高阶掩码防护的设计实现安全性研究

李延斌[1] 唐 明[1] 李煜光[1] 胡晓波[2] 彭 敏[3] 张焕国[1]

[1](武汉大学计算机学院 武汉 430072)

[2](电力芯片设计分析国家电网公司重点实验室 北京 102200)

[3](国网新疆电力公司检修公司 乌鲁木齐 830063)

摘 要 掩码对抗方案自提出以来,从一阶对抗逐渐发展至高阶对抗阶段,安全性及通用性也不断提高.最早的一阶掩码方案主要针对 DES 算法提出,而后出现的一阶掩码方案则大多以 AES 为防护目标,并针对于不同的软硬件平台,同时不断减少时间和空间耗费.在追求更高安全性的同时,高阶掩码方案也不断朝着通用化的方向发展,主要工作在于设计通用化的 S 盒掩码方案,保证可应用于任何 S 盒设计且可抵抗任意阶侧信道攻击.高阶掩码方案已被普遍接受为一种算法级可证明安全的侧信道防护方法,出现以 ISW 安全性框架为代表的理论安全性证明,以及在此框架下的任意阶掩码方案.然而面向侧信道分析,密码算法设计实现的安全性无法仅仅基于算法安全,针对这种掩码方案理论安全与实际安全间的差距,Roche 与 Prouff 于 2011 年提出面向硬件设计的安全性掩码方案,但该方案无法运用于已有高阶掩码设计,只是对 Rivain 和 Prouff 在 CHES2010 上提出的 RivP 方案进行硬件级安全性实现.同时,以实现 d 阶安全的有限域乘法为例,实现需要加法和乘法的执行次数由 $O(d^2)$ 增加到 $O(d^3)$,由于增加过多的设计资源而对执行效率有较大的影响,降低了方案的实用性.在高效安全的硬件设计平台上,首先,作者分析由于时延不同导致的 glitch 有可能泄露敏感信息.相比于组合逻辑设计,时序设计下的电路不会产生降阶泄露.除了已有的 glitch 泄露外,文中还发现存在与硬件设计结构相关的泄露.作者从密码芯片设计者的角度出发,对掩码方案中关键部件的不同硬件设计结构进行分析.作者利用互信息的方法分析并行设计所产生的安全性问题,从理论上证明并行设计存在的安全隐患.在找出掩码设计隐患的基础上给出安全、轻量的安全设计建议,并最终通过实验对比不同设计结构下高阶掩码方案硬件设计的安全性,证明实验结果与理论研究结论一致.

关键词 侧信道;高阶掩码方案;glitch;硬件设计结构;安全性设计隐患

中图法分类号 TP309 **DOI 号** 10.11897/SP.J.1016.2018.00323

Study on the Security of Implementation of Higher order Masking Schemes

LI Yan-Bin[1] TANG Ming[1] LI Yu-Guang[1] HU Xiao-Bo[2] PENG Min[3] ZHANG Huan-Guo[1]

[1](*School of Computers, Wuhan University, Wuhan* 430072)

[2](*State Grid Key Laboratory of Power Industrial Chip Design and Analysis Technology, Beijing* 102200)

[3](*State Grid Xinjiang Maintenance Company, Urumqi* 830063)

Abstract Masking schemes have developed from the first order schemes to higher order schemes since it was proposed, with the improvement of its security and versatility. The first order masking schemes mostly focused on protecting DES and AES algorithm, while continually optimizing the area-time-memory trade-offs. As the first order masking schemes were not sufficient for long term security purpose, the higher order masking schemes have been widely studied, which tend to be more generic and mainly focus on designing the secure S-box against the attacks of any order. Higher order masking scheme is one of the most acceptable provably side channel counter-

收稿日期:2016-06-27;在线出版日期:2017-05-17.本课题得到国家自然科学基金(61472292,61332019)、国家"九七三"重点基础研究发展规划项目基金(2014CB340601)、面向智能电网新一代高速高等级安全芯片关键技术研究(526816160015)资助.**李延斌**,男,1992 年生,博士研究生,主要研究方向为密码学、侧信道分析.E-mail:lyb9205@163.com.**唐 明**,女,1976 年生,博士,教授,主要研究领域为信息安全、密码学、密码芯片等.**李煜光**,男,1992 年生,硕士,主要研究方向为密码学、侧信道分析.**胡晓波**,女,1977 年生,学士,主要研究方向为应用密码学、安全芯片等.**彭 敏**,女,1985 年生,硕士,主要研究方向为密码学、安全芯片等.**张焕国**,男,1945 年生,博士,教授,主要研究领域为信息安全、密码学、可信计算、容错计算.

measures against side channel attacks (SCAs). The security framework proposed by Ishai, Sahai and Wagner, known as the Ishai-Sahai-Wagner (ISW) scheme, is one of the most acceptable secure models of the existing d-order masking schemes, where d represents the masking order and plays the role of a security parameter. ISW includes an adversary model with a microprobe, which can be used in digital circuits. Relying on the framework of security proofs proposed by ISW, higher order masking schemes can be theoretically proven as ideal security. However, the security of cryptographic algorithm design does not only rely on arithmetic security, a gap may exist between theoretical and practical security. In order to solve this problem, Roche and Prouff proposed the "higher order glitch-free implementation" in 2011. However, various types of higher order masking cannot be designed under this framework. Unfortunately, it is difficult to implement on a hardware chip because of the very high time and area overheads, which replaces every field multiplication by $4d^3$ field multiplications and $4d^3$ additions. Excessive resource and time cost reduce the practicability of the scheme. In a combinational circuit, the glitch is generated by a certain design and is influenced by several factors, such as place and route, platforms, temperature, and other environment factors. The glitches or other intermediate values in the implementation of higher order masking scheme may contain a leakage related with two or even more shares. These can be considered as the leakage of sensitive information. Since the delays of internal signals are different, glitches may leak the sensitive information. Instead of combinational logic, the sequential logic of masking does not lead to smaller-order leakage. Beside glitches, we find there exists leakage in the hardware design structure. In this paper, we discuss whether there exists practical security problem of the provably secure masking schemes, which may lead masking schemes fail to resist attacks. In the view of cryptographic chip designer, we evaluated the security of different hardware design structures of key components in masking schemes theoretically. We analyze the security of parallel implementation using mutual information. It can be deduced that the implementation of parallel structure can not reach the security of masking scheme. Relying on these weaknesses, the secret key can be recovered through side channel attacks. To illustrate the security weakness on implementation, we analyze the different designs of different structures as contrast tests. Finally the result of experiment consists with theoretical analysis.

Keywords side channel; higher order masking; glitch; hardware design structure; risk of security design

1 引 言

侧信道分析经过近 20 年发展,基于功耗、电磁、时间等信息的侧信道分析方法已经成为密码算法软硬件设计的实际威胁[1-3].为了能有效防御侧信道攻击,迄今为止出现了多种算法级和实现级的侧信道防护方法,如掩码防护[4]和互补逻辑防护等[5].其中掩码防护方案因其可以从算法级别将加密过程中所有数据随机化,破坏功耗/电磁信号与敏感数据间的关联性,从而实现算法级别的可证明安全防护,成为现有侧信道防护方法中被最广泛使用的一种侧信道防护方法.

随着高阶侧信道攻击技术的发展,掩码方案自提出以来[4],从一阶对抗逐渐发展至高阶对抗,通过增加分析的时间和数据复杂度实现更高级别的安全性.现有可证明安全的高阶或任意阶掩码方案[6-11]主要基于 Ishai,Sahai 和 Wagner 等人 2003 年提出的 Ishai-Sahai-Wagner(ISW)证明框架[12].高阶掩码方案主要可被分为两类:以 Rivain 和 Prouff 于 2010 年提出的高阶掩码方案(RivP 方案)[10]为代表的多项式掩码和以 Coron 于 2014 年提出的高阶掩码方案(Coron14 方案)[11]为代表的查找表重构掩码.

