A Generic Method to Design Modes of Operation Beyond the Birthday Bound*

David Lefranc¹, Philippe Painchaud¹, Valérie Rouat², and Emmanuel Mayer²

¹ Cryptology Laboratory
Thales

160 Boulevard de Valmy – BP 82
92704 Colombes Cedex – France

`firstname.lastname@fr.thalesgroup.com`

² DGA / CELAR
BP 57419

35174 Bruz Cedex – France

`firstname.lastname@dga.defense.gouv.fr`

Abstract. Given a PRP defined over $\{0, 1\}^n$, we describe a new generic and efficient method to obtain modes of operation with a security level beyond the birthday bound $2^{n/2}$. These new modes, named NEMO (for New Encryption Modes of Operation), are based on a new contribution to the problem of transforming a PRP into a PRF. According to our approach, any generator matrix of a linear code of minimal distance d , $d \geq 1$, can be used to design a PRF with a security of order $2^{dn/(d+1)}$. Such PRFs can be used to obtain NEMO, the security level of which is of the same order ($2^{dn/(d+1)}$). In particular, the well-known counter mode becomes a particular case when considering the identity linear code (of minimal distance $d = 1$) and the mode of operation CENC [7] corresponds to the case of the the parity check linear code of minimal distance $d = 2$. Any other generator matrix leads to a new PRF and a new mode of operation. We give an illustrative example using $d = 4$ which reaches the security level $2^{4n/5}$ with a computation overhead less than 4% in comparison to the counter mode.

Keywords: symmetric encryption, modes of operation, PRP, PRF, birthday bound, counter mode, CENC.

1 Introduction

An encryption mode of operation is an algorithm which uses a pseudo-random permutation (PRP) defined over $\{0, 1\}^n$ to encrypt a message of size tn bits into a string of size tn bits. Several modes of operation exist such as electronic code book (ECB), chaining block cipher (CBC), counter (CTR). The latter is one of the most interesting since it presents both efficiency and security.

Using the framework of [4] for concrete security, Bellare *et al.* [2] proved the two following properties.

* Patent pending.

- The CTR mode used with a PRP defined over $\{0, 1\}^n$ cannot be used to encrypt more than $2^{n/2}$ blocks; this bound is generally called the *birthday bound*.
- The CTR mode used with a PRF is as secure as the PRF itself.

The birthday bound concerns almost all modes of operation when using a PRP as primitive. But, reaching a security level beyond such a bound can be easily obtained using a pseudo-random function (PRF) instead of a PRP. However, such an approach is not widespread. Some reasons can explain this fact: on one hand, block ciphers (PRP) have been studied and cryptanalyzed for several years so that they are implemented everywhere; on the other hand, designing a secure and efficient PRF from scratch is not so easy.

An alternative to this lack of consideration for PRFs consists in constructing a PRF from a given PRP. Such a problem has already been extensively analyzed. For example, in 1998, Bellare *et al.* [5] suggested the *re-keying construction* an illustrative special case of which the PRF F is defined from the PRP E by $F(K, x) = E(E(K, x), x)$. But, this solution significantly increases the number of calls to the PRP.

In 1998, Hall *et al.* [6] suggested the *truncate construction*. It truncates the output of the given PRP, but it does not preserve the security of the latter.

In 2000, Lucks [9] suggested the construction

$$Twin^d(K, x) = E(K, dx) \oplus E(K, dx + 1) \dots \oplus E(K, dx + d - 1)$$

for all $d \geq 1$ (the case $d = 2$ has also been independently analyzed in [3]). The security of $Twin^d$ depends on d : the larger d , the more secure the PRF. According to a targeted level of security, an adequate value d can be chosen. However, the computation overhead is also highly dependant on d .

Finally, in 2006, Iwata suggested the mode of operation CENC [7]. To our knowledge, it is the only mode (with a full¹ security proof) that is beyond the birthday bound. CENC is also based on a PRF built from a PRP. The main advantage of this PRF is that it outputs a string of several blocks of n bits (not only one n -bit block as $Twin^d$). However, the level of security can not be adjusted.

In this paper, we add a new contribution to the problem of constructing a PRF from a PRP. Our solution is the convergence of $Twin^d$ and CENC without their drawbacks. We propose a generic method to construct efficient PRFs with several n -bit output blocks (as the one involved in CENC) and with an adjustable security level (as $Twin^d$) of order $2^{dn/(d+1)}$ depending on a parameter d . Our approach is based on linear code theory. More precisely, it relies on the generator matrix associated to a linear code of minimal distance d . Our solution is both a theoretical generalisation and a practical method to obtain secure and efficient PRFs from PRPs.

¹ Two modes of operation beyond the birthday bound have been suggested, but one of them was proved in a weak security model [1] and the other one has no security proof [8].

With such a generalization, the PRF involved in CENC becomes a particular case of our method when used with the parity check code ($d = 2$). And, considering any linear code of minimal distance $d \geq 3$ leads to a new PRF with a level of security of order (at least) $2^{3n/4}$ which is beyond the security of Iwata's PRF.

The organisation of this paper is the following. In section 2, we recall security notions and we describe more precisely *Twin*^d and the PRF of CENC. In section 3, we describe our generic method to obtain new PRFs with a security level of order $2^{dn/(d+1)}$. In particular, we show that the PRF of CENC becomes a particular case of our construction, when considering the parity check code (of minimal distance $d = 2$). In section 4, we describe NEMO, our New Encryption Modes of Operation which preserve the security of our PRFs. Finally, in section 5, we present a direct application of our method to obtain a PRF with a security level of order $2^{4n/5}$. This PRF can be used to obtain a mode of operation with a security level of order $2^{4n/5}$ with a computation overhead around 4% (in comparison to the CTR mode).

