A note on the security of threshold implementations with $d + 1$ input shares

Santos Merino del Pozo and François-Xavier Standaert
{santos.merino, fstandae}@uclouvain.be

ICTEAM/ELEN/Crypto Group, Université catholique de Louvain, Belgium.

Abstract. Recently, threshold implementations (TI) with $d + 1$ input shares have been proposed at Crypto 2015. This optimization aims for more lightweight TI designs while keeping the glitch-resistance of the original concept. In this note, we consider such an approach and provide preliminary simulation-based evidence, backed by empirical results, of the existence of d^{th}-order leakages. We conclude that, while for first-order TI designs this solution can be overkill due to the extra randomness requirements, higher-order TIs can still benefit from it.

1 Threshold implementations

The major requirement to guarantee the security level of *masking schemes* is the so-called independent leakage assumption (jeopardized by glitches, cross-talk or memory transitions). Yet, practical investigations (cf. [6,7]) have shown the hardness to achieve such a property in hardware due to the propagation of glitches, i.e., bogus transitions in the output(s) of a combinatorial circuit within a clock cycle. In this context, *threshold implementation* (TI) is one of the few masking schemes that can ensure the resistance of masked hardware implementations to first-order attacks (cf. [10,11]), then extended in the so-called HOTI to any arbitrary order d (cf. [2]), based on sound security arguments. Typically, the d^{th}-order TI of, e.g., a single-bit output non-linear function $f : (x_1, \dots, x_n) \mapsto y$ with algebraic degree t, can be built with $s_{\text{in}} \geq t \cdot d + 1$ input shares and $s_{\text{out}} \geq \binom{s_{\text{in}}}{t}$ output shares. More recently, it has been suggested that d^{th}-order TIs can be designed using $s_{\text{in}} = d + 1$ input shares (i.e., independently of the algebraic degree t of the considered function f), therefore allowing for designs consuming less randomness and hardware resources (cf. [13]).

2 What can go wrong?

Early publications on TI designs claimed that one amongst their most noteworthy achievements was the fact that, when the output shares were uniformly distributed, no additional randomness was required. However, in more recent works this statement has been shown to be true only for first-order secure designs (cf. [12,13]). More concretely, since the HOTI concept did not consider *multivariate* scenarios, if a number of shares from different clock cycles are combined, then a d^{th}-order TI can leak (exploitable) information at the d^{th}-order statistical moment. To the best of our knowledge, the only known mitigation to this issue is to insert a *refreshing layer* injecting fresh random bits after shared functions. In this cautionary note, we show that a tweaked TI design using $d + 1$ input shares still can exhibit data-dependencies at the d^{th}-order statistical moment.

Table 1: First-order (2,4)-share TI output behavior in function of the given inputs.

(x_1,x_2,x_3)	(y_1,y_2)				μ	σ^2	$x_1^2 = X$	
	(0,0)	(0,1)	(1,0)	(1,1)			μ_0	μ_1
(0,0,0)	4	0	0	4	1	1	0	2
(0,1,0)	4	0	0	4				
(0,0,1)	4	0	0	4			1	1
(1,1,1)	4	0	0	4				
(0,1,1)	0	4	4	0	1	0	1	1
(1,0,0)	0	4	4	0				
(1,0,1)	0	4	4	0				
(1,1,0)	0	4	4	0				

3 The shared AND/XOR function with 2 shares in [13]

Let consider the AND/XOR module $y = f(x_1, x_2, x_3) = x_1 + x_2 \cdot x_3$ (i.e., with algebraic degree $t = 2$) in the KATAN block cipher, and its first-order TI representation with $s_{\text{in}} = 2$ input shares and $s_{\text{out}} = 4$ output shares as provided in [13]:

$$\begin{aligned} \tilde{y}^1 &= x_1^1 + x_2^1 \cdot x_3^1 \\ \tilde{y}^2 &= x_2^1 \cdot x_3^2 \qquad\qquad & y^1 &= \tilde{y}^1 + \tilde{y}^2 \\ \tilde{y}^3 &= x_1^2 + x_2^2 \cdot x_3^1 & y^2 &= \tilde{y}^3 + \tilde{y}^4. \\ \tilde{y}^4 &= x_2^2 \cdot x_3^2 \end{aligned} \tag{1}$$

