Università degli Studi di Trento

Dipartimento di Matematica

Primitivity of generalized translation based block ciphers.

Ilaria Zappatore

Supervisors:	Prof. Massimiliano Sala Dr. Riccardo Aragona Dr. Marco Calderini
Advisor:	Dr. Giancarlo Rinaldo

Anno accademico 2015/2016

Università degli Studi di Trento

Dipartimento di Matematica

Primitivity of generalized translation based block ciphers.

Thesis of: Ilaria Zappatore

Supervisors: Prof. Massimiliano Sala

Dr. Riccardo Aragona

Dr. Marco Calderini

Advisor: Dr. Giancarlo Rinaldo

Anno accademico 2015/2016

To the love of my life, Math.

Contents

Acknowledgments.

I would like to express my sincere gratitude to all my supervisors: Prof. Massimiliano Sala for his precious advices and encouragement throughout my Master's studies, Dr. Riccardo Aragona and Dr. Marco Calderini for their countless hours of reflecting, studying, reading and most of all patience during the entire process. Thank you for reassuring me in stressful moments.

Furthermore I would like to thank all my classmates and best friends which made me happy throughout these studies: Maria Chiara Molteni, Alessandro Melloni, Andrea Vinci, Davide Trotta, Alessandro Budroni, Giuseppe Giffone, Ermes Franch and Carlo Brunetta.

Finally, I must express my very profound gratitude to my family, especially to my granny Tina, and to Francesco who have supported with patience my passion for math and the process of writing this thesis. Thank you.

Introduction

In this thesis we introduce a new model of block ciphers, called *generalized translation based* block ciphers, which is, as the name suggests, a generalization of *translation based* block ciphers, firstly proposed by Caranti, Dalla Volta, Sala in [CVS09] and we prove the *primitivity* of the related *group generated by the round functions*.

Many block ciphers used in real life are *iterated* block ciphers, given by the composition of different key-dependent permutations of the plaintext/ciphertext space, called *round functions*. The authors in [CVS09], observed that some iterated block ciphers have a similar structure. Therefore they define a new class of iterated block ciphers, called *translation based* block ciphers, in which each round function is given by the composition of a non-linear *vectorial Boolean function* which is also a permutation, called *bricklayer transformation*, a linear permutation called *mixing layer* and a translation. This class of ciphers contains well-known ciphers, such as AES [Nat01], SERPENT [BAK98] and PRESENT [AKL$^+$07].

Coppersmith and Grossman in [CG75] studied a particular set of functions and their possible integration into a block cipher and analyzed the permutation group generated by them. These studies opened the way to many researchers that later analyzed the group generated by the round functions of block ciphers aiming to find properties which could reveal weaknesses of the cipher itself. In this direction Kalinski et al. in [KRS88] proved that if this group is too small, then the cipher is vulnerable to certain kind of attacks based on the *birthday paradox*. However, Murphy Paterson and Wild presented in [MPW94] an example of weak block cipher whose round functions generate the full symmetric group, thus proving that the fact that this group is big is not sufficient to build a strong cipher.
Furthermore, if the group has a known structure, for example if it is affine, or if it is *imprimitive*, it is possibile to embed a trapdoor, that is a hidden structure of the cipher which could be used to attack the cipher itself, as showed in [CS15] and [Pat99] respectively.

As we have previously remarked the study of the group generated by the round functions of an iterated block cipher is an interesting research problem, which involved many researchers. In 1983 Even and Goldreich defined certain DES-like functions, and proved that the permutation group generated by these functions is the alternating group [EG83]. After some years, Wernsdorf proved that the round functions of DES [Wer93] and SERPENT [Wer00] generate the alternating group, and Sparr and Wernsdorf showed that the same holds for KASUMI [SW15] and AES [SW08]. Moreover, in 2015, Aragona, Caranti, Sala introduced in [ACS17] a cipher that is an extension of GOST [Dol10] and studied the permutation group generated by its round functions, proving that under certain assumptions on the cipher components, this group is the alternating group.

In [CVS09] authors proved that, under some cryptographic conditions on bricklayer transformations, the group generated by the round functions of a translation based block cipher is *primitive*. This result is particularly important from the cryptographic point of view, since it allows to avoid the *Paterson attack*, which consists of insertion of a trapdoor based on the imprimitivity of the group generated by the round functions [Pat99]. Moreover in [CDS09], using the O'Nan Scott classification of primitive groups [Cam99], and adding another cryptographic assumption, authors proved that the group generated by the round functions is the *alternating* or the *symmetric* group. This allows to avoid the insertion of a trapdoor based on the affinity of the group generated by the round functions [CS15].
Since AES and SERPENT satisfy all these conditions, Caranti, Dalla Volta and Sala proved in a different way with respect to Sparr and Wernsdorf, that the group generated by the round functions of these two block ciphers is the alternating group [CDS09]. Nevertheless, these results are not applicable to PRESENT, since its bricklayer transformation does not verify those cryptographic assumptions. Thus, in [ACTT15], authors introduced another primitivity theorem slightly modifying cryptographic assumptions of [CDS09]. With these new hypotheses also the group generated by the round functions of PRESENT is primitive. Moreover they proved that in some cases, which include the PRESENT cipher, in order to prove that the group generated by the round functions of a translation based block cipher is the alternating or the symmetric group, it is enough to check the same conditions used to show its primitivity.

In this work we study and model generalized translation based block ciphers, in which each round function is given by the composition of a non-linear vectorial Boolean function from a vector space V to another one with bigger dimension W, also called *bricklayer transformation*, with a *mixing layer*, that is a linear transformation from W to V, and a translation (Definition 3.1.4). This is a generalization of translation based block ciphers, since if we consider $V = W$ the two definitions coincide. We also give an example which shows that the class of generalized translation based block ciphers properly contains the translation based one. This proves that there are some generalized translation based block ciphers that cannot be reconducted to "classical" ones and made relevant the study of this new category, from the cryptographic point of view. We analyze and give some results regarding the structure of the group generated by the round functions of generalized translation based block ciphers and then we prove its primitivity using some hypotheses that are the generalization of cryptographic assumptions of [CVS09] (Theorem 3.2.5).

This thesis is organized as follows.

- In Chapter 1, we introduce some algebraic concepts of finite fields and group theory. We also fix notations used during the work and give some definitions and properties of vectorial Boolean functions.

- In Chapter 2, starting from a brief introduction on block ciphers and attack scenarios, we describe iterated block ciphers. In particular, we give some important examples (AES and PRESENT), we describe the related group generated by the round functions and an attack based on the imprimitivity of this group, the Paterson attack. In conclusion we define translation based block ciphers and give some results on them, with a focus on the primitivity of the group generated by the round functions.

- In Chapter 3, we define generalized translation based block ciphers and then we prove the primitivity of the related group generated by the round functions.

- In Appendix A.1, we recall some additional results on the group generated by the round functions of translation based block ciphers. In particular, we introduce theorems which prove that it is isomorphic to the symmetric or the alternating group.

Preliminaries.

In this chapter we introduce notations and some algebraic concepts that we will use in the course of this work. In particular, in the first part, after a briefly introduction on finite fields we discuss about *Boolean functions* and state some properties that we will use later. On the other hand, in the second part we recall some basic results on *group actions* focusing on *primitive* and *imprimitive* ones.

1.1 Finite fields.

In all this section we refer to [LN03], unless otherwise specified.
A finite field is a field with a finite number of elements, called *order* of the field.
A basic example of a finite field is $\mathbb{F}_p := \mathbb{Z}_p$ with p prime. In this case, the two operations of the field are the integer addition modulo p and the integer multiplication modulo p.

Recall that:

- The *characteristic* of a ring R is the least positive integer n such that $nr = 0$, for any $r \in R$.

  In particular, a finite field has prime characteristic.

- The *degree* of the *extension field* $\mathbb{L}$ of the field $\mathbb{K}$, is the dimension of $\mathbb{L}$ as a $\mathbb{K}$-vector space. We denote it with $[\mathbb{L} : \mathbb{K}]$.

- The *prime subfield* of a field $\mathbb{F}$, that is the intersection of all its subfields, is isomorphic to $\mathbb{F}_p$ or $\mathbb{Q}$, according as the characteristic of $\mathbb{F}$ is a prime p or 0.

  Thus, the prime subfield of a finite field is isomorphic to $\mathbb{F}_p$.

The order of a finite field $\mathbb{F}$ is p^n where:

- p is the characteristic of the field,

- n is the degree of the field over its prime subfield, that is $[\mathbb{F} : \mathbb{F}_p] = n$.

Therefore, the dimension of $\mathbb{F}$ as an $\mathbb{F}_p$-vector space is n and we have that $\mathbb{F} = \mathbb{F}_{p^n} \cong (\mathbb{F}_p)^n$, where the symbol "$\cong$" stands for "is isomorphic as vector space to".

We state a basic result for finite fields theory.

Theorem 1.1.1 (Existence and uniqueness of finite fields). *For every prime p and every positive integer n there exists a finite field with p^n elements. Any finite field with $q = p^n$ elements is isomorphic to the splitting field of $x^p - x$ over $\mathbb{F}_p$.*

Proof. See Theorem 2.5 of [LN03]. □

For the uniqueness part of the previous theorem we may speak of *the* finite field with q elements. We will denote this field with $\mathbb{F}_q$.

Throughout this work we will deal only with *binary* finite fields $\mathbb{F}_{2^n}$, for some positive integer n.

1.1.1 Boolean functions.

A function $f : (\mathbb{F}_2)^n \longrightarrow \mathbb{F}_2$, with $n \geq 1$, is called *Boolean function.*
We denote with $\mathcal{B}_n$, the set of all Boolean functions from $(\mathbb{F}_2)^n$ to $\mathbb{F}_2$.

Each Boolean function $f \in \mathcal{B}_n$, can be uniquely written as a polynomial $f(x_1, \dots, x_n) \in \mathbb{F}_2[x_1, \dots, x_n]$. In particular,

$$f(x_1, \dots, x_n) = \sum_{S \subseteq \{1, \dots, n\}} a_S X_S,$$

where $X_S = \prod_{i \in S} x_i$. This representation is called *Algebraic Normal Form* of f, or simply *ANF*.

A function $f : (\mathbb{F}_2)^n \longrightarrow (\mathbb{F}_2)^m$, with $n, m \geq 1$, is called *vectorial Boolean function* (v.B.f for short).
Thus any Boolean function $f \in \mathcal{B}_n$ is a v.B.f with $m = 1$.

The notion of ANF of a Boolean function can easily be extended to vectorial Boolean functions. Precisely, the ANF of $f : (\mathbb{F}_2)^n \longrightarrow (\mathbb{F}_2)^m$ is

$$f(x_1, \dots, x_n) = \sum_{S \subseteq \{1, \dots, n\}} a_S X_S,$$

where, in this case, $a_S \in (\mathbb{F}_2)^m$ and $X_S = \prod_{i \in S} x_i$.

Recall that if $\mathbb{F}_q$ is a finite field, any function $f : \mathbb{F}_q \longrightarrow \mathbb{F}_q$ can be represented as a polynomial in $\mathbb{F}_q[x]$, with degree at most $q - 1$ (see [LN03] for further details).

Therefore, if $m = n$, we can also identify any v.B.f $f : (\mathbb{F}_2)^n \longrightarrow (\mathbb{F}_2)^n$, with an univariate polynomial in $\mathbb{F}_{2^n}$, since $\mathbb{F}_{2^n} \cong (\mathbb{F}_2)^n$ as remarked in the previous section.

In the following example we introduce the vectorial Boolean function used in AES (Section 2.3.1).

Example 1.1.2. Let n be a positive integer and $f : \mathbb{F}_{2^n} \longrightarrow \mathbb{F}_{2^n}$ be the map

$$f(x) = \begin{cases} x^{-1} & \text{if } x \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

This function is called *patched inversion* and since $\mathbb{F}_{2^n} \cong (\mathbb{F}_2)^n$, it is a vectorial Boolean function. Its polynomial representation is given by the univariate polynomial

$$f(x) = x^{2n-2} \in \mathbb{F}_{2^n}[x].$$

For any v.B.f $f : (\mathbb{F}_2)^n \longrightarrow (\mathbb{F}_2)^m$ and $u \in (\mathbb{F}_2)^n$, $u \neq 0$, we define the function

$$\begin{array}{rccc} \hat{f}_u : & (\mathbb{F}_2)^n & \longrightarrow & (\mathbb{F}_2)^m \\ & x & \longmapsto & f(x+u) + f(x) \end{array}$$

and we denote the image of this function by

$$\mathrm{Im}(\hat{f}_u) = \{f(x+u) + f(x) \mid x \in (\mathbb{F}_2)^n\}.$$

Moreover, for any $v \in (\mathbb{F}_2)^m$ we denote by

$$\delta_f(u,v) := |\hat{f}_u^{-1}(v)| = |\{x \in (\mathbb{F}_2)^n \mid \hat{f}_u(x) = v\}|,$$

the cardinality of the preimage of v with respect to the function $\hat{f}_u$.

Now, we are ready for the following definition.

Definition 1.1.3 ([Car10])**.** *Let δ be a positive integer. Any vectorial Boolean function $f : (\mathbb{F}_2)^n \longrightarrow (\mathbb{F}_2)^m$ is called differentially δ-uniform (δ-uniform for short) if, for any $u \in (\mathbb{F}_2)^n$, $u \neq 0$ and $v \in (\mathbb{F}_2)^m$ then*

$$\delta_f(u,v) \leq \delta.$$

The smallest such δ is called differential uniformity of the function f.

Observe that, for any v.B.f, $\delta \geq 2$ ([Car10]). If $\delta = 2$, we have *almost perfect non-linear* vectorial Boolean functions (shortly APN).
Furthermore, if $f : (\mathbb{F}_2)^n \longrightarrow (\mathbb{F}_2)^m$ is a linear v.B.f and fixed $u \in (\mathbb{F}_2)^n$, $u \neq 0$ and $v \in (\mathbb{F}_2)^m$ then

$$\delta_f(u,v) = \begin{cases} 2^n & \text{if } f(u) = v \\ 0 & \text{otherwise} \end{cases}$$

Thus, for linear v.B.f we have the highest value of uniformity. Since, in block ciphers are used non-linear vectorial Boolean functions (Section 2.3), we are interested to small values of δ.

Definition 1.1.4 ([CVS09])**.** *Let δ be a positive integer. A vectorial Boolean function $f : (\mathbb{F}_2)^n \longrightarrow (\mathbb{F}_2)^m$ is weakly δ-differentially uniform (weakly δ-uniform for short) if, for any $u \in (\mathbb{F}_2)^n$, $u \neq 0$, then*

$$|\mathrm{Im}(\hat{f}_u)| > \frac{2^{n-1}}{\delta}$$

If $\delta = 2$, the function f is called *weakly APN.*

Remark 1.1.5. Observe that any δ-uniform v.B.f. $f : (\mathbb{F}_2)^n \longrightarrow (\mathbb{F}_2)^m$ is also weakly δ-uniform. In fact, by the δ-uniformity, for any $u \in (\mathbb{F}_2)^n, u \neq 0$ and $v \in \mathrm{Im}(\hat{f}_u)$, $\delta_f(u,v) \leq \delta$.
Now, since

$$2^n = \bigcup_{v \in \mathrm{Im}(\hat{f}_u)} \delta_f(u,v) \leq \delta |\mathrm{Im}(\hat{f}_u)|,$$

then

$$|\mathrm{Im}(\hat{f}_u)| \geq \frac{2^n}{\delta} > \frac{2^{n-1}}{\delta}.$$

Throughout this work we will denote the dimension of a vector space V by $\dim(V)$.

