Speed-up of SCA attacks on 32-bit multiplications

Robert NGUYEN[1], Adrien FACON[1,3], Sylvain GUILLEY[1,2,3], Guillaume GAUTIER[4], and Safwan ELASSAD[5]

[1] Secure-IC S.A.S. - Think Ahead Business Line, 35 510 Cesson-Sévigné, France
[2] LTCI, Telecom ParisTech, COMELEC department 75 013 Paris, France
[3] École Normale Supérieure Département d'Informatique 75 005 Paris, France
[4] INSA RENNES, 35 708 Rennes Cedex 7
[5] IETR Laboratory, UMR CNRS 6164; VAADER team, NANTES

Abstract. Many crypto-algorithms, Deep-Learning, DSP compute on words larger than 8-bit. SCA attacks can easily be done on Boolean operations like XOR, AND, OR, and substitution operations like s-box, p-box or q-box, as 8-bit hypothesis or less are enough to forge attacks. However, attacking larger hypothesis word increases exponentially required resources: memory and computation power. Considering multiplication, 32-bit operation implies 2^{32} hypothesis. Then a direct SCA attack cannot be efficiently performed. We propose to perform instead 4 small 8-bit SCA attacks. 32-bit attack complexity is reduced to 8-bit only complexity.

Keywords: SCA · arithmetic multiplication · 32-bit · divide and conquer · 8-bit · reduce partition size · fault model · neural network · Deep learning · signal processing · PID · automotive · avionic · LFSR · PUF · chaotic pseudo-random generator

1 Introduction

Following the low cost of 32-bit microcontrollers that substitute to 8-bit and 16-bit microcontrollers in embedded product, more and more algorithms use 32-bit operators. IoT firmware may then embed technical secret values of processing, meaning then key-knowledge of the product. SCARE approach (SCA+RE) is a way to retrieve such secret. It uses Side Channel Analysis (SCA) [1] to extract statistical information from product behavior (consumption and/or EM radiation) to perform Reverse Engineering (RE) and the retrieve secret.

Initial work has been done on a Vernam-like cipher using a PRNG based on Chaotic cell [2], [3], [4], [5]. The purpose of work was to retrieve 15 words of 32-bit from the secret keys of the PRNG. 12 words are used in a sum of products for a linear feedback. This article describes a side-channel attack on 32-bit multiplication, alone multiply operation or multiply-and-add operation. The attack has been performed on "ma" instruction of ARM-v2 which computes a multiply-and-add operation.

This 32-bit multiplication vulnerability can be applied on multiple other targets and for a large spectrum of applications. One can consider targets using

