计算机研究与发展
Journal of Computer Research and Development
DOI: 10.7544/issn1000-1239.202220172
61(1): 261−270, 2024

基于多源数据聚合的神经网络侧信道攻击

张润莲　潘兆轩　李金林　武小年　韦永壮
（密码学与信息安全重点实验室（桂林电子科技大学）　广西桂林　541004）
（zhangrl@guet.edu.cn）

A Side Channel Attack Based on Multi-Source Data Aggregation Neural Network

Zhang Runlian, Pan Zhaoxuan, Li Jinlin, Wu Xiaonian, and Wei Yongzhuang
（*Key Laboratory of Cryptography and Information Security* (*Guilin University of Electronic Technology*), *Guilin*, *Guangxi* 541004）

Abstract　Side channel attack based on deep learning needs to model and train each key byte of the cryptographic algorithm, which costs a lot of data acquisition and model training. To solve this problem, a side channel attack method based on multi-source data aggregation neural network is proposed. In order to screen the leaked data of key byte with good generalization quality for data aggregation, taking AES-128 algorithm as an example, firstly 16 single key byte models are trained based on the leaked data of 16 key bytes, and models are used to recover 16 key bytes respectively. Secondly, a scoring mechanism is designed to evaluate the generalization effect of each single key byte model, and models with the best recovery effect for each key byte are selected according to score sorting. Finally, a multi-source data aggregation model is constructed based on the key byte leaked data sets corresponding to the selected models to realize key recovery. The tested results show that the multi-source data aggregation model has good generalization effect, effectively improves the accuracy and efficiency of key recovery, reduces the number of traces used to recover the key, and the model also has good attack effect in the case of less traces.

Key words　side channel attack; neural network; scoring mechanism; data aggregation; data labeling

摘　要　基于深度学习的侧信道攻击需要针对密码算法的每一个密钥字节建模并训练，数据采集和模型训练开销大.针对该问题，提出一种基于多源数据聚合的神经网络侧信道攻击方法.为筛选具有良好泛化效果的密钥字节泄露数据进行数据聚合，以AES-128算法为例，先基于16个密钥字节的泄露数据训练16个单密钥字节模型，分别实现对16个密钥字节的恢复；其次，设计一种打分机制评估各单密钥字节模型的泛化效果，通过得分排序筛选出对各密钥字节恢复效果最好的单密钥字节模型；最后，以筛选模型所对应的各密钥字节泄露数据集构建多源数据聚合模型进行训练，实现密钥恢复.实验测试结果表明，多源数据聚合模型具有良好的泛化效果，有效提高了密钥恢复的准确率和效率，降低了恢复密钥所需的能量迹数量，其在采集能量迹较少的情况下依然具有较好的攻击效果.

关键词　侧信道攻击；神经网络；打分机制；数据聚合；数据标签

中图法分类号　TP309.7

收稿日期：2022-03-01；**修回日期**：2022-12-23

基金项目：国家自然科学基金项目（62062026, 61872103）；广西创新研究团队项目（2019GXNSFGA245004）；广西青年创新人才科研专项（桂科AD20238082）；广西自然科学基金项目（2020GXNSFBA297076）；广西研究生创新项目（2022YCXS082）
This work was supported by the National Natural Science Foundation of China (62062026, 61872103), the Innovation Research Team Project of Guangxi (2019GXNSFGA245004), the Scientific Research Project of Young Innovative Talents of Guangxi (guike AD20238082), the Guangxi Natural Science Foundation (2020GXNSFBA297076), and the Graduate Innovation Project of Guangxi (2022YCXS082).

侧信道攻击通过对密码算法在设备实际运行过程中泄露的物理信息(如时间信息、能量信息、电磁信息)进行采集和分析,从而恢复密钥信息[1]. 侧信道攻击实用性强且破坏性大,严重威胁着密码设备的安全,是近年来密码分析的一个研究热点. 随着密码设备功能的完善以及新防护手段的应用,传统侧信道攻击的效率和成功率越来越低. 为提高侧信道攻击的效率和成功率,深度学习被引入侧信道攻击.

深度学习具有学习能力强、覆盖范围广、适应性好等优点. 基于密码设备运行过程中采集的数据,深度学习可以在一定程度上学习和提取密码算法在设备中进行数据加密时的特征信息,实现对密码算法的密钥恢复[2-4]. 2016 年, Maghrebi 等人[5]将深度学习用于 AES(advanced encryption standard)算法在无掩码和有掩码情况下的攻击,证明基于深度学习的侧信道攻击具有更好的攻击效果. 2017 年, Cagli 等人[6]基于卷积神经网络实现在不需要对能量迹对齐和精确特征点提取的情况下的攻击. 2020 年, Benadjila 等人[7]提出基于 ASCAD 数据集的最优神经网络模型,并通过实验证明神经网络模型的攻击效果优于被优化的模板攻击; Wang 等人[8]提出基于联邦学习的侧信道攻击,使用 9 个同型号的不同设备采集数据进行联合训练,并成功完成攻击; Perin 等人[9]分析了基于深度学习的侧信道分析模型的泛化能力并以集成技术增强模型的攻击能力. 2021 年, Won 等人[10]提出将 1 维数据的原始能量迹的测量值转化为具有数据位置信息的 2 维数据,改进了基于深度学习的侧信道分析; Zaid 等人[11]则为了减少深度学习衡量指标与侧信道评估指标的差异,提出一种排名损失函数并论证了成功攻击所需的能量迹上界.