Now we are ready for the last definition of this section.

Definition 1.1.6. *A vectorial Boolean function $f : (\mathbb{F}_2)^n \longrightarrow (\mathbb{F}_2)^n$ is:*

1. *l-anti-invariant if, for any subspace U of $(\mathbb{F}_2)^n$ such that $f(U) = U$, we have that dim$(U) < n - l$ or $U = (\mathbb{F}_2)^n$;*

2. *strongly l-anti-invariant, if for any two subspaces U, W of $(\mathbb{F}_2)^n$ such that $f(U) = W$, we have either dim$(U) =$ dim$(W) < n - l$ or $U = W = (\mathbb{F}_2)^n$.*

1.2 Group actions and primitivity.

In this section we refer to [DM96], unless otherwise specified. Let us begin with the definition of *group actions*.

Definition 1.2.1. *Let $(G, \cdot)$ be a group with neutral element e and X a nonempty set. An action of the group G over the set X is a map*

$$\begin{array}{rccc} \varphi : & G \times X & \longrightarrow & X \\ & (g, x) & \longmapsto & \varphi(g, x) \end{array}$$

such that:

1. *$\varphi(e, x) = x$, for all $x \in X$,*
2. *$\varphi((g \cdot h), x) = \varphi(g, \varphi(h, x))$, for all $g, h \in G$ and $x \in X$.*

In all this work we will denote the action of an element g of a group over an element x of a set, simply as

$$\varphi(g, x) := xg.$$

Moreover, whenever we speak about a group acting on a set we will implicitly assume that the set is nonempty.

We will denote with $\mathrm{Sym}(X)$ the *symmetric group* over a nonempty set X, that is the set of all permutations from X to itself.
A *permutation group* of a set X, is a subgroup of the symmetric group $\mathrm{Sym}(X)$. Moreover, the *degree* of a permutation group of X is the cardinality of the set X.

Now, let us consider an example of action.

Example 1.2.2. Any subgroup G of $\mathrm{Sym}(X)$, or in other words, any permutation group of X, acts naturally on X. In fact in this case, given $g \in G$ and $x \in X$, the action is determined by

$$xg := g(x),$$

that is the image of x under the permutation g.

If a group G acts on a set X then each $g \in G$ determines the bijection

$$\begin{array}{rccc} \psi_g : & X & \longrightarrow & X \\ & x & \longmapsto & xg \end{array}$$

whose inverse is $\psi_{g^{-1}}$. Hence $\psi_g \in \mathrm{Sym}(X)$ and we can consider the function:

$$\begin{array}{rccc} \psi : & G & \longrightarrow & \mathrm{Sym}(X) \\ & g & \longmapsto & \psi_g \end{array}$$

This map is a group homomorphism, since:

$$\psi(g \cdot h) = \psi(g)\psi(h), \text{ for } g, h \in G.$$

In general any homomorphism of G into $\mathrm{Sym}(X)$ is called *(permutation) representation of G on X*.

Therefore we have seen that each action of G on X determines a representation of G on X. Conversely, representations correspond to actions. In fact, let $\psi : G \longrightarrow \mathrm{Sym}(X)$ be a representation, we define the action

$$xg := x\psi(g).$$

It is easy to verify that this is an action, indeed. Specifically:

1. $xe = x\psi(e) = x$, for any $x \in X$, since $\psi(e)$ is the *identity* map $id : X \longrightarrow X$ such that $id(x) = x$, for any $x \in X$.

2. $x(g \cdot h) = x\psi(g \cdot h) = (x\psi(g))\psi(h) = (xg)h$, for any $g, h \in G$ and $x \in X$, since ψ is a group homomorphism.

Therefore, we speak equivalently of group actions and representations.

The *kernel* of an action is the kernel of the corresponding representation, that is:

$$\ker \psi = \{g \in G \mid \psi_g = id\},$$

where *id* is the identity map. In other terms we have that

$$\ker \psi = \{g \in G \mid xg = x, \forall x \in X\}.$$

Recall that in general, $\ker \psi$ is a subgroup of G and $\mathrm{Im}(\psi)$ is a subgroup of $\mathrm{Sym}(X)$.

An action is called *faithful* if the kernel of the corresponding representation is trivial, that is $\ker \psi = \{e\}$. In this case, for the *first isomorphism theorem* (see [LN03] for more details), G is isomorphic to the image of ψ, denoted as $\mathrm{Im}(\psi)$.

Let us consider an example:

Example 1.2.3 (Cayley representation)**.** For any group G, we can consider $X := G$ and define an action φ of G into itself

$$\begin{aligned} \varphi : \quad G \times G &\longrightarrow G \\ (g, h) \quad &\longmapsto \quad g \cdot h \end{aligned}$$

where $\cdot$ denotes the operation of the group G. The corresponding representation ψ of G into $\mathrm{Sym}(G)$ is called *regular representation.* The action is faithful, since the kernel is

$$\ker \psi = \{g \in G \mid hg = h, \forall h \in G\} = \{e\}.$$

This result shows that every group is isomorphic to a permutation group.

Given a group G acting on a set X we can define the following relation ρ on X

$$\text{for any } x, y \in X, x\rho y \iff \exists g \in G \text{ s.t. } y = xg.$$

This is an equivalence relation [DM96].
Equivalence classes with respect to this relation are called *orbits.* In particular, the orbit of an element $x \in X$ is

$$xG := \{y \in X \mid y = xg \text{ for certain } g \in G\} = \{xg \mid g \in G\}.$$

On the other hand, the stabilizer of $x \in X$ in G is the set

$$G_x = \{g \in G \mid xg = x\}$$

or, in other terms, the set of elements of G that fix x.

Properties of orbits and stabilizers are summarized in the following theorem.

Theorem 1.2.4. *Suppose that G is a group acting on a set X. Let $g, h \in G$ and $x, y \in X$. Then*

1. *Two orbits xG and yG are either equal or disjoint, so the set of all orbits is a partition of X.*

2. *The stabilizer G_x is a subgroup of G and $G_y = g^{-1}G_xg$ whenever $y = xg$. Moreover, $xg = xh$ if and only if $G_xg = G_xh$.*

3. *The cardinality of xG, is equal to the index of the stabilizer G_x over the group G or, in other terms $|xG| = (G : G_x)$ for all $x \in X$.*

Proof. See Theorem 1.4A of [DM96]. □

The property (3) of the previous theorem is called *orbit-stabilizer property.*

The action of a group G acting on a set X is called *transitive* if there is only one orbit:

$$xG = X, \text{ for any } x \in X.$$

Equivalently, for any $x, y \in X$,

$$\exists g \in G \text{ s.t. } xg = y.$$

Furthermore the transitive action of a group G on X is called *regular* if only the neutral element fixes any point of X, that is

$$\forall x \in X, G_x = \{e\}.$$

This is equivalent to say that, for any $x, y \in X$,

$$\exists! g \in G \text{ s.t. } xg = y.$$

Let G be a finite group acting transitively on a set X. A partition $\mathcal{B}$ of X is

- *trivial* if $\mathcal{B} = X$ or $\mathcal{B} = \{\{x\} \mid x \in X\}$,
- *G-invariant* if for any $B \in \mathcal{B}$ and $g \in G$ then $Bg \in \mathcal{B}$.

Any partition $\mathcal{B}$ non-trivial and G-invariant is called *block system* and in particular any $B \in \mathcal{B}$ is said *block.*

Definition 1.2.5 (imprimitive action). *Let G be a finite group acting transitively on a set X. Then G is imprimitive (equivalently G acts imprimitively on X) if there exixts a block system.*

On the other hand, if the group is not imprimitive we call it *primitive.*

Let us return for a while to general actions in order to introduce some concepts that we will use later.

Definition 1.2.6 ([DM96]). *Let G be a group acting on X and $S \subseteq X$. Then*

- *the pointwise stabilizer is*

$$G_{(S)} := \{g \in G \mid xg = x, \forall x \in S\}.$$

- *the setwise stabilizer is*

$$G_{\{S\}} := \{g \in G \mid Sg = S\}.$$

We observe that both $G_{(S)}$ and $G_{\{S\}}$ are subgroups of G, that is $G_{(S)}, G_{\{S\}} \leq G$, and in particular $G_{(x)} = G_{\{x\}} = G_x$.

Now we are ready for the following result which connects blocks of imprimitivity to subgroups.

Theorem 1.2.7. *Let G be a finite group acting transitively on a set X and $x \in X$. Then blocks B of G containing x are in one-to-one correspondence with proper subgroups H of G properly containing the stabilizer of x in G. In particular, denoted with Ψ such a bijection, then*

- *for any block B such that $x \in B$, we have $\Psi(B) = G_{\{H\}}$ with $G_x < H < G$.*
- *The inverse is $\Psi^{-1}(H) = xH$, for any H proper subgroup of G properly containing G_x.*

Proof. See Theorem 1.5A of [DM96]. □

Therefore a block of imprimitivity is of the form xH, for some $x \in X$, where H is such that $G_x < H < G$.

Let V be a vector space over the finite field $\mathbb{F}_q$, where $q = p^f$ for some p prime and $f \geq 1$, with dimension $\dim(V) = d \geq 1$. For $v \in V$, the map

$$\begin{array}{rccc} \sigma_v : & V & \longrightarrow & V \\ & w & \longmapsto & w + v \end{array}$$

is the *translation* of v. We have that $\sigma_v \in \mathrm{Sym}(V)$ and we denote with

$$\mathrm{T}(V) := \{\sigma_v \mid v \in V\}$$

the set of all translations. This is a group with respect to the composition of functions and in particular, it is a subgroup of $\mathrm{Sym}(V)$.
As we have observed in the Example 1.2.2, $\mathrm{T}(V)$ acts naturally on V.
Furthermore, this action is transitive and regular. In fact, for any $v \in V$:

$$v = 0_V \sigma_v \tag{1.1}$$

where 0_V is the zero vector of V. Thus any vector belongs to the orbit of 0_V and so $0_V \mathrm{T}(V) = V$.

Corollary 1.2.8. *The group $\mathrm{T}(V)$ is always imprimitive, unless $f = 1$ and $d = 1$ ($\mathbb{F}_q = \mathbb{F}_p$). Moreover, a block system is of the form $\{W + v \mid v \in V\}$ for some proper non trivial subspace W of V.*

Proof. This is a direct consequence of Theorem 1.2.7. □

Finally, we introduce some other notations that we will use during this work. In particular, we denote with

- $\mathrm{GL}(V)$ the group of all *linear* permutations of V,
- $\mathrm{AGL}(V)$ the group of all *affine* permutations of V.

For more other details about these two groups see [DM96].

Translation based block ciphers and primitivity.

In this chapter, starting from a brief introduction to block ciphers, we will describe *iterated block ciphers* and then we will examine some of the most important and widely used: AES and PRESENT. After that we will analyze weaknesses, with a focus on the *Paterson attack* and finally we will introduce a new class of iterated block ciphers, called *translation based block cipher* and give some important results about them.

2.1 Introduction to block ciphers.

Cryptography, as defined in [Kob94], is the study of methods of sending messages in disguised form so that only the intended recipients can remove the disguise and read the message. Thus, the main aim of this science is to assure that a private message is not made available to unauthorized people or, in other terms, to guarantee the message *confidentiality.* The message we want to send is called *plaintext* and the disguised message is called *ciphertext.* The *encryption* is the process of converting a plaintext into a ciphertext, and the *decryption* is the reverse.
Formally, we introduce the following definition [BFKR10]:

Definition 2.1.1 (Cryptosystem). *A cryptosystem is a tuple* $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ *where:*

- $\mathcal{P}$ *is the set of all possible plaintexts, called plaintext space.*
- $\mathcal{C}$ *is the set of all possible ciphertexts, called ciphertext space.*
- $\mathcal{K}$ *is the key space, each element* $k \in \mathcal{K}$ *is called key.*
- $\mathcal{E}$ *is the set of injective maps* $f_k : \mathcal{P} \longrightarrow \mathcal{C}$ *indexed by the key space, called encryption maps. Hence, for each* $k \in \mathcal{K}$ *there is an injective map* $f_k : \mathcal{P} \longrightarrow \mathcal{C}$*.*
- $\mathcal{D}$ *is the set of maps* $g_{k'} : \mathcal{C} \longrightarrow \mathcal{P}$ *indexed by the key space, called decryption maps.*
- *for each* $k \in \mathcal{K}$ *(encryption key) there exists a corresponding key* $k' \in \mathcal{K}$ *(decryption key) and a decryption map* $g_{k'} \in \mathcal{D}$ *that is the left inverse of* f_k*, or in other terms:*

$$\forall k \in \mathcal{K} \ \exists k' \in \mathcal{K}, g_{k'} \in \mathcal{D} \text{ such that } g_{k'} \circ f_k = id$$

Each encryption map must be injective in order to guarantee the existence of the left inverse of the encryption map itself, that is the decryption map.

Using this model, we can easily distinguish *symmetric* from *asymmetric* or *public key cryptosystems.* In symmetric cryptosystems, given the encryption key it is easy to find the corresponding decryption key as they are equal. On the other hand, in public key cryptosystems even if the encryption key is known (usually it is public), it is infeasible to compute the decryption one (which is kept secret, known only to the authorized receipt). Since we will focus only on symmetric cryptography, we refer to [Kob94] or [BFKR10] for an introduction to the public key one.

The most common symmetric cryptosystems used in modern cryptography are *stream* and *block ciphers.* In stream ciphers, the ciphertext is obtained combining each plaintext digit with a *pseudorandom* digit stream, called *keystream.* Throughout this study we will analyze only block ciphers and so we refer to [Rue92] for further details on stream ciphers.
On the other hand in block ciphers, given a key, the corresponding encryption function acts on fixed-length groups of plaintext digits, called *blocks.* Thus the first step of the encryption consists of the division of the plaintext into blocks of a certain number of digits, for example n, with $n \geq 1$. Therefore the plaintext of the block cipher becomes the block itself. We will assume in all this work, without loss of generality, that the plaintext space coincides with the ciphertext one, $V := \mathcal{P} = \mathcal{C} = (\mathbb{F}_2)^n$. In this case, given a key $k \in \mathcal{K} = (\mathbb{F}_2)^l$ with $l \geq 1$, the encryption map $f_k \in \mathcal{E}$ is a permutation, so $f_k \in \mathrm{Sym}((\mathbb{F}_2)^n)$ (using the same notation of Section 1.2), and we denote the corresponding decryption function $g_k := f_k^{-1}$.

