

Non-Malleable Codes from the Wire-Tap Channel*

Hervé Chabanne ^{†‡} Gérard Cohen ^{†§} Jean-Pierre Flori ^{†§}
 Alain Patey ^{†‡§}

November 14, 2018

Abstract

Recently, Dziembowski *et al.* introduced the notion of *non-malleable codes* (NMC), inspired from the notion of non-malleability in cryptography and the work of Gennaro *et al.* in 2004 on tamper proof security. Informally, when using NMC, if an attacker modifies a codeword, decoding this modified codeword will return either the original message or a completely unrelated value.

The definition of NMC is related to a family of modifications authorized to the attacker. In their paper, Dziembowski *et al.* propose a construction valid for the family of all bit-wise independent functions.

In this article, we study the link between the second version of the Wire-Tap (WT) Channel, introduced by Ozarow and Wyner in 1984, and NMC. Using coset-coding, we describe a new construction for NMC w.r.t. a subset of the family of bit-wise independent functions. Our scheme is easier to build and more efficient than the one proposed by Dziembowski *et al.*

1 Introduction

In cryptography, the non-malleability property [1] requires that it is impossible, given a ciphertext, to produce another different ciphertext so that the corresponding plaintexts are related to each other. Non-malleability under adaptive chosen-ciphertext attack (NM-CCA2) is one of the strongest computational security property that is required from an asymmetric encryption scheme (it is

*This work has been partially funded by the ANR SPACES project.

[†]Identity & Security Alliance (The Morpho and Télécom ParisTech Research Center)
 Télécom ParisTech – 46, rue Barrault - 75013 Paris - France – Email: {chabanne, cohen, flori, patey}@telecom-paristech.fr

[‡]Morpho – 11, boulevard Gallieni - 92130 Issy-Les-Moulineaux - France – Email: {herve.chabanne, alain.patey}@morpho.com

[§]CNRS-LTCI

equivalent to indistinguishability under adaptive chosen-ciphertext attack (IND-CCA2)).

Recently, Dziembowski *et al.* [2] proposed a transposition of the cryptographic definition of non-malleability to the field of coding theory. Informally, they define a NMC as a code such that, when a codeword is subject to modifications, its decoding procedure either corrects these errors and decodes to the original message or returns a value that is completely unrelated to the original message.

The property of non-malleability, as defined in [2], is subject to a choice of a family of modifications that we allow an adversary to make on the codewords. Dziembowski *et al.* also proved that it is impossible for a code to be non-malleable w.r.t. the set of all possible modifications of codewords.

The motivation for NMC is tamperproofness. The authors of [2] were indeed much influenced by the work of Gennaro *et al.* [3]. Non-malleability can be useful in real-life applications. Some storage devices may be assumed to be “read-proof” because of a sufficient amount of physical or algorithmic protections to prevent anyone from learning the data stored on them. However, even if one cannot read the data, injecting faults in the data and observing the way it affects functions using these data can help to recover them. Injecting faults can be done for instance using lasers [4]. There exists an important literature on how to use Differential Fault Analysis to break cryptosystems (e.g. [5, 6]).

Dziembowski *et al.* studied deeply the non-malleability w.r.t. bit-wise independent tampering functions, *i.e.* modifications that affect each bit of the codeword independently: flipping the bit or setting it to 0 or 1. This is typically what can be done using fault injections and, consequently, focusing on this family of tampering functions is worthwhile.

In [2], a construction for NMC w.r.t. all bit-wise independent functions is proposed. However, an implementable construction is left as an open problem. Our goal is to propose NMC that can be explicitly built. To this end, we exploit a relation that can be established between the model for NMC and the second version of the Wire-Tap channel [7]. This allows us to prove how coset-coding can be used to build a NMC. Furthermore, the decoding procedure of linear-coset coding consists uniquely of one matrix-vector product. Our construction is thus computationally efficient. Moreover, unlike their solution, our procedure always decodes messages whereas theirs is closer to error detection and often returns an error symbol.