由于掩码方案中常用的异或运算对线性函数而言满足结合律,分组密码掩码防护的重点在于非线

性运算的掩码设计. 在多项式逻辑实现掩码方案中，掩码防护将非线性运算拆分成多个子运算，通过对逻辑/算术运算的防护实现掩码运算过程中所有中间结果均为独立随机数，以实现 ISW 安全性框架，重点解决多项式乘法运算的安全性设计，并利用 RefreshMasks 对相关数据进行隔离. 这种实现方式对掩码方案的适用性更强，且执行效率较高；而以 Coron14 方案为代表的基于重构表型掩码方案，则更贴近 S 盒的查找表实现方式，由于敏感信息基于查找表地址，这种掩码防护的核心在于防止原始地址信息的泄露以及通过获得单次查找内容得到多个地址联立的结果，所以基于查找表方式的掩码方案设计中主要是随机化表格以及表格地址. 综上，本文主要研究多项式逻辑实现下掩码方案的安全性.

ISW 防护框架以探测次数与掩码方案中共享因子数量间的比例关系为安全性衡量的依据，以框架中假设的中间结果为掩码方案运算的中间结果. 而面向不同设计实现，任何一个算法都会在原有算法中间结果的基础上，出现算法以外的其它中间值.

现有高阶或任意阶掩码防护方法主要以 ISW 框架作为安全性证明的依据，ISW 可证明框架保证了高阶掩码方案在算法级别的安全性，但不能保证方案在实际环境下的安全性. 许多研究工作分析掩码方案在设计实现环节可能存在的泄露，主要包括 Mangard 等人 2005 年提出的“glitches”[13] 和“toggle count”[14]. 由于 glitches 产生的时间较短，我们称为瞬时泄露，而 toggle count 的泄露在一个时间区域内. toggle count 统计在一个时间区域内的跳变总数，既包含正常跳变，又包含 glitches 跳变. 正常跳变与掩码方案的逻辑电路对应，而 glitches 是由具体设计产生，并且受布局布线、温度和其它外界因素影响. toggle count 必须依赖于后仿的网表和相同的芯片元件库，利用功能仿真建模的跳变数与掩码前的值不相关[14]，因此 glitches 是 toggle count 成功攻击掩码方案的关键因素，glitches 在硬件中对可证明安全掩码方案的安全性具有不可忽视的威胁.

考虑到 glitches 对硬件电路设计的安全性影响，Roche 和 Prouff 在 2011 年提出了高阶抗 glitches 的掩码设计实现(RocP)[15]，旨在解决理论上可证明安全掩码方案在设计实现的安全性问题. 方案将多方计算协议应用于侧信道分析背景下，每个共享因子的操作分散在不同的子电路中，构成多方计算电路. 然而，它的复杂度比 RivP 方案多了一个数量级，对于 d 阶安全的方案，一个有限域乘法需要 $4d^3+8d^2+3d$ 乘法、$4d^3+8d^2+7d+2$ 加法，而 RivP 方案只需要 $2d^2+2d$ 乘法和 d^2+d+1 加法，资源和时间的开销限制了 RocP 方案的应用. 由于该方法是针对 RivP 方案分析基础上形成新的高阶掩码方案，不能完全解决其它高阶掩码方案在设计实现时存在的安全性问题，无法作为一种对现有其它类型掩码方案进行安全性改造的通用型方法.

现有对掩码方案设计实现特别是高阶掩码方案的硬件设计实现缺少系统且与硬件设计结构直接相关的安全性研究. 本文就任意阶掩码方案的硬件设计实现中的安全性问题进行研究，分析理论上可证明安全的掩码防护方案在不同硬件设计结构的实际安全性，以 RivP 方案[10] 为例进行仿真环境下任意阶功耗分析. 除 glitches 泄露外，本文还发现一种存在于硬件设计实现中的泄露. 通过本文的分析，我们可以看到：第一，不同硬件设计结构对于高阶掩码方案的设计实现影响不同；第二，掩码方案中不同模块的硬件安全性设计要求也不同；第三，在更有效进行泄露定位的基础上，可以提出更安全轻量的硬件设计方案. 这些都提示我们，如果合理选择硬件设计结构，并对掩码方案中的关键部件进行有效防护设计，即使在存在 glitches 的电路环境中，仍可以保证硬件设计掩码方案达到理论上的可证明安全.

本文第 2 节介绍现有高阶掩码方案和理论安全性证明框架；第 3 节分析可证明安全掩码方案在硬件设计实现的存在的两类安全性问题，主要包括 glitches 和并行设计结构带来的安全性问题；第 4 节针对这两类安全性问题进行安全性测试和防护增强设计与测试；最后是全文总结.

2　高阶掩码方案

2.1　掩码方案及可证明安全

掩码技术的目标在于使得密码设备的能量消耗不依赖于设备所执行的密码算法的中间结果，掩码对抗方案利用了秘密共享的思路，其原理在于将每个敏感的中间状态 x 拆分成多个共享因子，且满足 $x=x_1\oplus x_2\oplus\cdots\oplus x_n$. 每个加入掩码中间值均与原始中间值相独立，且每个共享因子 x_i 都单独执行相

应操作，从而保证不泄露任意 x 的信息，这也是掩码可抵抗 SCA 攻击的本质. 对于线性运算容易保证每个共享因子独立执行，而难点就在于利用共享因子重构非线性部件 S 盒来保证不泄露原始中间值.

Ishai，Sahai 和 Wagner 等人提出了一种高阶掩码方案安全性证明的模式[12]. ISW 模型利用 d 个输入共享因子来仿真攻击者能探测到的任意 d 个中间变量，通过这种仿真来证明中间变量和部分输入之间的相关性，也就证明了中间变量和秘密信息的独立性. 如果可以完美仿真，那么说明算法中任意 d 个中间变量都只与部分输入（d 个 x_i）相关，因此算法中任意 d 个中间变量与秘密信息相互独立，即算法满足 d 阶安全. ISW 模型的证明过程如下：

首先构造 d 阶下标集合 I. 分析算法中的所有中间变量，根据中间变量在算法中出现的位置定义每个中间变量的行下标和列下标. 任选 d 个攻击者可能探测到的中间变量，将这 d 个中间变量的行列下标添加到集合 I 中. 最后得到集合 I.

得到集合 I 后，利用下标集合中存在的部分输入共享因子，即 $a_{|I}:=(a_i)_{i\in I}$、$b_{|I}:=(b_i)_{i\in I}$ 来仿真这 d 个中间变量. 仿真过程完全按照掩码方案的步骤进行. 当算法使用到 $a_{|I}$ 或 $b_{|I}$ 时，使用正确值. 当算法使用到 $(a_i)_{i\in I}$，$(b_i)_{i\in I}$. 或未被探测到的随机数时，使用独立随机数代替. 如果所有 v_h 都可以被完美仿真，就证明掩码方案满足 d 阶可证明安全.

2.2 硬件设计安全性框架

Roche 和 Prouff 提出了高阶掩码方案硬件设计在抗 glitches 下的安全性评价模型（dth-order Glitches Adversary Model）[15]. 首先定义一个电路 C_f，攻击者在 d 阶 glitches 攻击模型中可以选取 d 个时间点 $t_1,t_2,\cdots,t_d$，并且可以观测到 d 个时间点的电路内部跳变状态 $(L_i(C_f(t_i)))_{i<d}$，从而进行攻击. 其中，$L(\cdot)$ 是泄露函数.

设 d 为整数，对于电路 C_f 所有中间状态变量，对于其中任意大小为 d 的子集合 V，对于任意敏感变量 Z，有 $p(Z|V)=p(Z)$，则电路 C_f 满足 d 阶安全.

为了统一评价硬件设计电路的安全性，我们在 ISW 框架下利用互信息（MI）评价模型进行评价[19]. 互信息是在假设已知密钥下，通过获取泄露的条件密度函数来描述设备可被利用的泄露量；当定量一个实际设备时，泄露的条件密度函数是未知的，只能通过估计得到一个估计值，由此引出了感应信息的概念，它用来评价评估者对互信息的最好预测.