Now we can give the following definition ([Rim09]):

Definition 2.1.2 (Algebraic block cipher)**.** *An algebraic block cipher is a function f*

$$f : (\mathbb{F}_2)^l \times (\mathbb{F}_2)^n \longrightarrow (\mathbb{F}_2)^n.$$

such that for any $k \in (\mathbb{F}_2)^l$ the function f_k:

$$\begin{array}{rccc} f_k : & (\mathbb{F}_2)^n & \longrightarrow & (\mathbb{F}_2)^n \\ & x & \longmapsto & f(k, x) \end{array}$$

is a permutation of $(\mathbb{F}_2)^n$.

Under these conditions we can also consider a block cipher C as an indexed set of permutations:

$$C := \{f_k \in \mathrm{Sym}((\mathbb{F}_2)^n) \mid k \in (\mathbb{F}_2)^l\}$$

For practical purposes it is important that block ciphers fulfill two requirements ([DR13]):

- *Efficiency*: given the key, the application of the encryption or the decryption map must be as efficient as possible, preferably on a wide range of platforms.
- *Security*: it must be impossible to exploit the internal structure of the cipher.

2.2 Attack scenarios.

Cryptanalysis is a science that study cryptosystems aiming to find weaknesses and vulnerabilities which allow to retrieve the plaintext from the ciphertext without necessarily knowing the key. There are different scenarios, depending on the access the attacker has to the plaintext, ciphertext or other aspects of the cipher itself. Below are listed some of the most common types of attacks:

- *brute-force attack:* this is the simplest kind of attack which consists of checking all possible keys in order to find the correct one. The resources required for a brute-force grow exponentially with increasing key size and so it is very expensive, from the computational point of view.
- *ciphertext-only:* the attacker has access to a sample of ciphertext, without the associated plaintext, and using this, he tries to find information about the plaintext itself. This data is relatively easy to obtain in many scenarios, but in general a successful ciphertext-only attack is difficult since it requires a very large ciphertext sample.
- *known-plaintext:* in this case the attacker knows a sample of ciphertext and the corresponding plaintext. Using them he attempts to deduce the key or some information about that.
- *chosen-plaintext:* the attacker is able to choose a certain quantity of plaintext and the corresponding encrypted ciphertext. Through them he tries to reconstruct the key or part of it. This is also a common scenario since now most of the cryptosystems are implemented into electronic devices or smartcards. Therefore it could be easy for an attacker to use them for the encryption of chosen data. The *Paterson attack* (Section 2.3.3) represents an example of this kind of attack.
- *adaptive-chosen-plaintext:* this is a special case of *chosen-plaintext* attack. In particular in this situation the attacker chooses plaintext samples dynamically, according to results obtained by encryptions made during the attack itself.

2.2.1 Trapdoors.

A trapdoor is a hidden structure of the cipher. The knowledge of this structure allows an attacker to decrypt some ciphertexts or to obtain information on the key. Rijmen and Preneel introduced this concept in [RP97]. In particular they distinguished different kind of trapdoors, depending on information the attacker retrieves from the trapdoor itelf:

- *full trapdoor:* this trapdoor allows an attacker to obtain knowledge of the key.
- *partial trapdoor:* this trapdoor does not necessarily work for all keys and gives an attacker only partial information on the key.

Moreover, always according to [RP97], the trapdoor is *detectable (undetectable):* if it is computationally feasible (infeasible) to find it even if one knows the general form of the trapdoor itself.
Paterson gave an example of full trapdoor in [Pat99]. We will study it in Section 2.3.3.

2.3 Iterated block ciphers.

Modern block ciphers adopt the structure of *iterated block ciphers*, introduced by Shannon in 1949 in his paper [Sha49]. The idea is to build a strong cipher by iterating simple encryption maps, called *round functions*, for a certain number of *rounds*. In each round it is used a different key, called *round key* which has the same length of the plaintext. In particular each round key is obtained through another function, the *key schedule*, whose input is the so-called *master key*.

Formally, if $C = \{f_k \mid k \in (\mathbb{F}_2)^l\}$ is an iterated block cipher with plaintext space $V = (\mathbb{F}_2)^n$, then each f_k is given by the composition of $f_{\phi(k,0)}, f_{\phi(k,1)}, \dots, f_{\phi(k,r)}$, where:

1. $r \geq 1$, and $r + 1$ is the number of rounds[1];
2. $\phi : (\mathbb{F}_2)^l \times \{0, \dots, r\} \longrightarrow (\mathbb{F}_2)^n$ is the key scheduling function, so that $\phi(k, h)$ is the h-th round key given by the master key k;
3. $\forall h \in \{0, \dots, r\}$, $f_{\phi(k,h)} \in \mathrm{Sym}((\mathbb{F}_2)^n)$.

[1] We start from the round zero following the specifics of block ciphers that we will analyze in detail in this chapter.

Figure 2.1 illustrates the general scheme of an iterated block cipher with plaintext p, master key k and ciphertext c.

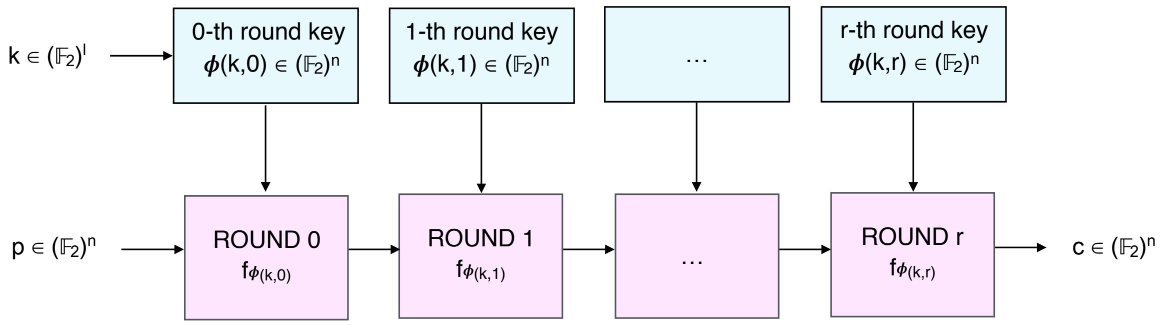

Figure 2.1: Iterated block cipher.

In the design of iterated block ciphers are applied two important concepts: the *diffusion* and *confusion* [Sha49]. Diffusion means that the output bits of the ciphertext should depend on the input bits of the plaintext and the key in a very complex way. In other words, in a cipher with a good diffusion if we change only a bit of the plaintext, then several bits of the ciphertext should change in a pseudorandom manner. The effect of the application of this concept is that the enemy must intercept a huge amount of material to tie down the input structure. On the other hand, confusion refers to making the relationship between the plaintext and the ciphertext as involved as possibile. The main aim of confusion is to make the key very hard to find given a huge amount of pairs plaintext-ciphertext produced using the same key. In practice in each round, the confusion is provided by a non-linear transformation, while the diffusion is given by a permutation (usually a linear or affine transformation) which acts on the whole data. If the round functions achieve good diffusion and confusion, after sufficiently many rounds the block cipher is expected to behave like a random permutation resistant to many attacks.

2.3.1 Some examples of iterated block ciphers: AES and PRESENT.

In this section we will describe in detail two of the most used iterated block ciphers: AES and PRESENT.

AES (Advanced Encryption Standard)

In 1997 the U.S. National Institute of Standards and Technology (NIST) conducted a competition to develop a replacement of the DES, Data Encryption Standard ([Nat77]), which became vulnerable to some attacks. The winner, announced in 2000, was the Rijndael algorithm, developed by J. Daemen and V. Rijmen, destined to become the Advanced Encryption Standard, AES [Nat01]. The AES became effective as a federal government standard later, in 2002.
Next, we will describe in details how AES works following [DR02] and [Ald13].

AES belongs to the class of Rijndael block ciphers. In particular, it has the same structure of Rijndael with the only difference that while Rijndael has variable block and master key length (any multiple of 32 bits with a minimum of 128 bits and a maximum of 256), AES fixes this two parameters.
Precisely, it has a fixed block length of 128 bits hence, following the previous notations $V = (\mathbb{F}_2)^{128}$. Moreover we can see V as the direct sum:

$$V = V_1 \oplus \ldots \oplus V_{16}$$

where $V_i \cong (\mathbb{F}_2)^8$, for $1 \leq i \leq 16$. The master key length l of 128, 192 and 256 bits varies according to the version: `AES-128`, `AES-192` or `AES-256`. Also the number of rounds $r+1$ depends on the version, as reported in the following table:

`AES-128`	$l = 128$	$r = 10$
`AES-192`	$l = 192$	$r = 12$
`AES-256`	$l = 256$	$r = 14$

Each round transformation operates on an intermediate result called *state*, that is a 4×4 matrix in $\mathbb{F}_{2^8}$. In particular, in the encryption, the plaintext $p \in V$ is divided into 16 bytes, $p = p_0 p_1 \ldots p_{15}$, where $p_l \in (\mathbb{F}_2)^8$ for $0 \leq l < 16$. Each byte is then inserted into the state matrix $S = (s_{i,j})_{0 \leq i,j < 4}$ according to:

$$s_{i,j} = p_{i+4j}, 0 \leq i, j < 4 \tag{2.1}$$

Thus:

$$S = \begin{bmatrix} p_0 & p_4 & p_8 & p_{12} \\ p_1 & p_5 & p_9 & p_{13} \\ p_2 & p_6 & p_{10} & p_{14} \\ p_3 & p_7 & p_{11} & p_{15} \end{bmatrix} \tag{2.2}$$

Recall that the key schedule is a function that, given in input a master key returns $r+1$ round keys and that each of them belongs to the $\mathbb{F}_2$-vector space V (see point

(2) of the formalization of an iterated block cipher, Section 2.3). Thus, according to the AES version, the AES-key scheduling function returns respectively $11, 13$ or 15 round keys, that are vectors of $(\mathbb{F}_2)^{128}$ (for futher details see [DR02]).
Now, denoted as $k^{(h)} := k_0^{(h)} k_1^{(h)} \dots k_{15}^{(h)}$ the h-th round key, with $0 \leq h \leq r$ we define the matrix $K = (k_{i,j})_{0 \leq i,j < 4}$ as :

$$K = \begin{bmatrix} k_0^{(h)} & k_4^{(h)} & k_8^{(h)} & k_{12}^{(h)} \\ k_1^{(h)} & k_5^{(h)} & k_9^{(h)} & k_{13}^{(h)} \\ k_2^{(h)} & k_6^{(h)} & k_{10}^{(h)} & k_{14}^{(h)} \\ k_3^{(h)} & k_7^{(h)} & k_{11}^{(h)} & k_{15}^{(h)} \end{bmatrix} \tag{2.3}$$

After the encryption each byte of the ciphertext $c = c_0 c_1 \dots c_{15}$ is extracted from the state according to:

$$c_i = s_{i \bmod 4, \lfloor h/4 \rfloor}, 0 \leq i < 16. \tag{2.4}$$

where $\lfloor h/4 \rfloor$ denotes the largest integer less or equal than $h/4$.

In the same way for the decryption, the ciphertext $c = c_0 c_1 \dots c_{15}$ is inserted into the state matrix according to:

$$s_{i,j} = c_{i+4j}, 0 \leq i, j < 4. \tag{2.5}$$

and at the end the plaintext is derived from the state following:

$$p_i = s_{i \bmod 4, \lfloor h/4 \rfloor}, 0 \leq i < 16. \tag{2.6}$$

Encryption. Fixed a master key $k \in (\mathbb{F}_2)^l$ and denoted with
$\phi : (\mathbb{F}_2)^l \times \{0, \dots, r\} \longrightarrow V$ the AES-key scheduling function, the encryption function f_k, is given by the composition of $r + 1$ round functions $f_{\phi(k,0)}, \dots, f_{\phi(k,r)}$.
Specifically:

$$f_{\phi(k,h)} = \begin{cases} \sigma_{\phi(k,0)} & \text{if } h = 0 \\ \gamma \lambda \sigma_{\phi(k,h)} & \text{if } 1 \leq h \leq r - 1 \\ \gamma \tilde{\lambda} \sigma_{\phi(k,r)} & \text{if } h = r \end{cases}$$

where:

1. $\sigma_{\phi(k,h)}$ is a translation of the h-th round key $\phi(k, h)$ for $0 \leq h \leq r$. Following notations of the Section 1.2, $\sigma_{\phi(k,h)} \in \mathrm{T}(V)$.

2. $\gamma \in \mathrm{Sym}(V)$ is a non-linear transformation.

3. $\lambda \in \mathrm{GL}(V)$ is a linear map represented in practice as the composition of two transformations: the *MixColumns* and *ShiftRows*, that we will describe later;

4. $\tilde{\lambda} \in \mathrm{GL}((\mathbb{F}_2)^{128})$ is a linear map with only the *MixColumns* transformation.

Now we describe in detail what happens in each round of this cipher.

- ROUND 0: this first round consists only of the translation $\sigma_{\phi(k,0)}$. In particular, if $S = (s_{i,j})_{0\leq i,j<4}$ is the state matrix obtained from the plaintext according to (2.1) and $K = (k_{i,j})_{0\leq i,j<4}$ the matrix obtained as in (2.3) from the first round key $\phi(k, 0)$, then the new state is given by:

$$s'_{i,j} := s_{i,j} \oplus k_{i,j}, 0 \leq i, j, < 4. \tag{2.7}$$

  where in this case $\oplus$ denotes the bitwise XOR operation.

- ROUND h, for $1 \leq h \leq r - 1$: since in these r rounds we apply the same encryption function we can omit the round index.

  The first transformation applied in each round is γ given by the parallel application of some permutations $\gamma_i \in \mathrm{Sym}(V_i)$ for $1 \leq i \leq 16$. Any γ_i is equal to a fixed permutation $\overline{\gamma}$ where $\overline{\gamma} = fg$. In particular, f is the patched inversion over $\mathbb{F}_{2^8}$ (Example 1.1.2) and g is an invertible affine transformation, $g \in \mathrm{AGL}(V_i)$ (using the same notations of Section 1.2), defined as:

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} \longmapsto \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \tag{2.8}$$

  Thus, the new state $S' = (s'_{i,j})_{0\leq i,j<4}$ after the application of this transformation is given by:

$$s'_{i,j} := s_{i,j}\overline{\gamma}, 0 \leq i, j, < 4. \tag{2.9}$$

  As we can see, g has a very simple algebraic expression. This could allow algebraic manipulations that can be used to mount attacks such as interpolation ones [DR02]. Therefore J. Daemen and V. Rijndael chose to build each γ_i, for $1 \leq i \leq 16$ as the composition of f with g. In fact the affine transformation has no impact on the non-linearity properties of the patched inversion but allows the entire γ to have a more complex algebraic expression, which prevents these kind of attacks.

After this transformation we apply λ, a linear transformation given by the composition of two operations: ShiftRows and MixColumns.

The ShiftRows is a byte transposition that cyclically shifts the rows of the state over different offsets. In details, the first row is not changed, the second is shifted of one position to the left, the third of two and the fourth of three. If $S' = (s'_{i,j})_{0\leq i,j<4}$ is the new state after this operation then:

$$S = \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} \xrightarrow{\texttt{ShiftRows}} S' = \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,1} & s_{1,2} & s_{1,3} & s_{1,0} \\ s_{2,2} & s_{2,3} & s_{2,0} & s_{2,1} \\ s_{3,3} & s_{3,0} & s_{3,1} & s_{3,2} \end{bmatrix} \tag{2.10}$$

Formally:

$$s'_{i,j} = s_{i,i+j \bmod 4}, 0 \leq i, j, < 4. \tag{2.11}$$

J. Daemen and V. Rijndael underlined that the choice of different offsets, respectively 0, 1, 2 and 3 was made to guarantee an optimal diffusion, which provides resistance against differential and linear attacks.

