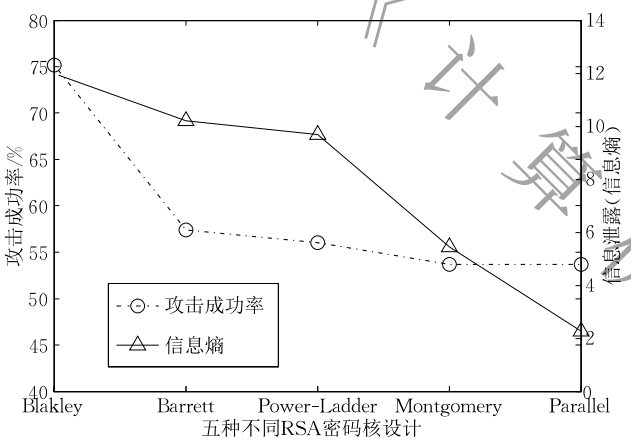

图 7 五种 RSA 设计信息熵和攻击结果

具体分析五种 RSA 密码核结构，Blakley 设计解密时间受到密钥和模乘运算的影响变化很大，实现 Blakley 算法的模乘器对每个密钥位的处理时间随密文变化显著. 而 Barrett 设计在实现模乘器时首先进行乘法运算，然后进行模运算，但所用的乘法器运算时间固定，只有做模运算时才会有时间变化，其模乘时间随密文变化较小. 因此 Barrett 设计信息熵和攻击成功率较 Blakley 设计减小. Power-Ladder 设计将模平方操作移入条件分支语句内，并改进算法流程，使每次密钥位的处理都要实施模乘和模平方的计算，有效减小了因为密钥位值的不同引起的 IF-ELSE 条件分支语句的延时差异. 如图 7 所示，其时间隐通道安全性获得提高. Montgomery 设计的模乘的时间只随着模变化，对不同密文的解密时间是不变的，只有当密钥发生变化时，解密时间才会改变. 而 Parallel 设计利用算法 1 模幂算法的特殊性（在同一个密钥位的处理时间范围内，模乘和模平方运算之间不存在数据依赖关系；但在相邻密钥位处理时，模乘和模平方运算存在数据依赖关系）和 FPGA 资源丰富的特点（可在空间上增加模乘器数量）实现模乘器的并行性运算，而做模平方运算的模乘器时间总是多于或者等于做模乘运算的模乘器时间，且时间趋于一致，解密时间几乎完全由密钥的最高位位置决定，解密时间受密文影响变化微弱. Montgomery 设计和 Parallel 设计时间隐通道特征的共同之处在于二者的解密时间都（几乎）不受密文的影响. 不同之处在于，Montgomery 设计的解密时间会随着密钥的变化而变化，不仅受到密钥最高位位置的影响，还受到其他密钥位的影响；而 Parallel 设计的解密时间受到密钥最高位位置影响显著. 在攻击 RSA 核时，主要利用不同密文解密的时间差异获得的信息熵破解密钥，因此，若攻击 Montgomery 设计，当猜测密钥的密钥位值给定时，所有密文解密时间都相同. 此时攻击过程 1 中步骤 3 密钥位 0 值获得的熵和密钥位 1 值的熵都等于 0，攻击者无法判断密钥位取 0 值或取 1 值；当攻击 Parallel 设计时，所有密钥位猜测的值都为 1. 基于熵和方差的攻击方法对这两种设计是失效的. 在量化分析 RSA 核时，信息熵量化分析的计算包含了不同密钥引起的解密时间的变化，导致 Parallel 设计的信息熵较 Montgomery 设计减小，而两种设计的信息熵都不会减小为 0. 但 Montgomery 设计的密钥依旧面临被破解的威胁. 例如，使用暴力破解方法找到猜测密钥，使其解密时间与正确密钥解密时间相同即可. 而 Parallel 设计即使使用暴力破解方法依然无法利用时间隐通道破解完整的密钥，因其解密时间几乎完全由密钥的最高位 MSB 位置确定，确实只有密钥最高位才能被准确破解.

