Correlation-Immune Boolean Functions for Leakage Squeezing and Rotating S-Box Masking against Side Channel Attacks

Claude Carlet

LAGA, Universities of Paris 8 and Paris 13; CNRS, UMR 7539;
Department of Mathematics,University of Paris 8, 2 rue de la liberté
93526 Saint-Denis cedex 02, France
claude.carlet@univ-paris8.fr

Boolean functions, from $\mathbb{F}_{2^n}$ to $\mathbb{F}_2$, have been playing an important role in stream ciphers, because they can be used in their pseudo-random generators to combine the outputs to several LFSR (in the so-called combiner model). Recall that the keystream (which is bitwise added to the plaintext for producing the ciphertext) is in such framework the sequence output by the function during a sufficient number of clock-cycles. The combiner Boolean function must then be balanced, that is, have uniform output distribution, for avoiding some straightforward distinguishing attack; and it should be *correlation-immune* of highest possible order.

An n-variable Boolean function $f(x_1, \ldots, x_n)$ is correlation-immune of some order $m < n$ (in brief, m-CI) if fixing at most m of the n input variables $x_1, \ldots, x_n$ does not change the output distribution of the function, whatever are the positions chosen for the fixed variables and the values chosen for them (a balanced m-CI function is called m-resilient). Such m-th order correlation immunity allows resisting the Siegenthaler attack [15] at the order m, which is a divide and conquer cryptanalysis using the existence of a correlation between the output to the function and m input bits $x_{i_1}, \ldots, x_{i_m}$, to make an exhaustive search of the initialization of the LFSRs of indices $i_1, \ldots, i_m$ (given a sub-sequence of the keystream), without needing to know the initialization of the other LFSRs. The initialization of these other LFSRs can subsequently be recovered by diverse methods, allowing rebuilding the whole keystream. Of course, a correlation attack at the order $m + 1$ is possible if the function is not $(m + 1)$-CI, but the attacker needs then to know a longer part of the keystream for recovering the initialization of $m + 1$ LFSR in order to rebuild the rest of the keystream. The function must also have large algebraic degree for allowing resistance to the Berlekamp-Massey attack [12] and lie at large Hamming distance from affine functions, that is, have large nonlinearity, for allowing resistance to the fast correlation attack [13] and its variants.

These constraints were roughly the only ones needed on the combiner function at the end of the last century (recall that all that can be done for asserting the security of a stream cipher is to ensure that it resists all the known attacks and has enough randomness for hoping resisting future attacks; indeed, no attempt has been successful for building an efficient stream cipher whose security

B. Gierlichs, S. Guilley, and D. Mukhopadhyay (Eds.): SPACE 2013, LNCS 8204, pp. 70–74, 2013.

could be proved like for block ciphers). Since the beginning of this century, a series of cryptanalyses called algebraic attacks has been discovered: the standard algebraic attack [9], the fast algebraic attack [8] and the Rønjom-Helleseth attack [14] (see a survey in [2] recalling the corresponding constraints on Boolean functions). No construction of infinite classes of resilient functions allowing resistance to all attacks on the combiner model is known yet, even when the notion of correlation-immunity is weakened as proposed in [7]. The study of CI functions has become then (maybe temporarily) more theoretical than really practical, from cryptographic viewpoint.

A new role in cryptography for correlation-immune functions, which renews their interest in cryptography, has appeared very recently in the framework of *side channel attacks* (in brief, SCA).

The implementation of cryptographic algorithms over devices like smart cards, FPGA, ASIC, leaks information on the secret data, leading to very efficient SCA. These attacks allow recovering the key from few plaintext-ciphertext pairs in a few seconds if no counter-measure is included in the algorithm and/or the device. Recall that for being considered robust, a cryptosystem should not be cryptanalysed by an attack needing less than 2^{80} elementary operations (which represent thousands of centuries of computation with a modern computer) and less than billions of plaintext-ciphertext pairs. This high level of security is achievable when the attacker has no information on the data processed by the algorithm when the cryptosystem is run. This model of attack is called the *black box* cryptanalysis model. In practice, as soon as the cryptosystem runs over some device, some information on this data leaks through electromagnetic waves or power consumption; a more appropriate model is then that of *grey box* cryptanalysis, in which the attacker does not have access to the exact data processed by the device (this would correspond to a white box model) but to a partial information on this data, which can be for instance a noisy version of the Hamming weight of some *sensitive variable* depending on a few bits of the secret key. He can then measure the leakage during a series of implementations with the same key and different plaintexts, and apply statistical methods to determine the most probable values of these few bits. This is particularly problematic when the cipher is implemented in smart cards or in hardware. And it is particularly true for iterative ciphers like block ciphers since it is then possible to attack the first round, when the diffusion is not yet optimal.