The MixColumns is a linear operation which acts on the state column by column. Each column of the state is considered as a polynomial over $\mathbb{F}_{2^8}$ and multiplied modulo x^4+1 with a fixed polynomial $c(x)$:

$$c(x) = (\alpha+1)x^3 + x^2 + x + \alpha, \tag{2.12}$$

where $\alpha \in (\mathbb{F}_2)^8$ is such that $\alpha^8 = \alpha^4 + \alpha^3 + \alpha + 1$. If we denote with $s_j(x) \in \mathbb{F}_{2^8}[x]$ the polynomial corresponding to the j-th column of the state then the polynomial $s'_j(x) \in \mathbb{F}_{2^8}[x]$ of the new state after the MixColumns operation is given by the multiplication:

$$s'_j(x) = c(x)s_j(x) \bmod x^4 + 1, 0 \leq j < 4. \tag{2.13}$$

As described in [Ald13] the modular multiplication with a fixed polynomial can be expressed as a matrix multiplication. So the (2.13) becomes:

$$\begin{bmatrix} s'_{0,j} \\ s'_{1,j} \\ s'_{2,j} \\ s'_{3,j} \end{bmatrix} = \begin{bmatrix} \alpha & \alpha+1 & 1 & 1 \\ 1 & \alpha & \alpha+1 & 1 \\ 1 & 1 & \alpha & \alpha+1 \\ \alpha+1 & 1 & 1 & \alpha \end{bmatrix} \begin{bmatrix} s_{0,j} \\ s_{1,j} \\ s_{2,j} \\ s_{3,j} \end{bmatrix}, 0 \leq j < 3. \tag{2.14}$$

Even the choice of coefficients of the polynomial c was made to guarantee an optimal diffusion, as remarked in [DR02].

After the application of γ and λ, there is the translation $\sigma_{\phi(k,h)}$ that is the bitwise XOR of all bytes of the state with the h-th round key $\phi(k,h)$ $(0 \leq h \leq r)$.

- ROUND r: this last round is is slightly different with respect to the others. It differs from the linear transformation part, $\tilde{\lambda}$, which is made only of the MixColumns.

Decryption. The algorithm for decription can be found in a straightforward way by using the inverses of each round function and reversing the order of application. Hence following the previous notations, round functions of the decryption are $f'_0, \dots, f'_r$:

$$f'_h = (f_{\phi(k,r-h)})^{-1}, 0 \leq h \leq r. \tag{2.15}$$

and in particular:

$$f'_h = \begin{cases} \sigma_{\phi(k,r)}\tilde{\lambda}^{-1}\gamma^{-1} & \text{if } h = 0, \\ \sigma_{\phi(k,h)}\lambda^{-1}\gamma^{-1} & \text{if } 1 \leq h \leq r-1, \\ \sigma_{\phi(k,0)} & \text{if } h = r. \end{cases} \tag{2.16}$$

where:

- $\sigma_{\phi(k,h)} \in \mathrm{T}(V)$, as in the encryption.
- λ^{-1} is the inverse of the linear transformation λ. It is given by the composition of the inverses of the MixColumns and ShiftRows, called *InvMixColumns* and *InvShiftRows*.

  Referring to the same notations used for MixColumns and ShiftRows, InvMixColumns consists of the multiplication modulo $x^4 + 1$ with the inverse $d(x)$ of $c(x)$:

$$d(x) = (\alpha^3 + \alpha + 1)x^3 + (\alpha^3 + \alpha^2 + 1)x^2 + (\alpha^3 + 1)x + (\alpha^2 + \alpha^2 + \alpha), \tag{2.17}$$

  which corresponds to the matrix multiplication:

$$\begin{bmatrix} s'_{0,j} \\ s'_{1,j} \\ s'_{2,j} \\ s'_{3,j} \end{bmatrix} = \begin{bmatrix} \alpha^3+\alpha^2+\alpha & \alpha^3+\alpha+1 & \alpha^3+\alpha^2+1 & \alpha^3+1 \\ \alpha^3+1 & \alpha^3+\alpha^2+\alpha & \alpha^3+\alpha+1 & \alpha^3+\alpha^2+1 \\ \alpha^3+\alpha^2+1 & \alpha^3+1 & \alpha^3+\alpha^2+\alpha & \alpha^3+\alpha+1 \\ \alpha^3+\alpha+1 & \alpha^3+\alpha^2+1 & \alpha^3+1 & \alpha^3+\alpha^2+\alpha \end{bmatrix} \begin{bmatrix} s_{0,j} \\ s_{1,j} \\ s_{2,j} \\ s_{3,j} \end{bmatrix}, \tag{2.18}$$

  for $0 \leq j < 3$.

On the other hand, the InvShiftRows corresponds to the cyclical shift of each row of the state of 0, 1, 2 and 3 positions to the right. Formally:

$$s'_{i,j} = s_{i,j-i \bmod 4}, 0 \le i,j < 4. \tag{2.19}$$

- γ^{-1} is the inverse of the non-linear transformation γ. In this case, any γ_i^{-1}, for $1 \le i \le 16$, is equal to $\overline{\gamma}^{-1} = g^{-1}f^{-1} = g^{-1}f$, since the inverse of the patched inversion is the patched inversion itself, $f^{-1} = f$. Moreover g^{-1} is given by:

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} \longmapsto \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.20}$$

PRESENT.

PRESENT is a block cipher developed in 2007 by Bogdanov et al. [AKL$^+$07]. This cipher was not designed for a wide-spread use since for almost all block cipher applications the AES is an excellent and preferred choice. However AES is not suitable for constraint environments such as sensor networks and embedded devices. For these purposes, students and researchers of the Orange Labs, Ruhr University of Bochum and Technical University of Denmark proposed this new cipher which is a lightweight one or, in other terms, able to run on devices with very low computing power.
Let us describe this cipher in details.

PRESENT works on plaintexts of 64 bits $V = (\mathbb{F}_2)^{64}$ and

$$V = V_1 \oplus \ldots \oplus V_{16}$$

where $V_i \in (\mathbb{F}_2)^4$ for $1 \le i \le 16$. The number of rounds is fixed to 32 ($r = 31$) and according to the key length l there are two versions:

PRESENT-80	$l = 80$	$r = 31$
PRESENT-128	$l = 128$	$r = 31$

Fixed a master key $k \in (\mathbb{F}_2)^l$ and denoted with $\phi : (\mathbb{F}_2)^l \times \{0, \ldots, r\} \longrightarrow V$ the PRESENT- key schedule (for details see [AKL$^+$07]), round functions $f_{\phi(k,0)}, \ldots, f_{\phi(k,r)}$ are given by:

$$f_{\phi(k,h)} = \begin{cases} \gamma\lambda\sigma_{\phi(k,h)} \text{ if } & 0 \le h \le r \\ \sigma_{\phi(k,r)} \text{ if } & h = r. \end{cases} \tag{2.21}$$

where:

- $\sigma_{\phi(k,h)} \in \mathrm{T}(V)$, is a translation of the h-th round key $\phi(k,h)$, for $0 \leq h \leq r$.
- $\gamma \in \mathrm{Sym}(V)$ is a non-linear transformation given by the parallel application of $\gamma_i \in \mathrm{Sym}(V_i)$, for $1 \leq i \leq 16$. In particular any γ_i coincides with a fixed permutation $\overline{\gamma}$, whose action in hexadecimal notation is given by the following table.

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\overline{\gamma}(x)$	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

- $\lambda \in \mathrm{GL}(V)$ is a linear transformation. The following table shows how it works: in particular the i-th bit of the current state is moved to bit position denoted with $\lambda(i)$.

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\lambda(i)$	0	16	32	48	1	17	33	49	2	18	34	50	3	19	35	51
i	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
$\lambda(i)$	4	20	36	52	5	21	37	53	6	22	38	54	7	23	39	55
i	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
$\lambda(i)$	8	24	40	56	9	25	41	57	10	26	42	58	11	27	43	59
i	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
$\lambda(i)$	12	28	44	60	13	29	45	61	14	30	46	62	15	31	47	63

Even in this case the decryption is obtained by the reverse application of the inverse of each encryption function.

2.3.2 Group generated by the round functions.

In 1975 Coppersmith and Grossman studied a particular set of functions and their possible integration into a block cipher and analyzed the permutation group generated by them [CG75] opening the way to many researchers that later studied the group generated by permutations of block ciphers aiming to find properties which could reveal weaknesses of the cipher itself, as proved in [KRS88], [Pat99] and [CS15].
In particular, since the most used block cipher are iterated ones (see Chapter 2.3), they focused on the group generated by the round functions of these kind of ciphers. Now, we describe this group.

Let $C = \{f_k \mid k \in \mathcal{K}\}$ be an iterated block cipher with plaintext space $V = (\mathbb{F}_2)^n$ and $n \geq 1$. We denote with $\Gamma = \Gamma(C)$, the subgroup of $\mathrm{Sym}(V)$ generated by all permutations f_k:

$$\Gamma = \Gamma(C) = \langle f_k \mid k \in \mathcal{K} \rangle. \tag{2.22}$$

As we have seen in Section 2.3, each encryption function is given by the composition of a certain number (equal to the number of rounds) of round functions. Since any round function depends on the key schedule, the whole encryption map depends on the key schedule itself. This is why the study of the group Γ appears a difficult problem.

Hence, we will analyze an "easier" permutation group related to Γ. In detail, assuming that $r \geq 1$ is the number of rounds and that $f_{k,1}, \ldots, f_{k,r}$ are the r round functions[2] related to the encryption map f_k and fixed a round h, with $1 \leq h \leq r$, we define the group:

$$\Gamma_h = \Gamma_h(C) = \langle f_{k,h} \mid k \in \mathcal{K} \rangle. \tag{2.23}$$

Thus we can define a new subgroup of $\mathrm{Sym}(V)$ containing Γ, that is:

$$\Gamma_\infty = \Gamma_\infty(C) = \langle \Gamma_h \mid 1 \leq h \leq r \rangle, \tag{2.24}$$

called *group generated by the round functions.*

Kalinski et al. in [KRS88] proved that if this group is too small, then the cipher is vulnerable to certain kind of attacks.

On the other hand, if this group has a known structure, for example it is a subgroup of $\mathrm{AGL}(V)$, it is possible to embed a trapdoor (Section 2.2.1) into the cipher as shown in [CS15].

In conclusion, Paterson in [Pat99], proved that if the group generated by the round functions is imprimitive (Definition 1.2.5), then it is possible to attack the cipher as we will see in the following paragraph.

2.3.3 Paterson attack.

The Paterson attack [Pat99] is a chosen-plaintext attack defined on an iterated block cipher. It is independent on the number of rounds and works, as we have previously remarked, if we suppose that the group generated by the round functions is imprimitive. Now, let us describe the attack in detail. We will use notations of the previous section.

[2] Precisely round functions are $f_{\phi(k,1)}, \ldots, f_{\phi(k,r)}$ where $\phi : \mathcal{K} \times \{1, \ldots, r\} \longrightarrow V$ is the key schedule. In this case we put $f_{k,h} := f_{\phi(k,h)}$ for $1 \leq h \leq r$, omitting the key schedule function in order to simplify notations.

Let C be an iterated block cipher with plaintext space V such that Γ_∞ is imprimitive. Let $k \in \mathcal{K}$ be a key and $f_k \in \Gamma_\infty$ the corresponding encryption function. Assume also that $B_1, \dots, B_d$, with $d \geq 1$, is a block system for the group Γ_∞. It is important to remark that this attack works only if we now an efficient algorithm, called *block sieving* which, given a vector $v \in V$ allows to obtain the unique block containing it.

Following the description of the attack given in the article [ACMS16] we will distinguish two steps:

1. *Preprocessing performed ones per key:* chosen one plaintext m_i in each block B_i $(1 \leq i \leq d)$, with the application of the encryption function we get the corresponding ciphertext c_i. Using the block sieving we obtain B_j such that $c_i \in B_j$, for a certain $j \neq i$, $1 \leq j \leq d$. Therefore the effect of the encryption function f_k on blocks is determined by:

$$c_i = m_i f_k \in B_j \Longrightarrow B_i f_k = B_j$$

   where this implication follows from the imprimitivity of Γ_∞.

2. *Real-time processing:* given any ciphertext c, at first we use the block sieving to compute the block B_t such that $c \in B_t$ for a certain t, $1 \leq t \leq d$. Then it is possible to find the corresponding plaintext m by examining the block $B_t f_k^{-1}$.

In other terms, through the preprocessing step the attacker, starting from d couples plaintext-ciphertext, collects information about blocks after the application of the encryption function. When the attacker intercepts a ciphertext $c \in B_t$ (in the real time processing phase), he tries to get the related plaintext by the study of the block $B_t f_k^{-1}$ which he knows for preprocessing step.
The preprocessing costs l encryptions. For any intercepted ciphertext, the search of the related plaintext is limited to a block, whose size is $\frac{|V|}{l}$, requiring at most $\frac{|V|}{l}$ encryptions.

2.4 Translation based block ciphers and primitivity.

In 2009 Caranti, Dalla Volta and Sala introduced in [CVS09] a new class of iterated block ciphers, called *translation based block ciphers*, which includes some well known ciphers such as AES, *SERPENT* (see [BAK98] for more details) and PRESENT. For this new class they also proved the primitivity of the group generated by the round functions by only checking some properties on the transformation working within the cipher. However in practice, we will see that these properties are verified only by AES and SERPENT.

Recently, in 2016 some researchers of universities of Trento and Salerno, in [ACTT15], extended some hypotheses, including also the PRESENT cipher and some other lightweight block ciphers.

In this section we will describe in detail this new category of block ciphers and then we will state primitivity theorems.
At first, we introduce notations and some important concepts following both articles [CVS09] and [ACTT15].
Let $C = \{f_k \mid k \in \mathcal{K}\}$ be an iterated block cipher with plaintext space $V = (\mathbb{F}_2)^n$ and $n = mp$ with $m, p > 1$. We can see the $\mathbb{F}_2$-vector space V as the direct sum:

$$V = V_1 \oplus \ldots \oplus V_m, \tag{2.25}$$

where each $V_i \cong (\mathbb{F}_2)^p$ for $1 \leq i \leq m$. Any $v \in V$ can be uniquely written as $v = v_1 + \ldots + v_m$ with $v_i \in V_i$ and $1 \leq i \leq m$.

Definition 2.4.1 (Bricklayer transformation). *A function $\gamma \in \mathrm{Sym}(V)$ is called bricklayer transformation if for any $i \in \{1, \ldots, m\}$ there exists $\gamma_i \in \mathrm{Sym}(V_i)$, called brick, such that:*

$$v\gamma = v_1\gamma_1 + \ldots + v_m\gamma_m.$$

In symmetric cryptography the permutation γ is traditionally called *parallel S-box* and each γ_i simply *S-box*.