基于深度学习的侧信道分析,目前普遍的方法是针对采集的能量迹,基于每个密钥字节对应的泄露数据建模,再恢复该密钥字节. 例如,针对 AES-128 算法,基于深度学习的侧信道分析将 128 b 密钥以每组 8 b 划分为 16 个密钥字节,为了恢复 16 个正确密钥需要分别对每个密钥字节建模再进行恢复[12]. 深度学习高度依赖数据,数据量越大,其学习效果越好. 因此,为提高攻击的准确率,上述方式需要采集大量数据且单独训练 16 个模型,效率较低. 而对侧信道攻击来说,数据采集由于外部因素的限制,攻击者可能无法采集到足够的数据,而数据不足会严重影响神经网络的训练效果. 如何在数据采集要求无法被满足的情况下,提高深度学习侧信道攻击的效率和准确率,成为开展深度学习侧信道攻击的关键. 针对该问题,数据增强技术被使用. 2020 年,王恺等人[12]通过对原始功耗曲线增加抖动和高斯噪声实现数据增强,并将增强后的数据用于残差网络训练实现密钥恢复; 同年, Wang 等人[13]将条件生成对抗网络(conditional generative adversarial network, CGAN)用于侧信道领域,通过条件生成对抗网络生成新的数据集进行训练数据集的扩充,从而提高能量迹不足情况下的攻击效率. 2021 年, Luo 等人[14]和 Karim 等人[15]运用 Mixup技术[16]生成能量迹,增加模型攻击效果. 然而,数据增强方法需要先训练网络生成所需能量迹,增加了侧信道攻击中数据训练的复杂性.

针对上述问题,提出一种基于多源数据聚合的神经网络侧信道攻击方法,以 AES-128 算法为攻击对象,先基于 16 个密钥字节的泄露数据训练 16 个模型,并分别实现对相应 16 个密钥字节的恢复; 进一步以每个模型分别恢复其他的密钥字节,了解每个模型在特征学习过程中的泛化效果; 针对每个模型对所有密钥字节的攻击效果打分并排序,筛选出对所有密钥字节攻击效果好的模型所对应的泄露数据,构建和训练一个基于多密钥字节泄露数据的多源数据聚合模型,再恢复所有的密钥字节. 实验结果表明,该方法在较少能量迹情况下不需要数据增强处理也能够提高侧信道攻击的准确率,降低了模型训练的开销,减少了成功恢复密钥所需要的能量迹数量.

1 MLP 神经网络简介

多层感知器(multi-layer perceptron, MLP)是一种由多个感知器单元组成的神经网络,包括输入层、输出层和 1 个或多个隐藏层,每一层包含 1 个或多个感知器单元,每一层的所有感知器与下一层的所有感知器进行全连接, MLP 的结构如图 1 所示.

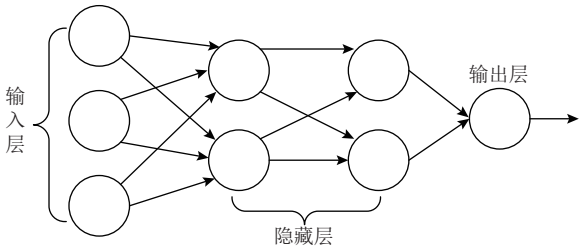


Fig. 1 The structure of MLP

图 1 MLP 结构

MLP 网络模型的核心思想是通过前向传播得到误差,再把误差通过反向传播实现权重的修正,最终得到最优模型. 在反向传播过程中通常使用随机梯

度下降法对权重值进行修正，梯度下降法的原理是计算损失函数关于所有内部变量的梯度，并进行反向传播.

2 多源数据聚合模型

目前采用神经网络针对 AES 算法的侧信道分析，主要利用采集的能量迹分别为每一个密钥字节在第 1 轮 S 盒处的泄露点进行建模，再恢复该密钥字节，如对 AES-128 算法需要建立和训练 16 个模型. 这种方法需要采集大量的数据进行训练，能量迹采集和训练开销大、效率低；若采集的能量迹不足，会导致模型的准确率低. 针对上述问题，提出基于多源数据聚合的模型构建方法，针对 AES-128 算法筛选出对各个密钥字节攻击效果较好的数据集建立多源数据聚合模型，再进行密钥恢复.

2.1 基于单密钥字节的泄露数据集建模

在 AES-128 算法加密过程中，16 个密钥字节加密明文的操作是一个时序操作，所有的泄露都在采集的一条能量迹中. 在整个时序中，由于每个密钥字节在加密不同明文时所泄露的能量信息具有相同特征，因此在采用深度学习方法进行侧信道分析时，通常是对单个密钥字节的泄露数据建模，在攻击阶段根据训练结果恢复出对应字节的密钥.

在传统侧信道攻击中，通常使用能量消耗模型模拟真实情况下泄露的能量消耗. 能量消耗模型有不同的精度级别，一般而言，精度越高的模型攻击效果越好，但所需要的资源也越多. 汉明重量模型是攻击者刻画总线与寄存器能量消耗常用的能量消耗模型，它的基本原理是假设能量消耗与寄存器中存储数据比特 1 的数目成正比. 而密码设备 S 盒置换的非线性操作在寄存器中引发的能量变化较为突出，因此主要针对 S 盒的输出通过能量消耗模型寻找相关特征.

能量迹的特征依赖于密钥与明文，理论上针对单密钥字节所建立的模型并以其恢复对应密钥字节的效果是最佳的. 模型的建立需要先对采用的数据集进行数据处理，即完成数据标签. 数据的标签实际上是将每条能量迹的特征通过能量消耗模型重新刻画，即将每条能量迹的分类类别由明文和密钥转化为标签值函数计算后的标签值. 假设一条能量迹中某 S 盒泄露点 x 的明文数据为 p，该单密钥字节为 k，假设采用汉明重量函数 $HW(\ )$ 计算，则标签值 z 计算方法为

$$z = f(p,k) = HW(Sbox(p \oplus k)). \tag{1}$$

对于其他能量迹，同样在该泄露位置 x 计算对应的标签值 z，完成对所有能量迹在泄露点 x 的标签. 采用同样的方法，继续针对所有能量迹的其他泄露点完成数据标签. 以所有能量迹在某个泄露点标签后的数据，训练建立一个单密钥字节模型，并以该模型恢复该泄露点对应的密钥字节.

