Linear Regression Side Channel Attack Applied on Constant XOR

Shan Fu[ab], Zongyue Wang[c⋆], Fanxing Wei[b], Guoai Xu[a], An Wang[d]

[a]National Engineering Laboratory of Mobile Internet Security, Beijing University of Posts and Telecommunications;
[b]China Academy of Information and Communications Technology;
[c]Shandong University;
[d]School of Computer Science, Beijing Institute of Technology

Abstract. Linear regression side channel attack (LRA) used to be known as a robust attacking method as it makes use of independent bits leakage. This leakage assumption is more general than Hamming weight/ Hamming distance model used in correlation power attack (CPA). However, in practice, Hamming weight and Hamming distance model suit most devices well. In this paper, we restudy linear regression attack under Hamming weight/ Hamming distance model and propose our novel LRA methods. We find that in many common scenarios LRA is not only an alternative but also a more efficient tool compared with CPA. Two typical cases are recovering keys with XOR operation leakage and chosen plaintext attack on block ciphers with leakages from round output. Simulation results are given to compare with traditional CPA in both cases. Our LRA method achieves up to 400% and 300% improvements for corresponding case compared with CPA respectively. Experiments with AES on SAKURA-G board also prove the efficiency of our methods in practice where 128 key bits are recovered with 1500 traces using XOR operation leakage and one key byte is recovered with only 50 chosen-plaintext traces in the other case.

Keywords: linear regression; side channel attack;

1 Introduction

Nowadays, embedded devices such as smart cards, mobile phones, RFID tags and even sensor networks are widely used in our daily lives. Though these devices are extensively used, the sensitive data contained in them might be easily recovered by adversaries. Side channel attacks, especially power analysis attacks provide an access to data in cryptographic implementations, which are well-known threat to these devices. The most famous example is Differential Power Analysis introduced by Kocher et al. in 1999[9]. In their experiment, they monitored the power consumption of a smart card and extracted the secret key efficiently. Later, the correlation between power consumption traces and modeled values (e.g. under Hamming weight model) of handled data was taken into account. The CPA was proposed in 2004 by Brier et al.[2]. Subsequently, several works applied this idea to practical environments and achieved good results[3, 13]. Other attack models such as partitioning power analysis(PPA)[10], collision attack[1] and mutual information analysis[6] have also been studied.

Related works Linear regression side channel attack(LRA) has been introduced by Schindler et al. in 2005[15]. Initially, they describe an efficient profiling method for SCA. The attack can recover the IBL (Independent bits leakage) function coefficient based on the known subkey k using Linear Regression. Linear Regression is viewed as an alternative to the template attacks. With coefficient of determination R^2, Doget et al.[5] further developed a non-profiled key-recovery attack. The Linear regression attacks can be applied in the same context as the CPA, but with weaker assumption on the device behavior. In some papers[17], this extension is noted as Linear Regression Attack (LRA). Furthermore, these results of LRA have been extended to apply against first-order masking techniques[4], which are the main SCA

⋆ Corresponding author.

countermeasures. In parallel, linear regression attacks have been used to model the deterministic part of the information leakage for complex circuits. Other contributions [7][8][11]improve the efficiency and effectiveness of LRA when applied to real attack procedure.

Our Contributions Even though independent bits leakage model is more general, the Hamming weight model and the Hamming distance model suit most devices well in practice, especially for leakages from registers and data bus. A general understanding is that Hamming weight model is applicable to software while Hamming distance model to hardware implementations. In this paper, we restudy linear regression side channel attack under Hamming weight/ Hamming distance model. We find that in many common scenarios LRA is not only an alternative but also a more efficient tool compared with CPA. Two typical cases are recovering keys with XOR operation leakage and chosen plaintext attack on block ciphers in T-table software or round based hardware realizations.