It is noteworthy that when $s_{\text{out}} > s_{\text{in}}$, the uniform sharing of non-linear functions cannot be achieved. Therefore, following the rules of HOTI designs, after storing the output shares $\tilde{y}^{j\in\{1\dots,4\}}$ in registers (i.e., to avoid the propagation of glitches), a compression layer (i.e., a 4-to-2 resharing with no randomness) needs to be introduced to meet the uniformity property (see Equation (1)). When the inputs are drawn from a uniform distribution and assuming a Hamming Weight leakage model, the shared outputs are distributed as depicted in Table 1, obviously satisfying the uniformity requirements imposed by TI designs. Expectedly for a first-order masking scheme, only the second-order analysis reveals information on the inputs. However, as shown by the two most right columns, when the first-order analysis is elaborated in function of the x_1^2 share, then the means of the leakages are not constant anymore over all unshared inputs.

4 Empirical analysis

In order to practically examine this leakage behavior, we have considered KATAN (cf. [3]), a family of hardware-oriented block ciphers featuring a fully-serialized architecture and which has been extensively considered in the context of TI designs (cf. [2,9]). Concretely speaking, we have targeted KATAN-32, the smallest version with 32-bit plaintext and 80-bit key. After saving the plaintext (resp. key) in a serial

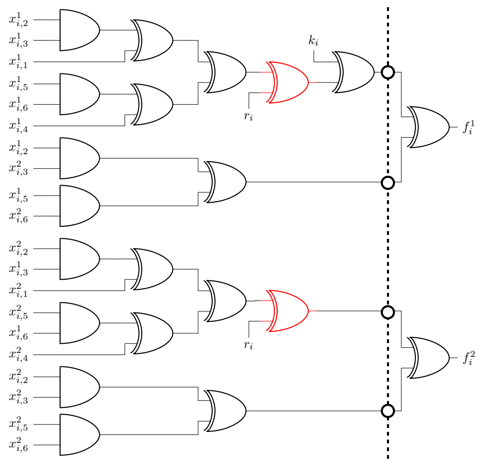


Fig. 1: Considered first-order (2,4)-share TI design. The registers are marked by bold circles, the XOR gates on their right hand side implement the compression layer, and the optional XOR gates for the refreshing layer are depicted in red.

fashion into the corresponding state (resp. key) shift registers, the crypto core is clocked 254 times to produce the ciphertext (which is then taken from the contents of the state register). The basic building block of KATAN is the AND/XOR module required by the two non-linear functions $f_1(\dots)$ and $f_2(\dots)$ such that:

$$f_i(x_{i,1}, x_{i,2}, x_{i,3}, x_{i,4}, x_{i,5}, x_{i,6}) = \\ (x_{i,1} + x_{i,2} \cdot x_{i,3}) + (x_{i,4} + x_{i,5} \cdot x_{i,6}) + k_i$$

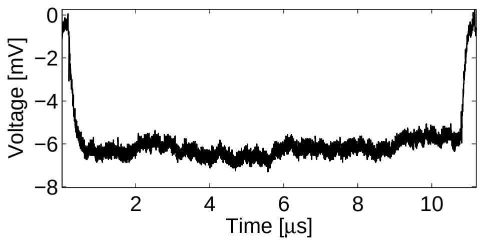
where each $x_{i,j}$ represents a single bit of the state register. On each clock cycle, k_1 and k_2 are generated from the initial 80-bit key by means of and LFSR so then the state register is updated using the two output bits produced by $f_1(\dots)$ and $f_2(\dots)$.

We do not represent the key register and the key schedule in a shared form (the common trend in the TI literature), so only the state registers need to be doubled. In the datapath, the linear functions, which are limited to the additional XORs in $f_i(\dots)$, are repeated on a per-share basis, and the non-linear AND/XOR instances are implemented using the formulas given in Equation (1). The resulting TI construction of $f_i(\dots)$, including the refreshing machinery, is shown in Figure 1. Besides, we have followed a glitch-free FPGA-based prototyping methodology (cf. [8]), so we make sure to not be biased by the impact of glitches.