Organization of the Paper

In Section 2, we explain and give the formal definitions for NMC as established in [2]. We describe the model of the WT channel in Section 3 and explain the use of coset-coding. We show how the second version of the WT channel and NMC w.r.t. bit-wise independent functions are related and prove why coset-coding can be used as a NMC in Section 4. We finally conclude in Section 5.

2 Non-Malleable Codes

In this section, we intend to give an easy-to-understand description of NMC and their goals. All definitions come from [2].

In the following, we consider a randomized encoding function $\text{Enc} : \{0, 1\}^k \mapsto \{0, 1\}^n$, which is associated to a deterministic decoding function $\text{Dec} : \{0, 1\}^n \mapsto \{0, 1\}^k \cup \{\perp\}$, where \perp means that the codeword cannot be decoded. Let \mathbb{F}_2 denote the field with two elements.

2.1 The Tampering Experiment

Let us first introduce the situation considered in NMC. In this model, a source message m is encoded using Enc , in order to be later decoded using Dec . The codeword $c = \text{Enc}(m)$ is stored on a device or sent over a channel before being decoded. During this phase, an attacker applies some tampering function f belonging to a given family of functions $\mathcal{F} \subset \mathbb{F}_2^{n \times n}$. A tampered codeword $\tilde{c} = f(c)$ is thus obtained. This erroneous codeword is then decoded to $\tilde{m} = \text{Dec}(\tilde{c})$. This process is described in Figure 1.

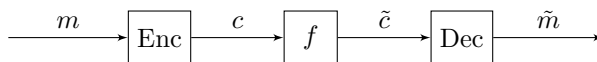


Figure 1: The Tampering Experiment

Now focus on the behaviour of the attacker, called Eve in the following. Eve applies a function $f \in \mathcal{F}$ to the codeword c , but she does not read c . In the real world, this can be seen as injecting faults on a device that you cannot read (e.g. a smart-card) using, for instance, a laser. In this experiment, Eve can however read the resulting decoded message \tilde{m} and try to learn as much as possible about m from \tilde{m} . Let us also specify that f is a deterministic function and, furthermore, that Eve knows which function she has chosen in \mathcal{F} .

2.2 Defining Non-Malleability

Let us now give the formal definition of non-malleability. Let \mathcal{F} be a family of tampering functions. For each $f \in \mathcal{F}$, we define a random variable Tamper_s^f corresponding to the tampering experiment described in the previous section:

$$\text{Tamper}_s^f = \left\{ \begin{array}{l} c \leftarrow_R \text{Enc}(s), \tilde{c} = f(c), \tilde{s} = \text{Dec}(\tilde{c}) \\ \text{Output} : \tilde{s} \end{array} \right\}$$

The randomness is induced by the encoding function Enc .

The *Non-Malleability* property is defined as follows:

Definition 1 (Non-Malleability). *Let (Enc, Dec) be a coding scheme, where $\text{Enc} : \{0, 1\}^k \mapsto \{0, 1\}^n$ is random and $\text{Dec} : \{0, 1\}^n \mapsto \{0, 1\}^k \cup \{\perp\}$ deterministic. Let $\mathcal{F} \subset \mathbb{F}_2^{n \times n}$ be a family of tampering functions.*

We say that the coding scheme (Enc, Dec) is non-malleable w.r.t. \mathcal{F} if for each $f \in \mathcal{F}$, there exists a distribution \mathcal{D}_f over $\{0, 1\}^k \cup \{\perp, \mathbf{same}\}$ such that, $\forall s \in \{0, 1\}^k$, we have:

$$\text{Tamper}_s^f \approx \left\{ \begin{array}{l} \bar{s} \leftarrow \mathcal{D}_f \\ \text{Output} \left\{ \begin{array}{l} s \text{ if } \bar{s} = \mathbf{same} \\ \bar{s} \text{ otherwise} \end{array} \right\} \end{array} \right\} \quad (1)$$

where \approx denotes computational or statistical indistinguishability.

2.3 Explaining the Definition

First, notice that the definition is relative to a family \mathcal{F} of tampering functions, but the property of indistinguishability concerns each function f separately. Non-malleability w.r.t. a family is in fact non-malleability w.r.t. each function in this family.