Definition 2.4.2 (Wall). *Any non trivial proper subspace V' of V such that:*

$$V' = V_{i_1} \oplus \ldots \oplus V_{i_j},\ 1 \leq i_1 < \ldots < i_j \leq m$$

is called wall.

Moreover, we call any linear map $\lambda \in \mathrm{GL}(V)$ used in composition with a bricklayer transformation, *mixing layer*.

Example 2.4.3.

1. In AES (Section 2.3.1) the bricklayer transformation is the non-linear function $\gamma \in \mathrm{Sym}((\mathbb{F}_2)^{128})$ and each brick $\gamma_i \in (\mathbb{F}_2)^8$, for $1 \leq i \leq 16$, is equal to the composition of the patched inversion with an affine transformation. On the other hand the mixing layer is the linear transformation denoted as λ which belongs to $\mathrm{GL}((\mathbb{F}_2)^{128})$. As we have seen it is given by the composition of ShiftRows and MixColumns except for the last round, for which consists only of the MixColumns part.

2. In PRESENT (Section 2.3.1) the bricklayer transformation, is the non-linear transformation γ. In particular, any brick γ_i, for $1 \leq i \leq 16$, is a fixed permutation of $(\mathbb{F}_2)^4$. On the other hand the mixing layer is the linear transformation, called λ, which belongs to $\mathrm{GL}((\mathbb{F}_2)^{64})$.

We introduce two definitions that we will use later:

Definition 2.4.4 (Proper mixing layer). *Any linear transformation $\lambda \in \mathrm{GL}(V)$ is a proper mixing layer if no wall V' is such that $V'\lambda = V'$, or in other words if no wall V' is λ-invariant.*

Definition 2.4.5 (Strongly proper mixing layer). *Any linear permutation $\lambda \in \mathrm{GL}(V)$ is a strongly proper mixing layer if there are no walls V' and V'' of V such that $V'\lambda = V''$.*

As the name suggests, the role of a good mixing layer, from a cryptographic point of view, consists of mixing all bits of the state as much as possible. As a matter of fact, a proper mixing layer destroys in a certain sense the structure of any wall, making it hard for an attacker to retrieve the initial state given the state after the application of the mixing layer itself.

Now we are ready for the most important definition of the chapter:

Definition 2.4.6 (Translation based block ciphers). *An iterated block cipher C is a translation based block cipher (tb for short) if:*

1. *each f_k is the composition of a finite number, say $r \geq 1$, of round functions $f_{k,h}$ with $k \in \mathcal{K}$ and $1 \leq h \leq r$, such that each $f_{k,h}$ can be written as the composition $\gamma_h \lambda_h \sigma_{\phi(k,h)}$ of three permutations of V, where*
   - *γ_h is a bricklayer transformation not depending on k and $0\gamma_h = 0$,*
   - *λ_h is a mixing layer not depending on k,*
   - *$\phi : \mathcal{K} \times \{1, \dots, r\} \longrightarrow V$ is the key scheduling function;*
2. *at least one round is proper, that is,*
   - *λ_h is a proper mixing layer for some h,*
   - *the map $\phi_h : \mathcal{K} \longrightarrow V$ given by $k \longmapsto \phi(k, h)$ is surjective.*

Remark 2.4.7. We can always assume that the bricklayer transformation of each round, γ_h for $1 \leq h \leq r$, sends 0 in 0, (as in point (1) of the definition) since we can possibly add $0\gamma_h$ to the round key of the previous round.

From the AES description given in Section 2.3.1 we can easily see that AES itself verifies point (1) of the definition of tb ciphers. Moreover we have the following result:

Proposition 2.4.8. *The AES-mixing layer is proper.*

Proof. For `AES-128` see the proof of Lemma 3.7 of [CVS09]. The same holds for the other two versions. □

Therefore since this proposition proves that AES verifies also point (2) of the definition of tb ciphers, we can conclude that AES is a tb cipher itself.

We can easily prove that the PRESENT is a tb cipher too. In particular, in this case the fact that the mixing layer is proper can be easily deduced from the definition of the mixing layer itself.

Finally, also the SERPENT is a tb cipher, as remarked in [CVS09].

Let us return to tb ciphers in general. Recall definitions of differential-uniformity (Definition 1.1.3) , weakly differential-uniformity (Definition 1.1.4) and strongly anti-invariance (Definition 1.1.6, point (2)) given for vectorial Boolean functions, introduced and discussed in Section 1.1.1.

Now we are ready to state the following important primitivity theorem [CVS09]:

Theorem 2.4.9. *Let C be a translation based block cipher with a proper round h and $1 \leq r < m$. If any brick of γ_h is:*

1. *weakly 2^r-uniform* and
2. *strongly r-anti-invariant,*

then $\Gamma_h(C)$ is primitive and hence so is $\Gamma_\infty(C)$.

Proof. See the proof of Theorem 4.4 of [CVS09]. □

We have two important corollaries:

Corollary 2.4.10. *Any typical round h of the* AES *cipher verifies the hypotheses of Theorem 2.4.9. As a consequence, both $\Gamma_h(\mathrm{AES})$ and $\Gamma_\infty(\mathrm{AES})$ are primitive.*

Proof. See the proof of Corollary 4.6 of [CVS09]. □

Corollary 2.4.11. *Any typical round h of the* SERPENT *cipher satisfies the hypothesis of Theorem 2.4.9. Therefore, both* $\Gamma_h(\text{SERPENT})$ *and* $\Gamma_\infty(\text{SERPENT})$ *are primitive.*

Proof. See the proof of Corollary 4.7 of [CVS09]. □

In particular bricklayer transformations of AES and SERPENT are weakly 2-uniform and strongly 1-anti-invariant, thus they verifies hypotheses of Theorem 2.4.9 with $r = 1$.

On the other hand by a computer check it is possible to see that the bricklayer transformation of the PRESENT cipher is 4-uniform and thus weakly 4-uniform (Remark 1.1.5) and strongly 1-anti-invariant, as underlined in [ACTT15]. Hence it does not verify hypotheses of the Theorem 2.4.9.
For these purposes, as we have remarked at the beginning of this section, Aragona, Calderini, Tortora and Tota in [ACTT15] introduced another primitivity theorem using a stronger hypotheses with respect to the weakly differential uniformity and relaxing the strongly anti-invariance one. As a matter of the fact, we have:

Theorem 2.4.12. *Let C be a translation based block cipher with a proper round h. Suppose that for some $1 < r < m$ each brick of γ_h is:*

1. *2^r-uniform* and
2. *strongly $r - 1$-anti-invariant,*

then $\Gamma_h(C)$ is primitive and hence so is $\Gamma_\infty(C)$.

Proof. See the proof of Theorem 4.1 of [ACTT15]. □

Therefore, with these new hypotheses we can deduce that the group of round functions of the PRESENT is also primitive.

The primitivity of the group generated by the round functions is a very important cryptographic requirements: it allows to avoid the Paterson attack but not only.
In fact using the O'Nan-Scott classification [Cam99] of primitivity groups and using the same hypotheses of the Theorem 2.4.9 adding only the assumption that the mixing layer is strongly proper (see Definition 2.4.5) and another condition on the bricklayer transformation, in [CDS09], the authors proved that the group generated by the round functions is the alternating or the symmetric group.
AES and SERPENT satisfy also these conditions, thus their group generated by the round functions is the alternating group [CDS09].

Even in this case PRESENT and some other lightweight ciphers do not satisfy hypotheses added in [CDS09]. However in [ACTT15], authors added some group theoretical assumption to avoid cryptographic conditions not verified by PRESENT and the others lightweight block ciphers and then proved that this assumption is implied by the hypotheses sufficient to prove the primitivity of the group. In other terms, for these ciphers, in order to prove that the group generated by the round functions is the symmetric or the alternating one it is enough to check the same conditions used to show its primitivity. Further details about these topics are discussed in the Appendix A.1.

These results are particularly important since allow to exclude some algebraic weaknesses of the cipher itself, as already discussed in Section 2.3.2. In particular, recall that if the group generated by the round functions is small or has a known structure, for example it is a subgroup of $\mathrm{AGL}(V)$ (or a conjugate of it), then it would be easy to break the cipher ([KRS88], [CS15]). But if Γ_∞ turns out to be the alternating or the symmetric group, this possibility is automatically avoided.

Generalized translation based block cipher and primitivity.

In this chapter we will introduce new results. In particular, we define a new class of block ciphers, which is a generalization of the translation based ones (Section 2.4.6), called *generalized translation based block ciphers.* Moreover we will prove the primitivity of the group generated by the round functions of this new category of ciphers.

3.1 Generalized translation based block ciphers.

We start fixing notations and introducing some important definitions.
Recall that if $C = \{f_k \mid k \in \mathcal{K}\}$ is an iterated block cipher with plaintext space $V = (\mathbb{F}_2)^n$ and $n = mp$ with $m, p > 1$ then we can see the $\mathbb{F}_2$-vector space V as the direct sum:

$$V = V_1 \oplus \ldots \oplus V_m \tag{3.1}$$

where $V_i \cong (\mathbb{F}_2)^p$, for $1 \leq i \leq m$. In particular any vector $v \in V$ can be uniquely written as $v = v_1 + \ldots + v_m$ with $v_i \in V_i$, for $1 \leq i \leq m$.
We will also denote the *i-th projection* with respect to the decomposition (3.1):

$$\begin{array}{rccc} \pi_i : & V & \longrightarrow & V_i \\ & v & \longmapsto & v_i \end{array}$$

for $1 \leq i \leq m$.
On the other hand, let $W = (\mathbb{F}_2)^t$ be another $\mathbb{F}_2$-vector space, with $t \geq n$ and $t = ml$, with $l > 1$. We can also decompose W as the direct sum:

$$W = W_1 \oplus \ldots \oplus W_m$$

where $W_i \cong (\mathbb{F}_2)^l$, for $1 \leq i \leq m$.

Now we are ready for the following definition.

Definition 3.1.1 (bricklayer transformation). *A function $\gamma : V \longrightarrow W$ is called bricklayer transformation if there exists $\gamma_i : V_i \longrightarrow W_i$, called brick, for $1 \leq i \leq m$, such that for any $v \in V$:*

$$v\gamma = v_1\gamma_1 + \dots v_m\gamma_m.$$

Furthermore, observe that if $V = W$, we have exactly the Definition 2.4.1. Thus, this new definition is a generalization of the previous one.
As remarked in Section 2.4, in cryptography γ is usually called parallel S-box and each γ_i simply S-box.

Given a linear transformation $\lambda : W \longrightarrow V$, we denote with:

$$v\lambda^{-1} := \{w \in W \mid w\lambda = v\},$$

and

$$U\lambda^{-1} := \{w \in W \mid w\lambda \in U\}, \tag{3.2}$$

respectively the *preimage of a vector* $v \in V$ and the *preimage of a subset* $U \subseteq V$. For the linearity of λ follows that $v\lambda^{-1}$ and $U\lambda^{-1}$ are both vector subspaces of W.

Furthermore, we denote with $\ker \lambda$ the *kernel* of the linear transformation λ and with $\mathrm{Im}(\lambda)$ its image. In other words:

$$\ker \lambda := 0_V\lambda^{-1} = \{w \in W \mid w\lambda = 0_V\}$$

and

$$\mathrm{Im}(\lambda) := \{w\lambda \mid w \in W\}.$$

Remark 3.1.2. We observe that for any subspace U of V,

$$U\lambda^{-1} + \ker \lambda = U\lambda^{-1}.$$

where $U\lambda^{-1} + \ker \lambda = \{w + w' \mid w \in U\lambda^{-1}, w' \in \ker \lambda\}$.
In fact, in general we have that $U\lambda^{-1} \subseteq U\lambda^{-1} + \ker \lambda$. In order to prove the other inclusion let us consider $w' \in U\lambda^{-1}$ and $w'' \in \ker \lambda$. Since $\ker \lambda \subseteq U\lambda^{-1}$ and $U\lambda^{-1}$ is a vector subspace of W, then $w' + w'' \in U\lambda^{-1}$, therefore $U\lambda^{-1} + \ker \lambda \subseteq U\lambda^{-1}$.

Recall from the Definition 2.4.2 that a wall of the $\mathbb{F}_2$-vector space V is a non-trivial subspace of V given by the direct sum of some of the V_i, for $1 \leq i \leq m$. The same holds for walls of W. Furthermore, as in Section 2.4, we call mixing layer any linear function used in composition with a bricklayer transformation.

Now we are ready for the following definition:

Definition 3.1.3 (proper mixing layer). *A linear transformation $\lambda : W \longrightarrow V$ is a proper mixing layer if for any nontrivial wall $W' = \bigoplus_{i \in I} W_i$ of W and $V' = \bigoplus_{i \in I} V_i$ of V, where $I \subset \{1, \dots, m\}$, then*

$$V'\lambda^{-1} \not\subset W' + \ker \lambda.$$

In other terms, if $\pi : W \longrightarrow W/\ker(\lambda)$ is the *canonical projection of* W onto $W/\mathrm{ker}(\lambda)$, λ is proper if it does not exist any non-trivial wall $W' = \bigoplus_{i \in I} W_i$ of W and $V' = \bigoplus_{i \in I} V_i$ of V (where $I \subset \{1, \dots, m\}$), such that $\lambda(\pi(W')) = V'$.
Moreover we observe that if $V = W$ and $\lambda \in \mathrm{GL}(V)$ then $\ker \lambda = \{0_W\}$ and so the previous definition of proper mixing layer coincides with the Definition 2.4.4.

Even in this case the role of a proper mixing layer is to make it hard for an attacker to retrieve the initial state given the state after the application of the mixing layer itself. In fact such a mixing layer destroys the structure of a wall of W, not sending the related coset into the corresponding wall of V.

Recall from Section 2.3 that since C is an iterated block cipher, fixed a master key $k \in \mathcal{K}$, each encryption map f_k is given by the composition of $r \geq 1$ round functions $f_{k,1}, \dots, f_{k,r} \in \mathrm{Sym}(V)$.
Now we are ready for the main definition of our work.

Definition 3.1.4 (generalized translation based block cipher). *An iterated block cipher C is generalized translation based if*

1. *each round function $f_{k,h}$, with $1 \leq h \leq r$ and $k \in \mathcal{K}$, is determined by the composition $\gamma_h \lambda_h \sigma_{\phi(k,h)}$ where:*

   - *$\gamma_h : V \longrightarrow W$ is an injective bricklayer transformation, independent on k, and such that $0_V \gamma_h = 0_W$,*
   - *$\lambda_h : W \longrightarrow V$ is a surjective linear transformation, independent on k,*
   - *there is a one-to-one correspondence between $W/\ker \lambda_h$ and $\mathrm{Im}(\gamma_h)$. In other words, for any $v \in V$:*

$$|\mathrm{Im}(\gamma_h) \cap (v\lambda_h^{-1} + \ker \lambda_h)| = 1 \tag{3.3}$$

   - *$\phi : \mathcal{K} \times \{1, \dots, r\} \longrightarrow V$ is the key scheduling function, so that $\phi(k, h)$ is the h-th round key, given by the master key k.*

2. *at least one round $h \in \{1, \dots, r\}$ is proper, i.e.*

- λ_h *is a proper mixing layer,*
- *the map* $\phi_h : \mathcal{K} \longrightarrow V,\ k \longmapsto \phi(k,h)$ *is surjective.*

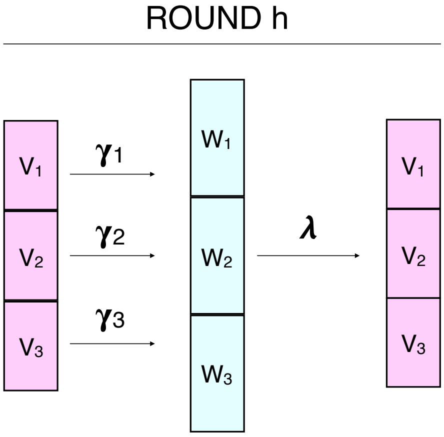

Figure 3.1: partial h-th round function of a generalized tb cipher with $1 \leq h \leq r$ and $m = 3$.