AES-128 算法中，使用了 16 个 S 盒，但这 16 个 S 盒的真值表是相同的. 这使得基于这 16 个 S 盒进行运算后的数据间可能存在一定的关联性，即所采集的能量迹不同泄露点的数据可能会表现出相似的特征. 对能量迹的标签计算，也会将这些相似特征表现出来，即在对这些能量迹进行标签时，通过标签值函数 $f(p,k)$ 标签，不同泄露点的标签结果也可能会被归于同一类. 基于这种特征，以某个泄露点训练的模型，若能够有效恢复出其他泄露点对应的密钥字节，则可以有效降低能量迹采集和模型训练的开销.

为评估以某个泄露点训练的模型恢复其他泄露点对应密钥字节，针对 AES-128 算法，先采集算法运行时的能量迹，再将其划分为 16 个密钥字节的泄露数据集，进行数据标签处理，以这 16 个密钥字节的数据集分别训练出 16 个模型. 进一步地，利用这 16 个模型，先分别恢复该泄露数据集对应的密钥字节，再分别去恢复其他密钥字节.

采用 Benadjila 等人[7] 于 2020 年对 ASCAD 数据集提出的 MLP-best 模型建立 MLP 神经网络模型，MLP 模型一共有 6 层，第 1 层为输入层，其是大小为 96 的 1 维向量，输入层神经元数对应单个密钥字节在 S 盒操作处的 96 个能量采样点数据. 中间 4 层为隐藏层，每层隐藏层都有 200 个神经元；为了在快速收敛的同时避免梯度消失并提高网络的精度，激活函数采用 RELU 函数. 最后一层是输出层，其大小与实验选择的能量消耗模型有关. ASCAD 公开数据集只提供第 3 个 S 盒处的泄露数据，本文方法需分析 AES 算法的 16 个 S 盒的所有泄露数据，在此选用提供了全部字节泄露的 DPAContest_V4 数据集. MLP-best 在 ASCAD 数据集上有局部最优效果，但并不适用于采用 RSM(rotating S-box masking)掩码方案[17] 的 DPAContest_V4 数据集[18]. 针对 DPAContest_V4 数据集，构造新的标签值函数计算标签值：

$$z' = f'(p,k) = p \oplus k \oplus Sbox(p \oplus k). \tag{2}$$

将标签值 $f'(p,k)$ 作为不同明文对应能量迹的标签值时，根据 AES 的 S 盒输入异或输出共 163 种异或结果，因此输出层设置为 163 个神经元，对应 163

种不同标签值的分类结果. 为提高分类的准确率，输出层的激活函数选择适用于多分类场景下的Softmax函数，以将输出映射为各个类别的分类概率.

基于上述建立的模型，利用AES-128算法16个密钥字节的标签好的数据集训练出16个模型，分别命名为 M_i(i=1,2,⋯,16). 为区分不同密钥字节的标签数据集，以 S_j (j=1,2,⋯,16)表示依据S盒操作顺序并完成标签的数据集. DPAContest_V4数据集中每个数据集 S_j 分别有10 000条能量迹，从中随机抽取8 000条作为训练集训练神经网络模型 M_i，剩下的2 000条作为测试集用以恢复对应密钥字节. 为保证效果一致，在每个模型训练过程中统一设置相关参数：训练的 *epoch*=300, *batchsize*=100, 学习率设为0.000 01，优化器使用RMSprop，使用交叉熵损失函数.

基于上述建立的16个模型，利用相应的测试集数据分别去恢复16个密钥字节，各个模型恢复对应密钥字节所用的最少能量迹数如表1所示.

进一步地，为评估以某个泄露点训练的模型是否能够恢复其他泄露点对应密钥字节，基于同样的实验环境，利用这16个模型中的每一个模型，分别以每个密钥字节对应的测试集数据去恢复相应的密钥字节，各模型完成对各密钥字节恢复的最少能量迹数如表2所示.

Table 1 Minimum Number of Traces for Sixteen Models Recovering the Corresponding Key Byte

表1 16个模型恢复对应密钥字节的最少能量迹数

模型	能量迹数	模型	能量迹数	模型	能量迹数	模型	能量迹数
M_1	50	M_5	31	M_9	18	M_{13}	28
M_2	40	M_6	15	M_{10}	50	M_{14}	33
M_3	49	M_7	27	M_{11}	68	M_{15}	41
M_4	30	M_8	39	M_{12}	35	M_{16}	33

Table 2 Number of Traces for Sixteen Models Recovering Sixteen Key Bytes

表2 16个模型恢复16个密钥字节的能量迹数

模型	密钥字节															
	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}	S_{16}
M_1	50	138	78	95	138	141	68	137	81	420	71	384	121	287	177	172
M_2	113	40	137	83	160	71	67	170	256	192	171	329	466	208	352	155
M_3	96	182	49	300	332	183	87	442	275	299	71	870	738	482	149	374
M_4	113	97	245	30	61	68	71	109	125	115	183	142	58	287	181	80
M_5	57	129	409	49	21	165	51	191	63	518	150	111	70	415	355	49
M_6	125	24	89	70	182	15	39	284	402	191	148	1 622	154	164	97	367
M_7	66	72	63	128	157	87	27	332	146	359	39	675	134	314	104	328
M_8	112	108	165	100	127	125	148	39	40	45	66	74	42	68	117	125
M_9	81	306	174	173	93	192	108	49	18	68	66	103	27	122	120	195
M_{10}	308	125	274	168	219	102	211	76	48	50	107	92	107	47	249	161
M_{11}	194	273	183	215	389	151	148	130	103	157	68	404	145	152	108	711
M_{12}	550	274	540	90	135	575	333	79	39	49	140	35	54	74	265	181
M_{13}	147	178	501	177	105	146	131	41	46	72	56	92	28	58	70	150
M_{14}	211	183	508	147	205	116	197	116	72	46	89	175	63	33	184	162
M_{15}	86	123	56	168	188	123	49	86	150	120	42	356	66	274	41	417
M_{16}	99	145	329	54	52	191	118	103	82	142	252	101	92	421	378	33

