文章编号：2096-1618(2020)01-0001-06

基于巴特沃斯滤波算法的侧信道分析

于天凯[1]， 王 敏[1]， 王 燚[1]， 吴 震[1]， 杜之波[1]， 习 伟[2]
(1. 成都信息工程大学网络空间安全学院，四川 成都 610225；2. 南方电网科学研究院有限公司，广东 广州 510080)

摘要：基于巴特沃斯滤波算法在现场可编程逻辑门阵列(FPGA)侧信道攻击中的使用，主要利用巴特沃斯滤波算法对功耗曲线进行预处理，然后用神经网络模型代替传统模板攻击的统计模型对功耗曲线进行侧信道攻击。该算法对模板攻击，随机方法，深层感知器以及深层卷积神经网络的功耗曲线预处理具有普适性，在实验部分针对 DPA CONTEST V2 数据进行了 4 种侧信道方法的分析，实验数据表明该方法提高了可攻击的信噪比，同时提高了侧信道攻击的成功率。

关 键 词：FPGA；巴特沃斯滤波；侧信道攻击
中图分类号：TN918 **文献标志码**：A
doi：10.16836/j.cnki.jcuit.2020.01.001

0 引言

随着物联网技术的日新月异，带有加密模块的嵌入式设备成为保护敏感数据的重要因素。尽管这些设备中使用的算法在数学理论上是安全的，但攻击者仍然可以对硬件工作时泄露的功耗分析来恢复密钥[1-3]。近年来随着深度学习领域的爆炸式发展，许多研究人员利用深度学习来进行侧信道分析。

侧信道分析是利用加密设备在运行中产生的时间消耗、功率消耗等泄露的敏感信息对密钥进行攻击的一种方法。Paul Kocher[4]提出简单能量分析攻击(simple power analysis，SPA)，为侧信道攻击奠定了基础。Paul Kocher 等[5]发现了差分能量分析攻击(differential power analysis，DPA)，利用统计规则进行侧信道分析。Suresh Chari[6]提出了模板攻击(template attack，TA)，大大加快了侧信道攻击的时间；Werner Schindler 等[7]提出了随机方法(stochastic approach，SA)，模板攻击和随机方法是最常用的侧信道分析方法，它们依据被攻击设备功耗的噪音模型符合多元高斯分布来建模。在实际环境中，这样的模型假设趋向理想化；基于噪音的多元高斯模型常带来奇异矩阵求逆问题。而使用多层感知器(multi-layer perceptions，MLP)[8]和卷积神经网络(convolutional neural networks，CNN)[9]进行侧信道分析，就不会受制于噪音模型假设。利用计算机的特征识别能力替代噪音建模，这样的尝试已经在侧信道攻击中取得不错成绩。在功耗曲线的预处理中，提高信噪比也是提高侧信道攻击效率的关注点之一[10]。利用巴特沃斯滤波算法进行数据预处理，对比有监督学习方法的实验后发现，巴特沃斯滤波算法不仅提高信噪比，而且提高了侧信道攻击效率。

收稿日期：2019-09-02
基金项目：国家重点研发计划资助(2018YFB0904900、2018YFB0904901)；“十三五”国家密码发展基金资助项目(MMJJ20180244)；四川省重点研发项目(2019YFG0096)

1 侧信道分析

假如被攻击设备的泄露模型已知，那么使用相关系数侧信道攻击是可取的方法。如果无法找到设备泄露模型中的相关性，可以尝试使用有学习方法构建设备的泄漏模型，然后进行攻击。

1.1 模板攻击

经典模板攻击假设功耗曲线泄露模型符合多元高斯分布，对加密过程中泄露的特定信息建立噪音模板。噪音模板反映了泄露信息在能耗上的特征。模板攻击分为：训练阶段和攻击阶段。

1.1.1 训练阶段

(1)选择兴趣点(points of interest，POI)：对 W 个操作$\{O_1,O_2,\cdots,O_W\}$，收集 W 个训练功耗曲线集合$\{S_1,S_2,\cdots,S_W\}$，每个集合均包含 I 条功耗曲线。计算各功耗曲线集的均值功耗曲线$\{M_1,M_2,\cdots,M_W\}$。计算均值功耗曲线的差分：$\delta=\sum_{i<j}(M_i-M_j)$，选择前 n 个差分值最大的点作兴趣点$\{P_1,P_2,\cdots,P_n\}$。