For the rest of this chapter, fixed a round index $1 \leq h \leq r$, we will omit h and denote by ρ the composition $\gamma\lambda$.

Remark 3.1.5. As already observed in Remark 2.4.7, also in this general case we can always assume that the bricklayer transformation of each round sends 0_V in 0_W (as in point (1) of the definition) since we can possibly add $0_V\gamma$ to the round key of the previous round.

Remark 3.1.6. By the hypothesis of the Definition 3.1.4 follows that

1. the function ρ is bijective.

   Since $\rho : V \longrightarrow V$, and V is finite, it is sufficient to prove only its injectivity. Given $v_1, v_2 \in V$ such that $v_1\gamma\lambda = v_2\gamma\lambda$ then $v_1\gamma, v_2\gamma \in V\lambda^{-1}$. Thus, there exists $v' \in V$ such that $v_1\gamma, v_2\gamma \in v'\lambda^{-1} \subseteq v'\lambda^{-1} + \ker\lambda$. From (3.3) follows that there is only an element in $\mathrm{Im}(\gamma)$ which also belongs to $v'\lambda^{-1} + \ker\lambda$. Therefore $v_1\gamma = v_2\gamma$, and since γ is injective $v_1 = v_2$.

2. $0_V\rho = 0_V$, since by hypotheses $0_V\gamma = 0_W$ and λ is a linear transformation.

The Definition 3.1.4 is a generalization of the Definition 2.4.6 of translation based block cipher, as the name suggests. In fact, if $\gamma, \lambda \in \mathrm{Sym}(V)$ the two definitions coincide.

In the following example we show that the class of generalized tb ciphers properly contains the tb cipher one. In fact we provide an instance of a generalized tb cipher that can not be expressed as a "classical" tb cipher for any bricklayer transformation (which acts parallelly on vector spaces V_i) and for any mixing layer (which acts on the whole space V).

Example 3.1.7. Assume that $V := (\mathbb{F}_2)^4$, $W := (\mathbb{F}_2)^6$, $V_i := (\mathbb{F}_2)^2$ and $W_i := (\mathbb{F}_2)^3$, for $1 \leq i \leq 2$. Let us consider a fixed proper round with

- bricklayer transformation $\gamma : (\mathbb{F}_2)^4 \longrightarrow (\mathbb{F}_2)^6$ such that the 2×3 S-boxes $\gamma_1 : (\mathbb{F}_2)^2 \longrightarrow (\mathbb{F}_2)^3$ and $\gamma_2 : (\mathbb{F}_2)^2 \longrightarrow (\mathbb{F}_2)^3$ are defined as follows:

γ_1	$(0,0)$	$(1,0)$	$(0,1)$	$(1,1)$
	$(0,0,0)$	$(0,1,0)$	$(1,1,0)$	$(1,0,1)$

γ_2	$(0,0)$	$(1,0)$	$(0,1)$	$(1,1)$
	$(0,0,0)$	$(0,1,0)$	$(0,0,1)$	$(1,0,0)$

The ANF of γ (Section 1.1.1), is given by

$$(x_2, x_1 + x_2, x_1x_2, x_3x_4, x_3 + x_3x_4, x_4 + x_3x_4).$$

- mixing layer $\lambda : (\mathbb{F}_2)^6 \longrightarrow (\mathbb{F}_2)^4$, represented by the matrix:

$$\lambda = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

By a computer check it is possible to prove that this mixing layer is proper.

In this case $\rho = \gamma\lambda$ is the permutation:

$$\begin{aligned}\rho = (&(1,1,0,0),(1,1,0,1),(0,1,1,1),(0,1,0,1),(1,0,1,1),(0,0,1,0),(1,1,1,0),\\ &(0,0,1,1),(0,1,0,0),(0,0,0,1),(1,0,1,0),(1,0,0,0),(0,1,1,0),(1,1,1,1),\\ &(1,0,0,1))\end{aligned}$$

which is *non linear* since:

$$\rho((1,0,0,0)+(0,1,0,0)) = (1,1,0,1) \neq \rho(1,0,0,0)+\rho(0,1,0,0) = (0,1,1,1).$$

We observe that in general the permutation group $\mathrm{Sym}((\mathbb{F}_2)^2)$ coincides with the affine group $\mathrm{AGL}((\mathbb{F}_2)^2)$.
Therefore any bricklayer transformation in $\gamma' \in \mathrm{Sym}(V)$ is linear. Since we saw that ρ is non linear, there does not exist any linear transformation λ' such that $\gamma'\lambda' = \rho$. This means that there does not exist any translation based block cipher which acts in the same way as the generalized one.

3.2 Primitivity.

In this section we will prove that the group generated by the round functions of a generalized tb cipher is primitive.
In order to do this let us introduce some important definitions and results.

Recall the definition of vectorial Boolean function given in Section 1.1.1. Then we have:

Definition 3.2.1 ([CVS09]). *Let $\gamma : (\mathbb{F}_2)^p \longrightarrow (\mathbb{F}_2)^l$ a vectorial Boolean function, with $p, l > 1$, such that $0_{(\mathbb{F}_2)^p}\gamma = 0_{(\mathbb{F}_2)^l}$. Given $1 \leq r < p$, the function γ is strongly r-anti-invariant if for any subspace U of $(\mathbb{F}_2)^p$ such that $U' := U\gamma$ is a subspace of $(\mathbb{F}_2)^l$, then dim$(U') < p - r$.*

As we can easily see this is a generalization of the Definition 1.1.6 (point (2)) of strongly anti-invariance for vectorial Boolean functions that are not permutations. Nevertheless we have to remark that, given U a subspace of $(\mathbb{F}_2)^p$, then $U\gamma$ is not a vectorial subspace of $(\mathbb{F}_2)^l$ in general.

Let us return to our case. For the rest of this section we use notations of the previous section. In particular recall that $V = (\mathbb{F}_2)^n$, $W = (\mathbb{F}_2)^t$ with $n = mp$ and $t = ml$ $(m, p, l > 1)$.
Now we introduce more specific definition of strongly anti-invariance, more useful for our purposes.

Definition 3.2.2 (generalized strongly anti-invariance). *Let $\gamma : V \longrightarrow W$ a bricklayer transformation and $\lambda : W \longrightarrow V$ a mixing layer and $1 \leq r < p$. A brick γ_i, with $1 \leq i \leq m$ is generalized strongly r-anti-invariant if, for any subspace U of V_i and U' of W_i such that $U\gamma_i \subseteq U'$, then:*

$$U' = U\gamma_i + (\ker\lambda \cap U') \tag{3.4}$$

and dim(U), dim$(U') < p - r$.

Remark 3.2.3. We observe that in this case:

$$U\gamma_i \cap (\ker \lambda \cap U') = \{0_{W_i}\}.$$

In fact if there exists $w \in U\gamma_i \cap (\ker \lambda \cap U')$, $w \neq 0_{W_i}$, $w = v\gamma_i$ for a certain $v \in U$. Now, $v \neq 0_{V_i}$ (otherwise $w = 0_{W_i}$) and this means that there exists a nonzero element $v \in V_i$ such that $v\gamma\lambda = 0_V$ contradicting the fact that ρ is a bijection and sends only zero in zero (Remark 3.1.6).

Moreover, in the same notations of the previous definition, if $\ker \lambda \cap U' = \{0_{W_i}\}$, then $U\gamma_i$ is a vector subspace of W_i and we can refer to the Definition 3.2.1 of standard strongly anti-invariance.

Recall from Section 2.3.2 that $\Gamma_h = \Gamma_h(C) = \langle f_{k,h} \mid k \in \mathcal{K}\rangle$, for $1 \leq h \leq r$, and $\Gamma_\infty = \Gamma_\infty(C) = \langle \Gamma_h \mid 1 \leq h \leq r\rangle$ is the group generated by the round functions. Then we state the following lemma.

Lemma 3.2.4. *Let C be a generalized translation based block cipher and let h be a proper round, $1 \leq h \leq r$. Then:*

1. *$\Gamma_h(C) = \langle \rho, \mathrm{T}(V)\rangle$, where $\mathrm{T}(V)$ is the translation group of V defined in Section 1.2,*
2. *$\Gamma_h(C)$ is imprimitive if and only if there exists a nontrivial proper subspace U of V such that $(u + v)\gamma + v\gamma \in U\lambda^{-1}$, for any $u \in U$ and $v \in V$. In this case a block system is $\{U + v \mid v \in V\}$.*

Proof.

1. In general we have that $\Gamma_h(C) = \langle \rho\sigma_v \mid v \in V\rangle$.

   Moreover $\rho = \rho\sigma_{0_V} \in \Gamma_h(C)$ and $\sigma_v = \rho^{-1}(\rho\sigma_v) \in \Gamma_h(C)$, for any $v \in V$.

2. Since $\mathrm{T}(V) \subset \Gamma_h(C)$, if $\Gamma_h(C)$ is imprimitive, a block system is of the form $\{U + v \mid v \in V\}$ for some nontrivial proper subspace U of V (Corollary 1.2.8).

   Let $v \in V$, since $0_V\rho = 0_V$ then $(U + v)\rho = U + v\rho = U + v\gamma\lambda$. Therefore for any $u \in U$ and $v \in V$
   $$(u + v)\gamma\lambda + v\gamma\lambda \in U$$
   and from the linearity of λ follows that:
   $$(u + v)\gamma + v\gamma \in U\lambda^{-1}.$$
   In particular for Remark 3.1.2 for any $u \in U$, $u \neq 0$, we have that
   $$\mathrm{Im}(\hat{\gamma}_u) + \ker \lambda \subset U\lambda^{-1} + \ker \lambda = U\lambda^{-1}$$

□

Now we state the primitivity theorem, the main result of this section.

Theorem 3.2.5. *Let $V = V_1 \oplus \ldots \oplus V_m$ and $W = W_1 \oplus \ldots \oplus W_m$ with $V_i = (\mathbb{F}_2)^p$, $W_i = (\mathbb{F}_2)^l$, $1 \leq i \leq m$ and $m, l, p > 1$. Let C be a generalized translation based cipher over V and suppose that each brick corresponding to a proper round h, $\gamma_{h,i} : V_i \longrightarrow W_i$ is:*

1. *2^r-uniform, and*

2. *generalized strongly $(r-1)$-anti-invariant.*

for some $1 < r < p$. Then $\Gamma_\infty(C)$ is primitive.

Proof. In the proof we will omit the round index h.
Suppose that $G := \Gamma_h(C)$ is imprimitive. For the previous lemma a block system is of the form $\{U + v | v \in V\}$ for any proper nontrivial subspace U of V.
Since U is a block, $U\rho\sigma_{0_V} = U\gamma\lambda = U + v$ for some $v \in V$. Moreover $0_V\gamma\lambda = 0_V$ and so $U\gamma\lambda = U$. We observe that $U\gamma$ is a proper subset of $U\lambda^{-1}$, as γ is not surjective. Since $U\gamma \subset U\lambda^{-1}$, then $U\gamma + \ker\lambda \subseteq U\lambda^{-1} + \ker\lambda$. Furthermore, given $w \in U\lambda^{-1}$ then there exists $u \in U$ such that:

$$w + u\gamma \in \ker\lambda.$$

In fact, as $w \in U\lambda^{-1}$, there exists $u' \in U$ such that $w\lambda = u'$. Remember that $U\gamma\lambda = U$, and so there exists $u \in U$ such that $u\gamma\lambda = u'$. Therefore, since $(w+u\gamma)\lambda = w\lambda + u\gamma\lambda = 0_W$, then $w + u\gamma \in \ker\lambda$ and hence $U\lambda^{-1} \subseteq U\gamma + \ker\lambda$. By Remark 3.1.2, then $U\lambda^{-1} + \ker\lambda \subseteq U\gamma + \ker\lambda$.
Thus we have that:

$$U\gamma + \ker\lambda = U\lambda^{-1} + \ker\lambda = U\lambda^{-1} \subseteq W \tag{3.5}$$

and so $U\gamma + \ker\lambda$ is a subspace of W.

Let $\pi_i : V \longrightarrow V_i$, for $1 \leq i \leq m$, be the i-th projection with respect to the decomposition $V = V_1 \oplus \ldots \oplus V_m$ and $I := \{i \in \{1, \ldots, m\} \mid \pi_i(U) \neq 0\}$. We distinguish two cases:

1. $U \cap V_i = V_i$, $\forall i \in I$;

2. $\exists \iota \in I$ such that $U \cap V_\iota \neq V_\iota$.

In the first case $U = \bigoplus_{i\in I} V_i$ is a wall. For (3.5) then:

$$(\bigoplus_{i\in I} V_i)\gamma + \ker\lambda = (\bigoplus_{i\in I} V_i)\lambda^{-1} + \ker\lambda = (\bigoplus_{i\in I} V_i)\lambda^{-1}. \tag{3.6}$$

We observe that, by definition:

$$(\bigoplus_{i\in I} V_i)\gamma \subset \bigoplus_{i\in I} W_i. \tag{3.7}$$

Thus from (3.6) and (3.7) follows that:

$$(\bigoplus_{i\in I} V_i)\lambda^{-1} \subset \bigoplus_{i\in I} W_i + \ker\lambda \tag{3.8}$$

which is impossible since λ is a proper mixing layer (Definition 3.1.3).

On the other hand, if there exists $\iota \in I$ such that $U \cap V_\iota \neq V_\iota$, U is not a wall. At first we prove that $U \cap V_\iota \neq \{0_V\}$.
Let $u \in U$ such that $\pi_\iota(u) := u_\iota \neq 0$ and $v \in V_\iota$. For the imprimitivity of G, by Lemma 3.2.4, follows that $(u+v)\gamma + v\gamma \in U\lambda^{-1}$. Moreover $u\gamma \in U\gamma \subset U\lambda^{-1}$ and so $u\gamma + (u+v)\gamma + v\gamma \in U\lambda^{-1}$. In particular the latter element has all zero components except possibly the ι-th one, that is $u_\iota\gamma_\iota + (u_\iota + v)\gamma_\iota + v\gamma_\iota \in U\lambda^{-1} \cap W_\iota$. If the latter is zero for all $v \in V_\iota$, then the $\hat{\gamma}_{\iota_{u_\iota}}$ would be constant, in contradiction to the fact that γ_ι is 2^r-uniform (Definition 1.1.3) and $r < p$.