In the first case, we recover key from the leakage $m \oplus k$ where m is the message and k is the whitening key. In Hamming weight model, leakage is expressed as a linear function of Hamming weight of the intermediate value $m \oplus k$. We find that in the expression of leakage, the signs of coefficients of every independent bit in m indicates the value of corresponding k bit. Multiple linear regression analysis is used to examine relation between bits in m and leakages, and recover the coefficients. Simulations of 8-bit, 32-bit, 64-bit and 128-bit leakage are given to make comparison between our method and CPA. The result shows that our method is much more efficient in multi-byte situation. For 128-bit leakage, we achieve a 400% improvement compared with CPA. Besides, note that keys are not guessed during regression analysis, the computational complexity is always $O(1)$, which is an another advantage.

In the second case, we focus on block cipher with leakage only in round output. This is a very common and very normal scenario. T-table based software and round-based hardware realizations are both examples. Typically, chosen plaintext trick is used to decrease key-guessing space where some bytes of plaintext are kept stable. However, these stable bytes result in unknown constant mask XORing to the calculated intermediate values used in CPA. As shown in our simulations, unknown constant mask have great impact on CPA efficiency. We prove that linear regression distinguisher can overcome this unknown mask. With this distinguisher, we achieve a 300% improvement compared with CPA.

Paper structure The remaining of the paper is organized as follows: Section 2 illustrates some preliminaries. Section 3 gives the attack on XOR operation. Section 4 gives the attack under constant XOR mask. Section 5 provides the experiments on SAKURA-G board. Finally, we conclude this paper in Section 6.

2 Preliminaries

2.1 Hamming Weight Model and Hamming Distance Model

The Hamming weight model is propose by Kocher[9] and Messerges[12] which generally assumed that leakage through the power side-channel depends on the number of bits set in the data. Let T be the leakage value of data X and $HW(\cdot)$ be the Hamming weight function, the Hamming weight model is described as follow:

$$T = a \cdot HW(X) + c + \sigma.$$

where a is a scalar coefficient, c is a constant consumption and σ is noise.

The Hamming distance model was proposed by Eric Brier et al. in CHES 2004[2] where the leakage is assumed to be depend on the number of bits switching from one state to another. The consumptions for a bit switching from 0 to 1 or from 1 to 0 are further assumed to be same. Let the current state be R and the next state be X. The number of flipping bits equals $HW(R \oplus X)$. The Hamming distance model is described as follow:

$$T = a \cdot HW(R \oplus X) + c + \sigma.$$

where a is a scalar coefficient, c is a constant consumption and σ is noise.

The Hamming weight and the Hamming distance model suit most devices well in practice, especially for leakages from registers and data buses. A general understanding is that Hamming weight

model is applicable to software while Hamming distance model to hardware implementations. As the Hamming weight model and the Hamming distance model are similar, we only describe our method under Hamming weight model in the following for simplification.

2.2 Correlation Power Analysis

Since the significant work of Kocher[9] of side channel attacks in late 1990s, a large amount of work have been devoted. As a most famous successor, Correlation Power Analysis (CPA) is proposed by Brier et al. in 2004.

The correlation coefficient between Hamming weight of target data X and the power consumption T is described as follow:

$$\rho_{HW(X),T} = \frac{cov\left(HW\left(X\right),T\right)}{\sigma_{HW(X)}\sigma_T}.$$

where $cov(\cdot)$ is the covariance between $HW(X)$ and T. σ_X and σ_T are standard deviation for $HW(x)$ and T respectively. When implemented, the target data X should relate to some unknown key bits and the correlation coefficient is used as a distinguisher. The attacker guesses the unknown key bits and calculates $\rho_{HW(X),T}$ for every key candidates. The correct key is supposed to indicate the biggest $\left|\rho_{HW(X),T}\right|$. For detail, we refer to [2].

2.3 Multiple Linear Regression

In statistics, multiple regression is an approach for modeling the relationship between a scalar dependent variable y and several explanatory variables denoted $X = (x_1, x_2 \dots x_p)$. For multiple linear regression (MLR), the relationships are modeled by linear predictor function:

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p. \tag{1}$$

where $\beta = (\beta_0, \beta_1, \cdots, \beta_p)'$ is the unknown model parameter which can be estimated by giving sample sets of y and X. Ordinary least square method is the most commonly used estimator. For given N sample sets

$$y_s = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}, X_s = \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ x_{21} & \cdots & x_{2p} \\ \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{Np} \end{pmatrix},$$

the ordinary least square method first generate a new matrix M as

$$M = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & \cdots & x_{Np} \end{pmatrix}.$$

Then the estimation of β is

$$\hat{\beta} = (M'M)^{-1}M'y_s \tag{2}$$

The confidence of determination, denoted R^2, indicates how well samples fit the linear model established with $\hat{\beta}$.