Now let us recall what we expect from a NMC. We want that, after the tampering experiment, either the codeword \tilde{c} is well-decoded to the original message s despite the tampering or the decoding procedure results in a value \tilde{s} that is unrelated to the original message. That is the idea behind the distribution \mathcal{D}_f : either it returns the symbol **same**, meaning that the decoding furnishes the original value or it returns a value $\tilde{s} \in \{0, 1\}^k \cup \{\perp\}$. As \mathcal{D}_f depends only on f and not on the message s , in the latter case, the value returned in the second part of Equation (1) is unrelated to s .

2.4 Basic Examples

We summarize here two examples developed in [2] that correspond to usual families of codes encompassed by the definition of NMC.

Error Correction

Let us assume that \mathcal{F} is a family of tampering functions and C an error-correcting code such that errors introduced by the application of a function $f \in \mathcal{F}$ on any codeword of C can be corrected. Then C is non-malleable w.r.t. \mathcal{F} . The distribution associated to every function $f \in \mathcal{F}$ is the constant distribution $\mathcal{D}_f = \mathbf{same}$, since erroneous codewords are always well-decoded.

Error Detection

The same idea can be applied to error-detecting codes. If there is a family \mathcal{F} of tampering functions such that each $f \in \mathcal{F}$ introduces errors in every codeword that are detected by a code C , then C is non-malleable w.r.t. \mathcal{F} . The distribution associated to every function $f \in \mathcal{F}$ is the constant distribution $\mathcal{D}_f = \perp$.

2.5 General (Im)Possibility Results

Impossibility

As proven in [2], no code is non-malleable w.r.t. the set of all possible tampering functions (*i.e.* $\mathcal{F} = \mathbb{F}_2^{n \times \mathbb{F}_2^n}$). Indeed there is, for instance, in \mathcal{F} a function that decodes the codeword, “increments” the message (*i.e.* adds 1 to its representation in \mathbb{F}_2^k) and re-encodes it. The result of the decoding of such a tampered codeword would always be $s + 1$ and thus would be neither the original message s nor an unrelated value.

Possibility

In [2], the authors prove that for any bounded-sized family of tampering functions, there exists a NMC. Their result is summed up in the following theorem:

Theorem 1 ([2]). *Let $\mathcal{F} \subset \mathbb{F}_2^{n \times \mathbb{F}_2^n}$ be a family of tampering functions such that $n > \log(\log(|\mathcal{F}|))$. Then there exists a non-malleable code w.r.t. \mathcal{F} .*

2.6 Bit-wise Independent Tampering

Bit-wise independent tampering is a special case of tampering where each bit of the codeword is tampered with independently. Formally a function $f : \{0, 1\}^n \mapsto \{0, 1\}^n$ is bit-wise independent if we can find n independent functions $f_1, \dots, f_n : \{0, 1\} \mapsto \{0, 1\}$ such that $\forall x \in \{0, 1\}^n, f(x) = (f_1(x), \dots, f_n(x))$. There are four possibilities for each f_i which we denote by **keep**, **flip**, **0** and **1** (**keep** and **flip** are explicit, **0** (resp. **1**) is the function that sets a bit to 0 (resp. 1) regardless of what it was before).

In [2], a construction for a NMC w.r.t. the family of all bit-wise independent functions is introduced. It uses Linear Error-Correcting Secret-Sharing (LECSS) schemes [8] and Algebraic Manipulation Detection (AMD) codes [9]. Both are quite new tools and even the authors of [2] leave the explicit construction of LECSS codes as an “interesting open problem”. Furthermore, their solution is quite close to error detecting codes as it decodes to \perp after a tampering in most cases¹.

In Section 4, we propose a new way to build NMC w.r.t. bit-wise independent functions. Our solution covers less tampering functions but uses more standard and efficient tools. Moreover, our scheme is neither error-correcting nor error-detecting (it never returns \perp) and so, to our opinion, is closer to the original definition of non-malleability, which is more generic than error detection or correction.