表2中，每一行的数值表示第 i 个模型 M_i 恢复不同密钥字节的最少能量迹数. 从表2中可以看出，以某个泄露点对应的密钥字节数据建立的模型，在恢复该泄露点对应的密钥字节时，除 M_{10} 外都具有最佳效果；而以某个模型去恢复其他密钥字节时，也都可以恢复，只是需要使用更多的能量迹数量.

2.2 数据筛选与多源数据聚合模型构建

表2结果表明，基于神经网络进行单密钥字节模型训练时，由于AES算法中S盒特性，密钥泄露数据集之间存在可能相似的特征，每个模型可以一定程度上学习到其他密钥字节的泛化特征，能够恢复出其他泄露点对应的密钥字节. 在因运行环境等原因

无法大量采集能量迹的情况下，为减少数据采集和模型训练，基于密钥字节泄露数据集之间可能相似的特征，可以选用部分密钥字节的泄露数据集训练一个多源数据聚合模型去恢复所有密钥字节.

为保证训练的聚合模型具有更好的泛化效果，需要筛选出对其他所有密钥字节的攻击效果总体上最好的数据集进行训练. 具体地，设计一种打分机制进行评估. 先基于不同泄露数据集构建的 16 个模型中的每个模型对所有密钥字节的攻击结果，根据每个模型恢复各个密钥字节的能量迹数量进行升序排序并进行打分，排序靠前的模型得分高，排序靠后的被认为对恢复密钥的作用较小，得分降低甚至忽略不计. 以 $C_x(M_i, S_j)$ 表示模型 M_i 在密钥字节 S_j 上排序位置为 x 时的得分，打分方法为

$$C_x(M_i,S_j)=\begin{cases}3-0.3x, 1\leqslant x\leqslant 10,\\ 0, 11\leqslant x\leqslant 16.\end{cases} \tag{3}$$

根据 $C_x(M_i, S_j)$ 计算出各个模型 M_i 在密钥字节 S_j 上的得分. 基于该方法，计算每个模型 M_i 在所有 16 个密钥字节上的得分并求总分，以 $CT(M_i)$ 表示模型 M_i 的总得分，则

$$CT(M_i)=\sum_{j=1}^{16}C_x\left(M_i,S_j\right). \tag{4}$$

根据表 2 中不同单密钥字节模型对各个密钥字节恢复所需要的能量迹数，采用式(4)分别计算每个模型在恢复所有密钥字节时的得分情况如表 3 所示.

Table 3　Scores of Each Model

表 3　各个模型的得分

模型	得分	模型	得分	模型	得分	模型	得分
M_1	13.2	M_5	17.7	M_9	21.9	M_{13}	22.5
M_2	13.5	M_6	15.9	M_{10}	15.9	M_{14}	12.9
M_3	7.2	M_7	18.6	M_{11}	6.0	M_{15}	19.2
M_4	19.5	M_8	28.5	M_{12}	16.8	M_{16}	14.7

根据表 3 中各个模型的得分进行升序排序，结果如表 4 所示.

Table 4　Ranking of Each Model According to Scores

表 4　由得分得到的各个模型的排序

排名	模型	排名	模型	排名	模型	排名	模型
1	M_8	5	M_{15}	9	M_{10}	13	M_1
2	M_{13}	6	M_7	10	M_6	14	M_{14}
3	M_9	7	M_5	11	M_{16}	15	M_3
4	M_4	8	M_{12}	12	M_2	16	M_{11}

为构建一个具有较好泛化效果的聚合模型，选取排序靠前的几个模型分别对应的密钥字节泄露数据集进行聚合，训练出一个多源数据聚合模型，再以聚合模型进行密钥恢复.

在多源数据聚合模型训练中，仍然采用 Benadjila 等人[7] 提出的 MLP-best 模型建立 MLP 神经网络模型，聚合模型的层数、激活函数和参数均与 2.1 节中基于单密钥字节构建模型相同.

3　实验结果与分析

采用 Python 语言，基于 Keras 深度学习开源库(keras 2.2.4)和 TensorFlow 后端(Tensorflow 1.12.0)在硬件配置为 Intel(R) Core™ i5-6300HQ CPU@2.30 GHz、内存 8 GB 和 NVIDIA GeForce GTX 960 M 显卡的计算机上实现 MLP 神经网络模型，基于 DPAContest_V4 数据集构建并训练 16 个单密钥字节模型和多源数据聚合模型，并进行密钥恢复测试. 针对多源数据聚合模型，完成了其进行密钥恢复的能量迹数量及与其他模型的对比测试.

3.1　不同多源数据聚合模型的攻击效果测试

根据表 4 中单密钥字节模型对不同密钥字节恢复的打分排名结果，按照排名情况选取各个模型对应的密钥字节数据进行聚合，训练多源数据聚合模型，再进行密钥恢复. 为更好地评估不同密钥字节数据集聚合的泛化攻击效果，根据表 4 的排名情况分别以得分排名前 6, 8, 10, 12, 14, 16 的模型对应的密钥字节数据集进行多源数据聚合，建立和训练聚合模型 *Mix*_6, *Mix*_8, *Mix*_10, *Mix*_12, *Mix*_14, *Mix*_16, 对不同密钥字节数据进行恢复，并与各个单密钥字节模型 M_i 恢复相应密钥字节数据的结果进行对比，各自所需要的能量迹数如图 2 所示.