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y_i})^2}{\sum_{i=1}^{N}(y_i - \bar{y_i})^2}, \tag{3}$$

where $\hat{y_i} = \hat{\beta}_0 + \hat{\beta}_1 x_{1i} + \cdots + \hat{\beta}_p x_{pi}$ is the estimated y_i with the linear model and $\bar{y_i}$ is the mean of y_s. R^2 has a value in the range of $[0, 1]$ with 1.0 being the best fit.

2.4 Linear Regression Distinguisher

Linear regression distinguisher is proposed by Doget et al. to perform "robust side channel attack"[5]. Let T be the leakage measurement. Choose an n-bit target value v_k which depends on part of key bits. V_k is further denoted as $(v_k[n], v_k[n-1], \ldots, v_k[1])$ which is its binary decomposition. Instead of correlation coefficient, the linear regression distinguisher test the linear relationship using R^2. The process is as follows:

1. For every key candidate $\hat{k}$, calculate $(v_{\hat{k}}[n], v_{\hat{k}}[n-1], \ldots, v_{\hat{k}}[1])$ for every measurement.
2. Construct the model between T and $(v_{\hat{k}}[n], v_{\hat{k}}[n-1], \ldots, v_{\hat{k}}[1])$ as

$$T = \beta_{\hat{k},0} + \beta_{\hat{k},1} v_{\hat{k}}[1] + \cdots + \beta_{\hat{k},n} v_{\hat{k}}[n].$$

   Estimate the parameter $\beta_{\hat{k}} = (\beta_{\hat{k},0}, \beta_{\hat{k},1}, \cdots, \beta_{\hat{k},n})'$ with ordinary least square method as shown in Sec.2.3.
3. Compute and store the confidence of determination $R^2_{\hat{k}}$ for $\hat{k}$.

The key candidate $\hat{k}$ with largest $R^2_{\hat{k}}$ is considered to be the right key with highest level of confidence.

3 Linear Regression Attack on XOR Operation

Original linear regression attack is known as an alternative to CPA using R^2 instead of ρ to distinguish key candidates. The advantage is that the model assumption is weaker. On Hamming weight compatible devices, LRA gives a similar efficiency compared with CPA. In this section, we give a novel attack method on XOR operation using multiple linear regression which achieve higher efficiency.

3.1 Leakage from XOR Operation

XOR Operation is one of the most commonly used operation in ciphers. For example, in most block ciphers, the plaintext m is XORed with whitening key k as the first step. Attackers can use leakage from $m \oplus k$ to launch a side channel attack.

Typically, CPA is used to recover k as follows:

1. For every key candidates $\hat{k}$, calculate $L_{\hat{k}} = m \oplus \hat{k}$.
2. Further calculate $\rho_{HW(L_{\hat{k}}),T}$ and sort key candidates according to $\left|\rho_{HW(L_{\hat{k}}),T}\right|$.
3. Output the key candidates with largest $\left|\rho_{HW(L_{\hat{k}}),T}\right|$.

In block ciphers, k is usually a 128-bit value. Considering the performing architecture, the leakage can be 8-bit, 16-bit, 32-bit, 64-bit and even 128-bit (most for hardware situation). In CPA procedure, guessing a multi-byte k value results in high computation cost which is almost impossible for 32-bit, 64-bit and 128-bit architectures. One solution is separating k into different parts and performing CPA in every part. When performing one part, leakages from other parts are considered noise. Although this solution works, the efficiency is reduced as only part of information is used.