¹In their proof of non-malleability, the authors of [2] distinguish different cases depending on the considered tampering function (more precisely its number q of **0** and **1** sub-functions) and the *secrecy* t of the LECSS scheme. When $t < q < n - t$, the tampering experiment always returns \perp and when $q \leq t$, the scheme is likely to often return \perp .

3 The Wire-Tap Channel

In the following, a $[n, k, d]$ linear code denotes a subspace of dimension k of \mathbb{F}_2^n with minimal Hamming distance d .

3.1 Linear Coset Coding

Coset coding is a random encoding used for both models of WT Channel. This type of encoding uses a $[n, k, d]$ linear code C with a parity-check matrix H . Let $r = n - k$. To encode a message $m \in \mathbb{F}_2^r$, one chooses randomly an element among all $x \in \mathbb{F}_2^n$ such that $m = H^t x$. To decode a codeword x , one just applies the parity-check matrix H and obtains the syndrome of x for the code C , which is the message m . This procedure is summed up in Figure 2.

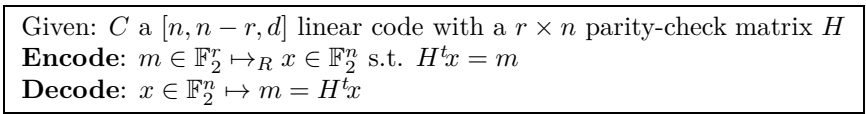


Figure 2: Linear Coset-coding

3.2 The Wire-Tap Channel I

The Wire-Tap Channel was introduced by Wyner [10]. In this model, a sender Alice sends messages over a potentially noisy channel to a receiver Bob. An adversary Eve listens to an auxiliary channel, the WT channel, which is a noisier version of the main channel. It was shown that, with an appropriate coding scheme, the secret message can be conveyed in such a way that Bob has complete knowledge of the secret and Eve does not learn anything. In the special case where the main channel is noiseless, the secrecy capacity can be achieved through a linear coset coding scheme. We summarize the WT Channel I in Figure 3.

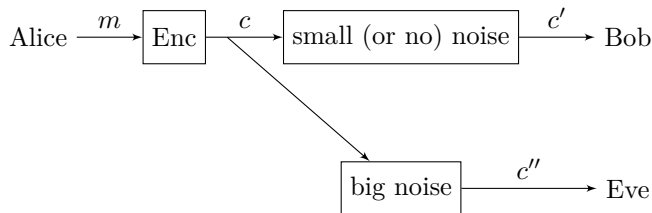


Figure 3: The Wire-Tap Channel I

3.3 The Wire-Tap Channel II

Ten years later, Ozarow and Wyner introduced a second version of the WT Channel [7]. In this model, both main and WT channels are noiseless. This time, the disadvantage for Eve is that she can only see messages with erasures: she has only access to a limited number of bits per codeword. She is however allowed to choose which bits she can learn. We summarize the Wire-Tap Channel II in Figure 4.

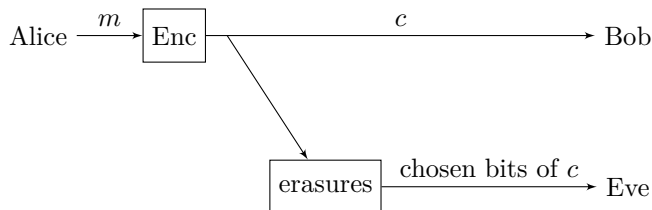


Figure 4: The Wire-Tap Channel II

The encoding used in this model is again a coset coding based on a linear code C , as in the Wire Tap Channel I with a noiseless main channel. Let d^\perp denote the minimal distance of the dual C^\perp of C . One can prove (see [11] for instance) that, if Eve can access less than d^\perp bits of a codeword, then she gains no information at all on the associated message.

Linear coset-coding for the WT channel can be efficiently implemented using LDPC codes [12, 13].

4 From the Wire-Tap Channel to Non-Malleable Codes

For our construction, we only deal with tampering functions that are bit-wise independent.

