

基于在线计数的改进快速泄露评估

时争光^{1,2}, 郑梦策^{1,2}, 胡红钢^{1,2}, 俞能海^{1,2}

(1. 中国科学院电磁空间信息重点实验室, 安徽 合肥 230027;

2. 中国科学技术大学网络空间安全学院, 安徽 合肥 230027)

摘要: 快速泄露评估 (FLA, fast leakage assessment) 在 2017 年被提出, 利用“计数”的思想极大地提升了计算效率。首先对快速泄露评估的性能进行分析, 然后提出基于在线计数的方法对评估流程进行改进, 并使用 FPGA 对该方法进行实现。根据实验结果对比, 所提方法不仅可以减少评估中所需要占用的内存, 而且能够缩短评估所用时间, 使评估效率得到明显提升。

关键词: 泄露评估; 在线处理; 高效计算; 侧信道防护

中图分类号: TP309

文献标识码: A

doi: 10.11959/j.issn.2096-109x.2021017

Improved fast leakage assessment based on online counting

SHI Zhengguang^{1,2}, ZHENG Mengce^{1,2}, HU Honggang^{1,2}, YU Nenghai^{1,2}

1. Key Laboratory of Electromagnetic Space Information, Chinese Academy of Science, Hefei 230027, China

2. School of Cyber Science, University of Science and Technology of China, Hefei 230027, China

Abstract: Fast leakage assessment (FLA) was proposed in 2017, which made use of the idea of "counter" and significantly improves the computing efficiency. The performance of FLA was analyzed, an online-counting based method was proposed to improve the evaluation flow, and the method was also validated via FPGA implementation. According to the comparisons of experimental results, the proposed method could not only decrease the required memory but also lower the runtime of the evaluation flow. Hence it can improve the evaluation efficiency significantly.

Keywords: leakage assessment, online processing, efficient computation, side-channel countermeasures

收稿日期: 2020-10-22; 修回日期: 2021-01-18

通信作者: 胡红钢, hghu2005@ustc.edu.cn

基金项目: 国家自然科学基金 (61632013, 61972370, 62002335); 中央高校基本科研业务费专项资金 (WK3480000007)

Foundation Items: The National Natural Science Foundation of China (61632013, 61972370, 62002335), Fundamental Research Funds for Central Universities in China (WK3480000007)

论文引用格式: 时争光, 郑梦策, 胡红钢, 等. 基于在线计数的改进快速泄露评估[J]. 网络与信息安全学报, 2021, 7(2): 141-150.

SHI Z G, ZHENG M C, HU H G, et al. Improved fast leakage assessment based on online counting[J]. Chinese Journal of Network and Information Security, 2021, 7(2): 141-150.

1 引言

侧信道分析攻击是目前密码学领域一个比较热门的课题,它揭示了密码算法本身的安全性与实际的安全性之间的差距,即使是理论上安全的密码算法,在实际运行过程中也不一定是安全的,它可能会存在其他形式的安全薄弱点,如侧信道信息泄露。侧信道分析技术始于 Kocher 等^[1]在 1999 年提出的差分能量分析,他们发现密码设备运行过程中能量消耗与密码算法的中间值之间存在联系,原因是密码设备的某些操作在处理“0”和“1”时会产生不同的能量消耗,而这些不同可能与密钥有关系。因此,攻击者可以通过侧信道信息间接获取到密码设备的中间值信息,从而破解出设备的密钥。

侧信道分析技术发展至今,给密码设备的安全性带来了极大的威胁,在 FIPS 140-3 标准草案中已经明确要求商用密码设备需要能够抵抗侧信道分析攻击^[2]。出于安全需求,研究者提出了一些防护措施,最常见的两种方法是隐藏^[3]和掩码^[4],其目的是掩盖能量消耗的不同或者打破能量消耗与密码算法中间值之间的联系,而如何验证防护措施的有效性成为一个新的问题。

泄露评估是检验密码设备侧信道信息泄露和验证防护措施有效性中最常用的手段。Coron 等^[5]在 2001 年首次提出了泄露评估的概念,其目的是检验两组数据是否有明显的不同。目前的主流方法是由密码学研究机构在 2013 年提出的测试向量泄露评估(TVLA, test vector leakage assessment)^[2],该方法基于 t 检验,它精心选择输入数据得到两组能量曲线作为待测数据,然后使用 t 检验对两组数据进行两组非特定的 Fixed-VS.-Random 检验与 896 组特定的 Fixed-VS.-Fixed 检验,以判断这两组待测数据是否有明显的不同,TVLA 因其标准的评估流程和对泄露的敏感性成为广泛接受的评估方法。之后, Ding 等在文献[6]中提出了一些 t 检验的使用建议,可以加速采集过程并增强对泄露的检测能力。然而, t 检验的方法存在一些局限性,它只能在检测到泄露时声明泄露存在,却不能在未检测到泄露时声明泄露不存在,并且 t 统计值的计算受到能量曲线条数等因素的影响。

因此, Bache 等^[7]提出了一种基于置信区间的评估方法对 t 检验进行了扩展,并在此基础上提出了一个评估流程的框架^[8]。

无论是 t 检验还是置信区间的方法,都需要先得到两组待检验数据各自的均值和方差。因此,在数据量较大时,如何尽可能提升计算效率成为一个研究重点。传统的基于定义的方法计算均值和方差需要多次遍历整个数据集合,效率较低^[9]。2015 年, Schneider 等^[10]提出了基于增量计算的方法,该方法只需要遍历整个数据集合一次便可以迭代地更新数据的均值和方差;在 2017 年, Reparaz 等^[11]基于计数的思想,提出了快速泄露评估,该方法遍历一遍数据集合,将数据处理为计数矩阵的形式并存储,之后使用计数矩阵可以快速计算出均值和方差,比文献[10]的方法计算效率高了几个量级。

本文对快速泄露评估的性能进行了分析,探究影响其性能的因素并给出了相应的使用建议;之后基于在线计数的思想对快速泄露评估进行了改进,并将改进前后的评估方法使用 FPGA 进行了实现;根据实验结果,改进后的方法能够显著地减少评估所用的内存和时间,极大地提升评估效率。