(2)功耗曲线向量化：对功耗曲线集合 S_i 中的任意一条功耗曲线 T，其噪音向量 X 为

$$X=(T[P_1]-M_i[P_n],\cdots,T[P_n]-M_i[P]_n]) \quad (1)$$

(3)计算模板参数：功耗曲线集合 S_i 的模板 TP_i 的多元高斯密度分布为

$$\begin{cases} p(X|TP_i)=\dfrac{1}{\sqrt{(2\pi)^n|\boldsymbol{\Sigma}_i|}}\exp\left(-\dfrac{1}{2}X^{\mathrm{T}}\boldsymbol{\Sigma}_i^{-1}X\right) \\ \boldsymbol{\Sigma}_i[j,k]=\dfrac{1}{n-1}X_j^{\mathrm{T}}X_k, X_j,X_k\in S_i \end{cases} \quad (2)$$

1.1.2 攻击阶段

经典的模板攻击在攻击阶段,采用一条攻击功耗曲线 T,利用求最大似然概率的方法来判断 T 所属的模板。首先将 T 转换为模板 i 中的向量 y

$$y=(T[P_1]-M_i[P_n],\cdots,T[P_n]-M_i[P_n]) \quad (3)$$

似然率为

$$p(y\mid T_i)=\frac{1}{\sqrt{(2\pi)^n\mid\Sigma_i\mid}}\exp\left(-\frac{1}{2}y^{\mathrm{T}}\Sigma_i^{-1}y\right) \quad (4)$$

取最大似然率的模板作为功耗曲线 T 的匹配模板,即 $\hat{T}=\mathrm{argmax}[p(y_i\mid T_i)]$。

操作 O_i 实际上表示它所泄露的信息。根据 O_i 的信息(如 AES-128 加密算法中的 S 盒置换输出的汉明重量)与密钥的关系,可以从 $\hat{O}$ 推算出一个或多个子密钥。

1.2 随机方法

随机方法攻击同样包含训练阶段,但与模板攻击不同的是:随机方法并不是建立多个模板,而是训练出一个概率识别器,用于预测猜测密钥正确的概率。密钥概率识别器的训练向量包含了选定中间值的所有可能噪音向量。$I_t(\mu,k)$ 对所产生的噪音向量具有识别功能。攻击时使用识别器,计算攻击功耗曲线相对于各个猜测密钥噪音向量的概率,其中概率最大的密钥即为攻击的结果。

1.2.1 估算能耗转换系数

使用 N_1 条功耗曲线,计算 POI 位置的“侧信道泄露的数据依赖部分”,亦称“位能耗转化系数向量”。随机模型假设 t 时刻的能耗包含数据有用部分和白噪音两个部分:

$$I_t(\mu,k)=h_t(\mu,k)+R_t \quad (5)$$

其中,$I_t(\mu,k)$ 是明文 μ 和密钥 k 在功耗曲线的 t 时刻产生的能耗,其中包括数据相关的能耗 $h_t(\mu,k)$ 和噪音 R_t。随机模型进一步假设 $h_t(\mu,k)$ 是数据位能耗的线性组合:

$$h_t(\mu,k)=\sum_{i=1}^{\gamma}\beta_i g_i(\mu,k) \quad (6)$$

其中 $g_i(\mu,k)$ 是选择函数,表示选定的加密过程中产生的某个中间值的第 i 位(例如 S 盒输出的第 i 位)。β_i 为“位能耗转化系数”。

首先使用 POI 上的实际能耗,采用线性回归的方法估算这些位置上的能耗转换系数,使数据到拟合直线的方差最小,即

$$\begin{aligned}\varepsilon&=\sum_{i}^{N_1}[I_t(\mu_i,k)-h_t(\mu_i,k)]^2\\&=\sum_{i}^{N_1}[I_t(\mu_i,k)-\sum_{j=1}^{\gamma}\beta_j g_j(\mu_i,k)]^2\\&=\|I_t(\mu_i,k)-\boldsymbol{A}b\|^2\end{aligned} \quad (7)$$

其中 $\boldsymbol{A}$ 是 $[N_1,\gamma]$ 的矩阵,$\boldsymbol{A}_{i,j}=g_j(\mu_i,k)$,$b=(\beta_1,\beta_2,\cdots,\beta_\gamma)$。