4.1 Motivations for Using Wire-Tap

Roughly speaking, in both models, codewords are modified either with random faults (WT I), adversary-controlled erasures (WT II) or an adversary-controlled tampering function (NMC). From these modified codewords or their decoding results, the adversary tries to learn information on the original messages.

The first WT is a little different from the other models because errors are random and so do not occur in the same number and bit positions every time. It could however be covered by the definition of NMC if every possible tampering caused by these random errors were included in the family of tampering functions taken into account by the code.

Let us now assume that we want to use a linear coset-coding scheme with a parity-check matrix H as NMC. We cannot be protected against tampering functions that only add errors (*i.e.* bit-wise independent functions where the only choices for each bit are **keep** or **flip**). To see why, let \mathcal{F} be a family of such functions. Obviously, for each $f \in \mathcal{F}$, there is an error vector $e \in \mathbb{F}_2^n$ such that $\forall c \in \mathbb{F}_2^n, f(c) = c + e$. Let us follow the tampering experiment. Let $m \in \mathbb{F}_2^r$ be a source message and c an encoding of m . Say c is tampered to $\tilde{c} = c + e$. Decoding results in $\tilde{m} = H^t c + H^t e = m + H^t e$. Thus, \tilde{m} is always m plus a constant offset. It is consequently related to m . Linear coset-coding cannot be non-malleable w.r.t. these “error-only” functions. There must be some $\mathbf{0}$ and $\mathbf{1}$ in the tampering.

This is why we consider WT II. Indeed, using $\mathbf{0}$ and $\mathbf{1}$ on some bits of the codewords is, in an information-theoretic sense, like having erasures at the corresponding locations, as we do not know what was originally there. As WT II guarantees that no information is leaked from erased codewords encoded using an appropriate coset-coding scheme, there will be no relation between the decoded tampered codeword and the original message. That is what motivates our proposal.

4.2 The Construction

As discussed before, we consider bit-wise independent functions where the sub-functions are not only **keep** or **flip**. Nevertheless, we authorize bit-flips because if the result of the tampering experiment is unrelated to the original message, then the result added to a constant offset will also be unrelated to this message.

We state the following theorem:

Theorem 2 (Linear coset-coding as NMC). *Let $\mathcal{F} \subset \mathbb{F}_2^{n \times \mathbb{F}_2^n}$ be a family of bit-wise independent tampering functions such that:*

$$\forall f = (f_1, \dots, f_n) \in \mathcal{F}, |\{i | f_i = \mathbf{0} \text{ or } f_i = \mathbf{1}\}| \geq D.$$

Let C be a $[n, k, d]$ -linear code such that $D > n - d^\perp$, where d^\perp is the minimal distance of its dual code C^\perp .

Then a linear coset-coding using C is non-malleable w.r.t. \mathcal{F} .

4.3 Proof of Non-Malleability

Our proof of non-malleability is inspired from the proof of security of the WT II in [14].

Let us consider we are in the situation of Theorem 2. Let $f = (f_1, \dots, f_n) \in \mathcal{F}$ be a tampering function. Let $S_{\mathbf{01}}$ be the set of all positions i such that $f_i = \mathbf{0}$ or $f_i = \mathbf{1}$. Let S_{keep} and S_{flip} be the equivalent sets for **keep** and **flip**. Let $e \in \mathbb{F}_2^n$ be such that $\forall i = 1, \dots, n, e_i = \chi_{S_{\text{flip}}}(i)$ (where χ_A denotes the indicator function of a set A) and $\epsilon \in \mathbb{F}_2^n$ be such that $\epsilon_i = 1$ if $f_i = \mathbf{1}$ and $\epsilon_i = 0$ otherwise. Let h_1, \dots, h_n denote the columns of the parity-check matrix H .