图 7 实验结果表明，信息熵量化分析结果完全符合相关 RSA 设计的时间隐通道密钥信息泄露的特征：Barrett 设计和 Montgomery 设计通过减小或消除模乘器运行时间的差异以减少时间隐通道信息泄露；Power-Ladder 设计通过减小条件分支语句执行时间差异以减少密钥信息泄露；Parallel 设计通过增加并行性操作减少信息泄露. 实验结果亦显示即使在已知攻击方法不能破解相关密钥情况下，信息熵量化分析亦可揭露 RSA 密码核潜在的密钥信息泄露的风险（如 Montgomery 设计）. 信息熵可有效量化分析 RSA 改进设计对时间隐通道的影响.

5.3.3 权衡信息熵与性能、资源利用率之间的关系

上述实验结果显示信息熵可以量化不同密钥信

息泄露的程度和不同 RSA 密码核密钥信息泄露程度，信息熵可以作为量化 RSA 密码核时间隐通道的指标. 为了更加完整评估密码核设计，本文利用 Xilinx ISE 综合相关设计，获得相应的资源利用状况，如表 2 所示. 并收集各个密码核的平均解密时间，如表 3 所示. 实验结果显示，Parallel 设计性能最高，平均每个解密操作需要 21896 个时钟周期，但是由于使用两个模乘器，消耗板上资源接近 Blakley 设计的两倍. Blakley 设计消耗资源最少，但是解密速度减小. Barrett 设计消耗最多的资源却只达到 Blakley 设计的速度. Power-Ladder 设计消耗资源比 Blakley 设计较多，性能却远低于 Blakley 设计，仅达到 Blakley 设计的 2/3. Montgomery 设计资源利用过多而性能只达到 Blakley 设计的一半.

表 2　FPGA 资源利用情况

	LUT	Slice register	DSP48E	Occupied Slices	LUT-FF pairs
Blakley	1683	1710	0	707	2099
Barrett	11944	5548	340	3511	13327
Power-Ladder	2400	1949	0	685	2524
Montgomery	5969	4882	0	1915	6325
Parallel	3484	1803	0	1156	3731

表 3　RSA 密码核平均运行时间

不同设计	平均时钟周期数
Blakley 设计	46292
Barrett 设计	46814
Power-Ladder 设计	62438
Montgomery 设计	95383
Parallel 设计	21896

我们将时间隐通道安全性纳入评估内容，并以 Blakley 密码算法核为参照. Power-Ladder 设计仅在损失性能情况下降低 RSA 密码核信息泄露. Barrett 设计同时损失资源与性能降低 RSA 时间隐通道信息泄露. Montgomery 设计在损失性能和资源的条件下获得了隐通道的安全性. 而 Parallel 设计则在只损失硬件资源的条件下，达到了性能和时间隐通道安全性的提高. 信息熵量化分析模型的指标可为设计者提供时间隐通道安全性的度量标准，如同资源利用率量化逻辑资源使用情况、时钟周期数量化加解密速度的性能. 设计者可根据信息熵指标评价或改善 RSA 密码核设计，实现资源利用、性能和时间隐通道安全性最佳权衡和取舍.

5.4　RSA 密码核时间隐通道抵抗措施下信息熵量化应用分析

5.3 节实验利用信息熵量化分析不同 RSA 硬件密码核结构的时间隐通道. 实验结果表明，在没有任何 RSA 时间隐通道抵抗措施的情况下，信息熵能够有效量化分析硬件架构对时间隐通道的影响. 本节将引入时间隐通道抵抗措施，使用信息熵量化分析方法研究时间隐通道迁移设计的信息泄露.