采用梯度为 0 的方法求解:$\frac{\partial\varepsilon}{\partial b}=(I_t-Ab)=0$,则 $I_t=\boldsymbol{A}b$。由于矩阵 $\boldsymbol{A}$ 不是方阵,求解方法为

$$b=(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})^{-1}\boldsymbol{A}^{\mathrm{T}}I_t \quad (8)$$

1.2.2 形成多元高斯噪音模型

使用另外 N_2 条功耗曲线,利用 $h_t(\mu,k)$ 的位能耗线性组合公式,计算在各 POI 上的数据依赖能耗,与实际能耗相减得到功耗曲线的噪音:

$$R_t=I_t(\mu,k)-h_t(\mu_i,k) \quad (9)$$

然后按照模板攻击的方法,利用 N_2 个噪音向量计算多元高斯分布的均值向量 ρ 和协方差矩阵 Σ。

1.2.3 攻击阶段

使用 N_3 条功耗曲线 $(T_1,T_2,\cdots,T_{N_3})$,设猜测密钥为 k'。和模板攻击相同的方式转换为噪音向量 $(r_1,r_2,\cdots,r_{N_3}\mid k')$,计算它们在高斯模型中的联合似然概率,最小似然概率的 k' 作为正确的密钥:

$$k^*=\underset{kk''}{\mathrm{argmax}}\prod_{t=1}^{N_3}p(r_t\mid k') \quad (10)$$

1.3 深层神经网络攻击

近年来,深层神经网络(DNN)的研究取得非常大的进展,在语义识别、文本分类等领域都能看到它的大放异彩。DNN 的概念来源于传统人工神经网络中一种典型的多层结构 MLP,其结构如图 1 所示。层与层之间的连接权值通过误差反向传播(back propagation,BP)算法进行修正。

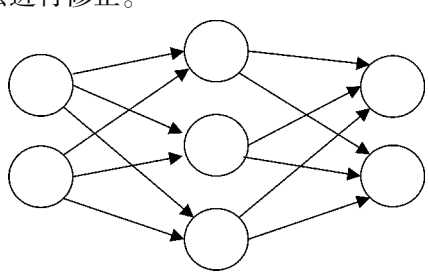

图 1 神经网络结构

训练深层感知器包括以下步骤:

Step1:将神经网络的权重 $Weight(k)_{m,n}$ 初始化为小的随机值,其中 (m,n) 表示第 k 层第 m 个节点与第 $k+1$ 层第 n 个节点之间的连接。

Step2:训练 (α,β),其中 α 为输入数据,β 为真实数据。计算误差函数 L(也称为损失函数),描述 β 与预测结果 $\hat{\beta}$ 之间的差值。

Step3:利用反向传播算法计算神经网络中损失函数相对于权值的梯度:

$$\nabla L=\frac{\partial L}{\partial Weight(k)_{m,n}} \quad (11)$$

Step4:更新权重,减小损失函数。其中 θ 为学习率,调整学习率来加速找到最优解。

$$\nabla Weight(k)_{m,n}=\theta\nabla L \tag{12}$$

Step5:重复上面 3 个步骤,直到训练结果达到预设阈值或者训练完成[12]。

1.4　深层卷积神经网络攻击

卷积神经网络为多层神经网络,其包括了卷积层,池化层以及全连接层,卷积层和池化层的个数可以自由设置。卷积层利用卷积核来提取功耗曲线的局部特征,并通过多个卷积层提取人眼不易识别的特征;池化层相当于对功耗曲线数据的下采样处理;全连接层是对经过上述步骤后得到的特征融合,并且计算各特征的概率。

卷积网络的结构:在卷积层上一般都包含着多个特征平面,在输入层上的一个窗口,通过卷积操作得到特征层上的一个特征。窗口移动的距离称为步长。窗口通过移动步长进行卷积得到特征层上其他的特征。但特征层上的特征并不是每一个都有效,通过将特征层上特征进行池化处理得到一个更有效的特征。这样处理也降低了模型的复杂度。卷积和池化层提取得到的数据特征最终在全连接层进行非线性组合,输出数据针对各类别的概率分布,具体结构如图 2 所示。