2 Preliminaries

2.1 PRFs and PRPs Security

We denote by $Rand(m, n)$ the set of all functions $F : \{0, 1\}^m \rightarrow \{0, 1\}^n$ and we denote by $Perm(n)$ the set of all permutations defined over $\{0, 1\}^n$.

Let $E : \{0, 1\}^k \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ be a block cipher. For each key $K \in \{0, 1\}^k$, we denote by E_K the bijection defined by $E_K(x) = E(K, x)$. A block cipher E determines the family

$$\mathcal{F}(E) = \{E_K, K \in \{0, 1\}^k\}.$$

Let $\mathcal{D}$ be an algorithm, called a distinguisher, having access to an oracle parametrized by a bit b . According to b , the oracle simulates a function randomly chosen in $\mathcal{F}(E)$ or in $Rand(n, n)$. We denote by $\mathcal{D}(t, q)$ an algorithm $\mathcal{D}$ making q queries to the oracle and with a running time bounded by t .

The adversarial (distinguisher) advantage $\text{Adv}_E^{\text{prf}}(t, q)$ in distinguishing the block cipher from a truly random function is a good estimate for the quality of a block cipher. It is defined by

$$\text{Adv}_E^{\text{prf}}(t, q) = \max_{\mathcal{D}(t, q)} \left\{ \Pr[\mathcal{D} = 1 \mid b = 1] - \Pr[\mathcal{D} = 1 \mid b = 0] \right\}.$$

In the same way, we now assume that the oracle simulates a function randomly chosen in $\mathcal{F}(E)$ or in $Perm(n)$. The adversarial (distinguisher) advantage $\text{Adv}_E^{\text{prp}}(t, q)$ in distinguishing the block cipher from a truly random permutation is a good estimate for the quality of a block cipher. It is defined by

$$\text{Adv}_E^{\text{prp}}(t, q) = \max_{\mathcal{D}(t, q)} \left\{ \Pr[\mathcal{D} = 1 \mid b = 1] - \Pr[\mathcal{D} = 1 \mid b = 0] \right\}.$$

2.2 Security Analysis of Modes of Operation

To analyze the security of a mode of operation used with a block cipher E , we consider the real or random indistinguishably notion [2] against a chosen plaintext attack (cpa). More precisely, let $\mathcal{A}$ be an adversary having access to an oracle parametrized by a bit b . According to b , the oracle encrypts the requested plaintext or a random string of the same size. We denote by $\mathcal{A}(t, q)$ an adversary making q requests to the oracle and $\mathcal{A}$ with a running time bounded by t .

The security of the mode of operation $\text{mode}[E]$ in the real or random model against a chosen plaintext attack is denoted by $\text{Adv}_{\text{mode}[E]}^{\text{ror-cpa}}(t, q)$ and is defined by

$$\text{Adv}_{\text{mode}[E]}^{\text{ror-cpa}}(t, q) = \max_{\mathcal{A}(t, q)} \left\{ \Pr[\mathcal{A} = 1 \mid b = 1] - \Pr[\mathcal{A} = 1 \mid b = 0] \right\}.$$

2.3 The Twind^d Construction

In [9], Lucks analyzes the security of the PRF Twind^d . Let $P \in \text{Perm}(n)$, Twind^d is defined by

$$\begin{aligned} \text{Twind}^d : \{0, 1\}^{n - \lceil \log_2 d \rceil} &\longrightarrow \{0, 1\}^n \\ x &\longmapsto P(dx) \oplus P(dx + 1) \oplus \cdots \oplus P(dx + d - 1). \end{aligned}$$

The security of Twind^d is given by $\text{Adv}_{\text{Twind}^d}^{\text{prf}}(t, q) \leq \frac{qd^2}{2^n} + \frac{d^d}{2^{dn-1}} \sum_{0 \leq i < q} i^d$, for any $q, q \leq 2^{n-1}/d^2$.

2.4 The CENC Construction

CENC is a mode of operation presented by Iwata [7]. It is based on a PRF, denoted by F^+ which has two parameters: a permutation P of $\text{Perm}(n)$ and an integer u . The PRF F^+ is defined by

$$\begin{aligned} F^+ : \{0, 1\}^n &\longrightarrow (\{0, 1\}^n)^u \\ x &\longmapsto (P(x) \oplus P(x+1), P(x) \oplus P(x+2), \dots, P(x) \oplus P(x+u)). \end{aligned}$$

The security of F^+ is given by $\text{Adv}_{F^+}^{\text{prf}}(t, q) \leq \frac{(u+1)^4 q^3}{2^{2n+1}} + \frac{u(u+1)q}{2^{n+1}}$ assuming all the q requests x_i , are such that for all i, j , $1 \leq i < j \leq q$, the sets $\{x_i, x_i + 1, \dots, x_i + u\}$ and $\{x_j, x_j + 1, \dots, x_j + u\}$ are disjoint. Such a constraint does not matter since it exactly reflects the different calls to F^+ in CENC. Indeed, given a message of size kun bits, the algorithm CENC uses k calls to the PRF F^+ . The first nu bits are encrypted using the output of $F^+(x)$, the nu following bits are encrypted using the output of $F^+(x + u + 1)$ and so on until the nu last bits encrypted using the output of $F^+(x + (k - 1)(u + 1))$.

