



















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.

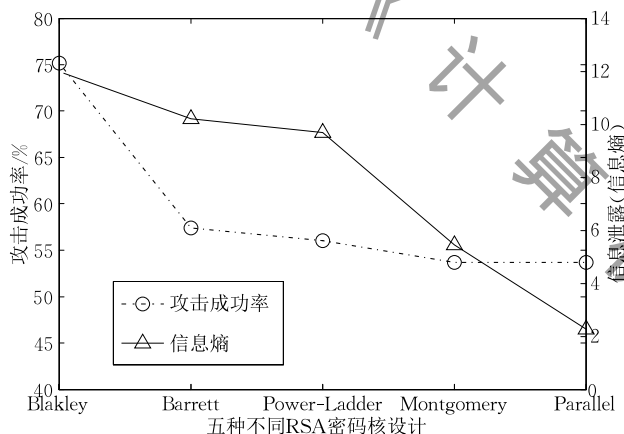


图 7 五种 RSA 设计信息熵和攻击结果

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	11 944	5548	340	3511	13 327
Power-Ladder	2400	1949	0	685	2324
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	46 292
Barrett 设计	46 814
Power-Ladder 设计	62 438
Montgomery 设计	95 383
Parallel 设计	21 896

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 46 187 个时钟周期, 而加入空操作离散化后的模幂时间为 46 200. 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.

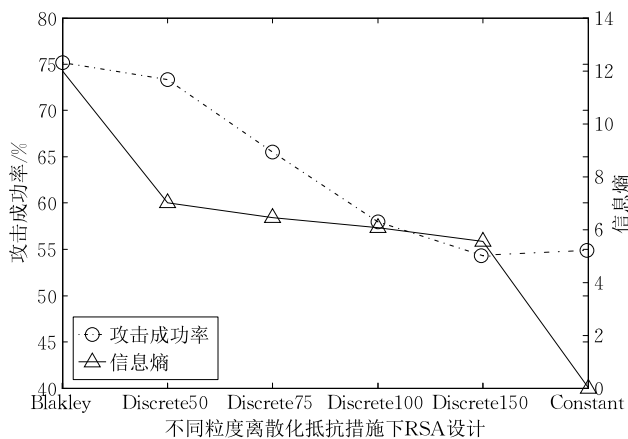


图 8 隐通道抵抗措施下 RSA 信息熵和攻击结果

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<sup>[7]</sup>. 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<sup>[5]</sup>. 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<sup>[4]</sup>.

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 <sup>[7]</sup>	Language 设计	信息流分析	硬件语言
Chen <sup>[4]</sup>	Bus, Divider, Cache 隐通道检测	事件相关性分析	微体系架构
Lee <sup>[5]</sup>	Cache 设计	量化分析	微体系架构
GLIFT <sup>[9, 25]</sup>	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<sup>[2-3, 32]</sup>, 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] Aciicmez 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] Aciicmez 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.