RSA 时间隐通道表现为密码核模幂运算的时间差异. 一种通用而简易的抵抗措施是在 RSA 模幂运算末尾引入空操作，增加单个模幂运算的时间延迟，减少甚至消除不同模幂操作的执行时间差异. 文献[20]中提出一种粗粒度离散化的添加空操作的时间延迟方案：例如，将原来以一个时钟周期进行计数的时间离散化为使用 10 个或者 100 个时钟周期进行阶梯式计数的时间. 例如，当使用 100 个时钟周期进行阶梯计数时，原始模幂时间是 46187 个时钟周期，而加入空操作离散化后的模幂时间为 46200. 推广到极端情况，所有的模幂时间都采用最大值，即所有模幂运算时间都为恒定值，则完全消除时间隐通道，但是因为时间都采用最大值，模幂运算效率低.

本节采用 Blakley 设计为基准 RSA 密码核完成相关时间隐通道抵抗措施的实验. 抵抗措施采用粗粒度离散化模幂时间延迟方法，在基准设计的基础上再增加五种不同粒度的离散化设计：Discrete50 设计、Discrete75 设计、Discrete100 设计和 Discrete150 设计分别采用的时间延迟阶梯间隔为 50、75、100 和 150 个时钟周期，Constant 设计则将所有模幂运算时间延迟到最大值. 然后利用 4.3 节中的信息熵量化分析方法测量密钥信息泄露，利用 4.4 节中的攻击过程 1 攻击相应 RSA 密码核，结果如图 8 所示.

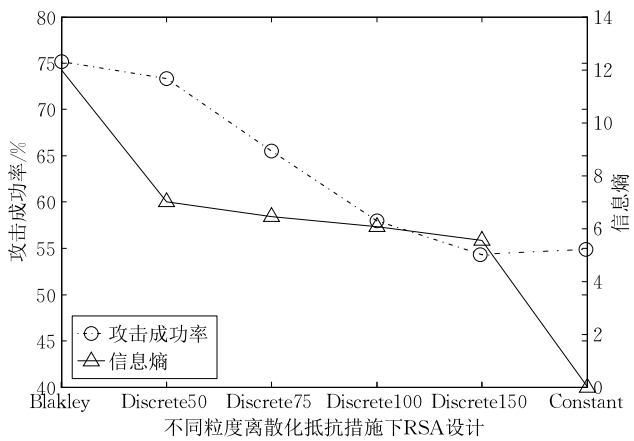

图 8　隐通道抵抗措施下 RSA 信息熵和攻击结果

图 8 实验结果显示，采用粗粒度离散化方案，随着离散化的时间间隔增大，信息熵和攻击成功率呈现下降的趋势. Constant 设计中信息熵结果

显示为 0,表征其不存在时间隐通道.实验结果表明信息熵可有效量化分析时间隐通道抵抗措施减少时间侧信道信息泄露的作用.同时在表 4 中列出相应密码核平均运行时间并进行简要分析:Discrete50 设计性能略微减小,但时间隐通道安全性获得改善;Discrete100 设计和 Discrete150 设计性能较 Discrete50 和 Discrete75 设计略有减小,而时间隐通道安全性获得大幅提高.Constant 设计是粗粒度离散化设计中性能最低但安全度最高的,实现了非干扰特性.

表 4 RSA 密码核平均运行时间

不同设计	平均时钟周期数
Blakley 设计	46 292
Discrete50 设计	46 350
Discrete75 设计	46 375
Discrete100 设计	46 400
Discrete150 设计	46 450
Constant 设计	51 729

6 与硬件时间隐通道工作的对比

为了更加清晰和直观的对比相关工作,本文从实验场景,分析方法和体系结构层次等方面进行了概括和分析,如表 5 所示.Myers 等人应用信息流分析方法从硬件设计语言迁移时间隐通道[7].Lee 等人利用信息论方法量化分析和验证改进的 Cache 设计减少 AES 密钥信息泄露[5].Chen 等人对共享硬件结构(除法器、高速缓存、总线)进行事件相关性分析以检测隐通道[4].