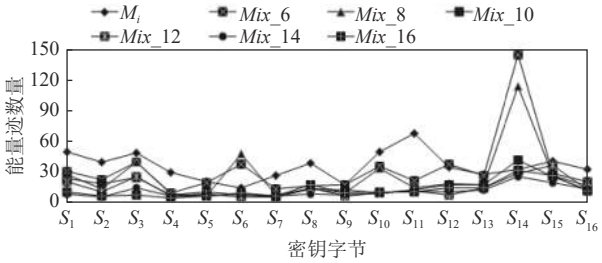


Fig. 2　Number of traces for each key byte recovered by different models

图 2　不同模型恢复各密钥字节的能量迹数

图 2 结果表明，多源数据聚合模型都能够以所聚合的部分密钥字节泄露数据集训练模型并有效恢复

出所有密钥字节, 包括那些没有参与聚合训练的密钥字节; 同时, 多源数据聚合模型较单密钥字节模型的攻击效果更好, 其恢复各密钥字节所需要的能量迹数量更少. 从对每一个密钥字节数据的攻击结果看, 随着聚合的数据集增加, 聚合模型的攻击效果提升, 恢复该密钥字节所需的能量迹数量减少, 这表明聚合模型可以在更多的数据集中更好地学习到泛化特征. 在聚合模型进行特征学习过程中, 当聚合的数据集较少时, 噪声容易干扰学习结果, 攻击效果并没有显著提升; 随着聚合数据集的增加, 噪声干扰降低, 在聚合 6 个数据集进行训练后, 对整体子密钥的恢复效果基本趋于稳定并优于单密钥字节模型的攻击结果.

多源数据聚合模型基于深度学习的强大学习能力, 能够基于部分采集的能量迹数据有效恢复算法整体密钥, 这在数据采集不便的情况下也能够实现高效的攻击; 由于不需要训练多个模型也使得模型训练和分析测试工作有效减少.

3.2 公用数据集对比实验与分析

现有针对密码算法的基于深度学习的侧信道分析方法, 更多的仍然是针对密码算法不同的密钥字节分别建模和分析. 针对 AES-256 算法, 文献 [13] 基于卷积神经网络 (convolutional neural network, CNN) 和 DPAV4 数据集, 以第 1 轮掩码 S 盒输出值的最低有效位作为训练标签, 针对其选择的某个密钥字节对应的泄露数据集分别建模和训练测试, 使用 1 000 条能量迹作为训练集, 2000 条作为验证集, 2000 条作为测试集, 完成对该密钥字节的恢复, 其恢复该密钥字节所需要的能量迹为 550 条.

采用多源数据聚合模型方法, 按表 4 分别取前 8, 10 个数据集聚合训练模型 *Mix*_8, *Mix*_10, 与文献 [13] 在同样数据集大小条件下进行模型训练并对前 16 个密钥字节进行恢复; 同时, 根据算法原始的密钥字节顺序选取前 8, 10 个密钥字节的泄露数据集, 采用本文的多源数据聚合思想以同样数据集大小聚合训练模型 *MO*_8, *MO*_10 并恢复前 16 个密钥字节, 与在同样数据集大小情况下采用单密钥字节泄露数据训练模型 M_i 的密钥恢复结果对比, 各自所需能量迹数如表 5 所示.

Table 5 Recovery Results of Six Models for 16 Key Bytes

表 5 6 个模型对 16 个密钥字节的恢复结果

模型	密钥字节															
	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}	S_{16}
文献 [13]	550	550	550	550	550	550	550	550	550	550	550	550	550	550	550	550
M_i	1 156	1 410	913	1 032	1 022	2 000	700	1 165	1 987	1 844	1 401	468	567	563	1 532	2 000
*Mix*_8	41	55	116	36	66	73	38	172	81	106	62	129	88	277	91	50
*MO*_8	42	73	41	77	120	65	40	190	93	109	93	172	137	108	100	155
*Mix*_10	27	39	71	32	37	41	25	115	60	68	65	124	84	117	91	33
*MO*_10	42	37	33	56	67	34	53	160	84	110	72	116	204	110	125	136

文献 [13] 通过已知的掩码计算掩码 S 盒的输出, 将其转换为一个不受保护的实现, 本实验中采用式 (2) 的标签方案并未对掩码进行额外操作, 因此在只有 1 000 条训练数据下训练的单密钥字节模型效果不太理想, 但在 2 000 条能量迹内也能恢复正确密钥.

表 5 结果也显示多源数据聚合模型恢复密钥字节的效果都优于文献 [13], 所需能量迹数量更少; 随着聚合数据集的增加, 聚合模型对密钥字节的恢复效果更好, 所需要的能量迹数减少. 模型 *Mix*_8, *Mix*_10 的攻击效果分别优于模型 *MO*_8, *MO*_10, 这表明基于打分排名后筛选的多源数据聚合模型具有更好的泛化特征学习能力, 其攻击效果要优于不做数据筛选的多源数据聚合模型, 恢复密钥字节所需使用的能量迹数更少.

文献 [9] 基于 MLP 网络和 DPAV4 数据集, 进行 50 次随机超参数搜索训练得到 50 个 MLP 网络并使用引导聚集 (bootstrap aggregating) 算法进行集成, 针对第 1 个密钥字节的数据进行训练与恢复, 以第 1 轮掩码 S 盒输出值的汉明重量作为训练标签, 34 000 条能量迹作为训练集, 1 000 条作为验证集, 1 000 条作为测试集进行实验. 采用同样的数据集规模, 基于多源数据聚合模型方法, 按表 4 分别取前 8, 10, 12 个数据集聚合训练模型 *Mix*_8, *Mix*_10, *Mix*_12 进行密钥恢复; 同时, 根据算法原始的密钥字节顺序选取前 8, 10, 12 个密钥字节的泄露数据集, 采用本文的多源数据聚合方法以同样数据集大小聚合训练模型 *MO*_8, *MO*_10, *MO*_12 并恢复密钥, 各自所需能量迹数如表 6 所示.