Our method is based on insight of Hamming weight model. m and k are denoted as $(m[n], m[n-1], \ldots, m[1])$ and $(k[n], k[n-1], \ldots, k[1])$ respectively, which are their binary decomposition. According to the Hamming weight model, the leakage of target value $m \oplus k$ is expressed as:

$$\begin{aligned} T &= a \cdot HW(m \oplus k) + c + \sigma \\ &= a \cdot \sum_{j=1}^{n} (m[j] \oplus k[j]) + c + \sigma. \end{aligned}$$

As k is a stable value, we have

$$m[j] \oplus k[j] = \begin{cases} m[j] & if\ k[j] = 0 \\ 1 - m[j] & if\ k[j] = 1 \end{cases}.$$

Hence,

$$\begin{aligned} T &= a \cdot \left(\sum_{k[j]=0} m[j] + \sum_{k[j]=1} (1 - m[j]) \right) + c + \sigma \\ &= a \cdot \sum_{k[j]=0} m[j] - a \cdot \sum_{k[j]=1} m[j] + a \cdot \sum_{k[j]=1} 1 + c + \sigma. \end{aligned}$$

Observation 1. *In the leakage expression, the bits with value '0' and the bits with value '1' in k give opposite sign of coefficient of corresponding bits in m.*

3.2 Attack Procedure

According to Observation 1, the sign of coefficient of every bit in m gives a predict to corresponding k bit. We use multiple linear regression to estimate the coefficient, taking T as the dependent variable and $m = (m[n], m[n-1], \ldots, m[1])$ as the explanatory variables. In practical terms, T is not a single value but a set $T = (T_1, T_2, \cdots, T_N)$ which formed a trace. The attack procedure is as follows:

1. Random choose plaintext m, perform encryption and record leakage traces.
2. For every trace point variable T_j, construct the model between T_j and $(m[n], m[n-1], \ldots, m[1])$ as
$$T = \beta_0 + \beta_1 m[1] + \cdots + \beta_n m[n].$$
Estimate the coefficient $\beta = (\beta_0, \beta_1, \cdots, \beta_n)'$ with ordinary least square method as shown in Sec.2.3. Calculate and store the corresponding confidence of determination R_j^2.
3. For the largest R_j^2, recover every bit in k as $\hat{k}[i] = 0$ if β_i is positive and $\hat{k}[i] = 1$ otherwise.
4. Test the $\hat{k}$ and $\neg\hat{k}$ where $\neg$ is flip. Output the correct one.

As we do not guess key, our method is applicable to leakage on any architecture. In multi-byte leakage situation, our method makes use of whole information of the leakage which brings a higher efficiency compared with CPA. Besides, the computational complexity is always $O(1)$ which is another benefit.

3.3 Simulation

To compare the efficiency between our method and CPA, we do simulation on 8-bit, 32-bit, 64-bit and 128-bit k with noise $\sigma = 2$. 1000 parallel experiments using different number of traces are performed to estimate success rates. As illustrated in Figure 1, the efficiency of our method and CPA almost match in 8-bit situation. But in multi-byte situation, our method is much more efficient. For 128-bit k, CPA requires more than 2000 traces to reach success rate 1 while our method needs only 500 traces which means a 400% improvement of efficiency.

4 Linear Regression Attack under Constant XOR Mask

When doing side channel attacks, the attacker may face a constant XOR mask in target value. Inspired from original linear regression distinguisher, we give a method to overcome this constant XOR mask efficiently.