用 S 表示侧信道攻击的目标密钥集样本量，s 则表示具体的密钥值. 令 $\boldsymbol{X}_q=[X_1,X_2,\cdots,X_q]$ 表示目标物理设备上一系列输入向量，即进行 q 次加密的明文，$x_q=[x_1,x_2,\cdots,x_q]$ 表示 q 个明文具体值. L_q 表示在侧信道攻击设备上采集的 q 条泄露数据，$l_q=[l_1,l_2,\cdots,l_q]$ 则表示具体 q 条泄露曲线值. 也就是说，泄露函数的一个实际输出曲线 l_q 对应于一个输入明文 x_q. 最后，我们定义 $\Pr[s|l_q]$ 表示在给定泄露 l_q 情况下密钥为 s 的条件概率. 我们定义条件熵矩阵如下：

$$H^q_{s,s^*}=-\sum_{l_q}\Pr[l_q|s]\cdot\log_2\Pr[s^*|l_q].$$

这里 s 和 s^* 分别表示正确密钥集和 $|S|$ 集合外可能的候选密钥集. 结合香农条件熵，可以直接得到互信息的表达方式：

$$MI(S;L_q)=H[S]-H[S|L_q].$$

注意到输入明文和输出样本曲线对侧信道攻击者一般是已知的，但一般密钥信息都会隐藏在中间公式中. 单纯从信息理论方面考虑，在已知明文和密文的情况下，正确密钥是可以被分析出来的，攻击者的攻击难易程度取决于加密过程的计算复杂度了. 条件熵正好可以描述中间过程中密钥参与运算的复杂程度，所以对熵的计算正好可以用来评价物理泄露量的大小.

敏感信息 x 的掩码电路 C_f 中存在很多中间变量，从中探测任意包含 d 个功耗点的子集合 P. 根据互信息与条件熵之间的关系，掩码电路安全的充要条件可表示为

$$MI(x,P)=0 \tag{1}$$

其中，d 满足 $2d+1\leq n$. 即当由 d 个探测的中间变量功耗点组成的集合 P 与敏感信息 x 之间的互信息理论值为零时，我们认为该掩码电路安全；否则，电路存在泄露.

为了保证方案的 d 阶安全性，Roche 和 Prouff 利用多方计算协议（MPC）来保证目标实现中的任意 d 对中间值都与敏感信息之间相互独立. 通过将每一个函数分成最小数量的子电路来实现，最终形成一个抗 glitches 的多方电路. 换句话说，利用 MPC 的掩码实现可以抵抗 glitches 攻击，但其复杂度是限制其性能的重要原因. 文献[15]中以有限域乘法为例，给出了 RocP 与 RivP 所需时钟周期数的复杂度对比，见表 1.

表 1　有限域乘法复杂度

方案	乘法	加法	随机数
RocP	$4d^3+8d^2+3d$	$4d^3+8d^2+7d+2$	$d(2d+1)$
RivP	$2d^2+2d$	d^2+d+1	$d(d+1)/2$

3　高阶掩码硬件设计的安全性问题

3.1　glitches

一阶掩码方案在组合电路中由于 glithces 的存在产生的安全性问题已经被 Mangard 等人验证[13]，这里我们以 RivP 方案为例，详细分析可证明安全的掩码方案在硬件设计阶段仍然存在的安全性问题.

基于 ISW 框架，RivP 方案在 $2d+1\leq n$ 时是安全的[9]. 其中，n、d 分别是共享因子和探测的数量. 考虑 $n=3$ 的情况，对 SecMult 算法[10]进行分析.

算法 1.　SecMult.

输入：$\oplus_i a_i=a$，$\oplus_i b_i=b$

输出：$\oplus_i c_i=ab$

FOR $i=0$ to $n-1$ do

　FOR $j=i+1$ to $n-1$ do

　　$r_{i,j}=rand$

　　$r_{j,i}=(r_{i,j}\oplus a_i b_j)\oplus a_j b_i$

　END FOR

END FOR

FOR $i=0$ to $n-1$ do

　$c_i=a_i b_j$

　FOR $j=0$ to $n-1$，$j\neq i$ do $c_i=c_i\oplus r_{i,j}$ ENDFOR

ENDFOR

根据算法 1，SecMult 算法的随机数矩阵可表示为 $\boldsymbol{R}=\begin{pmatrix}\varnothing & r_{01} & r_{02}\\ r_{10} & \varnothing & r_{12}\\ r_{20} & r_{21} & \varnothing\end{pmatrix}$. 其中：$r_{10}=r_{01}\oplus a_0 b_1\oplus a_1 b_0$，$r_{20}=r_{02}\oplus a_0 b_2\oplus a_2 b_0$，$r_{21}=r_{12}\oplus a_1 b_2\oplus a_2 b_1$. SecMult 输出可以表示为

$$C=\begin{pmatrix}c_0\\ c_1\\ c_2\end{pmatrix}=\begin{pmatrix}a_0 b_0\\ a_1 b_1\\ a_2 b_2\end{pmatrix}\oplus f(\boldsymbol{R})=\begin{pmatrix}a_0 b_0\oplus r_{01}\oplus r_{02}\\ a_1 b_1\oplus r_{10}\oplus r_{12}\\ a_2 b_2\oplus r_{20}\oplus r_{21}\end{pmatrix}.$$

为了缩短算法所用的时钟周期数，我们将上述算法计算部分通过组合逻辑实现，如图 1 所示. 随机数发生器在 3 个时钟周期内提供 3 个随机数 r_{01}、r_{02}、r_{12}，虚线框内表示组合逻辑设计的变量，c_0、c_1 和 c_2 在组合电路中一个时钟周期内根据上式完成计算. 当 $n=3$ 时，从产生随机数到输出 c_0、c_1、c_2，每次执行 SecMult 算法只占用了 4 个时钟周期. 而完全串行下，图 1 中每个变量的计算均独立占用一个时钟周期，因此，每次串行执行 SecMult 算法共占用 9 个时钟周期. 虽然这种设计较于串行情况下时钟周期数缩短了，但是后面的分析则显示这种设计实现具有潜在的安全性问题.

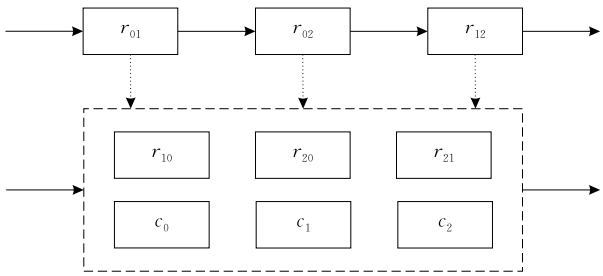


图 1　SecMult 组合电路设计

结合图 1 和算法 1 可以得到

$$\begin{cases}c_0=a_0 b_0\oplus r_{01}\oplus r_{02}\\ c_1=(r_{01}\oplus r_{12})\oplus a_1 b_1\oplus a_0 b_1\oplus a_1 b_0\\ c_2=(r_{02}\oplus r_{12})\oplus a_0 b_2\oplus a_1 b_2\oplus a_2 b_1\oplus a_2 b_0\oplus a_2 b_2\end{cases}\tag{2}$$

记 c_i^1、c_i^2 为 c_i 的连续状态，Δc_i 为两个状态间的跳变. 记 t_{a_1}，$t_{r_{01}}$，$t_{r_{02}}$，$t_{r_{12}}$，t_{b_1}，t_{a_0}，t_{b_0}，t_{a_2}，t_{b_2} 分别为各自信号的到达时间. 考虑下列情况：

(1) $t_{b_2}<t_{r_{01}},t_{r_{02}},t_{r_{12}},t_{b_1},t_{a_0},t_{b_0},t_{a_1},t_{a_2}$

$c_2^1=(r_{02}^1\oplus r_{12}^1)\oplus a_2^1 b_1^1\oplus a_2^1 b_0^1\oplus(a_0^1\oplus a_1^1\oplus a_2^1)b_2^1$，

$c_2^2=(r_{02}^1\oplus r_{12}^1)\oplus a_2^1 b_1^1\oplus a_2^1 b_0^1\oplus(a_0^1\oplus a_1^1\oplus a_2^1)b_2^2$，

$\Delta c_2=(a_2^1\oplus a_1^1\oplus a_0^1)(b_2^1\oplus b_2^2)=a^1(b_2^1\oplus b_2^2)$，

$MI(\Delta c_2,a^1)\neq 0$.

因此，当 $d=1$ 时，c_2 信号上发生的 glitches 导致功耗 P 与敏感信息 a 之间存在相关性，两个相关的变量之间必然不会相互独立，从而不满足式(1)，即 $MI(a,P)\neq 0$. 这种泄露存在于类似的情况如 $t_{b_2}<t_{a_2}$，t_{a_1},t_{a_0} 或 $t_{b_2}>t_{a_2},t_{a_1},t_{a_0}$ 中.