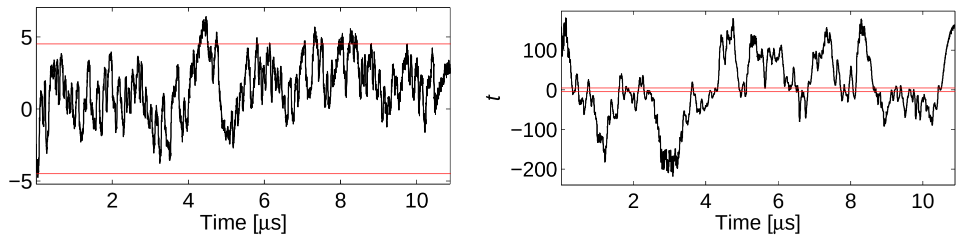
4.1 Setup

We have acquired power consumption traces from a SAKURA-G board (cf. [1]) using a 12-bit[1] high resolution oscilloscope, i.e., a Teledyne LeCroy HRO 66Zi in sequence mode at 500MS/s sampling rate.

[1] Note that, for comparison purposes with existing results, we have not exploited this feature. Therefore, we have considered the 8-bit outputs of the internal ADC.



(a) Sample power trace



(b) 1st-order (c) 2nd-order

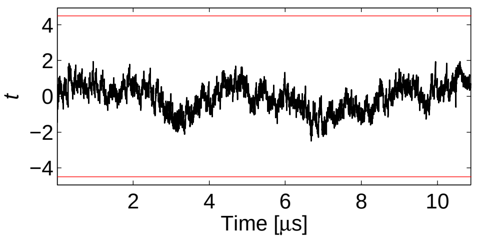
Fig. 2: (original design) Non-specific (fixed vs. random) t-test results using $25,000,000$ traces recorded with the PRNG enabled.

Besides, we have limited the bandwidth to 20 MHz and used a passive probe (i.e., a coaxial cable) plugged in the amplified output provided by the SAKURA-G so as to record high-quality (low-noise) signals. Moreover, in order to obtain relatively short power traces while covering the 254 encryption rounds of KATAN-32, the targeted design have been clocked at 48 MHz.

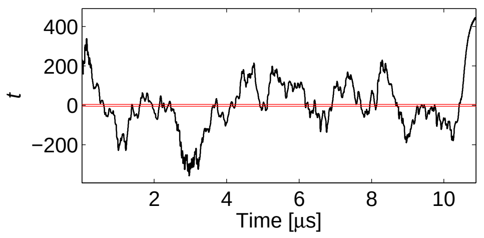
4.2 Leakage assessment

We have collected $25,000,000$ power traces to conduct a non-specific (fixed vs. random) t-test (cf. [4,14]) up to second order, with all the PRNGs and the glitch-freeness gadgetry enabled. A sample power trace is provided in Figure 2(a). The corresponding results depicted in Figure 2(b) and Figure 2(c) confirm the vulnerability at first- and second-order moments. Consequently, in order to disallow the first-order leakage, we have incorporated a refreshing layer as depicted in red in Figure 1, such that the shared outputs are computed as:

$$\begin{aligned} \tilde{f}_i^1 &= x_{i,1}^1 + x_{i,2}^1 \cdot x_{i,3}^1 + x_{i,4}^1 + x_{i,5}^1 \cdot x_{i,6}^1 + k_i + r_i \\ \tilde{f}_i^2 &= x_{i,2}^1 \cdot x_{i,3}^2 + x_{i,5}^1 \cdot x_{i,6}^2 & f_i^1 &= \tilde{f}_i^1 + \tilde{f}_i^2 \\ \tilde{f}_i^3 &= x_{i,1}^2 + x_{i,2}^2 \cdot x_{i,3}^1 + x_{i,4}^2 + x_{i,5}^2 \cdot x_{i,6}^1 + r_i & f_i^2 &= \tilde{f}_i^3 + \tilde{f}_i^4, \\ \tilde{f}_i^4 &= x_{i,2}^2 \cdot x_{i,3}^2 + x_{i,5}^2 \cdot x_{i,6}^2 \end{aligned} \tag{2}$$



(a) 1st-order



(b) 2nd-order

Fig. 3: (design with additional refreshing) Non-specific (fixed vs. random) t-test results using $25,000,000$ traces recorded with the PRNG enabled.

where r_i denotes the single bit of randomness required to perform the refreshing layer. So, in total, this approach consumes two fresh mask bits per encryption round. Therefore, we have integrated two additional PRNGs in our original design and recorded $25,000,000$ traces to conduct the same non-specific (fixed vs. random) t-test up to the second order. The corresponding resuts, given in Figure 3, confirm the effectiveness of applying the aforementioned refreshing layer to prevent the first-order leakage (as expected).