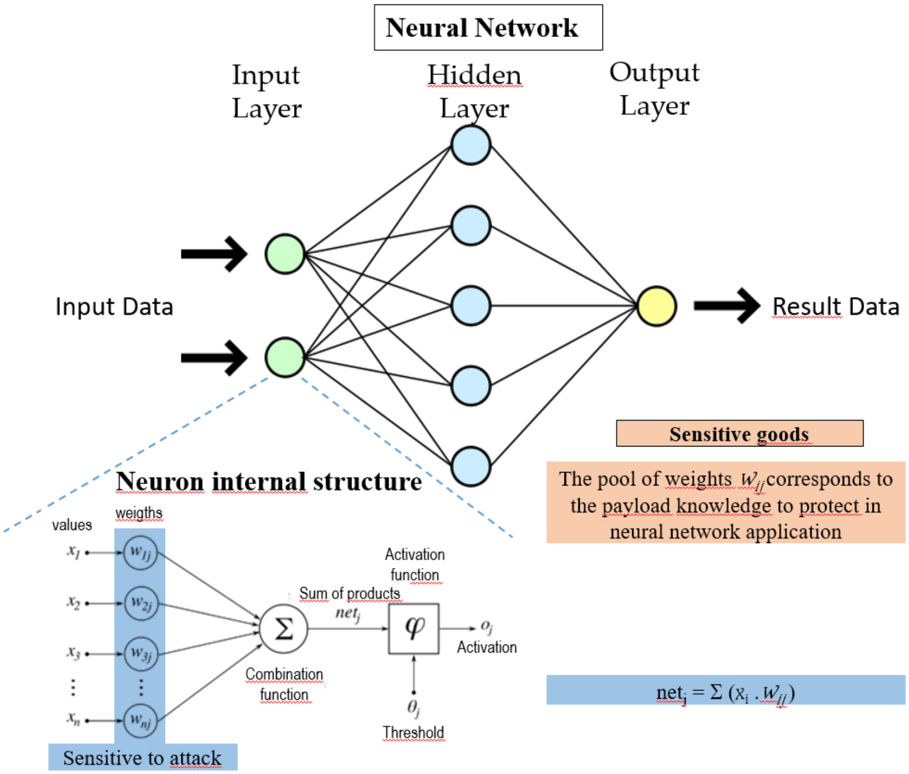

Fig. 1. Attack on sensitive data in neural network

neuronal network for deep learning [6], [7]. (see example in Fig. 1). Also coefficients of FIR-IIR filter for signal processing are sensitive goods (eg. FIR parameter used for preprocessing by a SCA attack at [8] could be retrieved by SCA counterattack). (see example in Fig. 2). Also coefficients of PID for control loop in avionic or automotive actuators ([9]) are goods for advanced functionalities. (see example in Fig. 3). Last examples of applications deal with cryptographic functions in TPM may also include such 32-bit operations for Linear Return Function (LRF) in LFSR (pseudo-random generator), for HASH function or for PUF[10] (post-processing of PUF measurements). (see example in Fig. 4).

2 Complexity of attacking 32-bit multiplication

The targeted operation to attack is an arithmetic multiplication of two 32-bit values. The result is truncated at 32 bits, a modulus 2^{32}. This 32-bit multiplication vulnerability against SCA has been identified on multiple targets. As the whole 32-bit word is needed for computation, following [11] statistical SCA

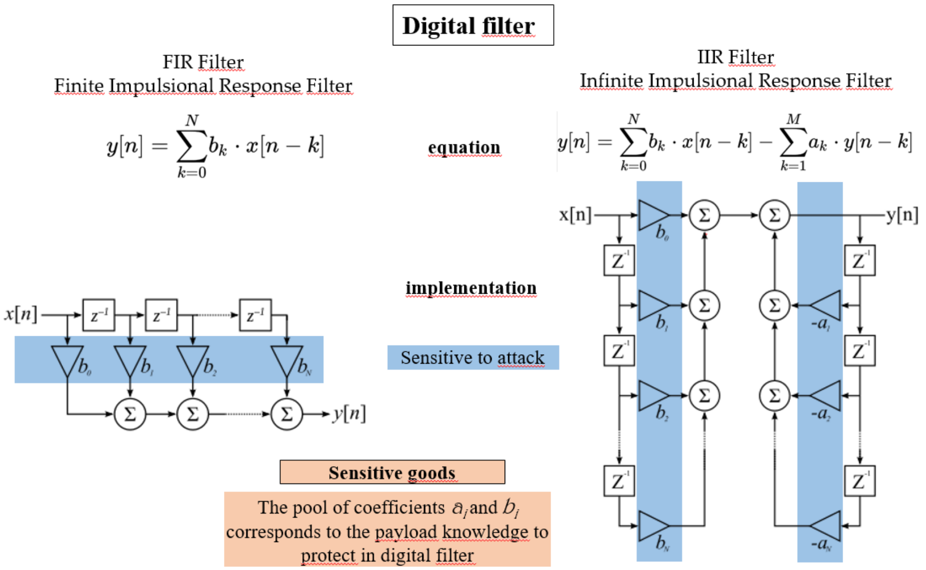

Fig. 2. Attack on sensitive coefficients of FIR-IIR Filter

attacks with a leakage model should need 2^{32} partitions to discriminate the secret multiplicand value. This implies a large memory resource to store 4 billion independent traces and associated computing power to calculate intermediate results for CPA or DPA at each new measurement of a multiplication activity.

Actually, current available computer resource can be enough for such partition and computation power. But it is still a waste of resources (memories and computation time). For example, attacking with 1k-points traces, makes $2^{32} =$ 4G partitions of 1k-points of 4 (or 8) bytes each. This imply to manage 16 TB of memory to store intermediate differential traces. When 10k-traces are enough to discriminate 8-bit hypothesis, 40k-traces will be needed at least for 32-bit hypothesis.