2 预备知识

本节首先对主要使用的符号进行说明,然后对 t 检验和快速泄露评估进行简要的介绍。

2.1 符号说明

本文主要使用的符号及其说明见表 1,在本文中, Fixed-VS.-Random 检验所使用的两组待检验数据集合分别被记为 $\mathbf{D}^f = \{T_i^f\}_{i=1}^{N_0}$ 和 $\mathbf{D}^r = \{T_i^r\}_{i=1}^{N_1}$, 其中, $T_i^f = \{\tau_{i,j}^f\}_{j=1}^{N_p}$, $T_i^r = \{\tau_{i,j}^r\}_{j=1}^{N_p}$ 。

2.2 t 检验

t 检验是 TVLA 中常用的检验方法,它于 2011 年在文献[12]中首次被应用于泄露评估。 t 检验通过计算两组待检验数据的相似性,来检验设备的同一操作在处理不同数据时消耗的能量是否有明显的差异。对于两组待测数据, t 检验通常将它们取自同一分布作为零假设,之后通过计算得出一个概率来检验是否接受零假设。

表1 符号说明
Table 1 symbol description

符号名	说明
$\frac{\mathbf{D}^f}{\mathbf{D}^r}$	固定输入/随机输入对应的能量曲线集合
$\frac{T_i^f}{T_i^r}$	$\frac{\mathbf{D}^f}{\mathbf{D}^r}$ 中的第 i 条能量曲线
$\frac{N_0}{N_1}$	$\frac{\mathbf{D}^f}{\mathbf{D}^r}$ 中的能量曲线条数
$\frac{\tau_{i,j}^f}{\tau_{i,j}^r}$	$\frac{T_i^f}{T_i^r}$ 上第 j 个采样点的值
N_p	每条能量曲线上的采样点数量
$\frac{\mathbf{D}_{,j}^f}{\mathbf{D}_{,j}^r}$	$\frac{\mathbf{D}^f}{\mathbf{D}^r}$ 矩阵的第 j 列
$\frac{\mathbf{H}^f}{\mathbf{H}^r}$	$\frac{\mathbf{D}^f}{\mathbf{D}^r}$ 对应生成的计数矩阵
$\frac{\mathbf{H}_{,j}^f}{\mathbf{H}_{,j}^r}$	$\frac{\mathbf{H}^f}{\mathbf{H}^r}$ 矩阵的第 j 列
$\frac{M_j^f}{M_j^r}$	$\frac{\mathbf{D}_{,j}^f}{\mathbf{D}_{,j}^r}$ 中数据的均值
$\frac{S_j^f}{S_j^r}$	$\frac{\mathbf{D}_{,j}^f}{\mathbf{D}_{,j}^r}$ 中数据的方差

具体而言,对于两组待检验数据 $\mathbf{D}_{,j_0}^f$ 与 $\mathbf{D}_{,j_0}^r$, 它们分别是 $\mathbf{D}^f$ 和 $\mathbf{D}^r$ 中第 j_0 列,代表第 j_0 个采样点对应的操作在处理不同输入时的能量消耗,若令 $M_0 = M_{j_0}^f$ 与 $M_1 = M_{j_0}^r$ 分别代表两组数据的均值,令 $S_0 = S_{j_0}^f$ 与 $S_1 = S_{j_0}^r$ 分别代表两组数据的方差,使用 t 检验, t 统计值与自由度 ν 分别可以用如下公式计算。

$$t = \frac{M_0 - M_1}{\sqrt{\frac{S_0}{N_0} + \frac{S_1}{N_1}}}, \quad \nu = \frac{\left(\frac{S_0}{N_0} + \frac{S_1}{N_1}\right)^2}{\frac{\left(\frac{S_0}{N_0}\right)^2}{N_0 - 1} + \frac{\left(\frac{S_1}{N_1}\right)^2}{N_1 - 1}} \quad (1)$$

评估者通常将 $|t| = 4.5$ 作为一个粗略的阈值^[10],即如果在 j_0 处得到 $|t| > 4.5$,那么不需要考虑自由度,评估者就能以 0.999 99 的置信概率拒绝零假设,即两组数据存在明显差异,应用于泄露评估中,代表待检测设备 (DUT, device under test) 存在泄露。

2.3 快速泄露评估

根据式(1), t 检验需要提前得到两组待测数据各自的均值和方差,以 $\mathbf{D}_{,j_0}^f$ 为例,它代表 $\mathbf{D}^f$ 中的第 j_0 列,包含 N_0 个元素,根据定义,其均值 $M_{j_0}^f$ 和方差 $S_{j_0}^f$ 的计算方式分别为

$$M_{j_0}^f = \frac{1}{N_0} \sum_{i=1}^{N_0} \tau_{i,j_0}^f, S_{j_0}^f = \frac{1}{N_0 - 1} \sum_{i=1}^{N_0} (\tau_{i,j_0}^f - M_{j_0}^f)^2 \quad (2)$$

考虑到该方法需要遍历整个数据集多次,当数据量较大时,计算效率较低,Reparaz 等^[11]在 2017 年提出快速泄露评估的方法,该方法的核心思想是“计数”。根据前文说明,固定输入对应的原始数据 $\mathbf{D}^f$ 由 N_0 条能量曲线构成,每条曲线包含 N_p 个 Q 比特采样点,则该方法对应地构建一个计数矩阵 $\mathbf{H}^f[2^Q][N_p]$,该矩阵由 2^Q 行 N_p 列构成,其中每个元素的横坐标对应 $[0, 2^Q - 1]$ 的一个特定采样值,纵坐标对应时间采样点,而元素的值代表所有能量曲线中该采样值在对应时间采样点出现的次数。具体而言,当处理曲线在 T_i^f 时,对于其上每个采样点 $\tau_{i,j}^f$,将 $\mathbf{H}^f[\tau_{i,j}^f][j]$ 的值增加 1。