表 6 结果显示，相较于文献 [9] 中的结果，采用聚合模型进行攻击，各个模型中参与了训练的密钥字节的恢复所需要的能量迹在 4 条左右，未参与训练的密钥字节的恢复需要更多的能量迹数；但采用聚合模型仅需要训练 1 个模型，开销小、时间短，且在聚合 12 个密钥字节的泄露数据集后，对未参与训练的密钥字节恢复密钥的能量迹数量都有效减少，而文献 [9] 需要训练 50 个不同的模型才完成对第 1 个密钥字节的攻击，要完成对 16 个密钥字节的攻击，其模型训练的开销极大、时间过长.

Table 6 Recovery Results of Seven Models for 16 Key Bytes

表 6 7 个模型对 16 个密钥字节的恢复结果

模型	密钥字节数据															
	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}	S_{16}
文献 [9]	4															
*Mix*_8	32	16	45	4	4	40	5	4	5	16	9	5	5	37	7	9
*MO*_8	4	5	5	4	5	6	4	6	9	23	15	29	17	47	34	12
*Mix*_10	30	6	35	4	5	4	6	4	4	7	10	6	5	28	8	8
*MO*_10	4	4	5	5	5	4	5	7	6	9	15	18	11	28	34	11
*Mix*_12	13	6	21	4	5	4	5	4	4	6	9	5	6	20	6	6
*MO*_12	3	4	5	4	5	4	5	6	7	7	10	6	7	22	25	11

相比于表 5 中的结果，表 6 结果也显示聚合模型中相应密钥恢复所需要的能量迹数量进一步减少. 这表明随着数据规模的增加，神经网络可以充分地学习到训练数据的特征，聚合模型的攻击成功率更高；其次，采用基于打分排名后筛选的数据聚合模型具有更好的泛化特征学习能力，其总体攻击效果仍然优于不做数据筛选的聚合模型；同时，随着参与训练的数据量增加，泄露数据参与了训练的密钥字节更容易被恢复，因其特征能够被更好地提取出来.

3.3 实测数据集对比实验与分析

使用 MathMagic 侧信道分析仪，搭载目标芯片 STC89C52 单片机，实现无防护的 AES-128 算法，模拟数据采集困难、所采集的能量迹少的情景，共采集 3 500 条能量迹，以其中 3 000 条能量迹作为训练集，剩下 500 条能量迹作为测试集. 实验中先通过相关性分析确定密钥字节在能量迹中的泄露位置，采用式(1)进行计算，由于汉明重量模型对应 9 种类别，因此在保持 MLP 模型结构参数与 2.1 节模型参数其余部分不变的基础上，将输出层节点数设置为 9. 对每个密钥字节独立建模并攻击，各单密钥字节模型记为 M_i. 采用多源数据聚合模型方法，按表 4 选取前 8, 10 个泄露数据集聚合训练模型 *Mix*_8, *Mix*_10, 同时按密码算法原始密钥字节顺序选取前 8, 10 个数据集聚合训练模型 *MO*_8, *MO*_10, 并分别恢复各密钥字节，各个模型完成密钥恢复所需的能量迹数对比结果如表 7 所示.

表 7 结果表明，由于训练数据的不足，采用单密钥字节建模并进行密钥恢复时，仅恢复出其中的 3 个字节，其他模型因不能学习到足够的特征，无法恢复密钥；采用多源数据聚合模型方法都有效完成了密钥恢复，优于单密钥字节模型攻击结果. 在 *Mix*_8, *Mix*_10 中，恢复前 3 个密钥字节的能量迹数较 *MO*_8, *MO*_10 所需的能量迹数稍多，主要原因在于这些密钥字节泄露数据在 *Mix*_8, *Mix*_10 中未参与训练，而

Table 7 Recovery Results of Five Models for 16 Key Bytes

表 7 5 个模型对 16 个密钥字节的恢复结果

模型	密钥字节															
	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}	S_{16}
M_i	95		44													40
*Mix*_8	5	7	6	4	5	4	5	5	4	6	6	5	6	8	6	15
*MO*_8	4	6	4	4	4	5	6	7	7	6	8	7	8	8	8	20
*Mix*_10	5	6	7	5	4	4	4	5	4	4	8	6	7	6	8	12
*MO*_10	4	5	4	4	4	4	7	7	7	8	13	9	7	11	9	16

在 *MO*_8, *MO*_10 中参与了训练, 在训练数据不足的情况下未参与训练的密钥字节特征学习效果相对减弱. 但总体上, 与采用公用数据集训练模型一样, 采用实测数据集训练模型时, 基于打分排名后筛选的多源数据聚合模型仍然具有更好的泛化特征学习能力, 其攻击效果优于不做数据筛选的多源数据聚合模型, 恢复密钥字节所需使用的能量迹数更少. 此外, 相对于加掩防护的公用数据集实验, 实测实验针对的是未加掩码防护的 AES-128 算法, 因此更容易完成攻击, 恢复密钥所需的能量迹数大大减少.