3 New PRFs Based on Linear Code

3.1 Description

Let P be a permutation in $\text{Perm}(n)$. Our new PRFs are parametrized by a generator matrix $G = (g_{i,j}) \in \mathcal{M}_{u \times \ell}(GF(2))$, associated to a linear code defined over $GF(2)$ of length ℓ , of dimension u and of minimal distance d so that G is of size $u \times \ell$. Define $\omega = 1 + \lfloor \log_2 \ell \rfloor$. For any given generator matrix G and any permutation P , we construct a new PRF $F : \{0, 1\}^{n-\omega} \rightarrow (\{0, 1\}^n)^u$, defined by

$$F(x) = \left(\bigoplus_{\substack{1 \leq j \leq \ell \\ g_{1,j} \neq 0}} P(\ell x + j - 1), \bigoplus_{\substack{1 \leq j \leq \ell \\ g_{2,j} \neq 0}} P(\ell x + j - 1), \dots, \bigoplus_{\substack{1 \leq j \leq \ell \\ g_{u,j} \neq 0}} P(\ell x + j - 1) \right).$$

As for Twins^d and the underlying PRF of CENC, when using this PRF for encryption, we will rather use a modification of this PRF to be able to use n -bit input strings instead of $(n - \omega)$ -bit input strings. Thus, in the following we will consider and prove the security of the PRF $F^+ : \{0, 1\}^n \rightarrow (\{0, 1\}^n)^u$ defined by

$$F^+(x) = \left(\bigoplus_{\substack{1 \leq j \leq \ell \\ g_{1,j} \neq 0}} P(x + j - 1), \bigoplus_{\substack{1 \leq j \leq \ell \\ g_{2,j} \neq 0}} P(x + j - 1), \dots, \bigoplus_{\substack{1 \leq j \leq \ell \\ g_{u,j} \neq 0}} P(x + j - 1) \right).$$

The security analysis will be the same as for F , since during the proof we assume that the q requests x_i , $1 \leq i \leq q$, are such that for all i, j , $1 \leq i < j \leq q$, the sets $\{x_i, x_i + 1, \dots, x_i + \ell - 1\}$ and $\{x_j, x_j + 1, \dots, x_j + \ell - 1\}$ are disjoint.

3.2 Example

Let us consider the matrix G of size $u \times \ell$ with $\ell = u + 1$ associated to the parity check code of minimal distance $d = 2$. The canonical form of G corresponds to the identity matrix $u \times u$ with a last additional column filled with “1”. An equivalent form of the matrix G is

$$G' = \begin{bmatrix} 1 & 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ & & & & \vdots & & & \\ 1 & 0 & 0 & 0 & \dots & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & \dots & 0 & 0 & 1 \end{bmatrix}.$$

According to our method, this matrix defines a PRF $F^+ : \{0, 1\}^n \rightarrow (\{0, 1\}^n)^u$ such that

$$F^+(x) = (P(x) \oplus P(x + 1), P(x) \oplus P(x + 2), \dots, P(x) \oplus P(x + u)).$$

Thus, to encrypt u blocks, it requires $u + 1$ calls to the permutation P .

This PRF is exactly the same as the one from CENC (see section 2.4). The security bound given by Iwata is

$$\text{Adv}_{F^+}^{\text{prf}}(t, q) \leq \frac{(u+1)^4 q^3}{2^{2n+1}} + \frac{u(u+1)q}{2^{n+1}}$$

in comparison with our bound (given in theorem 1) equal to

$$\text{Adv}_{F^+}^{\text{prf}}(t, q) \leq \frac{(u+1)^4 q^3}{2^{2n}} + \frac{(u+1)^2 q}{2^n}.$$

A second example consists in considering the generator matrix $u \times u$ of the identity code (of minimal distance $d = 1$). Our PRF F^+ just corresponds to the PRP, and has the same security (i.e. $\frac{q u^2}{2^n} + \frac{u^2 q^2}{2^n}$). Indeed, we obtain $F^+(x) = (P(x), P(x+1), \dots, P(x+u-1))$. Our security bound is of same order as the birthday bound (security of any permutation). Our bound is not optimal because of the method used in the security proof (however, the significant terms are almost the same).

3.3 Security Theorem

The security of our new PRFs is given in the following theorem.

Theorem 1. *Let $G = (g_{i,j}) \in \mathcal{M}_{u \times \ell}(GF(2))$ a generator matrix associated to a linear code defined over $GF(2)$, of length ℓ , of dimension u and of minimal distance d . Let P be a random permutation with an n -bit output. Let F^+ be our PRF parametrized with G and P . Let q be the number of requests x_i ($1 \leq i \leq q$) sent to the oracle. If $q \leq 2^{n-1}/\ell^2$, and if for all i, j , $1 \leq i < j \leq q$, $\{x_i, x_i+1, \dots, x_i+\ell-1\} \cap \{x_j, x_j+1, \dots, x_j+\ell-1\} = \emptyset$, then*

$$\text{Adv}_{F^+}^{\text{prf}}(t, q) \leq \frac{q \ell^2}{2^n} + \frac{N \ell^d q^{d+1}}{2^{dn}}$$

with $N = \sum_{k=0}^{u-1} \binom{d+k-1}{d-1}$.