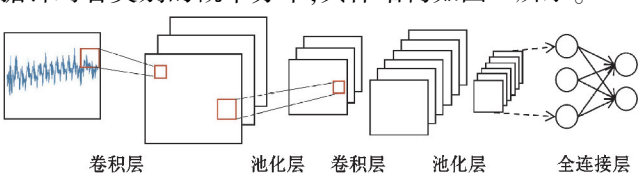

图 2　卷积神经网络结构

2　巴特沃斯滤波算法

功耗曲线的变化相对比较缓慢,而内部噪声涉及温度,电压等外部多种因素,同时变化快且不易发觉。假设所需功耗与噪音各自所占频带不同。因此尝试从频域进行滤波。首先巴特沃斯滤波的平方幅度响应函数公式为

$$|H(\tau T)|^2=A(T^2)=\frac{1}{1+\left(\frac{T}{T_c}\right)^{2N}} \tag{13}$$

其中,T_c 为巴特沃斯低通滤波器的截止频率,N 为阶数。其特点如下:通带内幅频特性平滑,且随频率增大平滑单调下降;函数中的阶数 N 越高,过渡带越窄[11]。同时传递函数无零点,极点等距离分布在以 $|S|=T_c$ 为半径的圆周上,从式(13)容易看出 T_c 以及 N 为影响滤波效果的两个重要参数。实际应用中,N、T_c 为

$$N=\frac{\lg\left(\sqrt{\frac{10^{\frac{R_s}{10}}-1}{10^{\frac{R_p}{10}}-1}}\right)}{\lg\left(\frac{T_s}{T_p}\right)} \tag{14}$$

$$T_c=\frac{T_s}{(10^{\frac{R_s}{10}}-1)^{\frac{1}{2N}}} \tag{15}$$

极点分配为

$$p_k=\exp\left(\tau\pi\frac{2k+N-1}{2N}\right),k=1,2,\cdots,N \tag{16}$$

接下来计算出滤波器在复平面的传递函数,将 $p=S/T_c$ 带入式(17)中即可求得最终滤波器传递函数 $G(s)$。

$$\begin{cases}G(p)=\prod_{k=1}^{N/2}G_k(p),N=2m\\ G(p)=\frac{1}{p+1}\prod_{k=1}^{N/2}G_k(p),N=2m+1\\ G_k(p)=\frac{1}{(p-p_k)(p-p_{N+1-k})}\\ \quad=\frac{1}{p^2-2p\cos\left(\pi\frac{2k+N-1}{2N}\right)+1}\end{cases} \tag{17}$$

巴特沃斯滤波器是在频域上解释数据的频谱特性。文中采用巴特沃斯滤波器对 FPGA 设备获取的功耗曲线进行滤波,其步骤为:

Step1:将功耗曲线数据进行变换处理,得到反映频谱特性的数据;

Step2:对变换后的数据进行巴特沃斯滤波处理;

Step3:将滤波后的数据进行逆变换处理。

$$L(t)=Tr(t)/Tr(1)-1 \tag{18}$$

使用功耗曲线中第一个采样点为参考,实现噪声信息提取。其中,$Tr(1)$ 为功耗曲线的第 1 个采样点,$Tr(t)$ 为 t 时刻功耗曲线数据(第 t 个采样点)。$L(t)$ 按照式(18)转换为噪声数据,频率特性保持不变,逐渐减小信噪比,只需对 $L(t)$ 进行滤波。同时,按照式(19)根据滤波器的阶数再确定第一个进行滤波处理的采样点:

$$L'(t)=b(N+1)\begin{bmatrix}L(t)\\ L(t-1)\\ \vdots\\ L(t-N)\end{bmatrix}-a(N)\begin{bmatrix}L'(t)\\ L'(t-1)\\ \vdots\\ L'(t-N)\end{bmatrix} \tag{19}$$

其中,a,b 分别为 $G(s)$ 进行变换后的分子和分母。$L'(t)$ 即为滤波后的数据,通过式(20)转换为最终数据 $Tr'(t)$。

$$Tr'(t)=[L'(t)+1]\cdot Tr(1) \tag{20}$$

3　实验结果

利用巴特沃斯滤波算法对功耗曲线进行滤波,对比了模板攻击、随机方法、深层感知器以及卷积神经网络这 4 种有学习的侧信道方法。

攻击对象:并行加载 AES-128 加密算法的 FPGA

设备。

攻击分析:在分析阶段,实验假设在 FPGA 设备执行加密算法时完全控制设备,能够获取执行时间内的功耗。在攻击阶段,通过收集功耗曲线,恢复未知密钥。

实验数据:为保证实验中侧信道攻击方法的复现,选择 DPA Contest V2 数据集。数据集中,每一条功耗曲线包括 3253 个采样点,测量带宽为5 GHz,采样率为 5 G采样点/秒。FPGA 设备以24 MHz频率运行,并行运行 AES-128 加密算法。其中包括 Public 和 Template 两个数据集,Public 数据集为训练数据集,其中包括 32 个随机密钥,每一轮密钥 20000 随机明文;Template 数据集为攻击数据,其中包括 32 个固定密钥。