当遍历一遍整个数据集之后,原始的数据都转化为计数矩阵的形式,此时, j_0 采样点的均值和方差计算公式如下。

$$M_{j_0}^f = \frac{1}{\sum_{i=0}^{2^Q-1} \mathbf{H}^f[i][j_0]} \sum_{i=0}^{2^Q-1} i \cdot \mathbf{H}^f[i][j_0] \quad (3)$$

$$S_{j_0}^f = \frac{1}{\left(\sum_{i=0}^{2^Q-1} \mathbf{H}^f[i][j_0]\right) - 1} \cdot \sum_{i=0}^{2^Q-1} (i - M_{j_0}^f)^2 \cdot \mathbf{H}^f[i][j_0] \quad (4)$$

对于随机输入对应的原始数据 $\mathbf{D}^r$,该方法构建计数矩阵 $\mathbf{H}^r$,之后数据的处理以及均值和方差的计算方式与 $\mathbf{D}^f$ 类似。这样,均值和方差的计算可以从对原始数据的处理转换到对计数矩阵的处理,根据 Reparaz 等的实验结果,从原始数据到计数矩阵的转换效率很高,且当原始数据规

模很大时,该方法可以极大地提升计算效率。

3 关于快速泄露评估的性能分析

基于 TVLA 的泄露评估方法通常采用分步的策略来部署,其流程主要分为以下几个阶段。

采集阶段:评估人员制定评估策略,采集密码设备运行时产生的能量曲线 T_i^f 与 T_i^r , 由于使用的采集设备内存有限,通常需要分多次采集。

发送阶段:分多次将采集到的能量曲线从采集设备发送到 PC 上并集中存储为 $\mathbf{D}^f$ 与 $\mathbf{D}^r$ 。

处理阶段:在 PC 上通过特定评估方法进行处理并得出评估结果。不同评估方法的效率不同主要体现在处理阶段,快速泄露评估的思想是利用“计数”的方法,将采集到的原始数据矩阵 $\mathbf{D}^f[N_0][N_p]$ 与 $\mathbf{D}^r[N_1][N_p]$ 转化为计数矩阵 $\mathbf{H}^f[2^Q][N_p]$ 与 $\mathbf{H}^r[2^Q][N_p]$, 因此,快速泄露评估的处理阶段可以分为两个部分。

预处理阶段:遍历原始数据矩阵将其转化为对应计数矩阵。

计算阶段:基于所得计数矩阵计算各列数据的均值和方差,并进行 t 检验。

计数矩阵的行数是固定的,因此,当原始数据规模很大时,该方法可以极大地提升计算效率。本节将对影响快速泄露评估性能的因素进行分析,并通过一些实验进行验证,在实验中,快速泄露评估的性能用其整个处理阶段所消耗的时间(记为 t^{process})进行量化,最后根据实验结果给出结论,并提出一些使用建议。

3.1 理论分析与实验

考虑原始数据矩阵 $\mathbf{D}^f$ 与预处理后对应的计数矩阵 $\mathbf{H}^f$, 两个矩阵的列数 N_p 相同,而行数从原本的能量曲线数 N_0 变为采样点所有可能取值的个数 2^Q 。从快速泄露评估的处理方法来看,原始数据矩阵的规模会对性能造成影响,因为当 N_0 增大时,预处理阶段所用的时间会增加,而由于快速泄露评估对于每一列的处理是相互独立的,且处理方式相同,因此,当 N_p 增大时,预处理阶段和计算阶段所用的时间都会增加。考虑到在实际的泄露评估场景中, N_0 由评估人员制定的评估策略所决定,而 N_p 由密码设备上算法的运

行周期和采样设备的采样率所决定,即对于既定的评估策略和硬件环境,原始矩阵 $\mathbf{D}^f$ 的规模是确定的,因此,对于 N_0 和 N_p 两个因素的分析意义不大。

此外,通过对快速泄露评估的流程进行分析,得知还有两个因素会影响到它的性能。由于快速泄露评估对每列数据的处理独立且处理方式相同,接下来将以 $\mathbf{H}^f$ 中的第 j_0 列 $\mathbf{H}_{j_0}^f$ 为例进行分析和实验,所得的结论也适用于 $\mathbf{H}^f$ 中其余的列和 $\mathbf{H}^r$ 中所有的列。为了便于书写,在之后的内容中 $\mathbf{H}_{j_0}^f$ 被记为 $\mathbf{H}_0 = \{h_i\}_{i=0}^{2^Q-1}$, 同时 $\mathbf{H}_0$ 所对应的原始数据被记为 $\mathbf{D}_0 = \{\tau_i\}_{i=1}^{N_0}$, 则两个影响因素及分析如下。

(1) 计数矩阵元素的变量类型, h_i 的类型不同,会影响到所占用的字节数 L , 进而影响到计算效率。例如,当 h_i 被定义为 2 字节无符号整数、4 字节无符号整数和 8 字节无符号整数时,即 L 的值分别是 2、4 和 8 时,在处理该元素时,因为变量长度不同,式(3)与式(4)计算所用时间会有所不同。

(2) 采样点的采样精度,即 Q 的大小,因为计数矩阵的行数为 2^Q , 所以采样精度的增加会影响到所构建的计数矩阵的规模。对于 $\mathbf{H}_0$ 来说, Q 值的增加会使集合中的元素个数以指数量级增长,之后计算阶段需要遍历的次数和所用时间会随之增加。

为了验证这两个因素对快速泄露评估性能的影响,本节分别进行了两组实验,其中每组实验都会重复 10 次,并记录每次对应的 t^{process} 值,最后使用得到的平均值进行分析。在实验中, $N_0 = 100\,000$, 两组实验说明及结果如下。

实验 1 设置采样精度为 10 bit, 即 $Q = 10$, 以随机数的方式生成 $\mathbf{D}_0$ 中 $[0, 2^Q - 1]$ 的 100 000 个元素,之后将 h_i 的数据类型分别设置为 2 字节无符号整数、4 字节无符号整数和 8 字节无符号整数。通过实验,3 种类型对应的 t^{process} 平均值分别为 57.2 ms、59.5 ms 和 62.2 ms。