Remark 1. The binomial coefficient $\binom{d+k-1}{d-1}$ involved in N can be bounded by $(d+k-1)^{d-1}$ so that

$$N = \sum_{k=0}^{u-1} \binom{d+k-1}{d-1} \leq \sum_{k=0}^{u-1} (d+k-1)^{d-1} \leq u(d+u-2)^{d-1} \leq \ell^d.$$

The last inequality relies on the Singleton bound recalled in definition 3. As a consequence,

$$\text{Adv}_{F^+}^{\text{prf}}(t, q) \leq \frac{q \ell^2}{2^n} + \frac{\ell^{2d} q^{d+1}}{2^{dn}}.$$

The proof of the theorem is given in appendix A.

4 NEMO: New Encryption Modes of Operation Beyond the Birthday Bound

4.1 Description

We describe how to use our new PRFs to obtain NEMO. The approach is the same as the one used in CENC and is a generalisation of the counter mode.

Let P be a n -bit permutation, G be a generator matrix of size $u \times \ell$ of a binary linear code (of minimal distance d), $F^+ : \{0, 1\}^n \rightarrow (\{0, 1\}^n)^u$ be our new PRF constructed from P and G , and M be a message of size m n -bit blocks denoted by $M_1, \dots, M_m$ ($m \geq 1$). Let α and r be such that $0 \leq \alpha$, $0 \leq r < u$ and $m = \alpha \times u + r$. To encrypt M , F^+ can be used to obtain a mode denoted $\text{NEMO}[F^+]$, as described in algorithm 1.

Algorithm 1. $\text{NEMO}[F^+] : \text{a mode of operation using our PRF } F^+$

Input: a message M of $\alpha \times u + r$ n -bit blocks denoted by M_j , $1 \leq j \leq \alpha \times u + r$.

Output: the encrypted message C of $\alpha \times u + r$ n -bit blocks associated to M .

Let x be an initial value.

for i **from** 0 **to** $\alpha - 1$ **do**

Compute $F^+(x + i \times \ell) = (S_1, \dots, S_u) \in (\{0, 1\}^n)^u$

for j **from** 1 **to** u **do**

$C_{i \times u + j} = M_{i \times u + j} \oplus S_j$

Compute $F^+(x + \alpha \times \ell) = (S_1, \dots, S_u) \in (\{0, 1\}^n)^u$

for j **from** 1 **to** r **do**

$C_{\alpha \times u + j} = M_{\alpha \times u + j} \oplus S_j$

Store $x + (\alpha + 1) \times \ell$ in place of x

Return $C_1, \dots, C_{\alpha \times u + r}$

4.2 Security of NEMO

We give the security level of NEMO using the framework recalled in section 2.2.

Theorem 2. *Let P be a n -bit random permutation. Let $G = (g_{i,j}) \in \mathcal{M}_{u \times \ell}(GF(2))$ a generator matrix associated to a linear code defined over $GF(2)$, of length ℓ , of dimension u and of minimal distance d . Let F^+ be the PRF parametrized with G . Let $\text{NEMO}[F^+]$ be the mode of operation described in algorithm 1. Then, we have*

$$Adv_{\text{NEMO}[F^+]}^{\text{ror-cpa}}(t, q) \leq \frac{(L/u + q)\ell^2}{2^n} + \frac{N \ell^d (L/u + q)^{d+1}}{2^{dn}}$$

with $N = \sum_{k=0}^{u-1} \binom{d+k-1}{d-1}$ and L is the overall number of n -bit blocks requested to the oracle.

Remark 2. The security level of the mode of operation relies on the term $\frac{N \ell^d (L/u + q)^{d+1}}{2^{dn}}$ which is of order $\mathcal{O}\left(\frac{q^{d+1}}{2^{dn}}\right)$. The security of NEMO is beyond the birthday bound for any $d \geq 2$.

The information rate of C is $247/256 \approx 0.965$. This means that the computation overhead in comparison to the counter mode is between 3% and 4%. In this construction, we need to compute and store 9 cipher blocks. The 247 next outputs will be the combination of one new cipher block with some of the first 9 cipher blocks.

6 Conclusion

In this paper we present a new contribution to the problem of transforming a PRP into a PRF. Our new construction allow to reach a security level beyond the birthday bound ($2^{n/2}$). It is based on a linear code with a minimal distance d , and its security level is of order $2^{dn/(d+1)}$. This work leads to New Encryption Modes of Operation, named NEMO, which generalize the CTR mode, and the CENC mode. Actually, the CTR mode can be built from a linear code, the minimal distance of which is 1, and the CENC mode can be seen as a special case of our model with a linear code, the minimal distance of which is 2. From a practical point of view, the computation overhead is very small and tends to zero.

References

1. Belal, A.A., Abdel-Gawad, M.A.: 2D-Encryption Mode. In: Schmalz, M.S. (ed.) SPIE 2003, vol. 4793, pp. 64–75 (2003)
2. Bellare, M., Desai, A., Jokipii, E., Rogaway, P.: A Concrete Security Treatment of Symmetric Encryption. In: FOCS 1997 (1997)
3. Bellare, M., Impagliazzo, R.: A tool for obtaining tighter security analyses of pseudorandom function based constructions, with applications to PRP to PRF conversion (1999)
4. Bellare, M., Kilian, J., Rogaway, P.: The Security of Cipher Block Chaining. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 341–358. Springer, Heidelberg (1994)
5. Bellare, M., Krovetz, T., Rogaway, P.: Luby-Rackoff Backwards: increasing Security by Making Block Ciphers Non-invertible. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 266–280. Springer, Heidelberg (1998)
6. Hall, C., Wagner, D., Kelsey, J., Schneier, B.: Building PRFs from PRPs. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 370–389. Springer, Heidelberg (1998)
7. Iwata, T.: New Blockcipher Modes of Operation with Beyond the Birthday Bound Security. In: Robshaw, M. (ed.) FSE 2006. LNCS, vol. 4047, pp. 310–327. Springer, Heidelberg (2006)
8. Knudsen, L.R.: Block Chaining Modes of Operation. NIST call for new modes of operation (2000)
9. Lucks, S.: The Sum of PRPs Is a Secure PRF. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 470–484. Springer, Heidelberg (2000)
10. MacWilliams, F.J., Sloane, N.J.A.: The theory of error-correcting codes. North-Holland, Amsterdam (1977)