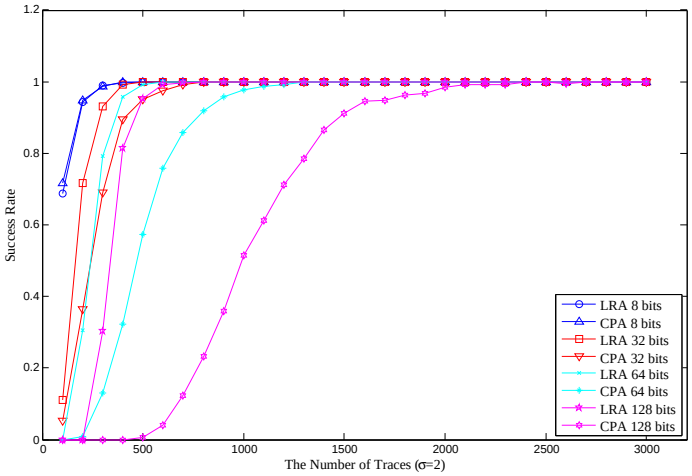

Fig. 1: Simulation of LRA and CPA on 8-bit, 32-bit, 64-bit and 128-bit k

4.1 Leakage under Constant XOR Mask

Leakage under constant XOR mask means that leakage T is caused by $x \oplus u$ where x is an intermediate value related to some guessed key bit and u is an unknown constant. This is a common scenario in side channel attacks, especially in chosen plaintext cases. To perform CPA, one way is to further guess u using $\rho_{HW(x\oplus u),T}$ as a distinguisher. This method is computationally infeasible when u is 32-bit or larger. In [16], Tu et al. suggests directly use $\rho_{HW(x),T}$ as a distinguisher as there exists some linear correlations between $HW(x)$ and $HW(x \oplus u)$. This is still not a perfect solution because $\rho_{HW(x),T}$ is highly effected by u, which may lose efficiency under some u values.

Similar as in Section 3.1, in the Hamming weight model, the leakage T can be expressed in bitwise. We have

$$T = a \cdot \sum_{u[j]=0} x[j] - a \cdot \sum_{u[j]=1} x[j] + a \cdot \sum_{u[j]=1} 1 + c + \sigma.$$

Even though the mask u changes the sign of coefficient of some bits in x, it doest not affect the linear relationship between x and T. So we can perform linear regression between x and T and takes the confidence of determination $R^2_{T,x}$ as distinguisher which is not affected by u. Hence linear regression distinguisher using $R^2_{T,x}$ overcome the mask without losing efficiency .

A simple simulation is made to make comparison between CPA distinguiser $\rho_{HW(x),T}$ and LRA distinguisher $R^2_{T,x}$. In the simulation, u is a 8-bit value and $T = HW(x \oplus u)$ without considering the noise and constant consumption for simplicity. The result is shown in Table 1. We can see that $\rho_{HW(X),T}$ changes over $HW(u)$ while $R^2_{T,x}$ is always stable.

Table 1: R^2 and ρ when $HW(u)$ changes

$HW(u)$	0	1	2	3	4	5	6	7	8
$R^2_{T,x}$	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
$\rho_{HW(x),T}$	1.000	0.746	0.495	0.245	-0.003	-0.248	-0.497	-0.747	-1.000

Observation 2. *The confidence of determination R^2 in linear regression distinguisher does not affected by constant XOR mask.*

4.2 Chosen Plaintext Linear Regression Attack

Chosen plaintext is a common technique attacking block ciphers. However, this technique usually leads a constant XOR mask to the target value, which is an obstacle recovering keys. We give an example attacking AES to show how linear regression distinguisher overcome this obstacle efficiently.

AES is block cipher supporting 128-bit blocks and 128/192/256-bit keys [14]. Based on substitution permutation network (SPN) structure, AES XORs whitening key first and performs 10/12/14 round functions. Except the last one, every round is consist of SubBytes (SB), ShiftRows (SR), MixColumn (MC) and AddKeys (AK). SubBytes works on each byte of cipher state, which is the only non-linear function. In software implements, T-table is usually used for higher efficiency where SubBytes and MixColumn operations are combined outputting 32-bit states.

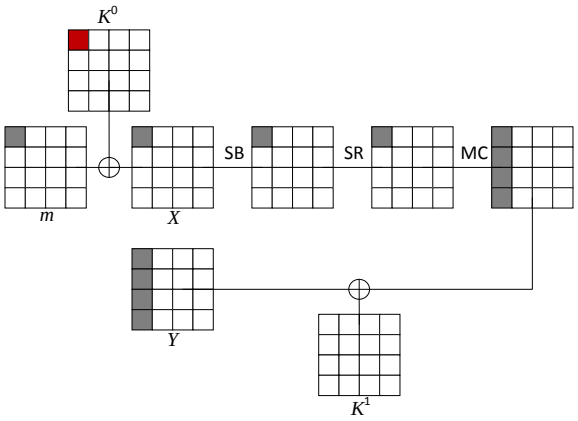

Fig. 2: Hardware implementation of AES

Figure 2 illustrates the first round of AES with whitening key. In T-table software or round based hardware implementations, there is leakage in Y, the output of first round. Taking one byte of Y as target value, the adversary has to guess 5 bytes of key (4 bytes of whitening key and 1 byte of first round key) in CPA procedure which result in high computational complexity . To deduce the guessing space, chosen plaintext technique is applied. The adversary can choose plaintext varying only in one byte e.g. the first byte. Except the first column, other columns in Y are constants. As MixColumn is linear, the first column of Y is expressed as