实验 2 设置 h_i 的数据类型为 4 字节无符号整数,将 Q 分别设置为 8、10 和 12,以随机数的方式生成 $\mathbf{D}_0$ 中 $[0, 2^Q - 1]$ 的 100 000 个元素。通过实验,3 种 Q 值对应的 t^{process} 平均值分别为 56.4 ms、59.5 ms 和 63.2 ms。

需要注意的是，每次重复实验都会重新生成原始数据，生成数据所用时间不包括在 t^{process} 中。

3.2 结论与建议

根据实验结果，本节得到一些结论，并给出如下使用建议。

根据实验 1，当计数矩阵元素所占字节数 L 增加时， t^{process} 随之增加，然而在使用时 L 并非设置得越小越好，因为 h_i 的值记录着采样值 i 出现的总次数，而其变量类型决定了能取得的最大值，考虑最极端情况， D_0 中所有 N_0 个元素值相同，为了实现正常计数， L 应满足 $2^{8L} \geq N_0$ 。考虑到评估中所采集的能量曲线数通常不会超过百万量级，该值小于 2^{32} ，因此 4 字节无符号整数类型足以满足计数需求。在实际使用时，评估者可以根据是否满足 $2^{16} \geq N_0$ ，选择使用 2 字节无符号整数或 4 字节无符号整数类型。

根据实验 2，当采样精度 Q 值增加时， t^{process} 随之增加（ Q 值会影响计数矩阵的规模）。在实际使用时， Q 值是由采样设备的采样精度决定的，常用的采样精度为 8 bit~12 bit，对于特定的采样设备， Q 为定值，虽然不会影响到快速泄露评估的性能，但会影响到是否采用快速泄露评估的方法，因为快速泄露评估的思想是将包含 N_0 个元素的 D_0 通过计数处理成包含 2^Q 个元素的 H_0 ，而均值和方差均采用遍历计算的方式，因此，当 $N_0 \leq 2^Q$ 时，使用式(2)基于定义直接计算的方法（因为不需要预处理阶段，反而比快速泄露评估的效率），为此，本节进行了简单的验证：设置 $Q=10$ ， $N_0=1024$ ，原始数据为 N_0 个随机数， h_i 的类型使用 4 字节无符号整数，首先基于式(2)直接计算均值和方差，然后使用快速泄露评估来预处理数据，再基于等式(3)和式(4)计算均值和方差，两种方法 10 次实验平均消耗的时间分别为 0.9ms 和 2.0ms，由此可知，在原始能量曲线数较少，满足 $N_0 \leq 2^Q$ 时，使用快速泄露评估的方法反而不如直接基于定义计算效率高。

综上，在使用快速泄露评估时，根据制定的评估策略，本文有两个使用建议，① 根据 N_0 与 2^Q 的大小关系，考虑是否使用快速泄露评估；② 根据 2^{16} 与 N_0 的大小关系，考虑计数矩阵的变量

类型使用 2 字节无符号整数或 4 字节无符号整数。

4 改进方法与方案实现

第 3 节中提到，基于 TVLA 的评估方法不同主要在于处理阶段，而快速泄露评估的处理阶段分为两个部分，即预处理阶段和计算阶段，在文献[11]中，Reparaz 等通过实验得出，对于已采集好的数据，在搭载酷睿 i5@3.3 GHz 处理器的桌面工作站上，快速泄露评估的预处理阶段可以达到 305 MB/s 的带宽，当预处理阶段结束后，计算阶段处理每列数据所用的时间是一个恒定值，与 N_0 无关，因此，当原始数据的能量曲线数很多时，快速泄露评估能够很大程度上提升计算效率。

基于 TVLA 的评估方法在采集能量曲线时通常需要分多次采集。例如，对于数据集 D^f ，评估者需要采集 N_0 条能量曲线，每条曲线上有 N_p 个 Q bit 采样值，那么存储 D^f 需要占用 $N_0 N_p Q$ bit 内存，该值随着 N_0 变化，当 N_0 很大时（如在百万量级），由于采集设备的内存有限，采集阶段通常需要分多次进行，并将每次的数据发送到 PC 上进行集中存储，这样不仅操作烦琐，而且在采集设备和 PC 上需要占用大量的内存，快速泄露评估也不例外，因此，该方法不适用于可用内存有限的评估场景，如片上泄露监控，因此，本节将“计数”思想与在线处理思想相结合，对该评估方法进行了改进。

4.1 改进方法

快速泄露评估的核心思想是将原始数据矩阵 D^f 转化为计数矩阵 H^f ，而 H^f 的规模不随 N_0 而改变，因此它能够在 N_0 很大时有效地提升计数效率，即快速泄露评估的关键是得到 H^f ，而如果能够在采集阶段就将原始数据转化为计数矩阵，那么不仅保留了快速泄露评估的特点，而且能够极大地减少存储数据所需的内存，从这个想法出发，本文提出了一种基于在线计数的方法对采集阶段进行了改进，即在采集设备上构建计数矩阵 $H^f[2^Q][N_p]$ 并初始化，之后每采集到一个采样点 $\tau_{i,j}^f$ 时，将 $H^f[\tau_{i,j}^f][j]$ 的值加 1。

该方法将原来的预处理阶段与采集阶段结合，这样无论是在采集设备上还是在 PC 上都只

需要存储 $\mathbf{H}^f$ 即可, 从占用内存看, 其占用的内存为 $2^Q N_p 8L$ bit, 当 N_p 确定时, 该值是固定的, 不会随着所采集能量曲线的条数而变动, 因此, 当 N_0 很大时, 该方法可以很大程度上减少存储数据所需的内存; 从运行时间来看, 该方法降低了需要发送的数据规模, 从而减少了发送阶段需要的时间, 并且之后无须单独的预处理阶段, 因此整体评估效率会得到一定的提升。

4.2 方案流程

为了进行对比分析, 本节将改进前后的快速泄露评估方案分别进行了实现, 并以固定输入对应的 $\mathbf{D}^f$ 和 $\mathbf{H}^f$ 为例给出了其具体的评估流程。方案定义及流程说明如下。