表 5 本文与相关工作对比

	目标/实验场景	分析精度/方法	体系结构层次
Myers[7]	Language 设计	信息流分析	硬件语言
Chen[4]	Bus,Divider,Cache 隐通道检测	事件相关性分析	微体系架构
Lee[5]	Cache 设计	量化分析	微体系架构
GLIFT[9,25]	时序逻辑电路	信息流定性分析	门级电路
本文	协处理器 RSA 核	量化分析	SoC 结构

与本文工作最密切的是能够检测所有硬件设计时间隐通道的门级信息流追踪的分析方法.因为本文除 Constant 设计外其余所有 RSA 设计都存在时间隐通道问题,能够被 GLIFT 全部检出.但是本文提出的分析方法能够有效量化鉴别相关设计的时间隐通道的危害大小,可以更加准确有效的辅助设计者在利用门级信息流分析技术基础上实现加解密硬件关于时间隐通道的安全设计.

7 结束语和讨论

本文研究了硬件 RSA 解密实现中的时间隐通道安全性问题,阐述了基于信息熵的方法在量化分析 RSA 硬件密码核时间隐通道中的应用,通过对不同 RSA 硬件结构进行量化分析和攻击测试,实验验证了信息熵量化分析方法的有效性.本文所提方法为 RSA 密码核时间隐通道的评估提供了一种有效的量化分析的理论依据和测试手段.

本文引入信息熵的攻击方法攻击硬件 RSA 时间隐通道,实验结果表明信息熵的攻击方法要优于基于方差的攻击方法.同时利用信息熵和方差的攻击方法能显著提高攻击的成功率.虽然攻击结果的成功率和量化分析结果都能够对硬件 RSA 的时间隐通道特征进行一定程度的定量化描述,但是攻击过程比量化分析过程消耗的时间有两个数量级的差异,量化分析方法更具效率.同时,信息熵量化分析方法能够有效的揭露 RSA 密码核潜在的时间隐通道风险(已有攻击方法无效时),并为非干扰特性(信息熵为 0)提供辅助的理论性证明,量化分析方法更具效果.

通过 RSA 密码核性能、资源利用和时间隐通道安全性关系的分析,设计者可依据信息熵指标改善 RSA 密码核设计,权衡和取舍密码核设计性能、资源利用和时间隐通道安全性.

本文 RSA 密码核只涉及基本 RSA 设计,近些年提出的高级模幂算法实现,如基于滑动窗口的 RSA 算法,基于中国剩余定理的 RSA 算法,基于掩码的 RSA 设计等需要采取不同于本文的攻击方案,虽然可以应用信息熵进行量化分析,但是由于现有攻击方案有限而且标准不统一[2-3,32],其攻击成功率无法与本文中的攻击成功率(攻击难易程度)进行统一的对比.即不同的攻击方案只针对特定一种或者几种 RSA 算法设计有效,并有效利用相应 RSA 设计的时间隐通道特征破解其密钥.如何将基于信息熵的量化分析方法推广应用到高级模幂算法,并建立衡量攻击难易程度的统一的攻击框架以验证量化分析的有效性是后续研究的一个重要探索方向.同时,本文的量化分析模型虽然只应用到 RSA 密码核设计,但是对其他涉及时间隐通道问题的算法如椭圆曲线密码算法、数字签名标准(Digital Signature Standard,DSS)等同样具有借鉴的意义,也是未来研究工作的重要方向.

致　谢　衷心感谢审稿专家和编辑部老师提出的宝贵意见和建议.同时衷心感谢加州大学圣迭戈分校计算机科学与工程系 Ryan Kastner 教授关于实验设计、分析的讨论和建议!

参　考　文　献

[1] Kocher P C. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems//Koblitz N eds. Advances in Cryptology-CRYPTO'96. Heidelberg, Germany: Springer, 1996: 104-113

[2] Schindler W. A timing attack against RSA with the chinese remainder theorem//Koç Ç K, Paar C eds. Cryptographic Hardware and Embedded Systems-CHES 2000. Heidelberg, Germany: Springer, 2000: 109-124

[3] Brumley D, Boneh D. Remote timing attacks are practical. Computer Networks, 2005, 48(5): 701-716

[4] Chen J, Venkataramani G. CC-Hunter: Uncovering covert timing channels on shared processor hardware//Proceedings of the 47th Annual IEEE/ACM International Symposium on Microarchitecture. Cambridge, UK, 2014: 216-228