3.4 针对 Present 算法的聚合模型实验与分析

为评估本文方法的通用性, 将聚合模型应用于 Present 算法, 验证数据聚合模型的分析效果. Present 算法是一个 4 b 的 S 盒且 SPN 结构的轻量级密码算法. 使用 osrtoolkit 功耗采集平台, 搭载目标芯片 ATMEGA2560, 实现无防护的 Present 算法, 采集 3 500 条能量迹, 以其中 3 000 条能量迹作为训练集, 500 条能量迹作为测试集. 实验中先通过相关性分析确定密钥半字节在能量迹中的泄露位置, 以真实密钥作为身份标签, 训练的 *epoch*=300, *batchsize*=100, 学习率设为 0.000 01, 优化器使用 RMSprop, 使用交叉熵损失函数, 将输出层节点数设置为 16. 对 16 个密钥半字节分别独立建模并攻击, 各密钥半字节模型记为 M_i. 利用这 16 个模型中的每一个模型, 分别以每个密钥半字节对应的测试集数据去恢复所有的密钥半字节, 各模型完成对各密钥半字节恢复的最少能量迹数如表 8 所示.

根据式(4), 基于表 8 的结果计算每个模型在恢复所有密钥半字节时的得分, 结果如表 9 所示.采用多源数据聚合模型方法, 对表 9 以得分高低排序后选取前 8, 10, 12, 16 个泄露数据集聚合训练模型 *Mix*_8, *Mix*_10, *Mix*_12, *Mix*_16, 分别恢复各密钥半字节, 所需的能量迹数与单密钥半字节建模恢复的结果对比情况如表 10 所示.

表 10 结果表明, 在聚合模型中, 由于密钥半字节 S_5 和 S_9 对应模型得分较低, 相应数据集未能在 *Mix*_8, *Mix*_10 中参与训练, 无法被恢复. 但在 *Mix*_12 模型中, S_5 对应的数据集尽管没有参与训练, 但由于有更多数据集参与训练, 相应特征被学习, 也能够被有效恢复. 而基于所有数据集的聚合模型, 能够以更少的能量迹数量恢复所有密钥半字节. 从所有聚合模型恢复的结果看, 若数据集参与了聚合模型训练, 其恢复相应密钥半字节所需要的能量迹数量均少于单密钥半字节模型的结果; 在单密钥半字节模型中无法恢复的其他密钥半字节, 在聚合模型中除 1 或 2

Table 8 Number of Traces for Sixteen Models Recovering Sixteen Key Nibbles for Present Algorithms

表 8 针对 Present 算法 16 个模型恢复 16 个密钥字节的能量迹数

模型	密钥字节															
	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}	S_{16}
M_1	4		169								357					
M_2		6		317		394	319	181		55		73		82	83	167
M_3	55		25										217			
M_4				5		14				78		489			130	
M_5					19				438		379					
M_6		337		15		11		159		79		184		341	240	421
M_7		270				236	4	75		257				97	352	
M_8							169	8		473				108	478	230
M_9									5							
M_{10}		72		114		54	442	198		9		35		139	65	463
M_{11}					340						6		150			
M_{12}		35		143		86	398			31		5		51	51	186
M_{13}	119		332								73		5			
M_{14}		19				145	165	166		45		402		6	189	163
M_{15}		130		403		176		209		118		172		219	11	
M_{16}		480						477		220		390		229	235	5

Table 9 Scores of Sixteen Models

表 9 16 个模型的得分

模型	得分	模型	得分	模型	得分	模型	得分
M_1	8.1	M_5	7.8	M_9	3	M_{13}	10.5
M_2	21	M_6	15.6	M_{10}	20.7	M_{14}	19.8
M_3	8.1	M_7	11.7	M_{11}	8.1	M_{15}	13.5
M_4	10.2	M_8	9.6	M_{12}	21.6	M_{16}	9.3

Table 10 Recovery Results of Different Models for Each Key Nibble

表 10 不同模型对各密钥半字节的恢复结果

模型	密钥字节数据															
	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}	S_{16}
M_i	4	6	25	5	19	11	4	8	5	9	6	5	5	6	11	5
*Mix*_8	272	3	171	11		3	2	8		2	54	3	2	2	3	129
*Mix*_10	250	2	397	4		3	3	3		2	104	1	4	2	5	254
*Mix*_12	2	2	13	2	474	1	2	4		3	38	2	2	6	2	3
*Mix*_16	4	3	5	5	4	3	2	4	4	5	4	3	4	3	3	6

个密钥半字节外，其他的都能够有效恢复，这表明多源数据聚合模型具有更好的泛化特征学习能力，相对于单密钥半字节模型，有效提升了其恢复密钥的能力. 针对不同算法的攻击结果，表明了本文提出的多源数据聚合模型在攻击时具有较好的通用性.

4 结束语

为降低在实际应用中基于深度学习进行侧信道攻击所需要的数据采集和模型训练开销，提出一种基于多源数据聚合的神经网络侧信道攻击方法. 该方法针对 AES-128 算法，通过评估单密钥字节模型的泛化效果，筛选出对各密钥字节恢复效果最好的部分密钥字节泄露数据集，构建多源数据聚合模型，再进行密钥恢复. 实验测试结果表明，多源数据聚合模型具有良好的泛化效果，有效提高了密钥恢复的准确率和效率，降低了恢复密钥所需的能量迹数量，可在能量迹采集困难、能量迹较少的情况下依然具有较好的攻击效果. 在今后工作中，将考虑结合数据增强技术，进一步提升在能量迹不足情况下的密钥恢复效果. 与此同时，也将研究针对深度学习侧信道攻击的防御措施，提高密码算法应用的安全性. 深度学习侧信道攻击主要针对 S 盒部件的泄露数据开展攻击. 因此，为有效预防深度学习侧信道攻击，一方面是在密码部件设计中，考虑 S 盒抗侧信道攻击的能力，并研究密码算法结构整体的安全性；另一方面是研究针对侧信道攻击的掩码防护技术，消除密码算法功耗泄露与中间值之间的可预测关系，达到防护的目的.

作者贡献声明：张润莲设计研究方案，撰写初稿；潘兆轩参与方案讨论、实验研究及结果分析；李金林采集数据，参与实验研究及结果分析；武小年参与设计研究方案及论文修改；韦永壮参与论文修改.