(2) $t_{a_2}<t_{r_{01}},t_{r_{02}},t_{r_{12}},t_{b_1},t_{a_0},t_{b_0},t_{a_1},t_{b_2}$

$c_2^1=(r_{02}^1\oplus r_{12}^1)\oplus a_1^1 b_2^1\oplus a_0^1 b_2^1\oplus(b_0^1\oplus b_1^1\oplus b_2^1)a_2^1$，

$c_2^2=(r_{02}^1\oplus r_{12}^1)\oplus a_1^1 b_2^1\oplus a_0^1 b_2^1\oplus(b_0^1\oplus b_1^1\oplus b_2^1)a_2^2$，

$\Delta c_2=(b_2^1\oplus b_1^1\oplus b_0^1)(a_2^1\oplus a_2^2)=b^1(a_2^1\oplus a_2^2)$，

$MI(\Delta c_2,b^1)\neq 0$.

同理，c_2 信号上发生的 glitches 导致子集合 P 与敏感信息 b 同样存在相关性，从而不满足式(1)，即 $MI(b,P)\neq 0$. 这种泄露存在于类似的情况如 $t_{a_2}<t_{b_2},t_{b_1},t_{b_0}$ 或 $t_{a_2}>t_{b_2},t_{b_1},t_{b_0}$ 中.

通过上述分析，可知当不同信号的时延不同，可能导致的泄露不同，而且导致泄露的阶数越高需要

满足的条件也越苛刻.理想情况下,当满足条件 t_{a_2}, $t_{a_1}, t_{a_0} < t_{b_2}, t_{b_1}, t_{b_0}$ 或 $t_{a_2}, t_{a_1}, t_{a_0} > t_{b_2}, t_{b_1}, t_{b_0}$ 时,说明设计既会泄露 a 也会泄露 b.掩码方案中会有包含所有共享因子的操作,特别是多项式掩码方案,如果这些操作采用组合电路设计,每个门级翻转的时间与数据、时延都有关系,电路内部会产生各种各样的 glitches,从而产生降阶的可能.当满足类似上述的时延条件时,掩码方案的安全性就会受到挑战.

3.2 并行设计结构

硬件设计的一个重要特点就是并行执行,通常设计者为了提高算法执行效率,可在保证不影响正确结果的情况下,采取并行设计.本节我们讨论硬件设计中并行设计对可证明安全掩码方案安全性的影响.

高阶掩码方案为了提高安全性,通常会引入大量的随机数来确保每个阶段的中间值都是相互独立的.Rivain 和 Prouff 在 RivP 方案中为了保证 SecMult 的输入相互独立[10],引入了 RefreshMasks,并被 Coron 在其高阶掩码方案中使用[11].无论是对于多项式掩码方案还是对于查找表重构掩码方案而言,RefreshMasks 设计实现的安全性对整个掩码方案的安全性都有重要影响.

算法 2. RefreshMasks.

输入:$\oplus_i x_i = x$

输出:$\oplus_i x_i = x$

FOR $i=1$ to $n-1$ do

$tmp = rand(8)$

$x_0 = x_0 \oplus temp -> tmp$

$x_i = x_i \oplus temp -> tmp$

ENDFOR

我们对 $n=3$ 的 RefreshMasks 算法的设计实现进行安全性分析.根据算法 2,x_0 和 x_i 的计算在每次循环中没有执行先后顺序的要求,因此为了节省时间开销,可以设计为并行运算结构,如图 2 所示.实线框内变量均采用寄存器存储.对于 n 阶的 RivP 方案,串行结构的 RefreshMasks 需要 $O(3\times(n-1))$ 个时钟周期才能完成所有共享因子的更新,而并行运算结构只需要 $O(n)$ 个时钟周期.下面对这种并行运算结构的设计进行安全性分析.

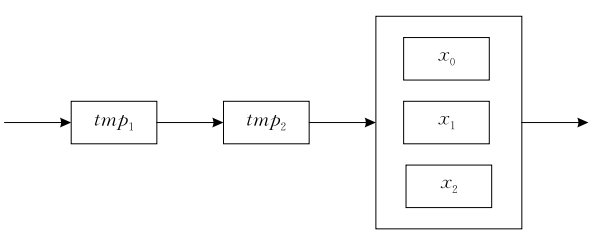


图 2 RefreshMasks 并行运算设计图

第 3 个时钟周期内寄存器的运算可以表示为

$$\begin{cases} x_0 = x_0 \oplus tmp_1 \oplus tmp_2 \\ x_1 = x_1 \oplus tmp_1 \\ x_2 = x_2 \oplus tmp_2 \end{cases} \tag{3}$$

寄存器翻转的 3 个中间值联立结果为 x.当功耗数据满足汉明重模型时,总功耗等于 3 个中间值的汉明重之和,只需要判断 3 个中间值汉明重之和与联立值的汉明重之间的关系即可说明是否存在泄露.

定理 1. 设密码算法执行过程中,存在 3 个中间值 m_0, m_1, m_2 并行执行,且满足 $m = m_0 \oplus m_1 \oplus m_2$.假设能量消耗符合汉明重模型,则互信息

$$MI\left(HW(m), \sum HW(m_i)\right) \neq 0.$$

证明. 设中间值 m 的汉明重为 $HW(m)$,可表示为函数形式:

$$\begin{aligned} HW(m) &= (1,1,\cdots,1)^{\mathrm{T}}(m[0], m[1], \cdots, m[7]) \\ &= (1,\cdots,1)^{\mathrm{T}}(m_0[0] \oplus m_1[0] \oplus m_2[0], \cdots, \\ &\quad m_0[7] \oplus m_1[7] \oplus m_2[7]), \end{aligned}$$

其中,$m[0], m[1], \cdots, m[7]$ 表示 m 的 8 个比特.同理,

$$\begin{aligned} \sum HW(m_i) &= (1,1,\cdots,1)^{\mathrm{T}}(m_0[0] + m_1[0] + \\ &\quad m_2[0], \cdots, m_0[7] + m_1[7] + m_2[7]), \end{aligned}$$

因此,$HW(m)$ 与 $\sum HW(m_i)$ 之间是否相互独立等价于 $m_0[i] \oplus m_1[i] \oplus m_2[i]$ 与 $m_0[i] + m_1[i] + m_2[i]$ 是否相互独立.穷举 $m_0[i], m_1[i], m_2[i]$,见表 2.

表 2 $m_j[i]$的统计信息

$m_0[i]$	$m_1[i]$	$m_2[i]$	$\sum_j m_j[i]$	$\oplus_j m_j[i]$
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	2	0
1	0	0	1	1
1	0	1	2	0
1	1	0	2	0
1	1	1	3	1

$$p(m_0[i] + m_1[i] + m_2[i] = 1) = \frac{3}{8},$$

$$p(m_0[i] \oplus m_1[i] \oplus m_2[i] = 1) = \frac{1}{2}.$$

$$p(m_0[i] + m_1[i] + m_2[i] = 1,$$

$$m_0[i] \oplus m_1[i] \oplus m_2[i] = 1) = \frac{3}{8} \neq \frac{3}{16}.$$

因此，$m_0[i]\oplus m_1[i]\oplus m_2[i]$与 $m_0[i]+m_1[i]+m_2[i]$不相互独立，$HW(m)$与$\sum HW(m_i)$之间不相互独立. 根据联合熵的性质，有

$$H\left(HW(m),\sum HW(m_i)\right)<$$
$$H(HW(m))+H\left(\sum HW(m_i)\right)$$

于是，对于条件熵

$$MI\left(HW(m),\sum HW(m_i)\right)=$$
$$H(HW(m))+H\left(\sum HW(m_i)\right)-$$
$$H\left(\left(HW(m),\sum HW(m_i)\right)\right)>0.\quad 证毕.$$

对于泄露模型满足汉明重模型的功耗分析，由于功耗 $P\propto\sum HW(m_i)$，功耗与敏感信息 m 之间的互信息不为零，即 $MI(m,P)\neq 0$，因此在并行设计下存在信息泄露. 对于 3 阶 RefreshMasks 的上述设计，互信息的理论值是 0.6956 bit. 由式(1)可知，若敏感信息与功耗之间互信息的理论值不为零，说明该掩码电路存在安全性问题. 当设计者采用并行运算设计结构时，攻击者利用互信息攻击[16]可以很容易地恢复敏感信息. 产生这种泄露的本质原因是掩码方案中同时进行了不同共享因子的运算，在时间维度上将不同共享因子的信息叠加，使得联立的中间值与功耗叠加之间不相互独立，因此，产生了降阶的泄露.