[5] Zhang T, Lee R B. New models of cache architectures characterizing information leakage from cache side channels//Proceedings of the 30th Annual Computer Security Applications Conference. New Orleans, USA, 2014: 96-105

[6] Ciet M, Neve M, Peeters E, et al. Parallel FPGA implementation of RSA with residue number systems-can side-channel threats be avoided?//Proceedings of the 2003 IEEE 46th Midwest Symposium on Circuits and Systems. Cairo, Egypt, 2003: 806-810

[7] Zhang D, Wang Y, Suh G E, Myers A C. A hardware design language for timing-sensitive information-flow security//Proceedings of the 20th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS2015). Istanbul, Turkey, 2015: 503-516

[8] Li X, Kashyap V, Oberg J K, et al. Sapper: A language for hardware-level security policy enforcement. ACM SIGARCH Computer Architecture News, 2014, 42(1): 97-112

[9] Oberg J, Meiklejohn S, Sherwood T, et al. Leveraging gate-level properties to identify hardware timing channels. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2014, 33(9): 1288-1301

[10] Goguen J A, Meseguer J. Security policies and security models//Proceedings of the 1982 IEEE Symposium on Security and Privacy. Oakland, USA, 1982: 11-11

[11] Clark D, Hunt S, Malacaria P. Quantified interference for a while language. Electronic Notes in Theoretical Computer Science, 2005, 112: 149-166

[12] Heusser J, Malacaria P. Quantifying information leaks in software//Proceedings of the 26th Annual Computer Security Applications Conference. Austin, USA, 2010: 261-269

[13] Newsome J, McCamant S, Song D. Measuring channel capacity to distinguish undue influence//Proceedings of the ACM SIGPLAN Fourth Workshop on Programming Languages and Analysis for Security (PLAS'2009). Dublin, Ireland, 2009: 73-85

[14] Doychev G, Köpf B, Mauborgne L, et al. CacheAudit: A tool for the static analysis of cache side channels. ACM Transactions on Information and System Security, 2015, 18(1): 4

[15] Standaert F X, Malkin T G, Yung M. A unified framework for the analysis of side-channel key recovery attacks//Joux A eds. Advances in Cryptology-EUROCRYPT 2009. Heidelberg, Germany: Springer, 2009: 443-461

[16] Batina L, Gierlichs B, Prouff E, et al. Mutual information analysis: A comprehensive study. Journal of Cryptology, 2011, 24(2): 269-291

[17] Aciiçmez O, Koç Ç K, Seifert J P. On the power of simple branch prediction analysis//Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security (ASIACCS). Singapore, 2007: 312-320

[18] Aciiçmez O, Koç Ç K, Seifert J P. Predicting secret keys via branch prediction//Abe M eds. Topics in Cryptology-CT-RSA 2007. Heidelberg, Germany: Springer, 2007: 225-242

[19] Liu F, Yarom Y, Ge Q, et al. Last-level cache side-channel attacks are practical//Proceedings of the 2015 IEEE Symposium on Security and Privacy. San Jose, USA, 2015: 605-622

[20] Köpf B, Dürmuth M. A provably secure and efficient countermeasure against timing attacks//Proceedings of the 22nd IEEE Computer Security Foundations Symposium. Port Jefferson, USA, 2009: 324-335

[21] Köpf B, Basin D. An information-theoretic model for adaptive side-channel attacks//Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS'07). Alexandria, USA, 2007: 286-296

[22] Köpf B, Mauborgne L, Ochoa M. Automatic quantification of cache side-channels//Proceedings of the 24th International Conference on Computer Aided Verification. Berkeley, USA, 2012: 564-580

[23] Askarov A, Zhang D, Myers A C. Predictive black-box mitigation of timing channels//Proceedings of the 17th ACM Conference on Computer and Communications Security. Chicago, USA, 2010: 297-307

[24] Wang Y, Ferraiuolo A, Zhang D, et al. SecDCP: Secure dynamic cache partitioning for efficient timing channel protection//Proceedings of the 53rd Annual Design Automation Conference (DAC). Austin, USA, 2016: 74