$$Y_0 = MC\left(S\left(X_{0,0}\right), 0, 0, 0\right) \oplus c,$$

where c is an unknown 32-bit value. Taking Y_0 as target value, only one byte of whitening key , $K_{0,0}$, need to be guessed. However, as we describe in Section 4.1, the constant XOR mask c have great effect on the result of CPA. According to Observation 2, we can keep the chosen plaintext attacking procedure but use linear regression distinguisher to bypass the effect of c.

4.3 Simulation

We simulate chosen plaintext attack on AES using both CPA and LRA. 1000 parallel experiments with random selected key are performed to estimate success rates. As shown in Figure 3, LRA achieves 100% success rate with about 400 traces. Because the effect of unknown constant mask, the highest success rate of CPA is about 60%, with 800 traces. When CPA failed, the adversary can chose another group of plaintext to change the value of constant XOR mask and repeat the attack. So on average, CPA reach 100% success rate with 1333 traces. This means, our LRA attack improves more than 300% efficiency compared with CPA.

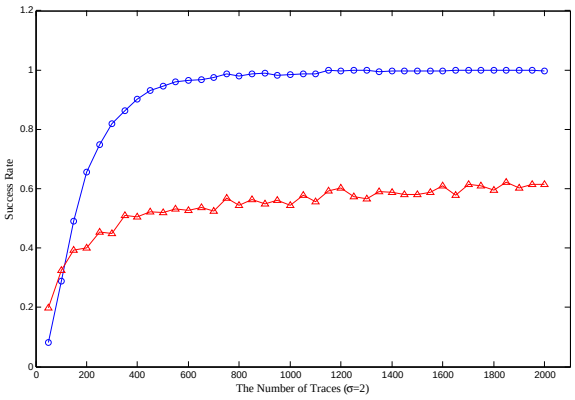

Fig. 3: Simulation of chosen plaintext attack on AES

5 Experiments

In this section, we test our methods in practice using SAKURA-G board, performing round-based AES implementation and acquiring the power consumption with an oscilloscope.

5.1 Description of AES Implementation

We synthesize the official code on SAKURA-G board to perform AES encryptions. This implementation

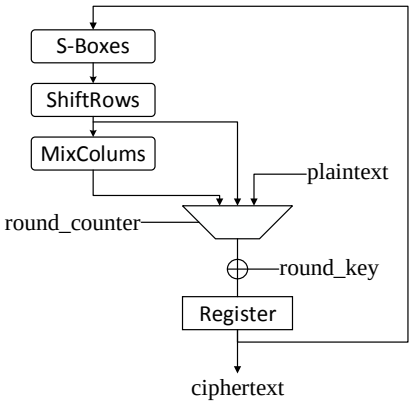

Fig. 4: Hardware implementation of AES

is a standard paralleled hardware realization of AES. As shown in Figure 4, the message is first XOR'd with the whitening key and stored into the register. Then for every clock cycle, the chip performs one AES round including **SubBytes**, **ShiftRows**,**MixColumns** and **AddRoundKey**. The **MixColumns** step is omitted from the last round. The power traces acquired by the oscilloscope is shown in Figure 5, where we can clearly recognize the pattern of round operation.

