LMV
Laboratoire de mathématiques
de Versailles - CNRS UMR 8100

UVSQ
université PARIS-SACLAY

cnrs

Dissertation présentée pour obtenir

L'Habilitation à Diriger des Recherches

Mention Informatique

Université Paris-Saclay

École doctorale Sciences et Technologies de l'Information et de la Communication

# Éléments de Cryptanalyse

Yann Rotella

Membres du Jury:

Alain Couvreur, rapporteur
Directeur de recherche, Inria, Paris

Joan Daemen, examinateur
Professeur, Radboud University, Nijmegen

Orr Dunkelman, examinateur
Professeur, University of Haifa

Caroline Fontaine, examinatrice
Directrice de recherche, CNRS, Université Paris-Saclay

Pierre-Alain Fouque, rapporteur
Professeur, Université de Rennes

María Naya-Plasencia, examinatrice
Directrice de recherche, Inria, Paris

Thomas Peyrin, rapporteur
Professeur, Nanyang Technological University, Singapour

Date de soutenance : le 6 février 2025

*Comprenez bien que je ne comprends pas tout*
*Je ne sais rien sur le rien, encore moins sur le tout*
*Qui ? Qui du soleil ou de la terre se tourne autour ?*
*Qui ? Je l'oublie toujours*
*...*
*Mais plus tu iras vers l'est*
*Tu te retrouveras à l'ouest*
*Tu t'avances et moins tu sais*
*...*
*Total à l'ouest, Philippe Katerine, 2024*

# Remerciements

Il faut que je fasse court, sinon ce sera trop long. Je suis super content de vous avoir ici, et c'est un honneur ! Alain, Joan, Orr, Caroline, Pierre-Alain, María et Thomas merci pour toutes les discussions que nous avons eues ces dernières années. Anne et María, merci pour tout : les encouragements, les retours, les conseils et la positivité !

Ensuite, je voudrais avoir un mot pour Irène et Frédéric et pour leur soutien ainsi que pour les précieux conseils et retours. Léo, merci pour le premier plan structurant ! Christina, merci pour ta confiance et encouragement depuis le début !

Je remercie particulièrement toutes les personnes de la communauté en cryptographie symétrique, avec qui j'ai pu travailler, discuter, réfléchir et en particulier tou.te.s les co-auteur.rice.s. C'est à chaque fois un plaisir de faire de la science ensemble ! Je souhaiterais aussi avoir un mot pour tou.te.s les étudiant.e.s à qui j'ai eu le grand plaisir d'enseigner plein de choses et avec qui j'ai réfléchi à des sujets super cool. J'aurais un mot en particulier pour Margot et Rachelle.

Ensuite je voudrais remercier l'équipe Crypto de Versailles où il fait bon vivre. De manière plus globale, je voudrais remercier le LMV et le département d'informatique pour l'environnement versaillais très agréable, en particulier Béatrice, Sandrine, Yann et Zoubida pour ces mots justes qui sont tombés au bon moment. L'équipe COSMIQ : merci de m'accueillir encore ! C'est toujours un grand plaisir ! Il y en a une autre équipe Inria, dans le grand Est (où il fait super froid), que je vais aussi remercier de m'avoir accueilli au bon moment. I would also like to thank the DS Group at Radboud University and the ESCADA team : I think I didn't learn that much stuff in a such short period of time !

Je voudrais aussi remercier le Smash, le Tarti, le Lol, Genneton, la bande à Benji, LPM et tou.te.s les copaines pour tous les moments super cools et pour le fun nécessaire, mais aussi pour vos encouragements. Vous êtes les meilleur.e.s ! Et enfin, alors que ce manuscrit et ma réflexion est en proie au doute, il y en a une avec qui il n'y en a pas : Camille, merci pour tout.

# Avant-propos

Ce document est rédigé en Français, le langage dans lequel je suis le plus à l'aise pour expliquer ma pensée de manière claire et (pas forcément) concise. De plus, les résultats présentés ici sont majoritairement expliqués et détaillés dans des articles écrits en anglais. Il est donc peut-être plus utile de les revisiter en français. Enfin, je suis relativement convaincu qu'une langue est faite pour être utilisée, à l'oral comme à l'écrit et dans l'ensemble des domaines possibles, reflétant une diversité que je considère comme souhaitable : une langue qui n'évolue plus est une langue qui est peut-être en soi déjà morte.