(1) Design_f : 原始的快速泄露评估方案, 其流程如下。

① 采集阶段: 使用采集设备采集 $N_0 N_p$ 个采样点, 对于每个采样点 $\tau_{i,j}^f$, 将其写入采集设备对应内存中。

② 发送阶段: 将采集数据发送到 PC 上并集中存储为 $\mathbf{D}^f$ (由于采集设备的内存有限, 阶段①和阶段②通常需分多次运行)。

③ 预处理阶段: 将采集到的原始数据 $\mathbf{D}^f$ 预处理为计数矩阵 $\mathbf{H}^f$ 。

④ 计算阶段: 根据计数矩阵 $\mathbf{H}^f$ 计算每个时间采样点的均值和方差, 并进行 t 检验。

(2) Design'_f : 基于在线计数方法改进后的快速泄露评估方案, 其流程如下。

① 采集阶段: 使用采集设备采集 $N_0 N_p$ 个采样点, 对于每个采样点 $\tau_{i,j}^f$, 将采集设备上 $\mathbf{H}^f[\tau_{i,j}^f][j]$ 对应内存的值加 1。

② 发送阶段: 将计数矩阵 $\mathbf{H}^f$ 发送到 PC 上。

③ 预处理阶段: 无。

④ 计算阶段: 根据计数矩阵 $\mathbf{H}^f$ 计算每个时间采样点的均值和方差, 并进行 t 检验。

需要注意的是, Design_f 和 Design'_f 需要事先的准备阶段, 如 Design_f 需要确定每条曲线上的采样点数量 N_p , 并根据采集设备的内存容量确定设备一次能够采集并存储的曲线条数; 而 Design'_f 需要确定每条曲线上的采样点数量

N_p , 根据采集设备的采样精度 Q , 在采集设备上构建二维数组 $\mathbf{H}^f[2^Q][N_p]$ 并初始化, 以及根据目标采集曲线条数 N_0 确定数组元素的变量类型, 即确定 L 。准备阶段的工作只需要进行一次, 因此在探究方案性能时, 该部分时间不予考虑。

5 性能探究

本节探究在线计数方法对占用内存和运行时间带来的提升, 并以固定输入对应的 $\mathbf{D}^f$ 和 $\mathbf{H}^f$ 为例来进行实验, 最终所得结论也适用于随机输入对应的 $\mathbf{D}^r$ 和 $\mathbf{H}^r$ 。

5.1 实验环境

考虑到方案实现需要几个核心功能, 即数据采集、在线处理 (即可编程) 和数据发送功能, 本文选择了一块 FPGA 作为开发平台, 其型号为赛灵思 Virtex-7XC7VX485T-2FFG1761, 简称 VC707 (如图 1 所示), 在该 FPGA 上主要实现两种方案采集阶段和发送阶段的功能。



图 1 实验所用 VC707FPGA 开发板
Figure 1 The VC707 FPGA board

实验中所选用的待测设备 (DUT) 为 ChipWhispererCW303 测试板, 在其上运行的是无防护的串行 AES 算法, 算法运行过程中测量板上的电压变化作为能量曲线。在 VC707 上有一个 1M 采样率的 ADC 模块——XADC, 实验使用专用外部差分模拟输入信道来采集能量曲线。在实验中, XADC 被设置工作在双端输入, 连续采样模式, 在每次采样的转换阶段, 它会在状态寄存器中产生一个 16 bit 的值, 其中高 12 bit 为有效值, 为了实验方便, 取其高 8 bit 作为采样结果。在发送阶段, FPGA 上的 UART 模块将所采集的数据发送到 PC 上。

为了实现 Design_f 和 Design'_f , 本文首先参考文献[13]使用 Vivado2013.3 在 VC707 上围绕

MicroBlaze 微处理器搭建了一个片上系统 (SoC, system on a chip), 其中包含了内存等基础模块以及评估方案需要使用的 XADC 模块和 UART 模块, 然后基于该 SoC 编写函数实现采集功能、在线计数功能以及数据发送等功能, 最后, 在一台搭载了酷睿 i5-4460 CPU @ 3.2 GHz 的台式计算机上使用 VisualStudio2017 实现处理阶段的功能, 该部分的代码使用 C++ 语言编写, 并以 Debug 版本运行 (Release 版本运行速度过快, 会导致测量运行时间时误差较大), 实验中采用 Windows 内置 clock 函数实现计时功能。

5.2 占用内存

根据之前的分析, Design_f 存储原始数据矩阵 D^f 需要 $N_0 N_p Q$ bit 内存, Design'_f 存储计数矩阵 H^f 需要 $2^Q N_p 8L$ bit 内存, 根据实验设置, 每一个采样值在 FPGA 上需要占用 8 bit 内存, 即 $Q=8$, 计数矩阵的类型定义为 4 字节无符号整数, 即 $L=4$, 这样 Design_f 和 Design'_f 存储采集数据所需的内存分别为 $N_0 N_p$ byte 和 $1024 N_p$ byte。

由此可知, 当 $N_0 > 1024$ 时, Design_f 需要的内存将会超过 Design'_f , 而根据之前的分析, $L=4$ 足以满足百万条曲线级别计数需求, 因此, 在线计数的方法会使存储数据所需要的内存减少很多, 并且, 对于确定的 N_p , $1024 N_p$ 是一个定值, 若令 $N_p=1024$, 即每条曲线上采集 1 024 个点, 那么 Design'_f 存储 H^f 需要的内存仅为 1 MB。在实际采集时, 若分开采集固定输入与随机输入的能量曲线, 由于 H^r 与 H^f 有相同的 N_p 和 Q , 则 H^r 复用 H^f 所用二维数组内存即可; 若采用交叉采集的方式, 则需要再构建一个相同规模的二维数组用于存储 H^r , 这样所需要的内存最多为 2 MB, 这对于采集设备来说是可以接受的。这样, 评估者在采集阶段可以一次采集所有能量曲线, 不必因内存有限而采用多次采集的方法, 极大地简化了采集过程; 另外, 该方法仅需有限的内存即可完成对百万级别条曲线的泄露评估, 因此可以满足可用内存有限场景下的评估需求。