Let $m \in \mathbb{F}_2^r$ be a message encoded to $c \in \mathbb{F}_2^n$. Let $\tilde{c} = f(c)$ and $\tilde{m} = H\tilde{c}$. We have

$$\begin{aligned}
\tilde{m} &= \sum_{i \in S_{01}} h_i \tilde{c}_i + \sum_{i \in S_{\text{keep}}} h_i \tilde{c}_i + \sum_{i \in S_{\text{flip}}} h_i \tilde{c}_i \\
&= \sum_{i \in S_{01}} h_i \epsilon_i + \sum_{i \in S_{\text{keep}}} h_i c_i + \sum_{i \in S_{\text{flip}}} h_i (c_i + e_i) \\
&= H^t \epsilon + H^t e + \sum_{i \in S_{\text{keep}} \cup S_{\text{flip}}} h_i c_i \\
& (= m + H^t \epsilon + H^t e - \sum_{i \in S_{01}} h_i c_i)
\end{aligned}$$

If we want \tilde{m} to be unrelated to m , then we want $\sum_{i \in S_{\text{keep}} \cup S_{\text{flip}}} h_i c_i$ to be unrelated to m . If the submatrix H_{kf} made of the columns $h_i, i \in S_{\text{keep}} \cup S_{\text{flip}}$ is of full rank $r = n - k$, then we gain no information on the corresponding bits of m , and all values are equiprobable. This is achieved in particular if $|S_{\text{keep}} \cup S_{\text{flip}}| < d^\perp$ (see chapter 9 of [14]).

If $D > n - d^\perp$, then $|S_{01}| > n - d^\perp$, *i.e.* $n - |S_{\text{keep}} \cup S_{\text{flip}}| > n - d^\perp$ or $|S_{\text{keep}} \cup S_{\text{flip}}| < d^\perp$. The condition of the previous paragraph is thus achieved if we use the parameters of Theorem 2.

Let us define more formally the distribution D_f associated to f . Let $K_i, i \in S_{\text{keep}} \cup S_{\text{flip}}$ be Bernoulli(1/2) distributions. Then $D_f = H^t \epsilon + H^t e + \sum_{i \in S_{\text{keep}} \cup S_{\text{flip}}} h_i K_i$. This distribution and the result of the tampering experiment are identically distributed.

The coset-coding scheme used in Theorem 2 is consequently non-malleable w.r.t. \mathcal{F} . □

4.4 Going Further

Towards a Larger Family of Tampering Functions

When comparing our construction to the one of [2], one can relate the LECSS and our coset-coding scheme. The only requirement that is not fulfilled by linear coset-coding is a large distance. As the distance of linear coset-coding is 1, we cannot assume $d > n/4$ as they do. That is why we cannot directly modify this construction and replace LECSS with coset-coding in the description of the code and the proof of non-malleability.

Both LECSS and coset-coding ensure non-malleability when the number of **0** or **1** sub-functions of the tampering function is high enough. To deal with the case where the number of such functions is low, Dziembowski *et al.* concatenated the LECSS with an AMD code. In such a case, the tampering function acts by adding an error following a fixed distribution (*i.e.* independent of the codeword) and the decoding procedure results in \perp with high probability because of the AMD code. Therefore, non-malleability is ensured. Following this idea, it might

also be possible to encapsulate our coset-coding scheme within an error-detecting or an error-correcting code. Thus we would achieve non-malleability w.r.t. a larger family of functions. In particular, functions with a small number of **0** or **1** sub-functions which cannot be dealt with by coset-coding alone could be included. For the error-detecting case, using an AMD code as in [2] seems to be feasible. However, for the error-correcting case, it is not clear which kind of correction strategy to use to deal with the effects of the linear coset-coding scheme. Nevertheless, if such functions are the only ones of interest, one must be aware that an error correcting or an error detecting code is sufficient by itself.

Relaxing the Notion of Non-Malleability

In the model for the WT II described in this paper, we require that Eve cannot obtain any bit of information on the messages sent over the channel. This strong security notion can be relaxed. Indeed, one could be satisfied even if Eve learned only a bounded amount of bits. This is possible if we consider generalized Hamming distances [11] instead of the dual distance d^\perp of the code considered in the linear coset-coding scheme. For $i \in \mathbb{N}$, the generalized distance d_i is such that if Eve cannot obtain more than d_i bits per message, then she gains no more than $i - 1$ bits of information per message. For instance, $d_1 = d^\perp$.