A Security Proof of Theorem 1

A.1 Notations and Definitions

To make easier the understanding of the proof, we introduce the following notations. From an n -bit input x , the computation of $F^+(x)$ can be decomposed in the two following steps:

- compute the ℓ -tuple $(P(x), P(x+1), \dots, P(x+\ell-1))$,
- apply to the above ℓ -tuple an application denoted $\tilde{F}^+ : (\{0, 1\}^n)^\ell \rightarrow (\{0, 1\}^n)^u$ such that $\tilde{F}^+(P(x), \dots, P(x+\ell-1))$ is equal to

$$\left(\bigoplus_{\substack{1 \leq j \leq \ell \\ g_{1,j} \neq 0}} P(x+j-1), \bigoplus_{\substack{1 \leq j \leq \ell \\ g_{2,j} \neq 0}} P(x+j-1), \dots, \bigoplus_{\substack{1 \leq j \leq \ell \\ g_{u,j} \neq 0}} P(x+j-1) \right).$$

The function $\tilde{F}^+$ is defined by the matrix G .

Lucks has introduced properties to prove the security of $Twin^d$ [9]. He only considers the case of an image set included in $\{0, 1\}^n$. Here we extend his definitions to fit with image sets included in $(\{0, 1\}^n)^u$.

Definition 1. Let ℓ and u be two integers and $f : (\{0, 1\}^n)^\ell \rightarrow (\{0, 1\}^n)^u$. The set $T \subseteq (\{0, 1\}^n)^\ell$ is fair for f , if for every $y \in (\{0, 1\}^n)^u$

$$\left| \{(t_1, \dots, t_\ell) \in T \mid f(t_1, t_2, \dots, t_\ell) = y\} \right| = \frac{|T|}{2^{un}}.$$

If $T \subseteq (\{0, 1\}^n)^\ell$ is fair for $f : (\{0, 1\}^n)^\ell \rightarrow (\{0, 1\}^n)^u$, there is a uniform distribution over the output of f when applied to an element randomly picked in T . However, we will consider sets that are not fair, but almost fair. Such a property is defined as follows.

Definition 2. Let ℓ and u be two integers and $f : (\{0, 1\}^n)^\ell \rightarrow (\{0, 1\}^n)^u$. The set $T \subseteq (\{0, 1\}^n)^\ell$ is z -fair for f , if:

- a set $V \subseteq (\{0, 1\}^n)^\ell$ exists with $|V| = z$ and $V \cap T = \emptyset$, such that $V \cup T$ is fair for f . The set V is called a "completion set" for T ;
- or if a set $U \subseteq T$ with $|U| = z$ exists such that $T \setminus U$ is fair for f . The set U is called an "overhanging set" for T .

During the proof, we will also require some linear code theory results. In particular, we recall the Singleton bound (see [10] for example).

Definition 3 (Singleton bound). Any linear code of length ℓ , of dimension u and of minimal distance d verifies $\ell - u \geq d - 1$.

An other important result is that any generator matrix of size $u \times \ell$ associated to a linear code defined over $GF(2)$, of length ℓ and of dimension u has some equivalent forms in which the identity matrix of $u \times u$ appears.

A.2 Overview of the Proof

As recalled in section 2.1, to analyse the security of our PRF, we consider a distinguisher making q requests x_i , $1 \leq i \leq q$, to an oracle. The latter simulates the PRF or a random function of $Rand(n, nu)$, depending on the value of a bit parameter b .

The general idea of the proof is the same as the one for *Twin*^d [9]. For a given request x_i , $1 \leq i \leq q$, we denote T_i the set of all possible instantiations of the ℓ -tuple $(P(x_i), P(x_i + 1), \dots, P(x_i + \ell - 1))$. To simulate our PRF, the oracle randomly picks an element in T_i and apply $\tilde{F}^+$ on it. For each request, if the set T_i is fair (see definition 1 on the previous page), there is a uniform distribution over the output of $\tilde{F}^+$ so that the distinguisher has no advantage (to determine the value of the bit b).

However, the set T_i is not fair. The goal of the proof is to show that T_i is almost fair, i.e. T_i is z_i -fair, for a given z_i to determine. Let us denote T_i^* the fair set obtained from T_i . We analyse the oracle simulation assuming it randomly picks an ℓ -tuple in T_i^* instead of T_i . In a second step of the simulation, the oracle will verify that the selected element can actually be used as an instantiation, i.e. it is also in T_i (so that the simulation is not altered).

As it will be proved, T_i^* and T_i will only differ from few elements; i.e. T_i is z_i -fair with a small z_i . Thus, if the selected ℓ -tuple is in $T_i^* \cap T_i$ (most of the time as proved later), the distinguisher has no advantage over the bit b . And, if the picked element is not in $T_i^* \cap T_i$, the probability of such an event (equals to $z_i/|T_i|$) bounds the advantage of the distinguisher for the request. By summing this advantage among the q different requests, we obtain the advantage of the distinguisher.