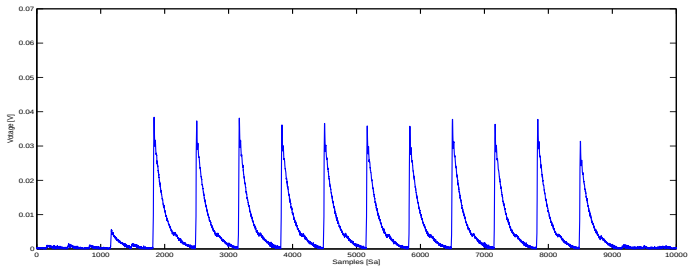

Fig. 5: Power consumption of one AES encryption

5.2 Experimental Results

We test the Linear regression side channel attack on XOR operation with 2000 traces to recover 128-bit key directly. The result in Fig 6 shows R^2 with all traces. Using trace point with the highest R^2, all 128 bits of key are recovered. Also, as Fig 7 indicates, as the number of test traces raises from 0 to 2000, the number of correct bits recovered increases. The correct bits we recovered would be more than 110 bits out of 128 bits in total within only 400 traces. The experiment proves that our method has incredibly high efficiency in multi-byte situation.

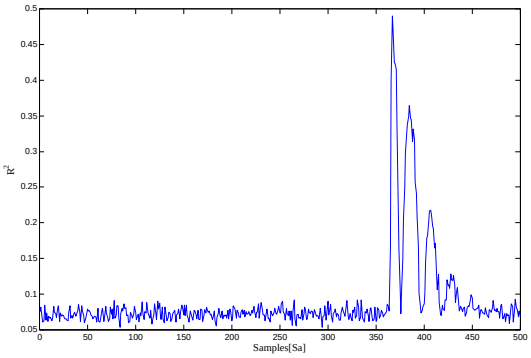

Fig. 6: R^2 with 2000 traces

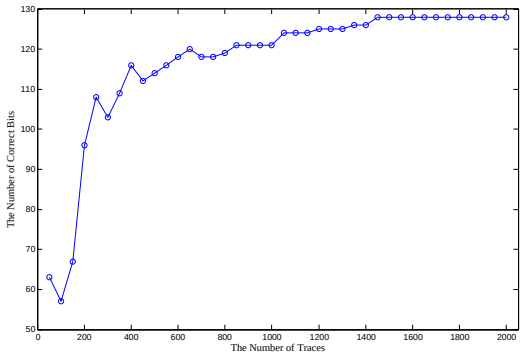

Fig. 7: Correct bits recovered

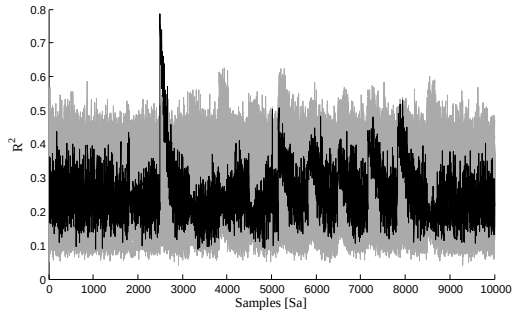

Fig. 8: Linear Regression Attack on AES

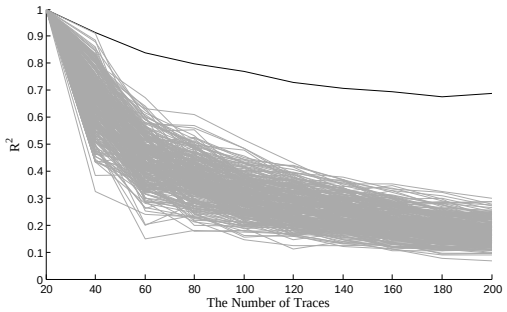

Fig. 9: Efficiency of Linear Regression Attack

For LRA under constant XOR mask situation, we test our linear regression attack with 100 traces. The experimental results is shown in Fig 8 where grey lines mean the R^2 of the wrong key, the black line means the R^2 of the correct. It is obviously that the correct key has a higher R^2 value.

To test the efficiency, we compute our linear regression attack under different number of traces. The result is illustrated in Fig 9. Grey lines mean the R^2 of the key candidates, the black line means the R^2 of the correct key. It shows that with only 50 original traces, the correct key can be recovered by LRA method.