参 考 文 献

[1] Wang An, Ge Jing, Shang Ning, et al. Practical cases of side-channel analysis[J]. Journal of Cryptologic Research, 2018, 5(4): 383−398 (in Chinese)
(王安, 葛婧, 商宁, 等. 侧信道分析实用案例概述[J]. 密码学报, 2018, 5(4): 383−398)

[2] Zhang Libang, Xing Xinpeng, Fan Junfeng, et al. Multi-label deep learning based side channel attack[C/OL] //Proc of the 2019 Asian Hardware Oriented Security and Trust Symp. Piscataway, NJ: IEEE, 2019[2022-01-10]. https://ieeexplore.ieee.org/document/9006657

[3] Ghandali S, Ghandali S, Tehranipoor S. Profiled power-analysis attacks by an efficient architectural extension of a CNN implementation[C] //Proc of the 22nd Int Symp on Quality Electronic Design. Piscataway, NJ: IEEE, 2021: 395−400

[4] Kim J, Picek S, Heuser A, et al. Make some noise: Unleashing the power of convolutional neural networks for profiled side-channel analysis[J]. IACR Transactions on Cryptographic Hardware and

Embedded Systems, 2019, 2019(3): 148−179

[5] Maghrebi H, Portigliatti T, Prouff E. Breaking cryptographic implementations using deep learning techniques[C] //Proc of the 5th Int Conf on Security, Privacy, and Applied Cryptography Engineering. Berlin: Springer, 2016: 3−26

[6] Cagli E, Dumas C, Prouff E. Convolutional neural networks with data augmentation against jitter-based countermeasures[C] //Proc of the 19th Int Conf on Cryptographic Hardware and Embedded Systems. Berlin: Springer, 2017: 45−68

[7] Benadjila, R, Prouff, E, Strullu, R. et al. Deep learning for side-channel analysis and introduction to ASCAD database[J]. Journal of Cryptographic Engineering, 2020, 10(2): 163−188

[8] Wang Huanyu, Dubrova E. Federated learning in side-channel analysis [C] //Proc of the 16th Int Conf on Information Security and Cryptology. Berlin: Springer, 2020: 257−272

[9] Perin G, Chmielewski L, Picek S. Strength in numbers: Improving generalization with ensembles in machine learning-based profiled side-channel analysis[J]. IACR Transactions on Cryptographic Hardware and Embedded Systems, 2020, 2020(4): 337−364

[10] Won Y S, Han D G, Jap D, et al. Non-profiled side-channel attack based on deep learning using picture trace[J]. IEEE Access, 2021, 9: 22480−22492

[11] Zaid G, Bossuet L, François D, et al. Ranking loss: Maximizing the success rate in deep learning side-channel analysis[J]. IACR Transactions on Cryptographic Hardware and Embedded Systems, 2020, 2021(1): 25−55

[12] Wang Kai, Yan Yingjian, Guo Pengfei, et al. Research on power analysis attack based on improved residual network and data augmentation technology[J]. Journal of Cryptologic Research, 2020, 7(4): 551−564 (in Chinese)
(王恺, 严迎建, 郭朋飞, 等. 基于改进残差网络和数据增强技术的能量分析攻击研究[J]. 密码学报, 2020, 7(4): 551−564)

[13] Wang Ping, Chen Ping, Luo Zhimin, et al. Enhancing the performance of practical profiling side-channel attacks using conditional generative adversarial networks[EB/OL]. 2020[2022-01-10]. https://eprint.iacr.org/2020/867

[14] Luo Zhimin, Zheng Mengce, Wang Ping, et al. Towards strengthening deep learning-based side channel attacks with mixup[EB/OL]. 2021[2022-01-10]. https://eprint.iacr.org/2021/312

[15] Abdellatif K M. Mixup data augmentation for deep learning side-channelattacks[EB/OL] 2021[2022-01-10] https://eprint.iacr.org/2021/328

[16] Zhang Hongyi, Cisse M, Dauphin Y N, et al. Mixup: Beyond empirical risk minimization[C/OL] //Proc of the 6th Int Conf on Learning Representations. Amherst, MA: OpenReview. net, 2018 [2022-01-10]. https://openreview.net/pdf?id=r1Ddp1-Rb

[17] Nassar M, Souissi Y, Guilley S, et al. RSM: A small and fast countermeasure for AES, secure against 1st and 2nd-order zero-offset SCAs[C] //Proc of the 15th Design, Automation and Test in Europe Conf and Exhibition. Piscataway, NJ: IEEE, 2012: 1173−1178

[18] Gilmore R, Hanley N, O'Neill M. Neural network based attack on a masked implementation of AES[C] //Proc of the 2015 IEEE Int Symp on Hardware Oriented Security and Trust. Piscataway, NJ: IEEE, 2015: 106−111



Zhang Runlian, born in 1974. PhD, associate professor. Her main research interests include information security and distributed computing.
张润莲, 1974 年生. 博士, 副教授. 主要研究方向为信息安全和分布式计算.



Pan Zhaoxuan, born in 1997. Master candidate. His main research interest includes side channel analysis.
潘兆轩, 1997 年生. 硕士研究生. 主要研究方向为侧信道分析.



Li Jinlin, born in 1997. Master candidate. His main research interests include network security and application, and side channel analysis.
李金林, 1997 年生. 硕士研究生. 主要研究方向为网络安全及应用和侧信道分析.



Wu Xiaonian, born in 1972. Master, professor. His main research interests include information security and distributed computing.
武小年, 1972 年生. 硕士, 教授. 主要研究方向为信息安全和分布式计算.



Wei Yongzhuang, born in 1976. PhD, professors, PhD supervisor. His main research interests include symmetric ciphers and security analysis of protocol.
韦永壮, 1976 年生. 博士. 教授. 博士生导师. 主要研究方向为对称密码和协议安全分析.