4　实　验

我们以 RivP 方案中 SecMult 和 RefreshMasks 为例，说明可证明安全的掩码方案在硬件阶段可能出现的安全问题，并给出安全性设计结构. 在 FPGA 上实现 RivP 方案中的 SecMult、RefreshMasks，整个实验工作基于 1.8 GHz Intel 处理器，Windows 7 操作系统.

4.1　SecMult 不同设计结构的安全性测试

4.1.1　存在安全性隐患的 SecMult 设计结构

为了提高高阶掩码的执行效率，实际应用中某些操作可以采用组合逻辑的方式实现. 我们下面以 RivP 方案中的 SecMult 算法的组合逻辑实现为例进行分析，硬件设计结构如图 3 所示. 整个电路分为随机数发生器和组合电路两部分. 其中，随机数发生器由时钟信号(CLK)和初始化信号(LOAD)控制，为组合逻辑的计算提供随机数；组合电路包括 SecMult 算法的输入数据之间的有限域乘法电路，最终计算输出 c_0，c_1，c_2. 根据 3.1 节的分析，组合电路中泄露发生在所有共享因子汇聚的操作. 由于 glitches 的作用，使得掩码电路功耗与原始值 a 相关. 由图 3 中可以发现，a 的共享因子 a_0，a_1，a_2 与 b_2 经过有限域乘法后输出 Z_5，Z_7，Z_4，均进入 Mxor_c21 参与 c_2 的计算，因此，Mxor_c21 可能导致信息泄露.

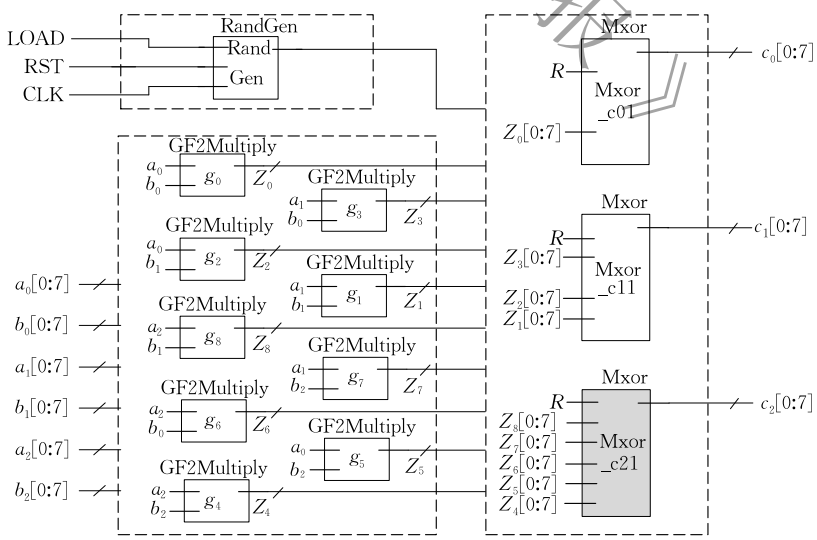


图 3　SecMult 组合电路设计结构

由于组合电路的功耗主要由静态功耗和动态功耗组成，而数据相关部分的动态功耗又和电路内部信号的跳变有关. 因此，掩码方案的目标是掩码电路所有中间节点的跳变均与原始输入值 a 无关. 经过 3.1 节的分析，这一目标可能被一些不可避免的 glitches 所打破. 为了检测当前组合电路设计

的安全性，我们对网表文件进行 H 组时序仿真，加密时间为 t_T，生成 VCD 文件. VCD 文件中包含了电路中所有信号在加密过程中任意时刻的跳变情况，记 $e^{s_i}_{h,t_j}$ 为信号 s_i 在第 h 次加密过程中时刻 t_j 的跳变情况，若发生跳变，则 $e^{s_i}_{h,t_j}=1$，反之为 0. 将信号 s_i 在一次加密过程中的跳变情况记录为一条长度为 t_T 的跳变曲线，记为 $e^{s_i}_h$. 由此，可得到信号 s_i 的跳变记录矩阵

$$\boldsymbol{E}^{s_i}=\begin{pmatrix}e^{s_i}_1\\ \vdots\\ e^{s_i}_H\end{pmatrix}=\begin{pmatrix}e^{s_i}_{1,1} & \cdots & e^{s_i}_{1,t_T}\\ \vdots & \ddots & \vdots\\ e^{s_i}_{H,1} & \cdots & e^{s_i}_{H,t_T}\end{pmatrix},$$

定义 $\Delta=p(e^{s_i}_{h,t_j}=1|a^k=0)-p(e^{s_i}_{h,t_j}=1|a^k=1)$ 表示跳变与原始值 a 之间的相关性，a^k 为 a 的其中一比特. 对于跳变 $e^{s_i}_{h,t_j}$ 与 a^k 无关的信号，有 $\Delta=0$，反之，则不为零. 受噪声的影响，区分 Δ 和 0 与所用的仿真数量 H 有关. 经过 Fisher 变换，Δ 可以表示为一个满足正态分布的变量 Z[4]，当以 $1-\alpha$ 置信度水平接受可区分 Δ 与 0 时，有

$$H=3+8\frac{z^2_{1-\alpha}}{\ln^2\dfrac{1+p(e^{s_i}_{h,t_j}=1|a^k=0)-p(e^{s_i}_{h,t_j}=1|a^k=1)}{1-p(e^{s_i}_{h,t_j}=1|a^k=0)-p(e^{s_i}_{h,t_j}=1|a^k=1)}},$$

仿真所需曲线量与相关性 Δ 和置信度水平之间的关系见图 4.

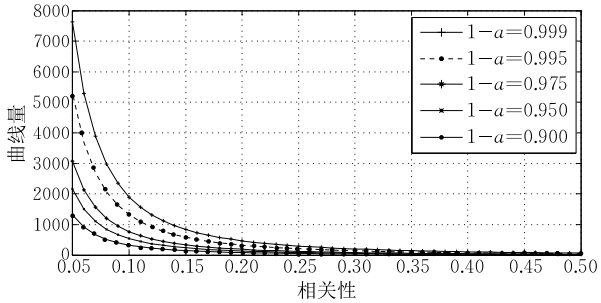


图 4 不同相关性下判别泄露所需曲线量

如图 4 所示，存在相关性越大的电路，所需要的仿真曲线量越少，相反，理论上相关性越弱的信号，将其区分出来就需要越多的曲线量. 此外，相关性相同情况下，置信度水平越高，所需曲线量越多. 这与功耗攻击中区分正确密钥所需要的曲线量与相关性的关系是相同的.

对于 8 比特的有限域乘法，$\Delta=\frac{1}{256}$，当置信度为 80% 时，大约需要 92 927 条仿真曲线. 对于信号 c_2，每个比特各有一个跳变记录矩阵 $\boldsymbol{E}^{c_2[k]}$. 由于 glitches 与数据相关，在不同数据作为输入时，跳变的时间不同，因此，需要先对跳变曲线进行整合预处理，即对一个时间段内的跳变取所有跳变之和. 当选择 a^k 作为中间值进行划分作差分曲线时，出现峰值对应泄露的信号跳变，即产生泄露的 glitches. 如图 5 所示. 仿真数量为 10 万条，横轴为中间值选取分别为 $a[0]\sim a[7]$，纵轴为 $c_2[k]$ 跳变与 $a[k]$ 的相关系数.