In the same spirit, one could relax the notion of non-malleability. After the tampering experiment, we could state that either the decoding procedure returns the original message or it enables to learn a bounded number of bits of information on this message. Using our construction, it is easy to build another scheme that would satisfy this requirement. One would only have to replace dual distances by generalized distances.

5 Conclusion

We established in this paper a parallel between Non-Malleable Codes and the Wire-Tap Channel. This relation enabled us to build an efficient non-malleable scheme, w.r.t. a family of bit-wise independent functions, that is neither error-correcting nor error-detecting.

Considering bit-wise independent tampering is a worthwhile first step for NMC. An interesting open problem would be now to build schemes that are non-malleable w.r.t. larger families of functions.

Acknowledgement

The authors would like to thank Julien Bringer for his helpful comments.

References

- [1] D. Dolev, C. Dwork, and M. Naor, “Non-malleable cryptography (extended abstract),” in *STOC*. ACM, 1991, pp. 542–552.
- [2] S. Dziembowski, K. Pietrzak, and D. Wichs, “Non-malleable codes,” in *ICS*, A. C.-C. Yao, Ed. Tsinghua University Press, 2010, pp. 434–452.
- [3] R. Gennaro, A. Lysyanskaya, T. Malkin, S. Micali, and T. Rabin, “Algorithmic tamper-proof (ATP) security: Theoretical foundations for security against hardware tampering,” in *TCC*, ser. Lecture Notes in Computer Science, M. Naor, Ed., vol. 2951. Springer, 2004, pp. 258–277.
- [4] S. P. Skorobogatov and R. J. Anderson, “Optical fault induction attacks,” in *CHES*, ser. Lecture Notes in Computer Science, B. S. K. Jr., Çetin Kaya Koç, and C. Paar, Eds., vol. 2523. Springer, 2002, pp. 2–12.
- [5] D. Boneh, R. A. DeMillo, and R. J. Lipton, “On the importance of checking cryptographic protocols for faults (extended abstract),” in *EUROCRYPT*, 1997, pp. 37–51.
- [6] H. Bar-El, H. Choukri, D. Naccache, M. Tunstall, and C. Whelan, “The sorcerer’s apprentice guide to fault attacks,” in *Workshop on Fault Diagnosis and Tolerance in Cryptography in association with DSN 2004 - The International Conference on Dependable Systems and Networks*, 2004, pp. 330–342.
- [7] L. H. Ozarow and A. D. Wyner, “Wire-tap channel II,” in *EUROCRYPT*, 1984, pp. 33–50.
- [8] H. Chen, R. Cramer, S. Goldwasser, R. de Haan, and V. Vaikuntanathan, “Secure computation from random error correcting codes,” in *EUROCRYPT*, ser. Lecture Notes in Computer Science, M. Naor, Ed., vol. 4515. Springer, 2007, pp. 291–310.
- [9] R. Cramer, Y. Dodis, S. Fehr, C. Padró, and D. Wichs, “Detection of algebraic manipulation with applications to robust secret sharing and fuzzy extractors,” in *EUROCRYPT*, ser. Lecture Notes in Computer Science, N. P. Smart, Ed., vol. 4965. Springer, 2008, pp. 471–488.
- [10] A. D. Wyner, “The wire-tap channel,” *The Bell System Technical Journal*, vol. 54, no. 8, pp. 1355–1387, October 1975.
- [11] V. K.-W. Wei, “Generalized hamming weights for linear codes,” *IEEE Transactions on Information Theory*, vol. 37, no. 5, pp. 1412–, 1991.
- [12] A. Thangaraj, S. Dihidar, A. R. Calderbank, S. W. McLaughlin, and J.-M. Merolla, “Applications of LDPC codes to the wiretap channel,” *IEEE Transactions on Information Theory*, vol. 53, no. 8, pp. 2933–2945, 2007.

- [13] A. Subramanian, A. Thangaraj, M. Bloch, and S. W. McLaughlin, “Strong secrecy on the binary erasure wiretap channel using large-girth LDPC codes,” *CoRR*, vol. abs/1009.3130, 2010.
- [14] G. Zémor, *Cours de cryptographie*. Cassini, 2000.