实验指标:信噪比(signal-noise ratio,SNR),SNR = Var(E(signal))/E(Var(signal)),其用来体现功耗曲线的泄露能力;猜测熵(guessing entropy,GE),ge = $|K \in \kappa | p(K) > P(K^*)|$,其中 $p(K)$ 表示猜测密钥的得分,$p(K^*)$ 表示正确密钥的得分。

信噪比越低,能耗中包含的特征越少。Template 数据集的 SNR 分布如图 3 所示,此时最大 SNR 为 0.0521,POI 为 2627,利用巴特沃斯滤波算法进行预处理后,最大 SNR 为0.0927,POI 为 2794,如图 4 所示。对 Public 数据集进行相同处理,初始数据的 SNR 泄露如图 5 所示,滤波后的数据 SNR 分布如图 6 所示。

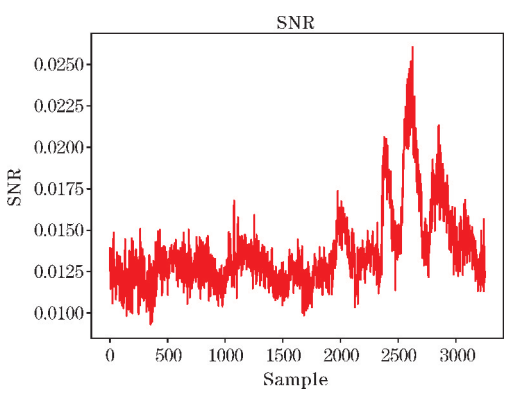

图 3 Template 数据集的 SNR

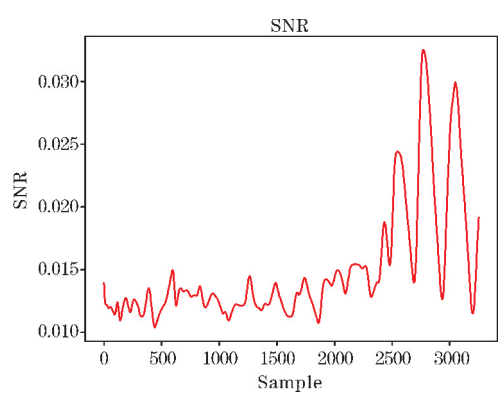

图 4 Template 数据集滤波后的 SNR

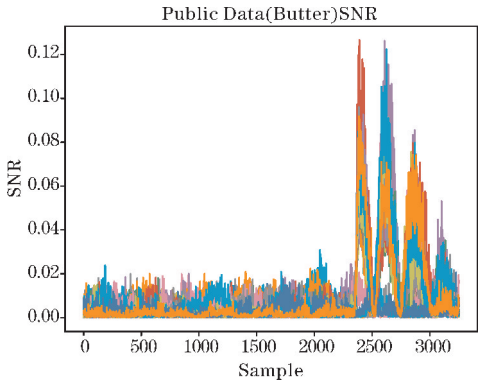

图 5 Public 数据集 SNR 分布

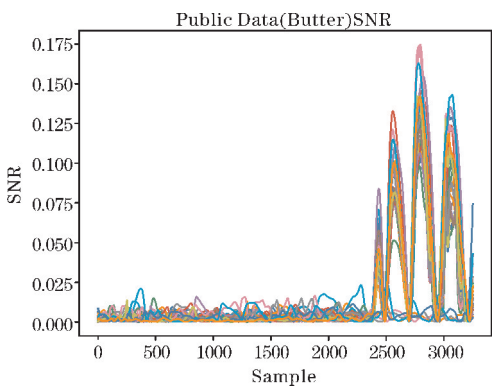

图 6 Public 数据集滤波后的 SNR

攻击步骤:

Step1:对 Public 和 Template 两个数据集进行巴特沃斯滤波,对比初始数据的 SNR,记录 SNR 泄露最大时的 POI;

Step2:改变 Step1 中选择 POI 时的步长(防止出现奇异矩阵),记录多组 SNR 泄露最大点时的 POI;