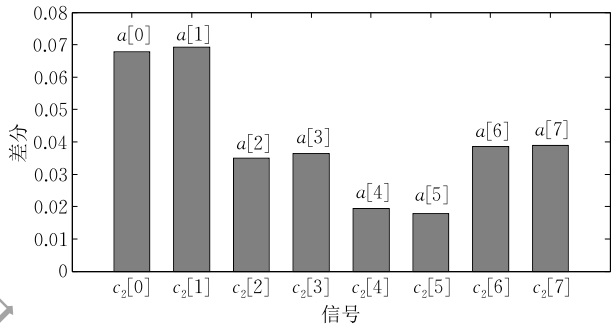


图 5 c_2 跳变与 a 之间的相关性

根据 3.1 节分析，若 c_2 跳变与 a 之间存在相关性，则 c_2 的所有比特都应与 a 的任一比特都存在相关性，这是由 8 比特有限域乘法特点决定的. 图 2 只给出了中间值为 $a[k]$ 时，与 $a[k]$ 相关性最大的信号 $c_2[k]$. 从图 5 中可以发现，对于 a 的任一比特都有对应的信号存在泄露，相关性水平不同，这是由于有限域乘法组合电路中每一比特参与运算的时延与逻辑门不同.

为了验证上述组合电路中某些信号的 glitches 对整体功耗的影响，我们将功耗曲线作为分析对象. 我们借助第三方工具 PrimePower 得到仿真功耗曲线. 前面的分析已经能够得到与 a 的每一比特存在相关性的 glitches，现在分析 a 的每一比特与功耗曲线之间的相关性，如图 6 所示. 可以看出，功耗与 a 之间存在相关性，且峰值点位置几乎相同，不同比特的峰值位置之间有微小的差异是由组合电路中延迟和门级翻转时间不同导致的.

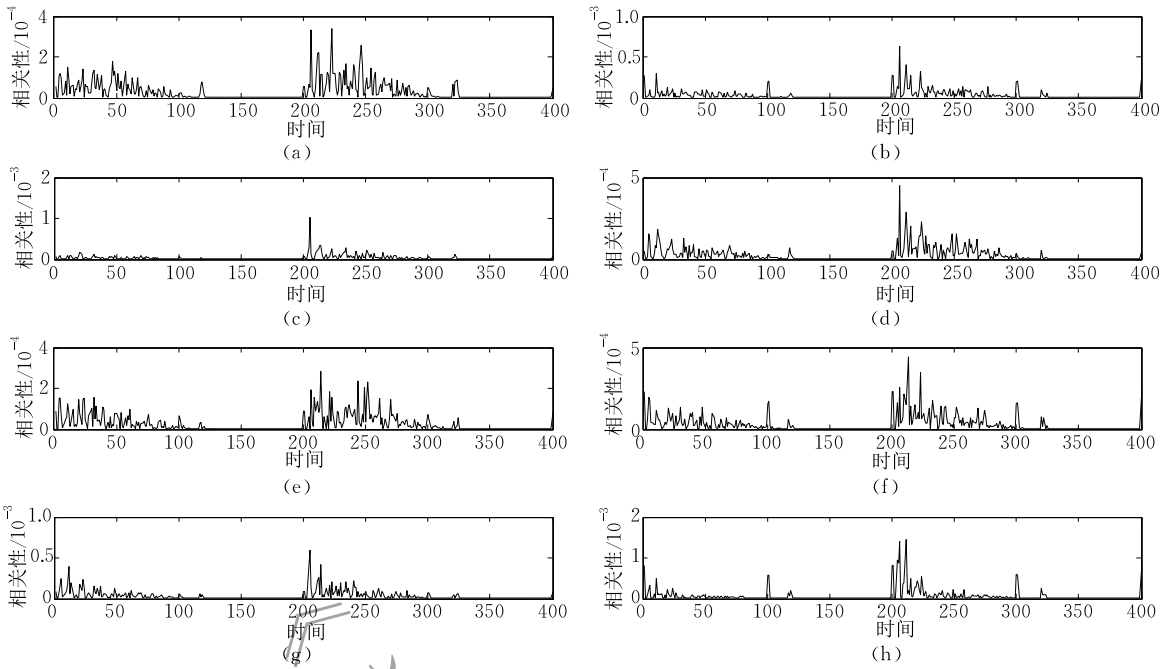


图 6　仿真功耗与敏感信息 a 的相关性((a)～(h)分别表示 $a[0]$～$a[7]$单个比特与仿真功耗的相关性分析结果)

4.1.2　安全的 SecMult 设计结构

SecMult 组合电路设计产生泄露的根本原因在于一片组合电路中包含了不同共享因子的运算. 为了避免 glitches 在组合电路中可能带来的降阶威胁,最直接的方法是在不同共享因子进入一片电路前设置寄存器,消除掉前面 glitches 的作用. 如图 3 结构中,r_{20},r_{21} 与 a_2b_2 包含了 a 所有共享因子的信息,如果用时钟控制将这三者计算完全隔离,将解决 Mxor_c21 的安全问题,同理对于 b 所有共享因子的相关操作也适用.

根据 SecMult 的流程,将产生随机数的过程作为第一阶段(GenerateRand),计算输出 c_0,c_1,c_2 的过程作为第二阶段(CalculateOutput). 两个阶段都采用时序逻辑设计,如图 7 所示.

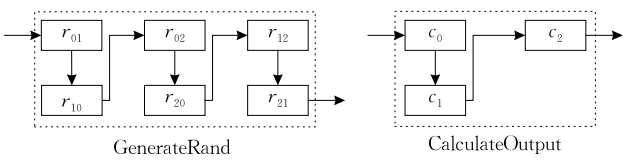


图 7　安全的 SecMult 硬件设计结构

表 3 显示了安全 SecMult 硬件设计结构与组合电路设计的资源对比. 为了消除组合电路中 glitches 的影响,则将算法中所有需要的有限域乘法串行执行,因此只需要一个有限域乘法器的资源,大大减少了整个电路的 LUT 开销.

表 3　SecMult 硬件设计结构的资源对比

	Flip-Flops	LUTs	CLK
组合电路设计	34	376	4
安全设计结构	134	158	11

类似上一节的功耗分析,分析 a 的每一比特与功耗曲线之间的相关性,如图 8 所示. 图中显示了所有比特与功耗之间的相关性,"*"代表安全设计的功耗与敏感信息的相关性结果,"-"代表组合电路设计,即存在安全性隐患设计对应的功耗分析结果. 相比于不安全的设计有明显的峰值,可以看出时序逻辑设计的功耗与 a 各个比特的相关性没有超过 10^{-4},且没有明显的峰值,说明功耗与敏感信息之间不存在相关性.

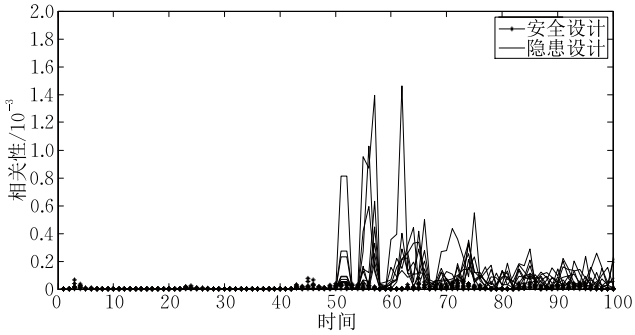


图 8　仿真功耗与敏感信息 a 的相关性

4.2　RefreshMasks 不同设计结构的安全性测试

4.2.1　存在安全性隐患的 RefreshMasks 设计结构

由于高阶掩码的中间值较多,因此很多过程都需要循环执行,当循环中的操作不存在先后执行顺

序时，为了节省时间开销，设计者可以将其设计改为并行运算，我们以 RivP 方案中的 RefreshMasks 为例，硬件设计结构如图 9 所示. 实线框内变量均采用寄存器存储. 我们对设计后的网表文件进行时序仿真，并利用第三方工具 PrimePower 得到 50 000 条仿真功耗曲线.