Fortunately, counter-measures exist, but they are costly in terms of running time (more in software applications), of implementation area (in hardware applications) and program executable file size (in software), all the more if they need to resist higher order side channel attacks (the d-th order attack computes the variance of the d-th power of the leakage for determining the most probable value of the sensitive variable; the complexity of the attack is exponential in the order). In fact, the cost overhead is then too high for real-world products.

The most commonly used counter-measure is a secret-sharing method called *masking*. It is efficient both for implementations in smart cards (which are software implementations including a part of hardware) and FPGA or ASIC (hardware implementation). The principle of masking is to replace every sensitive variable Z by a number of *shares* $M_0, \ldots, M_d$ such that the knowledge of some of them, but not all, gives no information on the value of Z, and the knowledge of all of them allows recovering this value. In other words, Z is a deterministic function of all the M_i, but is independent of $(M_i)_{i \in I}$ if $|I| \leqslant d$. The simplest way of achieving this is to draw $M_1, \ldots, M_d$ at random (they are then called *masks*) and to take M_0 such that $M_0 + \cdots + M_d$ equals the sensitive variable, where $+$ is a relevant group operation (in practice, the bitwise XOR). This counter-measure allows resisting the SCA of order d.

Correlation immune functions allow reducing, at least in two possible ways, the overhead of masking while keeping the same resistance to d-th order SCA, when the leak is simply (a noisy version of) a linear combination over the reals of the bits of the sensitive variable (such asumption is quite realistic in general):

- by applying a method called *leakage squeezing*, which allows achieving with one single mask the same protection as with d ones, with d strictly larger than 1. This method has been introduced in [11] and further studied in [10]; it has been later generalized in [3] to several masks. In its original single-mask version, it uses a bijective vectorial function F; the mask M_1 is not processed as is in the device, but in the form of $F(M_1)$. The condition for achieving resistance to d-th order SCA is that the graph indicator of F, that is, the $2n$-variable Boolean function whose support equals the graph $\{(x, y)/y = F(x)\}$ of F, is d-CI. Such graph is a *complementary information set* code (CIS code for short), in the sense that it admits (at least) two information sets which are complement of each other. The condition that the indicator of this CIS code is m-CI is equivalent to saying that the dual distance of this code is at least $d + 1$.
- an alternative way of resisting higher order SCA with one single mask consists in avoiding processing the mask M_1 at all: for every sensitive variable Z which is the input to some box S in the block cipher, Z is replaced by $Z + M_1$ where M_1 is drawn at random, and $Z + M_1$ is the input to a "masked" box S_{M_1} whose output is $S(Z)$ (or more precisely, is a masked value of $S(Z)$, since the process of masking must continue during the whole implementation). This method, called *Rotating Sbox Masking* (RSM), obliges, for each box S in the cipher, to implement a look-up table for each masked box S_{M_1} (in fact, this is costly in practice only for nonlinear boxes). To reduce the corresponding cost, M_1 is not drawn at random in the whole set of binary vectors of the same length as Z, but in a smaller set of such vectors. The condition for achieving resistance to d-th order SCA is that the indicator of this set is a d-CI function. Of course, given d, we wish to choose this non-zero d-CI function with lowest possible weight, since the size of the overhead due to the masked look-up tables is proportional to the Hamming weight of this d-CI function (note however that if the cipher is made like the AES, with identical substitution boxes up to affine equivalence, the substitution layer can be slightly modified so as to be masked at no extra cost: the affine equivalent boxes are replaced by masked versions of a same box). Moreover, with

RSM, some keys are indistinguishable; specifically, the attacker recovers the affine space equal to the set of null linear structures of the indicator of the masks, translated by the correct key. This means that an exhaustive search is required to finish the side-channel analysis [6].