5 Discussion

In this short note, we have been only able to achieve first-order resistance by injecting fresh random bits (in the very limited context of the KATAN block cipher). Compared to conventional first-order TI designs (where no fresh randomness is required), the increased resource utilization induced by the additional PRNGs can be a limiting factor for low-area low-power applications. Indeed, the main goal towards reducing the number of input shares is to alleviate the resource overhead incurred by TI circuits. However, when considering HOTI designs (where the refreshing layer is unavoidable), this approach becomes more relevant. As an example, let consider the following second-order (3,9)-share TI representation of the AND/XOR module:

$$\begin{aligned}
\tilde{y}^1 &= x_1^1 + x_2^1 \cdot x_3^1 \\
\tilde{y}^2 &= x_2^1 \cdot x_3^2 \\
\tilde{y}^3 &= x_2^2 \cdot x_3^1 \\
\tilde{y}^4 &= x_1^2 + x_2^2 \cdot x_3^2 \qquad & y^1 &= \tilde{y}^1 + \tilde{y}^2 + \tilde{y}^3 \\
\tilde{y}^5 &= x_2^2 \cdot x_3^3 & y^2 &= \tilde{y}^4 + \tilde{y}^5 + \tilde{y}^6 \\
\tilde{y}^6 &= x_2^3 \cdot x_3^2 & y^3 &= \tilde{y}^7 + \tilde{y}^8 + \tilde{y}^9. \\
\tilde{y}^7 &= x_1^3 + x_2^3 \cdot x_3^3 \\
\tilde{y}^8 &= x_2^3 \cdot x_3^1 \\
\tilde{y}^9 &= x_2^1 \cdot x_3^3
\end{aligned} \tag{3}$$

Table 2: Second-order (3,9)-share TI output behavior in function of the given inputs.

(x_1,x_2,x_3)	(y_1,y_2,y_3)								μ	σ^2	γ	$x_1^2 = X$			
	(0,0,0)	(0,0,1)	(0,1,0)	(0,1,1)	(1,0,0)	(1,0,1)	(1,1,0)	(1,1,1)				μ_0	μ_1	σ_0^2	σ_1^2
(0,0,0)	16	0	0	16	0	16	16	0	1.5	0.75	-1.15	1.38	1.63	0.86	0.61
(0,1,0)	16	0	0	16	0	16	16	0							
(0,0,1)	16	0	0	16	0	16	16	0							
(1,1,1)	16	0	0	16	0	16	16	0							
(0,1,1)	0	16	16	0	16	0	0	16	1.5	0.75	1.15	1.38	1.63	0.61	0.86
(1,0,0)	0	16	16	0	16	0	0	16							
(1,0,1)	0	16	16	0	16	0	0	16							
(1,1,0)	0	16	16	0	16	0	0	16							

As shown by Table 2, Equation (3) is a uniform second-order TI with 3 input shares. Indeed, the analysis must be performed up to the third order to detect the difference of the skewness for different unshared inputs. Yet, we face the same issue as with the formulas given in (1). In this case, a second-order analysis performed in function of the x_1^2 share reveals data-dependent information in the variance if no additional randomness is injected. However, as stated before, the refreshing layer is anyway required when implementing HOTIs, hence moving towards $d+1$ designs significantly reduces the area and randomness requirements compared to its counterpart using a (5,10)-share representation (cf. [2]).