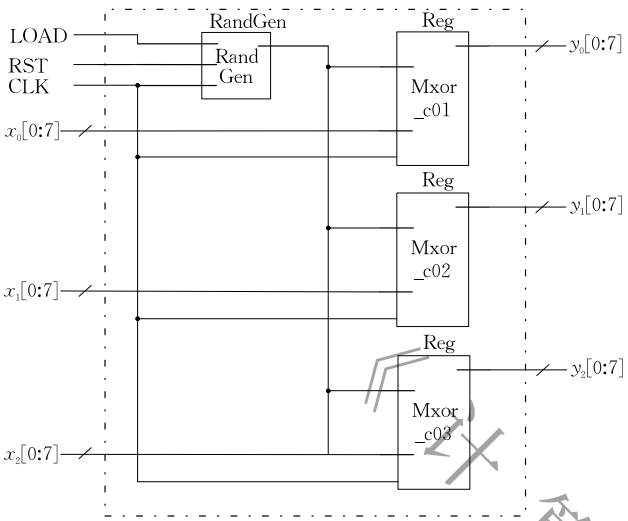


图 9 RefreshMasks 并行设计结构

如图 9 所示，x_0，x_1，x_2 作为 RefreshMasks 掩码电路的输入，$x=x_1\oplus x_2\oplus\cdots\oplus x_n$. 随机数发生器为其提供三个随机数 r_0，r_1，r_2，在下一个时钟周期生成输出 y_0，y_1，y_2. 基于汉明重模型，假设已知中间值的情况下，对 y_0，y_1，y_2 进行 Different of Means (DOM)[17-18] 定位. 毫无疑问，功耗 P 与这三个中间值均存在相关性，见图 9. 通常，攻击者对于安全的高阶掩码设计实现需要探测不同共享因子的功耗，从而联立恢复敏感信息 x. 从图 10(a)～(c)可以发现，三个中间值相关性峰值点在同一时刻(3001)，我们利用 Standaert 等人的互信息评价指标[19]，计算 $MI(HW(x),P)$进行泄露量评价，如图 10(d)所示.

如图 10(d)所示，当选择汉明重模型为泄露模型时，功耗曲线与敏感信息之间的互信息最大值为 0.6960，可视为与 3.2 节分析的理论值 0.6956 相等，即该掩码硬件电路产生的功耗泄露量与理论分析值近似相同，均不为零；而其它时刻与敏感信息相互独立的功耗点的互信息大约在 10^{-4} 数量级，趋近于零. 这说明时刻 3001 的功耗产生了泄露，攻击者进行一次探测即可恢复敏感信息 x. 实验结果还显示，当不采用汉明重模型时，敏感信息的值与功耗之间的互信息为 0.773，同样产生了泄露，这说明并行设计结构的泄露不局限于泄露模型本身，根本原因在于中间值与功耗之间并不是相互独立的关系.

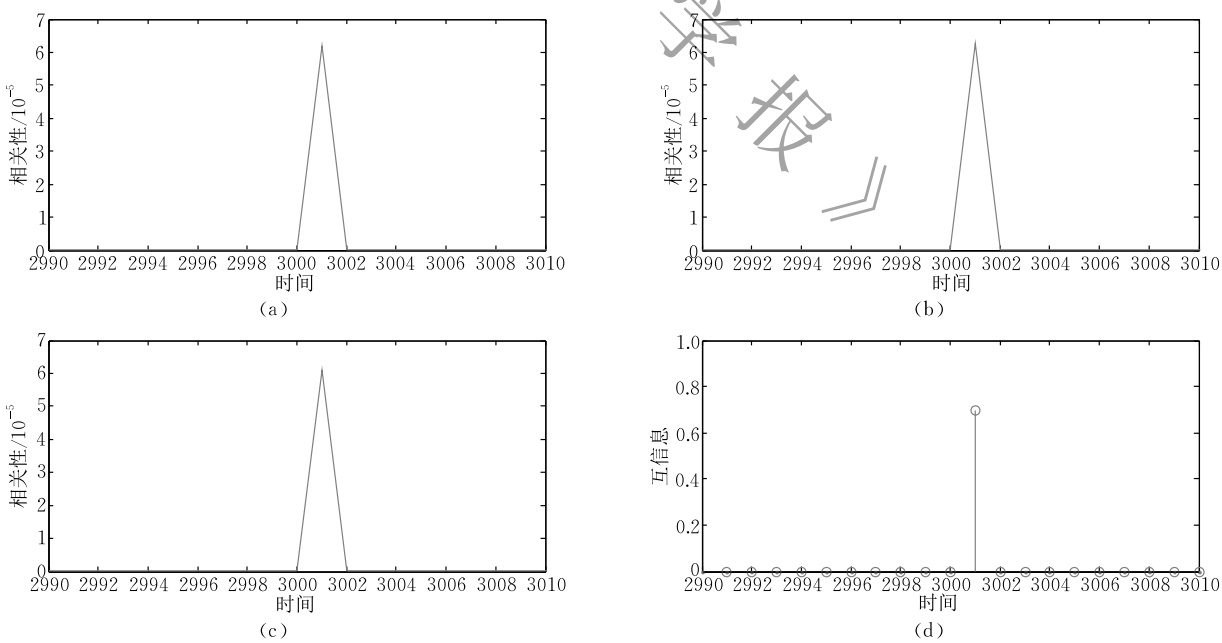


图 10 针对 RefreshMasks 的仿真功耗分析((a)～(c)分别为对 y_0，y_1，y_2 定位；(d)为功耗曲线与敏感信息之间的互信息)

4.2.2 安全的 RefreshMasks 设计结构

RefreshMasks 的作用是对掩码方案中的所有共享因子进行更新，由于在更新过程中不同共享因子的操作并行执行，导致对应的功耗点涵盖了多个共享因子的信息，因而也具有降阶的威胁.

同样，我们对产生随机数的过程定义为 Generate-Rand，对掩码的更新过程定义为 Refresh. 这两个过程中的操作均采用时序逻辑设计，结构采用流水设

计，如图 11 所示. 当 $n=3$ 时，整个 RefreshMasks 需要三个时钟周期. 第一个时钟周期产生一个随机数 tmp_1；第二个时钟周期进行寄存器 x_0 和 x_1 的更新，以及第二个随机数 tmp_2 的生成；第三个时钟周期进行寄存器 x_0 和 x_2 的更新. 寄存器在算法执行过程中的状态变化见表 4 所示.

表 4　寄存器状态变化

	x_0	x_1	x_2
1	—	—	—
2	$x_0 \oplus tmp_1$	$x_1 \oplus tmp_1$	—
3	$x_0 \oplus tmp_1 \oplus tmp_2$	$x_1 \oplus tmp_1$	$x_2 \oplus tmp_2$

表 5 显示了 RefreshMasks 的安全硬件设计结构与并行设计结构的资源对比. 可以发现，在 $n=3$ 的情况下，流水结构在资源消耗方面，占用 Flip-Flops 与 LUTs 的数量略小于并行结构；算法执行过程均耗费三个时钟周期. 由于随机数发生器两次产生的随机数相互独立，且共享因子相互独立，流水结构中的并行不会导致 x 的信息泄露.

表 5　RefreshMasks 硬件设计结构的资源对比

	Flip-Flops	LUTs	CLK
并行设计结构	52	57	3
安全设计结构	36	42	3

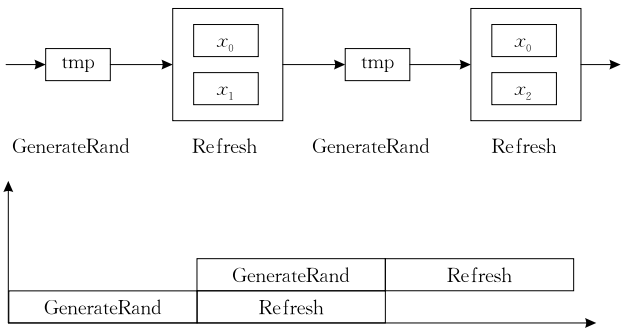
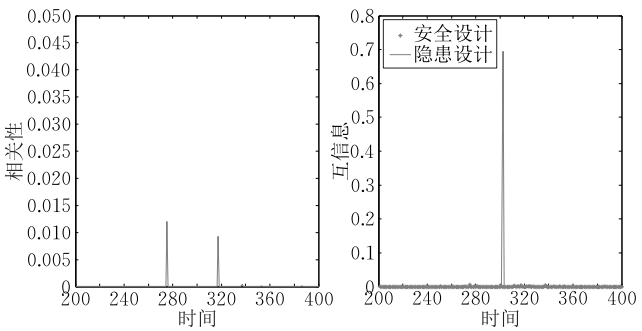


图 11　安全的 RefreshMasks 硬件设计结构

假设已知中间值，对仿真功耗进行定位 Refresh 操作位置，如图 12 所示. (a) 中定位的第一个峰值对应于 x_0 与 x_1 的操作，第二个峰值对应 x_0 与 x_2 的操作. (b) 为汉明重模型下敏感信息与功耗之间的互信息，“*”表示当前安全设计下的互信息，“-”表示上述存在安全隐患设计下的互信息 0.6960，可以发现由于将不同共享因子的运算时刻由一个时刻分散成两个时刻，互信息最大值远小于并行设计下互信息最大值，不会出现明显的峰值.