This will imply to manage $16 * 10^{12} * 40 * 10^{3} = 640 * 10^{15}$ Bytes, meaning more than 10^{18} operations (31 years of computation on 1GHz computer).

3 Split the attack

Instead of attacking the whole word, we propose a different approach based on divide and conquer. The single attack with 2^{32} partitions is substituted by 4 small and sequential attacks on 2^{8} partitions.

You can note this strategy to attack 32-bit word can be extended to larger word, (N x 8) bits word can be attacked through N successive attacks on 8-bit value.

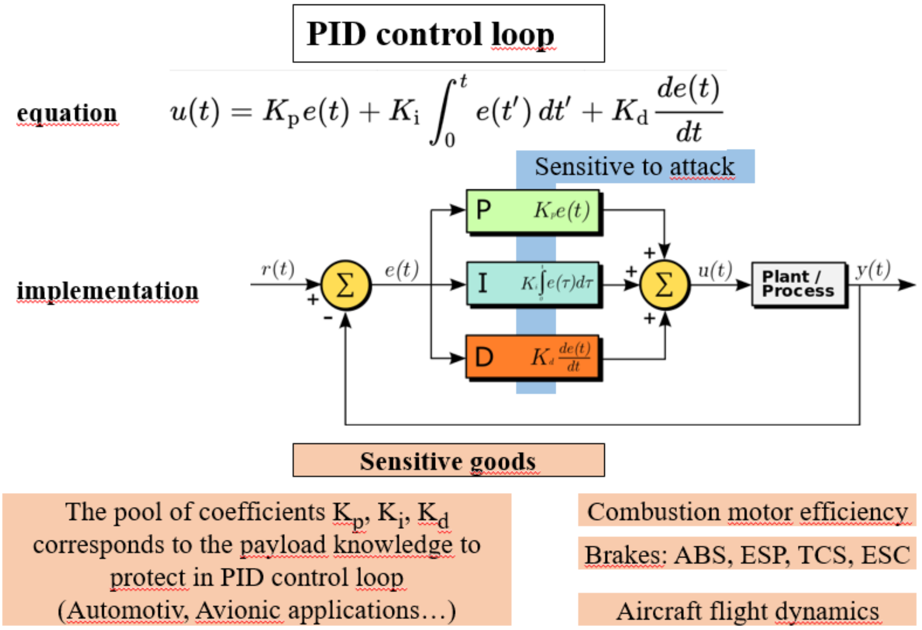

Fig. 3. Attack on sensitive coefficients of PID control loop

The proposed approach will split this single attack into 4 small attacks on 8 bits of secret key but computation still uses 32-bit multiplication[6].

First of all is to describe the operands and elementary operations of the multiplication.
Each 32-bit word can be assumed as a vector of four 8-bit bytes:

- $Y = [Y3, Y2, Y1, Y0]$: result Y = K * X
- $K = [K3, K2, K1, K0]$: secret key which is the multiplier constant
- $X = [X3, X2, X1, X0]$: data to multiply

Note: "$\ll$" operator corresponds to a bit-shifter operator, $c = a \ll b$ sets c to a value left shifted from b bits. The operation of "left shift from 1 bit" is equivalent to "multiply by 2". Using the "$\ll$" operator, Y can be rewrite in byte sub-operation as the following:

As result of multiplication is truncated to 32-bit, "Y" expression can be simplified as:

---

[6] Actually, for some cryptographic operations, such as AES, it is natural to cut the 128-bit datapath in 16 bytes, as the algorithm is programmed this way. But regarding the 32-bit multiplication, it is less obvious that the attacker can choose to focus specifically on sub-words, which actually normally have interactions between them (through carries). This is the point which makes our result remarkably non-obvious and interesting in terms of divide-and-conquer approach.