In both cases, this needs to use correlation-immune functions of low weights (with a particular shape in the case of leakage squeezing since the function must then be the indicator of the graph of a permutation). Most of the numerous studies made until now on CI functions dealt with resilient functions, and it happens that the known constructions of resilient functions do not work for constructing low weight CI functions. We shall review what is known on CI functions in this framework and on CIS codes, basing us on the survey work [1] on minimal weight CI functions in at most 13 variables and on the papers on CIS codes [5,4]; we shall investigate constructions.

Acknowledgement. All the work on counter-measures to side channel attacks has been made in collaboration with Sylvain Guilley. I take this opportunity to thank him deeply for our exciting collaboration.

References

1. Bhasin, S., Carlet, C., Guilley, S.: Theory of masking with codewords in hardware: low weight d-th order correlation-immune functions. IACR ePrint Archive 303 (2013)
2. Carlet, C.: Boolean functions for cryptography and error-correcting codes, in Boolean Models and Methods in Mathematics, Computer Science, and Engineering, ser. In: Crama, Y., Hammer, P.L. (eds.) Encyclopedia of Mathematics and its Applications, ch. 8, vol.134, pp. 257–397. Cambridge University Press, Cambridge (2010), `http://www.math.univ-paris13.fr/carlet/pubs.html`
3. Carlet, C., Danger, J.-L., Guilley, S., Maghrebi, H.: Leakage Squeezing of Order Two. In: Galbraith, S., Nandi, M. (eds.) INDOCRYPT 2012. LNCS, vol. 7668, pp. 120–139. Springer, Heidelberg (2012)
4. Carlet, C., Freibert, F., Guilley, S., Kiermaier, M., Kim, J.-L., Solé, P.: Higher-order CIS codes. IEEE Transactions on Information Theory (Submitted 2013)
5. Carlet, C., Gaborit, P., Kim, J.-L., Solé, P.: A new class of codes for Boolean masking of cryptographic computations. IEEE Transactions on Information Theory 58(9), 6000–6011 (2012)
6. Carlet, C., Guilley, S.: Side-channel indistinguishability. In: Proceedings of HASP 2013, 2nd International Workshop on Hardware and Architectural Support for Security and Privacy, Tel Aviv, Israel, pp. 9:1-9:8. ACM, New York (June 2013)
7. Carlet, C., Guillot, P., Mesnager, S.: On immunity profile of Boolean functions. In: Gong, G., Helleseth, T., Song, H.-Y., Yang, K. (eds.) SETA 2006. LNCS, vol. 4086, pp. 364–375. Springer, Heidelberg (2006)
8. Courtois, N.T.: Fast Algebraic Attacks on Stream Ciphers with Linear Feedback. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 176–194. Springer, Heidelberg (2003)
9. Courtois, N., Meier, W.: Algebraic Attacks on Stream Ciphers with Linear Feedback. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 346–359. Springer, Heidelberg (2003)

10. Maghrebi, H., Carlet, C., Guilley, S., Danger, J.-L.: Optimal first-order masking with linear and non-linear bijections. In: Mitrokotsa, A., Vaudenay, S. (eds.) AFRICACRYPT 2012. LNCS, vol. 7374, pp. 360–377. Springer, Heidelberg (2012)
11. Maghrebi, H., Guilley, S., Danger, J.-L.: Leakage Squeezing Countermeasure against High-Order Attacks. In: Ardagna, C.A., Zhou, J. (eds.) WISTP 2011. LNCS, vol. 6633, pp. 208–223. Springer, Heidelberg (2011)
12. Massey, J.L.: Shift-register analysis and BCH decoding. IEEE Transactions on Information Theory 15, 122–127 (1969)
13. Meier, W., Staffelbach, O.: Fast correlation attacks on stream ciphers. In: Günther, C.G. (ed.) EUROCRYPT 1988. LNCS, vol. 330, pp. 301–314. Springer, Heidelberg (1988)
14. Rønjom, S., Helleseth, T.: A new attack on the filter generator. IEEE Transactions on Information Theory 53(5), 1752–1758 (2007)
15. Siegenthaler, T.: Decrypting a Class of Stream Ciphers Using Ciphertext Only. IEEE Transactions on Computer C-34(1), 81–85 (1985)