6 Conclusion

In this paper, we give another look at linear regression side channel attack under Hamming weight/ Hamming distance model. We find that LRA has great advantages than CPA in many general cases. We propose two typical cases, recovering keys with XOR operation leakage and chosen plaintext attack on block ciphers in T-table software or round based hardware implementations. For the first case, in 128-bit leakage, we achieve as high as 400% improvement compared with CPA. Furthermore, the computational complexity is only $O(1)$. For the second case, we show that LRA is extremely powerful as it can overcome unknown constant mask. We believe that this characteristic of linear regression provides a feasible attacking method which could be used in many other cases. Experiments on AES are also given which verify the efficiency of two typical cases in practice.

References

1. Andrey Bogdanov. Improved side-channel collision attacks on aes. In *Selected Areas in Cryptography*, pages 84–95. Springer, 2007.
2. Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In *Cryptographic Hardware and Embedded Systems-CHES 2004*, pages 16–29. Springer, 2004.
3. Christophe Clavier, Benoit Feix, Georges Gagnerot, Mylene Roussellet, and Vincent Verneuil. Improved collision-correlation power analysis on first order protected aes. In *Cryptographic Hardware and Embedded Systems–CHES 2011*, pages 49–62. Springer, 2011.
4. Guillaume Dabosville and Emmanuel Prouff. A new second-order side channel attack based on linear regression. *IEEE Transactions on Computers*, 62(8):1629–1640, 2013.
5. Julien Doget, Emmanuel Prouff, Matthieu Rivain, and François-Xavier Standaert. Univariate side channel attacks and leakage modeling. *Journal of Cryptographic Engineering*, 1(2):123–144, 2011.
6. Benedikt Gierlichs, Lejla Batina, Pim Tuyls, and Bart Preneel. Mutual information analysis. In *Cryptographic Hardware and Embedded Systems–CHES 2008*, pages 426–442. Springer, 2008.
7. Annelie Heuser, Werner Schindler, and Marc Stottinger. Revealing side-channel issues of complex circuits by enhanced leakage models. pages 1179–1184, 2012.
8. M Kasper, W Schindler, and M Stottinger. A stochastic method for security evaluation of cryptographic fpga implementations. In *International Conference on Field-Programmable Technology*, pages 146–153, 2010.
9. Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *Advances in Cryptology-CRYPTO99*, pages 388–397. Springer, 1999.
10. Thanh-Ha Le, Jessy Clédière, Cécile Canovas, Bruno Robisson, Christine Servière, and Jean-Louis Lacoume. A proposition for correlation power analysis enhancement. In *Cryptographic Hardware and Embedded Systems-CHES 2006*, pages 174–186. Springer, 2006.
11. Victor Lomn, Emmanuel Prouff, and Thomas Roche. Behind the scene of side channel attacks. *Lecture Notes in Computer Science*, 8269:506–525, 2013.
12. Thomas S. Messerges, Ezzy A. Dabbish, and Robert H. Sloan. Investigations of power analysis attacks on smartcards. In *Usenix Workshop on Smartcard Technology on Usenix Workshop on Smartcard Technology*, pages 17–17, 1999.
13. Amir Moradi, Oliver Mischke, and Thomas Eisenbarth. Correlation-enhanced power analysis collision attack. In *Cryptographic Hardware and Embedded Systems, CHES 2010*, pages 125–139. Springer, 2010.
14. Vincent Rijmen and Joan Daemen. Advanced encryption standard. *Proceedings of Federal Information Processing Standards Publications, National Institute of Standards and Technology*, pages 19–22, 2001.

15. Werner Schindler, Kerstin Lemke, and Christof Paar. A stochastic model for differential side channel cryptanalysis. 3659:30–46, 2005.
16. Chenyang Tu, Neng Gao, Zeyi Liu, Lei Wang, Zongbin Liu, and Bingke Ma. A practical chosen message power analysis method on the feistel-sp ciphers with applications to clefia and camellia. *IACR Cryptology ePrint Archive*, 2015:174, 2015.
17. Carolyn Whitnall, Elisabeth Oswald, and Franois Xavier Standaert. The myth of generic dpa...and the magic of learning. *Lecture Notes in Computer Science*, 8366:183–205, 2014.