5.3 运行时间

在线计数的方法将预处理阶段的操作加入了

采集阶段, 这样可以直接获得计数矩阵 H^f , 一方面, 因为 H^f 通常比 D^f 的数据规模小很多, 所以在发送阶段将数据发送到 PC 上所需要的时间会减少; 另一方面, 采集阶段中已经得到 H^f , 所以不再需要单独的预处理阶段, 可以直接进行计算阶段, 综上, 对于相同的原始数据, 发送阶段和预处理阶段的运行时间得到降低。

考虑到在线计数对采集阶段也会造成影响, 本节对两种方案各个阶段的运行时间都进行了单独探究, 并对结果进行分析和对比。对于 $N_0 N_p$ 个 Q bit 采样值, 定义采集阶段的运行时间为 t^{collect} , 发送阶段的运行时间为 t^{send} , 预处理阶段的运行时间为 t^{pre} , 计算阶段的运行时间为 t^{cal} , 并定义从采集阶段开始到完成计算阶段所用的总时间为 t^{total} 。为了验证在线计数对运行时间带来的提升, 做了两组实验来探究 Design_f 和 Design'_f 的运行时间 t^{collect} 、 t^{send} 、 t^{pre} 和 t^{cal} 随 N_p 和 N_0 的变化情况。在实验中, 设置 $Q=8$, $L=4$, 两组实验说明如下。

实验 1 设置 $N_p=1000$, 令 N_0 依次取 100 到 2 000, 每次增加 100, 实验结果如图 2 所示。

实验 2 设置 $N_0=2000$, 令 N_p 依次取 100 到 1000, 每次增加 100, 实验结果如图 3 所示。

根据两组实验结果, 本文分别对各个阶段的运行时间进行单独分析。

(1) 采集阶段: 根据图 2(a)和图 3(a)可以发现, 采集阶段的运行时间 t^{collect} 随 N_0 和 N_p 均呈线性增长, 而因为 Design'_f 增加了在线计数的功能, 其采集阶段的运行时间有所增加, 根据实验结果, 对于一组确定的 N_0 和 N_p , Design'_f 较 Design_f 的 t^{collect} 增加了约 17.3%。

(2) 发送阶段: 发送阶段所用时间 t^{send} 基于硬件模块的发送速率, 且与所发送数据的规模呈线性关系。在实验中, Design_f 和 Design'_f 所需要发送的数据规模分别为 $N_0 N_p$ byte 和 $1024 N_p$ byte, 因此, 在图 2(b)中, 当 $N_p=1000$ 时, Design_f 的 t^{send} 随 N_0 线性增长, 而 Design'_f 的 t^{send} 几乎为恒定值, 且在 N_0 约为 1000 处被 Design_f 的 t^{send} 超越;

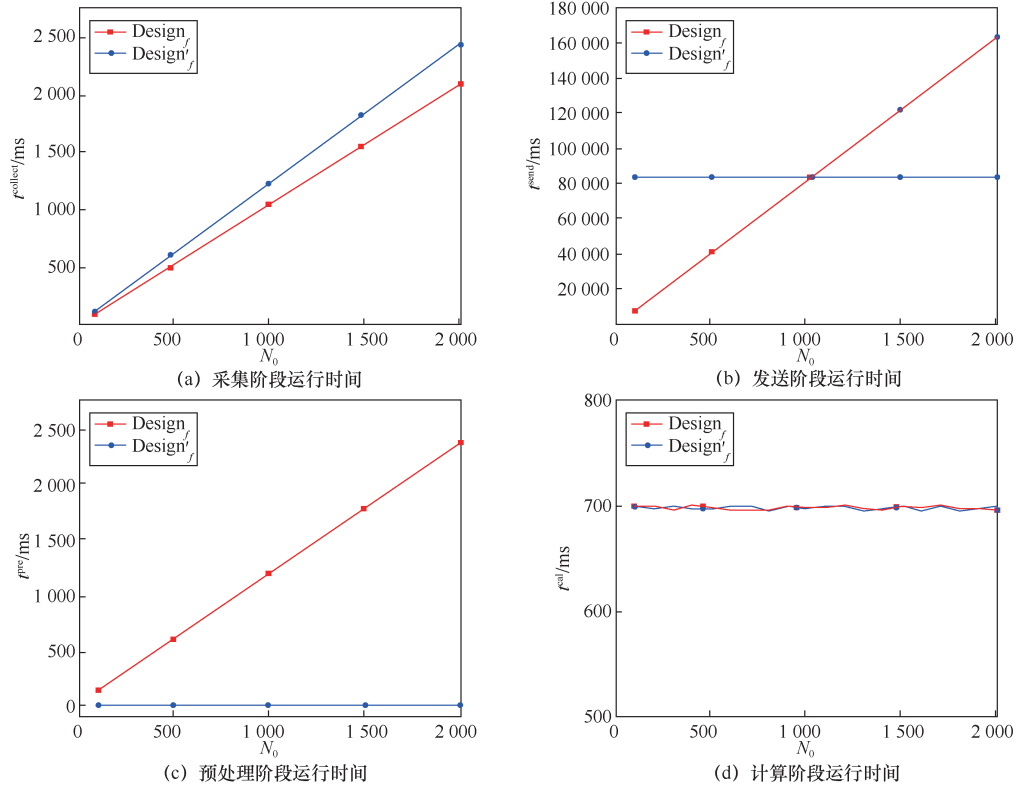


图2 在 $N_p = 1\ 000$ 时, 两种方案在不同 N_0 下各阶段的运行时间
Figure 2 Required time of each stage over N_0 with $N_p = 1\ 000$ in two designs