The main goal of the proof is to bound the value z_i , for each request x_i , $1 \leq i \leq q$.

A.3 Security Analysis of F^+ (and F)

In theorem 1, the hypothesis $\forall(i, j) \in \mathbb{N}^2, 1 \leq i < j \leq q, \{x_i, x_i + 1, \dots, x_i + \ell - 1\} \cap \{x_j, x_j + 1, \dots, x_j + \ell - 1\} = \emptyset$ ensures that among the q requests x_i , we will exactly have to instantiate $q \times \ell$ outputs of the permutation P since there will not exist collision over the input of the permutation P . For each request x_i , $1 \leq i \leq q$, we denote

- T_i the set of all possible instantiations of $(P(x_i), P(x_i + 1), \dots, P(x_i + \ell - 1))$;
- T_i^* the fair set constructed from T_i ;
- $(\pi_{i,1}, \dots, \pi_{i,\ell})$ the ℓ -tuple used to instantiate $(P(x_i), \dots, P(x_i + \ell - 1))$;
- L_i , the set of all the values $\pi_{k,j}$, $1 \leq k < i$, $1 \leq j \leq \ell$ appearing in the chosen instantiations of previous requests x_k , $1 \leq k < i$.

Remark 3. For all i , $2 \leq i \leq q - 1$, $|L_i| = \ell(i - 1)$.

Simulation Description. The oracle simulation can be summed up by algorithm 2 on page 340. In a first step of the simulation, for each request x_i ,

$1 \leq i \leq q$, we first accept to instantiate $(P(x_i), P(x_i + 1), \dots, P(x_i + \ell - 1))$ with ℓ -tuples containing eventually two equal components (which cannot exist since P is a permutation). Thus, we consider T_i defined by

$$T_i = \left\{ (t_1, \dots, t_\ell), \quad \forall j, \quad 1 \leq j \leq \ell, \quad t_j \in \{0, 1\}^n \setminus L_i \right\}.$$

Note that the cardinality of T_i verifies

$$\begin{aligned} |T_i| &= (2^n - \ell(i - 1))^\ell \\ &\geq 2^{\ell n} - \ell(\ell(i - 1) 2^{\ell(n-1)}) = 2^{\ell n} - \ell^2(i - 1) 2^{\ell(n-1)}. \end{aligned} \quad (1)$$

The fair set T_i^* also contains ℓ -tuples with eventually two equal components. As said in the overview of the proof, the oracle first randomly picks $(\pi_{i,1}, \pi_{i,2}, \dots, \pi_{i,\ell})$ in the fair set T_i^* . In a second step, the oracle checks if $(\pi_{i,1}, \pi_{i,2}, \dots, \pi_{i,\ell})$ is also in $T_i^* \cap T_i$. If not, (step denoted “Bad case 1” in algorithm 2 on the next page), the oracle then randomly picks a new ℓ -tuple in T_i .

Finally, let C^* be the subset of $(\{0, 1\}^n)^\ell$ containing ℓ -tuples with at least 2 equal components. The cardinality of C^* is bounded by

$$(2^n - |L_i|)^{\ell-1} \binom{\ell}{2} \leq (2^n)^{\ell-1} \ell^2 / 2.$$

The oracle checks if $(\pi_{i,1}, \pi_{i,2}, \dots, \pi_{i,\ell})$ is in C^* . In such a case (denoted “Bad case 2” in algorithm 2) a new ℓ -tuple with ℓ different components is randomly picked in $T_i \cap \overline{C^*}$. (Thus, for each request x_i , $1 \leq i \leq q$, the set L_i always contains exactly $\ell(i - 1)$ elements).

These two steps “Bad case 1” and “Bad case 2” ensure a valid oracle simulation and if no such “bad case” appends, the distinguisher has no advantage since the ℓ -tuple has been randomly picked in a fair set. The advantage of the distinguisher is bounded by the probability of the event “Bad case 1” or “Bad case 2”. The main technical point of the proof is to bound the value z_i such that the T_i , $1 \leq i \leq q$, is z_i -fair.

Fairness of T_i . We first give a useful lemma. Without loss of generality, we assume that the u columns of identity matrix $u \times u$ are already in G .

Lemma 1. *Let G be the generator matrix (of size $u \times \ell$) associated to a linear code defined over $GF(2)$, of length ℓ , of dimension u and of minimal distance d such that the u columns of the identity matrix are the u columns $i_1, \dots, i_u$ of G . Let $\tilde{F}^+$ be the function defined on page 337 and let $T \subseteq (\{0, 1\}^n)^\ell$. If the components $i_1, \dots, i_u$ of T are defined over $\{0, 1\}^n$, then the set T is fair for $\tilde{F}^+$.*

Proof. The components $i_1, \dots, i_u$ of T are associated to the columns of the identity matrix appearing in G . Thus, these components correspond to the terms $P(x - 1 + j)$, for all $j \in \{i_1, \dots, i_u\}$ and are used only once and each for only one of the u components of the output of $\tilde{F}^+$. Thus, for any instantiation of the $\ell - u$