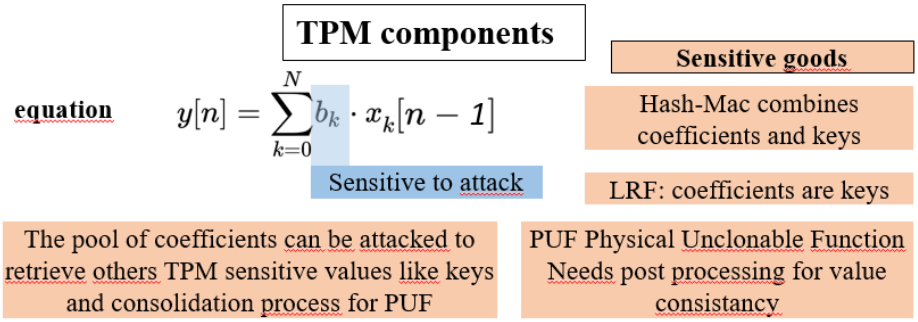

Fig. 4. Attack on sensitive goods inside a TPM

$$\begin{aligned}
Y = {} & (K3.X0) \ll 24 + (K3.X1) \ll 32 + (K3.X2) \ll 40 + (K3.X3) \ll 48 + \\
& (K2.X0) \ll 16 + (K2.X1) \ll 24 + (K2.X2) \ll 32 + (K2.X3) \ll 40 + \\
& (K1.X0) \ll 8 \ \ + (K1.X1) \ll 16 + (K1.X2) \ll 24 + (K1.X3) \ll 32 + \\
& (K0.X0) \ll 0 \ \ + (K0.X1) \ll 8 \ \ + (K0.X2) \ll 16 + (K0.X3) \ll 24
\end{aligned}$$

Amongst 16 initial intermediate multiplications, only 10 multiplications are really needed. This triangle representation reveals that part of the key can be selected in operation only by selecting Xi values.

4 Attack steps

4.1 Step 1 - Retrieve K0

If X0, X1 and X2 can be forced to zero (0), then
$Y = ((K0.X3) \ll 24)$ & `0xFF000000`.
A SCA attack with variation on X3 enables to retrieve K0 with only 256 partitions and up-to 256 traces. The leakage model is (only 8 low weight bits):
$\mathcal{L}(K0) : HW(Y) = HW((K0.X3)$ & `0xFF`$)$
$HW(Y)$ takes value in [0:8]
In case of noisy measurements, multiple traces can be acquired and average for each X3 value to reduced noise impact.

4.2 Step 2 - Retrieve K1

The attack strategy is the same but with different Xi forced to zero. If X0, X1 and X3 can be forced to zero (0), then
$Y = (K1.X2) \ll 24 + (K0.X2) \ll 16$.
A SCA attack with variation on X2 enables to retrieve K1 with only 256 partitions and up-to 256 traces. This attack needs to know the value of $K0$.

$$\begin{aligned}Y = {} & (K3.X0) \ll 24 + \\ & (K2.X0) \ll 16 + (K2.X1) \ll 24 + \\ & (K1.X0) \ll 8 \ \ + (K1.X1) \ll 16 + (K1.X2) \ll 24 + \\ & (K0.X0) \ll 0 \ \ + (K0.X1) \ll 8 \ \ + (K0.X2) \ll 16 + (K0.X3) \ll 24\end{aligned}$$

The leakage model is:
$\mathcal{L}(K1) : HW(Y) = HW(((K1.X2)\ \&\ \texttt{0xFF}) \ll 8 + (K0.X2))$
$\mathcal{L}(K1) : HW(Y) = HW(((K1 \ll 8 + K0).X2)\ \&\ \texttt{0x0000FFFF})$
$HW(Y)$ takes value in [0:16]
In case of noisy measurements, multiple traces can be acquired and average for each X2 value to reduced noise impact.

4.3 Step 3 - Retrieve K2

The attack strategy is the same but with different Xi forced to zero. If X0, X2 and X3 can be forced to zero (0), then
$Y = (K2.X1) \ll 24 + (K1.X1) \ll 16 + (K0.X1) \ll 8.$
A SCA attack with variation on X1 enables to retrieve K2 with only 256 partitions and up-to 256 traces. This attack needs to know the value of $K0$ and $K1$.
The leakage model is:
$\mathcal{L}(K2) : HW(Y) = HW(((K2.X1)\ \&\ \texttt{0xFF}) \ll 16 + (K1.X1) \ll 8 + (K0.X1))$
$\mathcal{L}(K2) : HW(Y) = HW(((K2 \ll 16 + K1 \ll 8 + K0).X1)\ \&\ \texttt{0x00FFFFFF})$

$HW(Y)$ takes value in [0:24]
In case of noisy measurements, multiple traces can be acquired and average for each X1 value to reduced noise impact.