References

1. Side-channel AttacK User Reference Architecture. `http://satoh.cs.uec.ac.jp/SAKURA/index.html`
2. Bilgin, B., Gierlichs, B., Nikova, S., Nikov, V., Rijmen, V.: Higher-Order Threshold Implementations. In: Sarkar, P., Iwata, T. (eds.) Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014, Proceedings, Part II. Lecture Notes in Computer Science, vol. 8874, pp. 326–343. Springer (2014), `http://dx.doi.org/10.1007/978-3-662-45608-8_18`
3. Cannière, C.D., Dunkelman, O., Knezevic, M.: KATAN and KTANTAN - A Family of Small and Efficient Hardware-Oriented Block Ciphers. In: Clavier, C., Gaj, K. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop, Lausanne, Switzerland, September 6-9, 2009, Proceedings. Lecture Notes in Computer Science, vol. 5747, pp. 272–288. Springer (2009), `http://dx.doi.org/10.1007/978-3-642-04138-9_20`
4. Cooper, J., De Mulder, E., Goodwill, G., Jaffe, J., Kenworthy, G., Rohatgi, P., et al.: Test Vector Leakage Assessment (TVLA) methodology in practice. In: International Cryptographic Module Conference (2013), `http://icmc-2013.org/wp/wp-content/uploads/2013/09/Rohatgi_Test-Vector-Leakage-Assessment.pdf`
5. Güneysu, T., Handschuh, H. (eds.): Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings, Lecture Notes in Computer Science, vol. 9293. Springer (2015), `http://dx.doi.org/10.1007/978-3-662-48324-4`
6. Mangard, S., Popp, T., Gammel, B.M.: Side-Channel Leakage of Masked CMOS Gates. In: Menezes, A. (ed.) Topics in Cryptology - CT-RSA 2005, The Cryptographers' Track at the RSA Conference 2005, San Francisco, CA, USA, February 14-18, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3376, pp. 351–365. Springer (2005), `http://dx.doi.org/10.1007/978-3-540-30574-3_24`

7. Mangard, S., Pramstaller, N., Oswald, E.: Successfully Attacking Masked AES Hardware Implementations. In: Rao, J.R., Sunar, B. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2005, 7th International Workshop, Edinburgh, UK, August 29 - September 1, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3659, pp. 157–171. Springer (2005), `http://dx.doi.org/10.1007/11545262_12`
8. Moradi, A., Mischke, O.: Glitch-Free Implementation of Masking in Modern FPGAs. In: 2012 IEEE International Symposium on Hardware-Oriented Security and Trust, HOST 2012, San Francisco, CA, USA, June 3-4, 2012. pp. 89–95. IEEE (2012), `http://dx.doi.org/10.1109/HST.2012.6224326`
9. Moradi, A., Wild, A.: Assessment of Hiding the Higher-Order Leakages in Hardware - What Are the Achievements Versus Overheads? In: Güneysu and Handschuh [5], pp. 453–474, `http://dx.doi.org/10.1007/978-3-662-48324-4_23`
10. Nikova, S., Rijmen, V., Schläffer, M.: Secure Hardware Implementation of Non-linear Functions in the Presence of Glitches. In: Lee, P.J., Cheon, J.H. (eds.) Information Security and Cryptology - ICISC 2008, 11th International Conference, Seoul, Korea, December 3-5, 2008, Revised Selected Papers. Lecture Notes in Computer Science, vol. 5461, pp. 218–234. Springer (2008), `http://dx.doi.org/10.1007/978-3-642-00730-9_14`
11. Nikova, S., Rijmen, V., Schläffer, M.: Secure Hardware Implementation of Nonlinear Functions in the Presence of Glitches. J. Cryptology 24(2), 292–321 (2011), `http://dx.doi.org/10.1007/s00145-010-9085-7`
12. Reparaz, O.: A note on the security of Higher-Order Threshold Implementations. Cryptology ePrint Archive, Report 2015/001 (2015), `http://eprint.iacr.org/`
13. Reparaz, O., Bilgin, B., Nikova, S., Gierlichs, B., Verbauwhede, I.: Consolidating Masking Schemes. In: Gennaro, R., Robshaw, M. (eds.) Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9215, pp. 764–783. Springer (2015), `http://dx.doi.org/10.1007/978-3-662-47989-6_37`
14. Schneider, T., Moradi, A.: Leakage Assessment Methodology - A Clear Roadmap for Side-Channel Evaluations. In: Güneysu and Handschuh [5], pp. 495–513, `http://dx.doi.org/10.1007/978-3-662-48324-4_25`