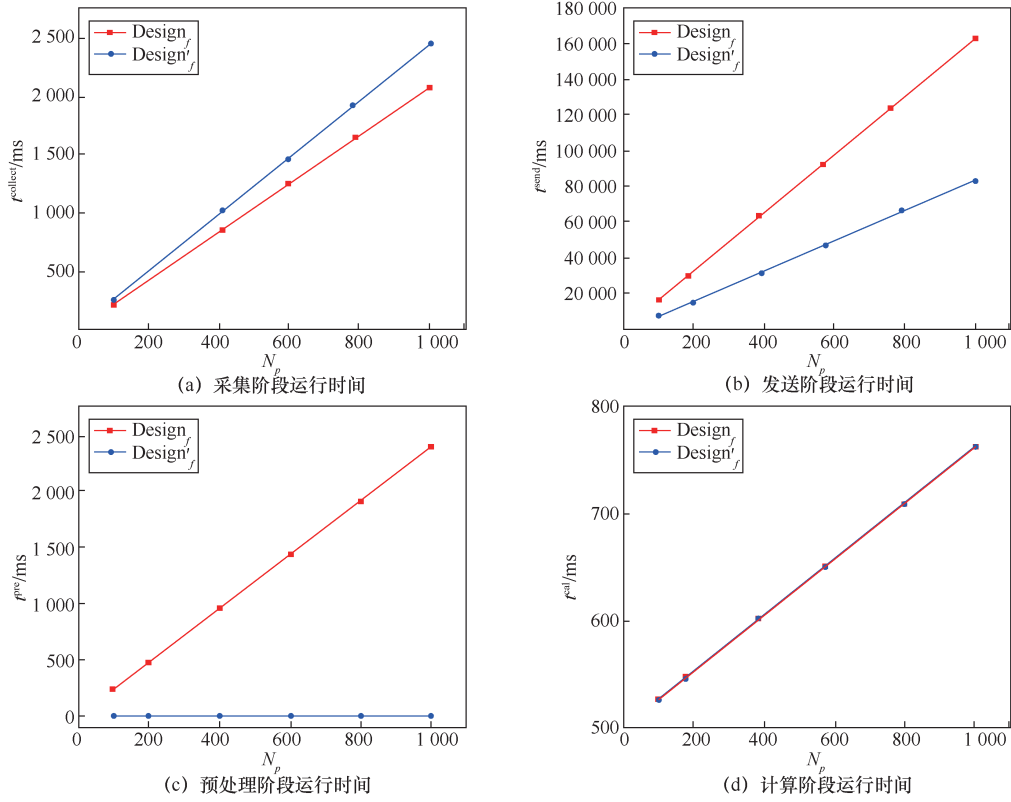


图3 在 $N_0 = 2\ 000$ 时, 两种方案在不同 N_p 下各阶段的运行时间
Figure 3 Required time of each stage over N_p with $N_0 = 2\ 000$ in two designs

在图 3(b)中, 当 $N_0 = 2\,000$ 时, 可以发现 Design_f 和 Design'_f 的 t^{send} 均随 N_p 呈线性增长, 且由于 $2000 > 1024$, 因此 Design_f 的 t^{send} 增长较快。在实验中, 所用 UART 模块的发送速度为 12 kbit/s。

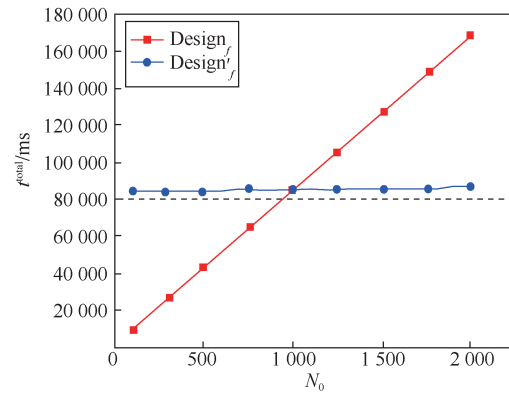
(3) 预处理阶段: 根据图 2(c)和图 3(c)可以发现, Design_f 的 t^{pre} 随 N_0 和 N_p 均呈线性增长, 而由于 Design'_f 不需要预处理阶段, 因此其 t^{pre} 值为 0。

(4) 计算阶段: 由于 Design_f 和 Design'_f 的计算阶段完全相同, 因此二者的 t^{cal} 也几乎保持一致, 根据之前分析, 在基于计数矩阵计算时, 快速泄露评估对 $\mathbf{H}^f$ 每一列的计算是相互独立的, 且计算方式相同, 而每一列的计算所用时间与 Q 相关, 与 N_0 无关, 因此, 在图 2(d)中, 两种方案的 t^{cal} 几乎为恒定值, 而在图 3(d)中, 两种方案的 t^{cal} 都随 N_p 呈线性增长。

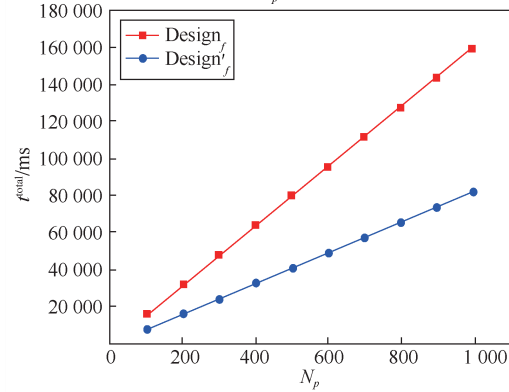
接下来, 分别将两种方案各个阶段的运行时间相加, 作为该方案整个评估过程的总运行时间 t^{total} 并进行对比分析, t^{total} 在两组实验中的结果如图(4)所示。

根据图 4(a), Design_f 和 Design'_f 的 t^{total} 大致上随 N_0 呈线性增长, 不同的是 Design_f 的 t^{total} 值起点较低, 增长较快, 而 Design'_f 的 t^{total} 值起点较高, 增长较慢, 这是因为对 Design'_f 来说, 计数矩阵 $\mathbf{H}^f$ 在采集阶段就已构建完毕, 无论需要采集的曲线条数 N_0 多大, $\mathbf{H}^f$ 数据规模都是固定的 $1024N_p$ byte, 当 N_p 固定时, 发送阶段、预处理阶段和计算阶段所用时间都是固定, 只有采集阶段的时间会随 N_0 改变; 而对 Design_f 来说, 除了计算阶段之外, 采集阶段、发送阶段和预处理阶段的时间都会随 N_0 呈线性增长。在文献[10]中提到, 采集设备和 PC 之间的通信速度通常会成为制约评估效率一个因素, 图 3(b)表明, Design_f 的 t^{send} 会随 N_0 快速增长。因此, 在添加了在线计数功能后, 虽然 Design'_f 比 Design_f 的采集阶段所用时间增加了 17.3%, 但是由于 Design'_f 的 t^{send} 为固定值, 且无须单独的预处理阶段, 所以 Design'_f