4.4 Step 4 - Retrieve K3

The attack strategy is the same but with different Xi forced to zero. If X1, X2 and X3 can be forced to zero (0), then
$Y = (K3.X0) \ll 24 + (K2.X0) \ll 16 + (K1.X0) \ll 8 + (K0.X0) \ll 0.$
A SCA attack with variation on X0 enables to retrieve K3 with only 256 partitions and up-to 256 traces. This attack needs to know the value of $K0$, $K1$ and $K2$.
The leakage model is:
$\mathcal{L}(K3) : HW(Y) = HW(((K3.X0)\&\ \texttt{0xFF}) \ll 24 + (K2.X0) \ll 16 + (K1.X0) \ll 8 + (K0.X0))$
$\mathcal{L}(K3) : HW(Y) = HW(((K3 \ll 24 + K2 \ll 16 + K1 \ll 8 + K0).X0)\&\ \texttt{0xFFFFFFFF})$
$HW(Y)$ takes value in [0:32]
In case of noisy measurements, multiple traces can be acquired and average for each X0 value to reduced noise impact.

4.5 Conclusion

The complex attack on K (32-bit) is replaced by 4 small attacks on 8-bit word: $K = [K3, K2, K1, K0]$. The order of the sequence of attacks remains as the last constraint to know few sub-keys K_i before attacking next sub-key K_j.

5 Benchmark

5.1 SCA attack on 8-bit multiplication

Each of 8-bit SCA attack presented in the previous chapter is based on the same attack scenario.

The 8-bit attack, used by the previous attacks, is a classical statistical SCA. CPA is chosen as distinguisher as it can converge quickly, even in noisy condition.

5.2 Performance on Software implementation

A single 8-bit attack on 1k-points traces requires $256 * 1024 * 8 = 2M$ bytes of memory and for computational resources $32 * 1024 * 256 = 8M$ multiplications and $256 * 1024 * 256 = 32\text{M}$ additions.
For the whole attack, this corresponds to 2M-bytes of memory, 32M-multiplications and 128M-Additions.

In comparison, a direct 32-bit attack needs 16 TB (16 Million of MB) of memory and 10^{18} operations (10^{12} * 1M operations).

6 Conclusion

By splitting big-word variables into an array of bytes, the complex attack of a N-Bytes word multiplication can be substituted by N small attacks on 8-bit words. The attack complexity $O(2^{32})$ is replaced by $4{*}O(2^8)$. The gain of memory is over 10 million and the gain of computation is 1 billion. Then the new method allows to compute the attack in 1 second on embedded computer (1GHz mono-core, 4MB of memory) instead of 31 years with 16 TB of memory.

7 Glossary

Chaotic Cell	Compute a value x(n+1) with $x(n+1) = f(x(n))$ that makes a prediction of x(n+p) very complex if p>1.
CPA	Correlation Power Analysis.
CEMA	Correlation Electro-Magnetic Analysis.
Double	an extended floating-point value on 64-bit (8 bytes), IEEE defined.
EM	ElectroMagnetic.
FIR	Finite Impulse Response, a filter defined by:

	$Y(n) = \sum_{i=1}^{N}[X(n-i) * a(i)]$
Float	a floating-point value on 32-bit, IEEE defined.
GB	Giga-Bytes = 10^9 Bytes (Billion).
HASH	Data transformation to produce a compressed signature. This signature is used to test data integrity.
HD	Hamming Distance, HW of the transition of a register value when update: $HD(reg(n)) = HW(\ reg(n)\ XOR\ reg(n-1)\ )$.
HW	Hamming Weight, number of "1" in binary representation of a number.
IRR	Infinite Impulse Response, a filter defined by $Y(n) = \sum_{i=1}^{N}[X(n-i) * a(i)] - \sum_{j=1}^{M}[Y(n-j) * b(j)]$
LFSR	Linear-Feedback Shift Register.
LRF	Linear Return Function.
MAC	Multiply-and-Accumulate, same as Multiply-and-Add.
MB	Mega-Bytes = 10^6 Bytes (Million).
Multiply-and-Add	Two operation executed by a single instruction $Y = a * X + b$.
Neural Network	In Artificial Intelligence (A.I.) context, set neurons organized and interconnected in layers to process and reduce number of values.
Neuron	Each neuron of a layer computes a value from sum of product of its inputs and propagate a post-processed value to upper layer of neurons.
PID	Proportional, Integral and Derivative; definite a three-term controller in a control loop feedback mechanism.
PRNG	Pseudo-Random Number Generator, produce a predetermined sequence of value that simulate random, an initial seed give the beginning of the sequence.
PUF	Physical Unclonable Function. Use silicon intrinsic property to produce a unique ID, even from the same logical gate/transistor definition. Post-processing using multiplication can be used to forge better quality PUF.
RE	Reverse Engineering.
RNG	Random Number Generator, can be a TRNG or a PRNG.
SCARE	Side-Channel Analysis for Reverse Engineering.
SCA	Side-Channel Analysis.
TB	Tera-Bytes = 10^{12} Bytes (Millions of million).
TPM	Trusted Platform Module.
TRNG	True Random Number Generator, use physical property to produce unpredictable random number (Eg. atomic desintegration).
XOR	eXclusive OR.

References

1. Kocher, P., Jaffe, J., and Jun, B. (1999). Differential Power Analysis. In Wiener, M., editor, Advances in Cryptology — CRYPTO' 99, volume 1666 of Lecture Notes in Computer Science book series (LNCS), pages 388– 397, Berlin, Heidelberg. Springer Berlin Heidelberg.
2. El Assad et al., "Chaos-based Block Ciphers: An Overview", IEEE, 10th International Conference on Communications, COMM-2014, Bucharest, Romania, May 2014, pp. 23-26.
3. El Assad, Farajallah, "A new Chaos-Based Image Encryption System". Signal Processing: Image Communication 41, (2016) 144-157.
4. G.Gautier,Safwan El Assad: Design and Efficient Implementations of a Chaos-based stream cipher for Securing Internet of Things"., IETR Laboratory, UMR CNRS 6164; VAADER team, NANTES, FRANCE:2017-12-08, Talk, Journée GDR-ISIS, IRISA, Rennes.
5. G.Gautier,Safwan El Assad: A Promising Chaos-based Stream Cipher, IETR Laboratory, UMR CNRS 6164; VAADER team, NANTES, FRANCE:2018-01-18, Talk at Polytech Nantes.
6. Lejla Batina[1], Shivam Bhasin[2], Dirmanto Jap[2], Stjepan Picek[3]: CSI Neural Network - Using Side-channels to Recover Your Artificial Neural Network Information: [1] Institute for Computing and Information Sciences, Radboud University, Nijmegen, The Netherlands; [2] Physical Analysis and Cryptographic Engineering, Temasek Laboratories at Nanyang Technological University, Singapore; [3] Delft University of Technology, Delft, The Netherlands: arXiv:1810.09076v1 [cs.CR] 22th Oct 2018.
7. Pierre-Alain Moellic,: The Dark Side of Neural Networks: an Advocacy for Security in Machine Learning: CEA Tech, Centre CMP, Équipe Commune CEA Tech - Mines Saint-Etienne, F-13541 Gardanne FRANCE pierre-alain.moellic@cea.fr: CESAR 2018 J1-05.
8. David Oswald and Christof Paar: "Improving Side-Channel Analysis with Optimal Pre-Processing", Horst Görtz Institute for IT Security, Ruhr-University Bochum, Germany, {david.oswald, christof.paar}@rub.de CARDIS 2012,16
9. Hari Om Bansal, Rajamayyoor Sharma, P. R. Shreeraman: "PID Controller Tuning Techniques - A Review", Electrical and Electronics Engineering Department, Birla Institute of Technology and Science, Pilani, India,: hobansal@gmail.com Journal of Control Engineering and Technology, JCET Vol. 2 Iss. 4 October 2012 PP. 168-176 www.vkingpub.com
10. Physically Unclonable Function - PUF, SR2I301 : https://perso.telecom-paristech.fr/danger/SR2I301/PUF.pdf
11. Brier, E., Clavier, C., Olivier, F.: Correlation power analysis with a leakage model. In: Cryptographic Hardware and Embedded Systems — CHES 2004. LNCS, vol. 3156, pp. 16–29. Springer (2004)