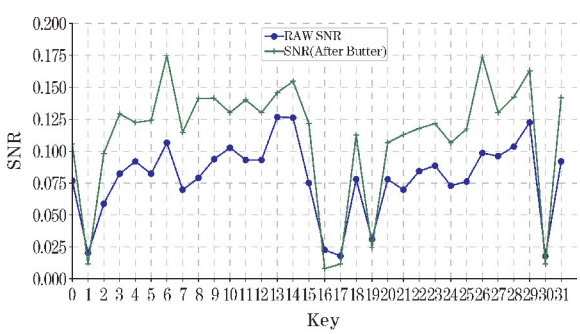

图 7 Public 数据集 32 个 Key 滤波后的 SNR

Step3:对比 Step1 中两组数据滤波后的曲线,进行基于 AES-128 最后一轮输出值的方差和均值的以此对比,找出相似曲线,进行下一步攻击;

Step4:模板攻击和随机方法结合 POI 进行最后一轮 S 盒输出值的攻击;

Step5:MLP 在预处理时结合 PCA(MLP 攻击时,相比 POI,PCA 不仅更便捷,还可以降维,降低运算复杂度);

Step6:CNN 网络结构中的卷积层对于功耗曲线之间的抖动具有识别能力,不再使用 POI 或者 PCA 进行预处理,实验中不断更换网络结构实现超参数优化。

实验证明:训练数据和攻击数据在巴特沃斯滤波后 SNR 泄露均得到明显提高,如图 7 所示。尽管 SNR 得到提高,但最终目标是提升侧信道攻击效果。因此,文中采用模板攻击,随机方法,深层感知器,深层卷积神经网络对 DPA Contest V2 数据进行侧信道攻击,4 种攻击效果如图 8 ~ 11 所示,其中在利用模板攻击,深层感知器以及深层卷积神经网络进行攻击时效果得到显著提升,同时,尽管随机方法的猜测熵下降,但是攻击效果依然不佳。经查阅文献以及 DPA Contest V2 竞赛文档,发现在功耗采集时,并行运算的加密模块会出现温度变化,对噪音模型的能耗转换系数影响不均匀。尽管有研究人员添加时间延迟函数代替温度影响,但攻击效果依然不尽如人意。