From (3.5) follows that $(U \cap V_\iota)\gamma_\iota \subset U\lambda^{-1} \cap W_\iota$. Therefore $(U \cap V_\iota)\gamma_\iota + \ker\lambda = (U\lambda^{-1} \cap W_\iota) + \ker\lambda$ and finally:

$$((U \cap V_\iota)\gamma_\iota + \ker\lambda) \cap W_\iota = ((U\lambda^{-1} \cap W_\iota) + \ker\lambda) \cap W_\iota. \tag{3.9}$$

We observe that:

$$((U\lambda^{-1} \cap W_\iota) + \ker\lambda) \cap W_\iota = (U\lambda^{-1} \cap W_\iota) + (\ker\lambda \cap W_\iota). \tag{3.10}$$

In fact given $w' \in U\lambda^{-1} \cap W_\iota$ and $w'' \in \ker\lambda$ such that $w := w' + w'' \in W_\iota$, since $w, w' \in W_\iota$ and W_ι is a vector space, we have that w'' also belongs to W_ι and so $w \in (U\lambda^{-1} \cap W_\iota) + (\ker\lambda \cap W_\iota)$.
On the other hand, if $w' \in U\lambda^{-1} \cap W_\iota$ and $w'' \in \ker\lambda \cap W_\iota \subset \ker\lambda$, then $w := w' + w'' \in (U\lambda^{-1} \cap W_\iota) + \ker\lambda$. Moreover $w', w'' \in W_\iota$, which is a vectorial space, thus $w \in W_\iota$ and finally $w \in ((U\lambda^{-1} \cap W_\iota) + \ker\lambda) \cap W_\iota$.
In the same way it is possible to prove that:

$$((U \cap V_\iota)\gamma_\iota + \ker\lambda) \cap W_\iota = (U \cap V_\iota)\gamma_\iota + (\ker\lambda \cap W_\iota). \tag{3.11}$$

From (3.9), applying (3.10) and (3.11) follows that:

$$(U \cap V_\iota)\gamma_\iota + (\ker \lambda \cap W_\iota) = (U\lambda^{-1} \cap W_\iota) + (\ker \lambda \cap W_\iota). \tag{3.12}$$

Now, in general we have that $U\lambda^{-1} \cap W_\iota \subseteq (U\lambda^{-1} \cap W_\iota) + (\ker \lambda \cap W_\iota)$.
On the other hand, since $\ker \lambda \subset U\lambda^{-1}$ then $\ker \lambda \cap W_\iota \subset U\lambda^{-1} \cap W_\iota$. Furthermore, given $w' \in U\lambda^{-1} \cap W_\iota$ and $w'' \in \ker \lambda \cap W_\iota$, then w'' also belongs to $U\lambda^{-1} \cap W_\iota$ and $w' + w'' \in U\lambda^{-1} \cap W_\iota$, since $U\lambda^{-1} \cap W_\iota$ is a vector subspace of W_ι. Therefore $(U\lambda^{-1} \cap W_\iota) + (\ker \lambda \cap W_\iota) \subseteq (U\lambda^{-1} \cap W_\iota)$ and so:

$$U\lambda^{-1} \cap W_\iota = (U\lambda^{-1} \cap W_\iota) + (\ker \lambda \cap W_\iota). \tag{3.13}$$

From (3.12) and (3.13) we can conclude that:

$$(U \cap V_\iota)\gamma_\iota + (\ker \lambda \cap W_\iota) = U\lambda^{-1} \cap W_\iota. \tag{3.14}$$

Let $u \in U \cap V_\iota$, $u \neq 0$, from the imprimitivity of G, by Lemma 3.2.4, follows that $\mathrm{Im}(\hat{\gamma}_{\iota_u}) \subset U\lambda^{-1}$. On the other hand, by definition we have $\mathrm{Im}(\hat{\gamma}_{\iota_u}) \subset W_\iota$ so that $\mathrm{Im}(\hat{\gamma}_{\iota_u}) \subset U\lambda^{-1} \cap W_\iota$. Now, γ_ι is 2^r-uniform and so $|\mathrm{Im}(\hat{\gamma}_{\iota_u})| \geq 2^{p-r}$ and in particular $|U\lambda^{-1} \cap W_\iota| \geq 2^{p-r}$. Furthermore since the map γ_ι is injective, $0_W \notin \mathrm{Im}(\hat{\gamma}_{\iota_u})$ hence $|U\lambda^{-1} \cap W_\iota| \geq 2^{p-r} + 1$, which implies that $\dim(U\lambda^{-1} \cap W_\iota) \geq p - r + 1$. By (3.14) this is impossible, since γ_ι is generalized strongly $(r-1)$-anti-invariant (Definition 3.2.2).
Hence $\Gamma_h(C)$ is primitive and, since it is a subgroup of $\Gamma_\infty(C)$, also $\Gamma_\infty(C)$ is primitive.
□

From the second part of the proof follows that if we suppose that:

$$\ker \lambda \cap W_i = \{0_{W_i}\}, \text{ for } 1 \leq i \leq m$$

then (3.14) becomes:

$$(U \cap V_\iota)\gamma_\iota = U\lambda^{-1} \cap W_\iota.$$

This means that $(U \cap V_\iota)\gamma_\iota$ is a vector subspace of W_ι, and we can refer to the standard definition of strongly anti-invariance (Definition 3.2.1) to prove the second part of the theorem. Thus we have the following result:

Theorem 3.2.6. *Let $V = V_1 \oplus \ldots \oplus V_m$ and $W = W_1 \oplus \ldots \oplus W_m$ with $V_i = (\mathbb{F}_2)^p$, $W_i = (\mathbb{F}_2)^l$, $1 \leq i \leq m$ and $m, l, p > 1$. Let C be a translation based cipher over V and suppose that each brick corresponding to a proper round h, $\gamma_{h,i} : V_i \longrightarrow W_i$ is:*

1. *2^r-uniform and*

2. *strongly $(r-1)$-anti-invariant.*

for some $1 < r < p$. Moreover, suppose that:

$$\ker \lambda \cap W_i = \{0_{W_i}\},$$

for $1 \le i \le m$. Then $\Gamma_\infty(C)$ is primitive.

In conclusion we observe that the hypothesis of Theorem 3.2.5 regarding the mixing layer is necessary.

Proposition 3.2.7. *Let C be a generalized translation based block cipher with λ a mixing layer for a round h. If λ is not proper, then $\Gamma_h(C)$ is imprimitive.*

Proof. If λ is not proper, then there exists a nonempty proper subset $I = \{i_1, \dots, i_s\}$ of $\{1, \dots, m\}$ such that:

$$\bigoplus_{i\in I} V_i\lambda^{-1} \subset \bigoplus_{i \in I} W_i + \ker \lambda. \tag{3.15}$$

Without loss of generality we can suppose that

$$I := \{1, \dots, s\}.$$

Let $U = \{(u_1, \dots, u_m) \in V \mid u_i = 0, i \ge s+1\}$. We observe that U is a vector subspace of V. We claim that $\mathcal{B} = \{U + v \mid v \in V\}$ is a block system for $\Gamma_h(C)$. By Corollary 1.2.8, $\mathcal{B}$ is a block system for $\mathrm{T}(V)$, thus we need only to prove that, fixed $v \in V$

$$(U+v)\gamma\lambda = U + v_1$$

for a certain $v_1 \in V$.
Since γ is a bricklayer transformation, then

$$\begin{aligned} (U+v)\gamma &= \{(u+v)\gamma \mid u \in U\} = \{((u_1+v_1)\gamma_1, \dots, (u_m+v_m)\gamma_m) \mid u \in U\} \\ &= \{((u_1+v_1)\gamma_1, \dots, (u_s+v_s)\gamma_s, v_{s+1}\gamma_{s+1}, \dots, v_m\gamma_m) \mid u_i \in V_i, 1 \le i \le s\}. \end{aligned}$$

We put

$$w := (0, \dots, 0, v_{s+1}\gamma_{s+1}, \dots, v_m\gamma_m) \in W$$

and then we have

$$\begin{aligned} (U+v)\gamma &= \{((u_1+v_1)\gamma_1, \dots, (u_s+v_s)\gamma_s, 0_{W_{s+1}}, \dots, 0_{W_m}) + w \mid u_i \in V_i, 1 \le i \le s\} \\ &= \{(u'_1\gamma_1, \dots, u'_s\gamma_s, 0_{W_{s+1}}, \dots, 0_{W_m}) + w \mid u'_i \in V_i, 1 \le i \le s\} \\ &= U\gamma + w \end{aligned} \tag{3.16}$$

since $0_{V_i}\gamma_i = 0_{W_i}$, for $1 \leq i \leq m$. We observe that, since γ is a bricklayer transformation and λ is not proper, the composition $\gamma\lambda$ sends U in U. Therefore $U \subseteq U\gamma\lambda$. On the other hand, since $\gamma\lambda$ is a permutation, $U\gamma\lambda$ and U have the same cardinality, hence $U\gamma\lambda = U$. Moreover $w\lambda \in V$, thus denoting $v_1 := w\lambda$, recalling that λ is linear and by (3.16) then

$$(U + v)\gamma\lambda = (U\gamma + w)\lambda = U\gamma\lambda + w\lambda = U + v_1$$

□

Conclusions.

The main aim of this work is the definition of a generalization of translation based block ciphers and the proof of the primitivity of the related group generated by the round functions. One of the possible future development of this work could be the study of round functions of this new cipher, aiming to find properties which imply that the group generated by them is the alternating or the symmetric group, as already proved for classical translation based block ciphers.

Nevertheless, this new model could seem not applicable from the computational point of view. Recall that in classical translation based block ciphers, any round function is given by the composition of a bricklayer transformation, a mixing layer and a translation that are also permutations. Since the inverse of a translation is the translation itself, in order to compute the inverse of any round function it is sufficient to compute the inverse of the matrix which represents the mixing layer and the inverse of any brick of the bricklayer transformation. The computation of inverses of these transformations is certainly less expensive, computationally speaking, with respect to the direct computation of the inverse of the entire round function. On the other hand, each round function of a generalized translation based block cipher is given by the composition of a bricklayer transformation and a mixing layer that are not permutations and a translation. The composition of all these functions is a bijection and invertible in the whole, but the bricklayer transformation and the mixing layer are not. Therefore, the only way to compute the inverse of any round function consists of computing the inverse of the entire function, which is very expensive, since this is a non linear function which usually acts on a big space. Hence, another possible future development of this work could be the study of a computationally efficient method to invert any round function of generalized translation based block ciphers.

Another interesting research topic to inspect could be the use of the round function of a generalized translation based as a round function of a Feistel network. In this case, the fact that the computation of the inverse of a round function is expensive, does not represent a problem, since any Feistel round is invertible even if the related round function is not.

For the sake of simplicity, let us denote a round of a Feistel network by $F : (\mathbb{F}_2)^{2n} \rightarrow (\mathbb{F}_2)^{2n}$, $n \geq 1$. Recall that we can see any $p \in (\mathbb{F}_2)^{2n}$ as the concatenation $p = l \mid\mid r$ with $l, r \in (\mathbb{F}_2)^n$, then

$$F(p) = F(l \mid\mid r) := r \mid\mid l \oplus f(r, k)$$

where $k \in (\mathbb{F}_2)^m$, $m \geq 1$, is the round key, $f : (\mathbb{F}_2)^n \times (\mathbb{F}_2)^m \rightarrow (\mathbb{F}_2)^n$ is the round function corresponding to the round F, and "$\oplus$" denote the XOR operation.
As we have previously remarked, F is invertible independently from the invertibility of the round function f and the inverse F^{-1} of F is defined by

$$F^{-1}(c) = F^{-1}(l' \mid\mid r') := r' \oplus f(l', k) \mid\mid l'$$

for any $c \in (\mathbb{F}_2)^{2n}$, seen as the concatenation $c = l' \mid\mid r'$, with $l', r' \in (\mathbb{F}_2)^n$. In fact,

$$F^{-1}(F(l \mid\mid r)) = F^{-1}(r \mid\mid l \oplus f(r, k)) = l \oplus f(r, k) \oplus f(r, k) \mid\mid r = l \mid\mid r.$$

In conclusion, it could be interesting to study the group generated by the Feistel rounds starting from known properties of the group generated by the round functions of a generalized translation based block cipher.

Appendix

A.1 More results on translation based block ciphers.

In this appendix we introduce some theorems that prove that the group generated by the round functions of some translation based block ciphers are isomorphic to the alternating group or the symmetric one.

We start introducing some definitions of group theory that we will use in this part of the work. In particular we will follow [Rom10] and [DM96] unless otherwise specified.

Definition A.1.1 (group extension)**.** *An extension of a pair (N, Q) of groups is a group G that has a normal subgroup N' isomorphic to N and for which G/N' is isomorphic to Q. We denote it by*

$$G = N.Q$$

Let G be a group. We denote by $\mathrm{Aut}(G)$, the set of all *automorphisms* of G.
Now, let us consider $a, b \in G$. The element aba^{-1} is called the *conjugate* of b by a.
The *conjugacy relation* is the binary relation $\sim$ on G defined by

$$a \sim b \Longleftrightarrow b = xax^{-1}, \text{ for some } x \in G.$$

This is an equivalence relation and if $a \sim b$, we say that a and b are *conjugate.*
Moreover, fixed $a \in G$ the map:

$$\begin{array}{rccc} f_a : & G & \longrightarrow & G \\ & x & \longmapsto & axa^{-1} \end{array}$$

is an automorphism, that is $f_a \in \mathrm{Aut}(G)$, called *inner* automorphism.
The set of all inner automorphisms is denoted by $\mathrm{Inn}(G)$. This group is a normal subgroup of $\mathrm{Aut}(G)$ and the quotient group

$$\mathrm{Out}(G) := \mathrm{Aut}(G)/\mathrm{Inn}(G)$$

is the *outer automorphism group.*

We recall the definition of *semidirect product.* In particular, let H and K be two groups and $\psi : H \longrightarrow \mathrm{Aut}(K)$ an action of H on K (Section 1.2).
Let $G := K \times H$ be the cartesian product of K and H. Let us define an operation

$$(k_1, h_1) \cdot (k_2, h_2) := (k_1(\psi(h_1)(k_2)), h_1h_2),$$

for any $k_1, k_2 \in K$ and $h_1, h_2 \in H$.
We have that $(G, \cdot)$ is a group called *semidirect product* of K by H and denoted by

$$G = K \rtimes H.$$

In what follows, in order to simplify notations, given a group G acting on a set X, we denote by x^g the action of an element $g \in G$ over an element $x \in X$.

Now we are ready to introduce a particular semidirect product, called *wreath product*, which plays an important role in the study of primitive permutation groups.

Given the two nonempty sets Γ and Δ, we denote with $\mathrm{Fun}(\Gamma, \Delta)$ the set of all functions from Γ to Δ. In particular, if Δ is a group, also $\mathrm{Fun}(\Gamma, \Delta)$ is a group with the *pointwise product*

$$(f \cdot g)\gamma := f(\gamma)g(\gamma),$$

for any $f, g \in \mathrm{Fun}(\Gamma, \Delta)$ and $\gamma \in \Gamma$.