[25] Hu W, Mu D, Oberg J, et al. Gate-level information flow tracking for security lattices. ACM Transactions on Design Automation of Electronic Systems, 2014, 20(1): 2

[26] Kastner R, Hu W, Althoff A. Quantifying hardware security using joint information flow analysis//Proceedings of the 2016 Design, Automation & Test in Europe Conference & Exhibition (DATE). Dresden, Germany, 2016: 1523-1528

[27] Shannon C E. A mathematical theory of communication. ACM SIGMOBILE Mobile Computing and Communications Review, 2001, 5(1): 3-55

[28] Blakely G R. A computer algorithm for calculating the product AB modulo M. IEEE Transactions on Computers, 1983, 32(5): 497-500

[29] Barrett P. Implementing the rivest shamir and adleman public key encryption algorithm on a standard digital signal processor //Odlyzko A M eds. Advances in Cryptology-CRYPTO'86. Heidelberg, Germany: Springer, 1986: 311-323

[30] Montgomery P L. Modular multiplication without trial division. Mathematics of Computation, 1985, 44(170): 519-521

[31] Joye M, Yen S M. The montgomery powering ladder// Kaliski B S, Koç Ç K, Paar C eds. Cryptographic Hardware and Embedded Systems. Heidelberg: Springer, 2002: 291-302

[32] Schindler W. Exclusive exponent blinding is not enough to prevent any timing attack on RSA. Journal of Cryptographic Engineering, 2016, 6(2): 101-119

MAO Bao-Lei, born in 1987, Ph. D. candidate. His current research interests include hardware security and information flow analysis of hardware design.

HU Wei, born in 1982, Ph. D., associate professor. His current research interests include hardware security, reconfigurable computing and embedded systems.

MU De-Jun, born in 1963, Ph. D., professor. His current research interests include information security and control system design.

ZHANG Hui-Xiang, born in 1981, Ph. D., associate professor. His current research interests include information security and pattern recognition.

TAI Yu, born in 1982, Ph. D. candidate. His current research interests include hardware security, logic synthesis and optimization.

HONG Liang, born in 1979, Ph. D., associate professor. His current research interests include information security and embedded system applications.

Background

Runtime for computing cryptographic functions can reveal a significant amount of information about secret data. Attackers can retrieve the secret key by observing through runtime measurements. Researchers look into the timing channels in two different directions. One way is to detect the timing channel by attacking the design. Another more important branch aims to reduce such leakage or protect it from attacks. Most research work focuses on programming language and mitigation algorithm design. Designers usually ignore timing channel during hardware architecture implementation.

Our research group has been focusing on building secure hardware architectures recent years using information flow tracking method. Although gate level information flow tracking (GLIFT) is efficient for detecting timing channel leakage qualitatively, it is not able to quantify how much the timing-based information flows is if any. We propose to measure this timing channel leakage using information theoretic methods to provide clues of security about different hardware implementations.

This paper focuses on timing channel leakage of RSA cryptographic cores. We measure this timing-based information flows leakage using entropy theory and verify the effectiveness of our proposed measurement by attacking the corresponding RSA architectures. This initial work indicates that information theoretic method can provide a kind of timing channel security guarantee, which will motivate the employment of information theoretic measures for secure hardware design.

Our previous work on information flow security have been published in Electronic Design Automation (EDA) and information security journals and conferences such as *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *IEEE Transactions on Information Forensics and Security*, *ACM Transactions on Design Automation of Electronic Systems*, *Design Automation Conference*, *Design Automation & Test in Europe Conference & Exhibition*, *International Conference on Computer-Aided Design* and *IET Information Security*, etc.

This work was supported by the National Natural Science Foundation of China under Grant No. 61672433, the Basic Research of Shenzhen Science and Technology Innovation Committee under Grant No. 201703063000517, the National Cryptography Development Fund under Grant No. MMJJ20170210, and the Fundamental Research Funds for the Central Universities under Grant No. 3102017OQD094.