(a) 对两次Refresh定位　　(b) 功耗曲线与敏感信息之间的互信息

图 12　针对 RefreshMasks 安全设计的仿真功耗分析

本文实现了高阶掩码中常用的两个算法，并在给定泄露模型的假设下评估了实现的安全性. SecMult 的组合电路实现可以看成是一片输入信号为敏感信息的所有共享因子的组合电路. 由前文分析可知，发生在组合电路内部的 glitches 会产生信息泄露，功耗与敏感信息之间的依赖关系使得设计实现易被攻击. 结合 RefreshMasks 在并行设计实现中产生的安全问题，我们建议高阶掩码设计实现避免不同共享因子在同一片组合电路中进行运算，可以通过时钟控制分离不同共享因子之间的操作，从而消除 glitches 的影响；为了提高算法的执行效率可以采用流水结构，不同共享因子的操作在时间上避免并行执行. 经过安全性设计，实验结果显示安全性设计保证了高阶掩码方案在 ISW 框架下的安全性，并且资源与原始设计实现相比未有明显增加，远未达到 RocP 实现方案的代价.

5　结　论

高阶掩码被普遍接受为一种理论上可证明安全的算法级侧信道防护方法，现有研究主要集中在如何在安全性框架下设计出任意阶可证明安全的掩码方案，以及相同阶数条件下掩码方案的资源速度优化问题. 然而，作为侧信道防护方法之一的掩码方案，其设计实现级别的安全性无法被算法级安全性覆盖，于是出现针对硬件设计实现中利用 glitches 进行掩码攻击的研究.

本文对现有任意阶掩码防护方案关键运算的不同硬件设计结构分析基础上，发现除 glitches 泄露外，算法硬件设计结构同样会对掩码方案的整体安

全性产生影响.对已有高阶功耗分析中认为安全的并行设计结构重点分析,发现当泄露模型满足汉明重模型时,关键运算的并行设计结构可能存在的安全性隐患,并对泄露量进行定量分析.

为了进一步验证高阶掩码方案硬件设计结构的安全性隐患,我们以 RivP 掩码方案为例,进行三阶 RivP 方案的不同设计结构分析与仿真环境下的验证性实验,实验结果不仅证实理论研究部分的安全性隐患确实存在,同时泄露量与理论分析完全一致.在此基础上,我们针对发现的硬件结构设计安全性问题,有针对性地给出相应的安全性设计增强,并对其进行分析测试证实其达到理论安全性.我们认为掩码防护方案的硬件设计实际安全性不能仅仅从算法级安全性框架上得到保证,而应该从硬件设计结构出发,由设计向算法提出相应的安全性要求.

参 考 文 献

[1] Kocher P, Jaffe J, Jun B. Differential power analysis//Proceedings of the Advances in Cryptology—CRYPTO'99. San Francisco, USA, 1999: 388-397

[2] Agrawal D, Archambeault B, Rao J R, et al. The EM side—Channel(s)//Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems. San Francisco Bay, USA, 2002: 29-45

[3] Kocher P. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems//Proceedings of the Advances in Cryptology—CRYPTO'96. Stanford, USA, 1996: 104-113

[4] Mangard S, Oswald E, Popp T. Power Analysis Attacks: Revealing the Secrets of Smart Cards. Berlin, Germany: Springer, 2007

[5] Razafindraibe A, Robert M, Maurine P. Analysis and improvement of dual rail logic as a countermeasure against DPA//Proceedings of the International Workshop on Power and Timing Modeling, Optimization and Simulation. Montpellier, France, 2007: 340-351

[6] Rivain M, Dottax E, Prouff E. Block ciphers implementations provably secure against second order side channel analysis//Proceedings of the International Workshop on Fast Software Encryption. Lausanne, Switzerland, 2008: 127-143

[7] Kim H S, Hong S, Lim J. A fast and provably secure higher-order masking of AES S-box//Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems. Nara, Japan, 2011: 95-107

[8] Carlet C, Goubin L, Prouff E, et al. Higher-order masking schemes for S-boxes//Proceedings of the Fast Software Encryption. Washington, USA, 2012: 366-384

[9] Coron J S, Prouff E, Rivain M, et al. Higher-order side channel security and mask refreshing//Proceedings of the International Workshop on Fast Software Encryption. Washington, USA, 2013: 410-424

[10] Rivain M, Prouff E. Provably secure higher-order masking of AES//Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems. Santa Barbara, USA, 2010: 413-427

[11] Coron J S. Higher order masking of look-up tables//Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques. Copenhagen, Denmark, 2014: 441-458

[12] Ishai Y, Sahai A, Wagner D. Private circuits: Securing hardware against probing attacks//Proceedings of the Annual International Cryptology Conference. California, USA, 2003: 463-481

[13] Mangard S, Popp T, Gammel B M. Side-channel leakage of masked CMOS gates//Proceedings of the Cryptographers' Track at the RSA Conference. San Francisco, USA, 2005: 351-365

[14] Mangard S, Pramstaller N, Oswald E. Successfully attacking masked AES hardware implementations//Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems. Edinburgh, UK, 2005: 157-171

[15] Roche T, Prouff E. Higher-order glitches free implementation of the AES using secure multi-party computation protocols//Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems. Nara, Japan, 2011: 63-78

[16] Gierlichs B, Batina L, Tuyls P, et al. Mutual information analysis//Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems. Washington, USA, 2008: 426-442

[17] Chari S, Rao J R, Rohatgi P. Template attacks//Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems. CA, USA, 2002: 13-28

[18] Fan G, Zhou Y, Zhang H, et al. How to choose interesting points for template attacks more effectively?//Proceedings of the International Conference on Trusted Systems. Beijing, China, 2014: 168-183

[19] Durvaux F, Standaert F X, Veyrat-Charvillon N. How to certify the leakage of a chip?//Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques. Copenhagen, Denmark, 2014: 459-476

LI Yan-Bin, born in 1992, Ph. D. candidate. His research interests include side channel analysis and cryptography.

TANG Ming, born in 1976, Ph. D., professor. Her research interests include information security, cryptography, and cryptographic chip.

LI Yu-Guang, born in 1992, M. S. His research interests include side channel analysis and cryptography.

HU Xiao-Bo, born in 1977, Bachelor. Her research interests include cryptography and cryptographic chip.

PENG Min, born in 1985, M. S. Her research interests include cryptography and cryptographic chip.

ZHANG Huan-Guo, born in 1945, Ph. D., professor. His research interests include information security, cryptography, trusted computing and fault-tolerant computing.

Background

Masking is an effective and popular countermeasure to several types of side-channel attacks. As for the higher-order masking case, it tends to be more and more generic to resist any order SCAs. Although the ISW model proposed by Ishai, Sahai and Wagner could prove the security of masking schemes theoretically, there remains a gap between the practical security and theoretical security. This paper aims at improving the security of masking implementation, researches the risks of hardware design on two aspects: glitches and hardware design structure. We use RivP to prove there exist leakage in the above scene in theory. On this basis, we make some experiments on simulation power traces. Our analysis shows that the risk actually occurs in the design of provably security masking scheme. This capability can help designers directly modify their designs to make them more secure.

Our research team has successively used power, electromagnetic and fault analysis different structure of block ciphers in SCA for 10 years. We completed a prototype system about side-channel analysis, protection and testing.

This research is supported by the National Natural Science Foundation of China (61472292, 61332019), the National Basic Research Program (973 Program) of China (2014CB340601) and the Key Technology Research of New-Generation High-Speed and High-Level Security Chip for Smart Grid (526816160015). It aims at putting forward a set of lightweight countermeasure scheme against side-channel analysis, to achieve the unification of light weight and security.