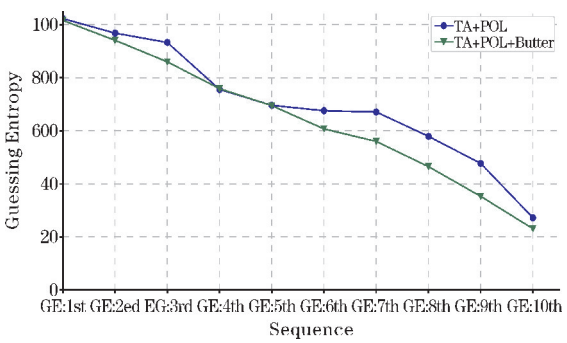

图 8　模板攻击滤波前后的攻击效果

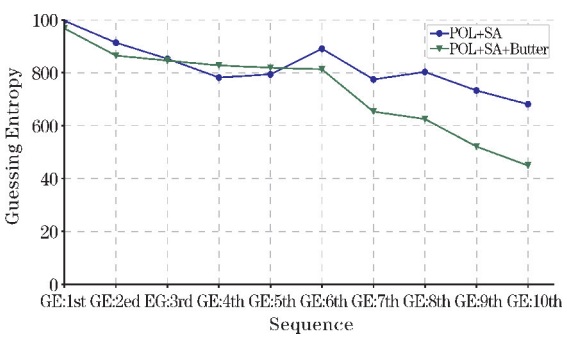

图 9　随机方法滤波前后的攻击效果

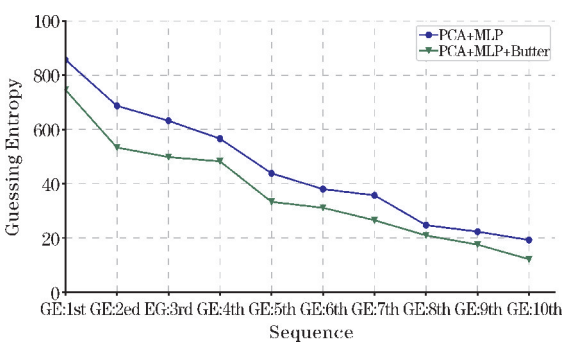

图 10　深层感知器滤波前后的攻击效果

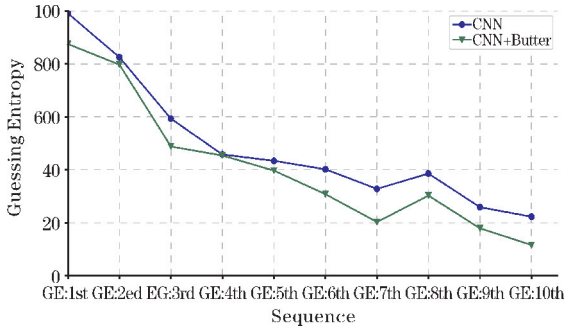

图 11　深层卷积神经网络滤波前后的攻击效果

4　结束语

不同的数据集在进行深层卷积神经网络训练时,都需要尝试多种网络结构,并不存在针对各类数据均适用的网络结构[13]。文中借助巴特沃斯滤波算法预处理降低了攻击的猜测熵,但还无法实现神经网络参数的最优化。最优网络结构以及超参数需要人工通过大量实验来寻找,今后的研究将专注挖掘 FPGA 侧信道分析的参数寻优。

参考文献:

[1]　杜之波,吴震,王敏,等. 针对基于 SM3 的 HMAC 的能量分析攻击方法[J]. 通信学报,2016,37(5):38-43.

[2]　杜之波,吴震,王敏,等. 基于 SM3 的动态令牌的能量分析攻击方法[J]. 通信学报,2017,38(3):65-72.

[3]　杜之波,吴震,王敏,等. 针对 SM4 轮输出的改进型选择明文功耗分析攻击[J]. 通信学报,2015,36(10):85-91.

[4]　Francois Koeune,Francois-Xavier Standaert. A Tutorial on Physical Security and Side-Channel Attacks[J]. Foundation of Security Analysis and Design Ⅲ,2005,3655:78-108.

[5]　Kocher,P. Introduction to differential power analysis and related attacks[EB/OL]. http://www. cryptography. com/dpa/technical/index. html,1998.

[6]　Chari S,Rao J R,Rohatgi P. Template attacks[J]. Cryptographic Hardware and Embedded Systems-CHES 2002,2003.

[7]　SchindlerW,Lemke K,Paar C. A Stochastic Model for Differential Side Channel Cryptanalysis[J]. Cryptographic Hardware and Embedded Systems-CHES 2005,3659:30-46.

[8] Bishop CM. Neural networks for pattern recognition[J]. Agricultural Engineering International the Cigr Journal of Scientific Research & Development Manuscript Pm, 1995, 12(5):1235-1242.

[9] O'Shea K, Nash R. An Introduction to Convolutional Neural Networks[J]. Computer Science, 2015.

[10] HermansM, Schrauwen B. Training and analyzing deep recurrent neural networks[C]. International Conference on Neural Information Processing Systems, Curran Associates Inc, 2013.

[11] Yang L J, Zhang B H, Xu-Zhen Y E. Fast Fourier transform and its applications[J]. Opto-electronic Engineering, 2004, 31:303-350.

[12] Maghrebi H, Portigliatti T, Prouff E. Breaking Cryptographic Implementations Using Deep Learning Techniques[J]. Springer, 2016, 10076:3-26.

[13] 吴震,杜之波,王敏,等. 密码芯片基于聚类的模板攻击[J]. 通信学报,2018,39(8):83-93.

Side-channel Analysis based on Butterworth Filtering Algorithm

YU Tiankai[1], WANG Min[1], WANG Yi[1], WU Zhen[1], DU Zhibo[1], XI Wei[2]

(1. School of Cybersecurity, Chengdu University of Information Technology, Chengdu 610225, China; 2. China southern power grid science research institute Co., Ltd., Guangzhou 510080, China)

Abstract: This paper introduced Butterworth filtering algorithm in field programmable logic gate array (FPGA) side channel attack. The power curve was preprocessed by Butterworth filtering algorithm, and the power curve was attacked by side channel using neural network model instead of traditional template model. This algorithm is universal for the power curve pretreatment of template attack, random method, deep perceptron and deep convolutional neural network, based on the experimental section in view of the DPA CONTEST2 data which were analyzed in four methods of side channel, the experimental data showed that the method increases the signal-to-noise ratio (SNR) attacked available, and improves the success rate of side channel attack.

Keywords: FPGA; Butterworth filtering; side channel attack