Now, let K and H be two groups and suppose that H acts on a set Γ. Then, the *wreath product* of K by H with respect to this action, denoted by $Kwr_\Gamma H$, is the semidirect product

$$\mathrm{Fun}(\Gamma, K) \rtimes H,$$

where H acts on $\mathrm{Fun}(\Gamma, K)$ via

$$f^x(\gamma) := f(\gamma^{x^{-1}}),$$

for any $f \in \mathrm{Fun}(\Gamma, K)$, $\gamma \in \Gamma$ and $x \in H$.

Let H be a group acting on a set Γ and K be a group acting on a set Δ. We can define an action of the wreath product $Kwr_\Gamma H$ on the set $\mathrm{Fun}(\Gamma, \Delta)$, called *product action.* Precisely, the product action is

$$\phi^{(f,x)}(\gamma) := \phi(\gamma^{x^{-1}})^{f(\gamma^{x^{-1}})},$$

for any $\phi \in \mathrm{Fun}(\Gamma, \Delta)$, $(f, x) \in Kwr_\Gamma H$ and $\gamma \in \Gamma$.

Now we are ready to introduce the Li version ([Li03]) of the O'Nan Scott Theorem (Section 4.5 [Cam99]), which classifies primitive groups with an abelian regular subgroup. In particular, following [ACTT15], we state the Li Theorem in the specific case where the degree of the primitive group is 2^n, with $n \geq 1$.

Theorem A.1.2. *Let G be a primitive permutation group of degree 2^n, with $n \geq 1$. Then G contains an abelian regular subgroup T if and only if either*

1. $G \leq \mathrm{AGL}((\mathbb{F}_2)^n)$ *or,*

2. *G is a wreath product, that is*

$$G = (S_1 \times \ldots \times S_d).O.P \text{ and } T = T_1 \times \ldots \times T_d$$

where $d \geq 1$ divides n, each $T_i < S_i$ with $|T_i| = 2^{n/d}$, the S_i are all conjugate, $O \leq \mathrm{Out}(S_1) \times \ldots \times \mathrm{Out}(S_d)$, P permutes transitively the S_i, and one of the following holds:

   (a) S_i is isomorphic to the projective linear group on $(\mathbb{F}_q)^\alpha$, that is $S_i \cong \mathrm{PGL}((\mathbb{F}_q)^\alpha)$ and T_i is a cyclic group of order $(q^\alpha - 1)/(q-1)$, for q a prime power, or

   (b) $S_i \cong \mathrm{Alt}(2^{n/d})$ or $\mathrm{Sym}(2^{n/d})$ and T_i is an abelian group of order $2^{n/d}$.

Proof. See the Theorem 1.1 of [Li03]. □

The notation $G = (S_1 \times \ldots \times S_d).O.P$ denotes that the group $N := S_1 \times \ldots S_d$ is normal in G and G/N is an extension of the group O by the group P (Definition A.1.1).

As consequence of Theorem A.1.2, we have the following corollary when T is *elementary abelian.* Recall that an elementary abelian group is an abelian group in which the order of every nontrivial element is the same prime.

Corollary A.1.3. *Let G be a primitive permutation group of degree 2^n, with $n \geq 1$. Assume that G contains an elementary abelian regular subgroup T. Then one of the following holds:*

1. $G \leq \mathrm{AGL}((\mathbb{F}_2)^n)$;

2. $G \cong \mathrm{Alt}(2^n)$ *or* $\mathrm{Sym}(2^n)$

3. *G is a wreath product, that is*

$$G = (S_1 \times \ldots \times S_d).O.P \text{ and } T = T_1 \times \ldots \times T_d$$

where $d > 1$ divides n, each T_i is an abelian subgroup of S_i of order $2^{n/d}$ with $S_i \cong \mathrm{Alt}(2^{n/d})$ or $\mathrm{Sym}(2^{n/d})$, the S_i are all conjugate, $O \leq \mathrm{Out}(S_1) \times \ldots \times \mathrm{Out}(S_d)$ and P permutes transitively the S_i

In particular, if $n \leq 5$, then G cannot be a wreath product.

Proof. See Corollary 3.4 [ACTT15]. □

Let C be a translation based block cipher (Section 2.4) over $V = (\mathbb{F}_2)^n$, with $n \geq 1$. The translation group $T(V)$ is an elementary abelian subgroup of the group generated by the round functions $\Gamma_\infty(C)$ ([CVS09]). Thus we can apply the classification of primitive groups of the previous corollary.

We introduce another definition that we will use later.

Definition A.1.4. *A vectorial Boolean function $f : (\mathbb{F}_2)^p \longrightarrow (\mathbb{F}_2)^p$ is called anti-crooked (AC for short), if for each $u \in (\mathbb{F}_2)^n$, $u \neq 0$, the set*

$$\mathrm{Im}(\hat{f}_u) = \{f(x+u) + f(x) \mid x \in (\mathbb{F}_2)^n\}$$

is not an affine subspace of $(\mathbb{F}_2)^p$.

Now we are ready to state the first basic result of this appendix.

Theorem A.1.5 ([CDS09]). *Let C be a translation based block cipher with a strongly proper round such that*

1. *each brick of the strongly proper round verifies hypotheses of Theorem 2.4.9, and*
2. *there exists a round such that any brick is AC.*

Then the group $\Gamma_\infty(C)$ is $\mathrm{Alt}(V)$ or $\mathrm{Sym}(V)$.

Using this theorem it is possible to prove that the group generated by the round functions of AES is the alternating group [CDS09].

Authors in [ACTT15] proved that using hypotheses of Theorem 2.4.12 instead of Theorem 2.4.9, we have the same results. In particular,

Theorem A.1.6. *Let C be a translation based block cipher with a strongly proper round such that*

1. *each brick of the strongly proper round verifies hypotheses of Theorem 2.4.12, and*
2. *there exists a round such that any brick is AC.*

Then the group $\Gamma_\infty(C)$ *is* $\mathrm{Alt}(V)$ *or* $\mathrm{Sym}(V)$.

As remarked in Section 2.4, bricks of PRESENT verify hypotheses of the Theorem 2.4.12. Nevertheless, PRESENT bricks are not AC, therefore we cannot apply the previous theorem.

By Corollary A.1.3, in order to prove that $\Gamma_\infty(C)$ is isomorphic to the alternating or the symmetric group it is necessary to exclude the affine and the wreath product case. In particular, in [CDS09] and in [ACTT15], authors prove that:

- $\Gamma_\infty(C)$ is not affine using the fact that each brick of a certain round is AC;
- $\Gamma_\infty(C)$ is not a wreath product using the fact that there is a strongly proper round and hypotheses on bricks of the Theorem 2.4.9 (or of Theorem 2.4.12 respectively).

In particular, they proved the following two results.

Proposition A.1.7. *Let* C *be a tb cipher with a proper round* h *where any brick is AC. If* Γ_h *is primitive, then it is not affine.*

Proposition A.1.8. *Let* C *be a tb cipher with a strongly proper round* h*. If the hypotheses of Theorem 2.4.9 (or Theorem 2.4.12 respectively) are satisfied, then the primitive group* $\Gamma_h(C)$ *is not a wreath product.*

As we have previously remarked, PRESENT bricks are not AC, thus authors in [ACTT15] introduced another condition which allows to avoid the affine case, as we can see in the following proposition.

Proposition A.1.9. *Let* C *be a tb cipher over* $V = (\mathbb{F}_2)^{pm}$*, with* $p \geq 3$ *and* $m \geq 2$*. Suppose that there exists a brick* γ_i *corresponding to a proper round* h *such that*

$$\mathrm{Alt}(V_i) \subseteq \langle \mathrm{T}(V_i), \gamma_i \mathrm{T}(V_i) \gamma_i^{-1} \rangle. \tag{A.1}$$

If Γ_h *is a primitive group, then it is not of affine type.*

Now we are ready for the last basic theorem of this appendix.

Theorem A.1.10. *Let* C *be a tb cipher over* $V = (\mathbb{F}_2)^{pm}$*,* $p \geq 3$ *and* $n \geq 2$*, with a strongly proper round* h *such that the corresponding bricks satisfy the hypotheses of Theorem 2.4.9 (or Theorem 2.4.12, respectively). Suppose further that one of these bricks satisfies condition A.1 of Proposition A.1.9. Then* $\Gamma_\infty(C) = \mathrm{Alt}(V)$.

Proof. By Theorem 2.4.9 (or Theorem 2.4.12 respectively), the group $\Gamma_h(C)$ is primitive. The claim follows applying Corollary A.1.3 and then Propositions A.1.9 and A.1.8 respectively. □

Moreover, we have the following corollary.

Corollary A.1.11. *Let C be a tb cipher over $V = (\mathbb{F}_2)^{pm}$, $p = 3, 4$ or 5 and $m \geq 2$, with a strongly proper round h such that the corresponding bricks satisfy the hypotheses of Theorem 2.4.9 (or Theorem 2.4.12, respectively). Then $\Gamma_\infty(C) = \mathrm{Alt}(V)$.*

Therefore when the dimension of S-boxes are $3, 4$ or 5, in order to prove that the group generated by the round functions is isomorphic to the alternating group, it is enough to check the same conditions used to show its primitivity.

Using this result we can conclude that the round functions of PRESENT generate the alternating group.

Bibliography

[ACMS16] R. Aragona, M. Calderini, D. Maccauro, and M. Sala, *On weak differential uniformity of vectorial boolean functions as a cryptographic cryterion*, AAECC **5** (2016), no. 27, 359–372.

[ACS17] R. Aragona, A. Caranti, and M. Sala, *The group generated by the round functions of a GOST-like cipher*, Annals of Pure and Applied Mathematics **196** (2017), no. 1, 1–17.

[ACTT15] R. Aragona, M. Calderini, A. Tortora, and M. Tota, *On the primitivity of PRESENT and other lightweight ciphers*, Arxiv preprint arXiv:1611.01346v1 (2015).

[AKL$^+$07] A. Andrey Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe, *PRESENT: An Ultra-Lightweight Block Cipher*, Proc. of CHES 2007, LNCS, vol. 4727, Springer, 2007, pp. 450–466.

[Ald13] F. Aldà, *The Partial Sum Attack on 6-round reduced AES: implementation and improvement*, Master's thesis (laurea specialistica), University of Trento, Department of Mathematics, 2013.

[BAK98] E. Biham, R. Anderson, and L. Knudsen, *Serpent: A new block cipher proposal*, Fast Software Encryption, Springer, 1998, pp. 222–238.

[BFKR10] G. Baumslag, B. Fine, M. Kreuzer, and G. Rosenberger, *A course in mathematical cryptography*, De Gruyter Graduate, 2010.

[Cam99] P. J. Cameron, *Permutation groups*, London Mathematical Society Student Texts, vol. 45, Cambridge University Press, Cambridge, 1999.

[Car10] C. Carlet, *Vectorial boolean functions for cryptography*, Boolean Models and Methods in Mathematics, Computer Science, and Engineering **134** (2010), 398–469.

[CDS09] A. Caranti, F. Dalla Volta, and M. Sala, *An application of the O'Nan-Scott theorem to the group generated by the round functions of an AES-like cipher*, Designs, Codes and Cryptography **52** (2009), no. 3, 293–301.

[CG75] Don Coppersmith and E. Grossman, *Generators for certain alternating groups with applications to cryptography*, SIAM J. Appl. Math. **29** (1975), no. 4, 624–627.

[CS15] M. Calderini and M. Sala, *On differential uniformity of maps that may hide an algebraic trapdoor*, Algebraic Informatics: 6th International Conference, CAI 2015, Stuttgart, Germany, September 1-4, 2015. Proceedings, LNCS, vol. 9270, 2015, pp. 70–78.

[CVS09] A. Caranti, F. Dalla Volta, and M. Sala, *On some block ciphers and imprimitive groups*, AAECC **20** (2009), no. 5-6, 229–350.

[DM96] J. D. Dixon and B. Mortimer, *Permutation groups*, vol. 163, Springer-Verlag, 1996.

[Dol10] V. Dolmatov, *GOST 28147-89: Encryption, decryption, and message authentication code (MAC) algorithms*, http://tools.ietf.org/html/rfc5830, 2010, Tech. report.

[DR02] J. Daemen and V. Rijmen, *The design of Rijndael*, Information Security and Cryptography, Springer-Verlag, Berlin, 2002, AES - the Advanced Encryption Standard.

[DR13] J. Daemen and V. Rijmen, *The design of rijndael: Aes-the advanced encryption standard*, Springer Science & Business Media, 2013.

[EG83] S. Even and O. Goldreich, *DES-Like functions can generate the alternating group*, IEEE Trans. Inform. Theory (1983), 863–865.

[Kob94] N. Koblitz, *A course in number theory and cryptography*, vol. 114, Springer, 1994.

[KRS88] Jr. B. S. Kaliski, R. L. Rivest, and A. T. Sherman, *Is the data encryption standard a group? (Results of cycling experiments on DES)*, J. Cryptology **1** (1988), no. 1, 3–36.

[Li03] C. H. Li, *The finite primitive permutation groups containing an abelian regular subgroup*, Proc. London Math. Soc. **87** (2003), no. 3, 725–747.

[LN03] R. Lidl and H. Niederreiter, *Finite fields*, Encyclopedia of mathematics and its applications, vol. 20, Cambridge university press, 2003.

[MPW94] S. Murphy, K. Paterson, and P. Wild, *A weak cipher that generates the symmetric group*, Journal of Cryptology **7** (1994), 61–65.

[Nat77] National Bureau of Standards, *The Data Encryption Standard*, Federal Information Processing Standards Publication (FIPS) 46, 1977.

[Nat01] National Institute of Standards and Technology, *The Advanced Encryption Standard*, Federal Information Processing Standards Publication (FIPS) 197, 2001.

[Pat99] K. G. Paterson, *Imprimitive permutation groups and trapdoors in interated block ciphers*, Fast software encryption, LNCS, vol. 1636, Springer, Berlin, 1999, pp. 201–214.

[Rim09] Anna Rimoldi, *On algebraic and statistical properties of AES-like ciphers*, Ph.D. thesis, University of Trento, 2009.

[Rom10] S. Roman, *Fundamentals of group theory: An advanced approach*, Birkhauser, 2010.

[RP97] V. Rijmen and B. Preneel, *A family of trapdoor functions*, Fast Software Encryption, Springer, 1997, pp. 139–148.

[Rue92] R. Rueppel, *Stream ciphers*, Contemporary cryptology - The science of information integrity, IEEE Press, 1992, pp. 65–134.

[Sha49] C. E. Shannon, *Communication theory of secrecy systems*, Bell System Tech. J. **28** (1949), 656–715.

[SW08] R. Sparr and R. Wernsdorf, *Group theoretic properties of Rijndael-like ciphers*, Discrete Appl. Math. **156** (2008), no. 16, 3139–3149.

[SW15] ______, *The round functions of KASUMI generate the alternating group*, Journal of Cryptology **9** (2015), 23–32.

[Wer93] R. Wernsdorf, *The round functions of DES generate the alternating group*, Proc. of EUROCRYPT92 **658** (1993), 9.

[Wer00] ______, *The round functions of SERPENT generate the alternating group*, 2000, http://csrc.nist.gov/archive/aes/round2/comments/20000512-rwernsdorf.pdf.