的 t^{total} 值起点较高, 但增长较慢, 而 Design_f 的 t^{total} 值增长较快, 并且在 N_0 大约为 1000 处 Design_f 的 t^{total} 值超过了 Design'_f 。



(a) t^{total} 在 $N_p=1\,000$ 时随 N_0 变化情况



(b) t^{total} 在 $N_0=2\,000$ 时随 N_p 变化情况

图 4 两种方案在不同 N_0 和 N_p 下的总运行时间
Figure 4 Total required time of two designs over N_0 and N_p

根据图 4(b), Design_f 和 Design'_f 的 t^{total} 值都随 N_p 呈线性增长, 这是因为根据实验 2 结果, Design_f 和 Design'_f 各个阶段的运行时间都随 N_p 呈线性增长, 结合图 4(a), 在 $N_0 = 2000$ 时, Design_f 处理每一列所用的时间较长, 因此在图 4(b)中 Design_f 的 t^{total} 值增长较快。

在实际使用中, N_p 的值依赖于评估策略和实验环境, 而 N_0 的值通常比 2000 大得多, 结合实验结果, 此时基于在线计数方法的 Design'_f 整个评估流程所用的时间明显少于 Design_f 。另外, 当 N_p 确定时, Design'_f 除了采集阶段之外, 其他阶段所用时间几乎是固定的, 因而其总运行时间比较稳定, 并且不需要预处理阶段, 因此, 即使

原始数据的能量曲线数较少, 仍然可以直接采用 Design'_f 进行评估。

6 结束语

本文首先分析了快速泄露评估能够提升评估效率的原因, 之后提出了两个影响其性能的因素, 分别是计数矩阵的变量类型和采样精度; 对这两个因素分别进行了分析和实验, 并根据结果给出了使用建议。

然后, 在保留快速泄露评估特点的基础上, 本文基于在线计数思想对其进行了改进, 通过将预处理阶段的操作加入采集阶段的方法, 可以在采集阶段直接获得相应的计数矩阵。这样不仅可以减少存储数据所需的内存, 简化采集流程, 使其适用于可用内存有限的评估场景, 而且可以在原始能量曲线数较多时显著减少评估所用时间, 提升整个评估流程的效率。

虽然在线计数带来了明显的提升, 但是有一些局限性, 因为所存储的是处理后的计数矩阵, 而非原始的能量曲线, 所以该方法在处理阶段只适用于使用快速泄露评估的方法来处理, 且后续难以进一步对原始数据进行分析 and 利用。此外, 当采集过程中出现错误时该方法不能发现和 处理, 因此该方法要求实验中有良好的采集环境。

参考文献:

- [1] KOCHER P, JAFFE J, JUN B. Differential power analysis[C]//Advances in Cryptology -CRYPTO'99. 1999: 388-397.
- [2] BECKER G, COOPER J, DEMULDER E, et al. Test vector leakage assessment (TVLA) methodology in practice[C]//International Cryptographic Module Conference 2013.
- [3] TIRI K, VERBAUWHEDE I. A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation[C]//Proceedings Design, Automation and Test in Europe Conference and Exhibition. 2004, 1: 246-251.
- [4] CNUDDÉ T D, REPARAZ O, BILGIN B, et al. Masking AES with $d+1$ shares in hardware[C]//Cryptographic Hardware and Embedded Systems-CHES. 2016: 194-212.
- [5] CORON J S, KOCHER P, NACCACHE D. Statistics and secret leakage[C]//International Conference on Financial Cryptography. 2001: 157-173.
- [6] DING A A. Simpler, faster, and more robust t-test based leakage detection[C]//Constructive Side-Channel Analysis and Secure Design. 2016: 163-183.
- [7] BACHE F. Confident leakage assessment—a side-channel evaluation framework based on confidence intervals[C]//Design, Automation Test in Europe Conference Exhibition-DATE. 2018. 1117-1122.
- [8] BACHEF. Evaluation of (power) side-channels in cryptographic implementations[J]. IT-Information Technology, 2019, 61(1): 15-28.
- [9] CHAN T F, GOLUB G H, LEVEQUE R J. Algorithms for computing the sample variance: analysis and recommendations[J]. American Statistician, 1983, 37(3): 242-247.
- [10] SCHNEIDER T, MORADI A. Leakage assessment methodology[C]//Cryptographic Hardware and Embedded Systems - CHES. 2015: 495-513.
- [11] REPARAZ O, GIERLICH B, VERBAUWHEDE I. Fast leakage assessment[C]//Cryptographic Hardware and Embedded Systems(CHES). 2017: 495-513.
- [12] GILBERT G, BENJAMIN J, JAFFE J, et al. A testing methodology for side-channel resistance validation [C]//NIST Non-Invasive Attack Testing Workshop, 2011.
- [13] UG960-7 series FPGA AMS targeted reference design[EB].

[作者简介]



时争光 (1994—), 男, 河南开封人, 中国科学技术大学硕士生, 主要研究方向为密码学、侧信道分析与泄露评估。



郑梦策 (1990—), 男, 浙江宁波人, 博士, 主要研究方向为应用密码学。



胡红钢 (1978—), 男, 四川彭州人, 中国科学技术大学教授、博士生导师, 主要研究方向为密码学、编码、网络安全。



俞能海 (1964—), 男, 安徽无为, 中国科学技术大学教授、博士生导师, 主要研究方向为多媒体数据处理与分析 和数字内容安全。