Algorithm 2. Oracle simulation

```

 $bad \leftarrow 0$ 
for  $i$  from 1 to  $q$  do
  Determine the fair set  $T_i^*$  from  $T_i$ 
  Randomly pick an element  $(\pi_{i,1}, \dots, \pi_{i,\ell})$  in  $T_i^*$ 
  {Bad case 1}
  if  $(\pi_{i,1}, \dots, \pi_{i,\ell}) \notin T_i^* \cap T_i$  then
     $bad \leftarrow 1$ 
    Randomly pick a new element  $(\pi_{i,1}, \dots, \pi_{i,\ell})$  in  $T_i$ 
  {Bad case 2}
  if  $(\pi_{i,1}, \dots, \pi_{i,\ell}) \in C^*$  then
     $bad \leftarrow 1$ 
    Randomly pick a new element  $(\pi_{i,1}, \dots, \pi_{i,\ell})$  in  $T_i \cap \overline{C^*}$ 
  Output  $\tilde{F}^+(\pi_{i,1}, \dots, \pi_{i,\ell})$ 

```

other components of T , there is a bijection between the image set $(\{0, 1\}^n)^u$ and the u components $P(x - 1 + j)$, for all $j \in \{i_1, \dots, i_u\}$. As a consequence, T is fair and each image element $y \in (\{0, 1\}^n)^u$ is reached as often as the number of possible instantiations of the $\ell - u$ other components.

The core of the proof consists in decomposing the set T_i into a union and/or difference of subsets of $(\{0, 1\}^n)^\ell$, $1 \leq j$, each verifying only one of the two following properties:

Property 1: u components $i_1, i_2, \dots, i_u$ are defined over $\{0, 1\}^n$ and there exists a generator matrix G' , equivalent to G , which contains the identity matrix $u \times u$ in columns $i_1, i_2, \dots, i_u$. Lemma 1 can be applied to conclude that the set is fair for $\tilde{F}^+$;

Property 2: d components are defined over L_i . These sets will be of negligible cardinality in comparison with the cardinality of T_i , and will correspond to completion or overhanging sets for T_i .

For the proof, we consider the list of images $\mathcal{L}(T_i)$ obtained by applying $\tilde{F}^+$ to T_i . In this list, an element of $(\{0, 1\}^n)^u$ appears as often as its number of pre-images in T_i .

The method to obtain an adequate decomposition consists in the recursive algorithm named *Decomposition*(MAT, T) and described in algorithm 3. It takes as input a generator matrix MAT and a subset T of $(\{0, 1\}^n)^\ell$. The algorithm is initialized with G and T_i .

Let us consider the tree of the recursive execution of the algorithm *Decomposition*(G, T_i). The root corresponds to the set T_i . At each generation of the tree, the definition set of one of the ℓ components of a given node is modified into $\{0, 1\}^n$ or L_i which leads to two child nodes. Thus, after $u + d - 1$ generations in the tree, each leaf verifies property 1 or 2 (the sets involved in the $(u + d - 1)^{\text{th}}$ generation of the tree contain $d - 1$ components defined over L_i and u components defined over $\{0, 1\}^n$). Using the Singleton bound ($\ell \geq u + d - 1$), and since

Algorithm 3. *Decomposition*(MAT, T)

Let k , $1 \leq k \leq u$ be the least integer such that the k^{th} row of MAT contains no “1” with a corresponding component in T defined over $\{0, 1\}^n$. {We select the first row involving no component of T defined over $\{0, 1\}^n$ }

Let j , $1 \leq j \leq \ell$ be the least integer such that $\text{MAT}_{k,j} = 1$ and the j^{th} component of T is defined over $\{0, 1\}^n \setminus L_i$.

Decompose T into the form $A \setminus B$ according to the j^{th} component such that the j^{th} component of A and B are now defined respectively over $\{0, 1\}^n$ and L_i {We obtain $\mathcal{L}(T) = \mathcal{L}(A) - \mathcal{L}(B)$ }

Compute the generator matrix MAT' , equivalent to MAT, such that the k^{th} row of MAT' is the only row with a “1” in column j {We obtain the k^{th} column of the identity matrix $u \times u$ }

if A verifies property 1 **then**

 return A and execute *Decomposition*(MAT, B)

else

if B verifies property 2 **then**

 return B and execute *Decomposition*(MAT', A)

else

 execute *Decomposition*(MAT', A) and *Decomposition*(MAT, B)

the algorithm *Decomposition* is applied to G and T_i , it is always possible to obtain $u + d - 1$ generations in the tree, i.e. *Decomposition*(G, T_i) always ends with sets verifying property 1 or 2.

Let us evaluate the number denoted N of sets verifying property 2. These sets have k , $0 \leq k \leq u - 1$, components defined over $\{0, 1\}^n$ among the first $k + d - 1$ generations in the tree. Thus, the number of sets verifying property 2 is given by

$$N = \sum_{k=0}^{u-1} \binom{d+k-1}{d-1}.$$

The cardinality of such a set with exactly k , $0 \leq k \leq u - 1$, components defined over $\{0, 1\}^n$ is $|L_i|^d \times 2^{nk} \times (2^n - |L_i|)^{\ell-d-k}$.

When the algorithm ends, we obtain one of the two following equalities, depending of the parity of d .

$$\mathcal{L}(T_i) = \sum_j \mathcal{L}(A_j^i) - \sum_j \mathcal{L}(B_j^i) - \sum_{j=1}^N \mathcal{L}(C_j^i) \quad \text{if } d \text{ is odd,} \quad (2)$$

$$\mathcal{L}(T_i) = \sum_j \mathcal{L}(A_j^i) - \sum_j \mathcal{L}(B_j^i) + \sum_{j=1}^N \mathcal{L}(C_j^i) \quad \text{if } d \text{ is even.} \quad (3)$$

In both equalities, the sets C_j^i verify property 2 and the sets A_j^i and B_j^i verify property 1: lemma 1 can be applied, i.e. the sets A_j^i and B_j^i are fair.

After a first step of the decomposition algorithm, we obtain $\mathcal{L}(T) = \mathcal{L}(A) - \mathcal{L}(B)$ where A has no component defined over L_i and B has one component defined over L_i . When applying the algorithm to A and B , we obtain $\mathcal{L}(A) = \mathcal{L}(A_1) - \mathcal{L}(A_2)$ and $\mathcal{L}(B) = \mathcal{L}(B_1) - \mathcal{L}(B_2)$ so that we obtain $\mathcal{L}(T) = \mathcal{L}(A_1) - \mathcal{L}(A_2) - \mathcal{L}(B_1) + \mathcal{L}(B_2)$. The set A_1 has no component defined over L_i , A_2 and B_1 have one component defined over L_i and B_2 has two such components. It is quite easy to see by induction that the sign of a term $\mathcal{L}(D)$ is directly linked to the parity of the number of components of D defined over L_i . Thus, in equalities (2) and (3), the sets A_j^i (resp. B_j^i) have an even (resp. odd) number of components defined over L_i . Since the sets C_j^i have exactly d components defined over L_i , the sign of $\mathcal{L}(C_j^i)$ depends on the parity of d . This justifies the distinction over the parity of d in equalities (2) and (3).

Let us first consider equality (2). The sets C_j^i , $1 \leq j \leq N$ are not necessarily disjoint. However, if $|T_i| + \sum_{j=1}^N |C_j^i| \leq 2^{n\ell}$, there are enough ℓ -tuples in $(\{0, 1\}^n)^\ell \setminus T_i$ to construct a set C such $\mathcal{L}(C) = \sum_{j=1}^N \mathcal{L}(C_j^i)$. In the same way for equality (3), if $0 \leq |T_i| - \sum_{j=1}^N |C_j^i|$ there is enough ℓ -tuples in T_i to construct a set $C \subseteq T_i$ such $\mathcal{L}(C) = \sum_{j=1}^N \mathcal{L}(C_j^i)$. Thus, we can rewrite equalities (2) and (3) as

$$\begin{aligned}\mathcal{L}(T_i \cup C) &= \sum_j \mathcal{L}(A_j^i) - \sum_j \mathcal{L}(B_j^i) && \text{if } d \text{ is odd,} \\ \mathcal{L}(T_i \setminus C) &= \sum_j \mathcal{L}(A_j^i) - \sum_j \mathcal{L}(B_j^i) && \text{if } d \text{ is even.}\end{aligned}$$

Since the sets A_j^i and B_j^i are fair, the set C is a completion set for T_i if d is odd or an overhanging set for T_i if d is even. Thus, T_i is z_i -fair, with

$$\begin{aligned}z_i = |C| &= \sum_{k=0}^{u-1} \binom{d+k-1}{d-1} |L_i|^d \times 2^{nk} \times (2^n - |L_i|)^{\ell-d-k} \\ &\leq \ell^d \times |L_i|^d \times 2^{n(\ell-d)} \leq \ell^{2d} (i-1)^d 2^{n(\ell-d)}\end{aligned}$$

(the first inequality uses the remark 1).

The inequalities $|T_i| + \sum_{j=1}^N |C_j^i| \leq 2^{n\ell}$ and $0 \leq |T_i| - \sum_{j=1}^N |C_j^i|$ are verified if

$$1 \leq i \leq q \leq 2^{n-1}/\ell^2. \quad (4)$$

For a given request x_i , an ℓ -tuple randomly picked in T_i^* may not be in $T_i^* \cap T_i$ with a probability $p_{1,i}$ verifying

$$p_{1,i} = \frac{z_i}{|T_i|} \leq \frac{\ell^{2d} (i-1)^d 2^{n(\ell-d)}}{|T_i|}.$$

Using inequality (1) and inequality (4), we obtain $|T_i| \geq 2^{n\ell-1}$ so that

$$p_{1,i} \leq \frac{\ell^{2d}}{2^{dn-1}} (i-1)^d.$$

As explained previously, an ℓ -tuple randomly picked in T_i may also be in $T_i \cap C^* \neq \emptyset$. This is a problematic case and a new ℓ -tuple must be chosen to leave the simulation correct. As seen previously, $|T_i| \geq 2^{n\ell-1}$, so the probability $p_{2,i}$ of this event verifies

$$p_{2,i} = \frac{|C^*|}{|T_i|} \leq \frac{(2^n)^{\ell-1} \ell^2 / 2}{2^{n\ell-1}} \leq \frac{\ell^2}{2^n}.$$

Thus, at each request x_i , $1 \leq i \leq q$, the advantage of the distinguisher is bounded by $p_{1,i} + p_{2,i}$. The overall advantage of the distinguisher is given by

$$\sum_{i=1}^q \left(\frac{\ell^{2d}}{2^{dn-1}} (i-1)^d + \frac{\ell^2}{2^n} \right) \leq \frac{q^{d+1} \ell^{2d}}{2^{dn}} + \frac{q \ell^2}{2^n}.$$

The security level is determined by the term $\frac{q^{d+1} \ell^{2d}}{2^{dn}}$ which is beyond the birthday bound for any $d \geq 2$.