Poser à l'écrit une partie de ma réflexion n'est pas chose aisée et j'espère que les futur.e.s lecteurs et lectrices ne m'en tiendront pas rigueur quand ce que j'écris ici deviendra peut-être caduque. Alors que l'on essaye en science de présenter de manière claire les hypothèses et leur contexte d'utilisation, il y aura dans ce document des avis qui sont encore assez flous, que j'hésite à écrire, afin de ne pas fixer dans le marbre des réflexions que j'espère en constante évolution. J'incite donc particulièrement le lecteur ou la lectrice au scepticisme quant aux remarques non prouvées présentes dans ce document. En effet, une réflexion personnelle est forcément en proie à certains biais venant de chacune de nos compétences ou spécificités. C'est la raison pour laquelle j'incite les lecteur.rice.s à prendre le temps et à douter le plus possible dans un but d'amélioration de la qualité de nos réflexions communes. Ceci est d'autant plus vrai car nous sommes ensemble dans un monde qui nous demande d'aller de plus en plus vite, augmentant drastiquement les probabilités d'erreur de raisonnement.

Ce document a pour ambition de présenter un ensemble de travaux de recherche, ces travaux étant principalement le fruit d'échanges avec un grand nombre de personnes (que je remercie encore une fois), échanges que j'ai trouvés très intéressants et instructifs. Ils m'ont permis de faire évoluer grandement ma manière de voir les choses y compris au delà du domaine de la cryptographie.

Pour conclure cet avant-propos, j'espère intimement que ce document ne marquera aucunement une fin, mais plutôt qu'il suscitera de nouvelles réflexions, convergeant de manière plus fine vers la réponse à la grande question : « Qu'est-ce qu'un algorithme cryptographiquement sûr ? »

# Introduction

La cryptographie est la science qui s'intéresse à assurer la confidentialité, l'intégrité et l'authenticité des messages transmis sur un canal non-sécurisé. Depuis les années 1940 et les travaux de Claude Shannon [Sha49], nous savons que tout chiffrement pratique ne peut être inconditionnellement sûr. Ceci se voit très simplement grâce au formalisme provenant de la théorie de l'information. Par conséquent aucun algorithme cryptographique ne peut être prouvé sans des hypothèses plus ou moins fortes et plus ou moins réalistes sur les primitives sous-jacentes. On ne sait pas à ce jour garantir que ces hypothèses sont vérifiées.

### De l'importance de la cryptanalyse

Afin d'avoir une intuition permettant de juger si ces hypothèses sont réalistes ou non, nous avons besoin de cryptanalyse, c'est-à-dire d'analyse de sécurité des primitives, faite par le plus de personnes possible. Plus il y a de cryptanalyse et de cryptanalystes, plus on peut avoir confiance dans les primitives et donc dans les algorithmes cryptographiques. Ceci conforte le premier grand principe de la cryptographie émis par Auguste Kerckhoffs en 1883 [Ker83] :

*La sécurité d'un cryptosystème doit reposer exclusivement sur le secret de la clef.*

Ce principe est une question de résilience : si la sécurité du système cryptographique repose sur un secret autre que la clef, par exemple une partie de ses spécifications qui n'est pas publique, alors il y a naturellement moins de cryptanalyse sur ledit système ce qui le rend moins sécurisé. En ne respectant pas le principe de Kerckhoffs, on prend donc le risque de ne pas identifier certaines faiblesses, laissant la porte ouverte à un grand nombre d'attaques potentiellement dévastatrices. Lorsque celles-ci sont mises en œuvre et mettent en danger les données des utilisateurs, il est déjà trop tard. Il faut donc déployer des primitives cryptographiques ayant fait l'objet de nombreuses cryptanalyses. Ceci requiert du temps.

Par conséquent, nous nous intéressons à nous prémunir contre les attaques qui pourraient mettre en danger la sécurité des communications. Comme nous ne pouvons pas garantir entièrement que nos hypothèses sont vérifiées, deux cas de figure se présentent : ou bien la primitive est « cassée » ou bien elle ne l'est pas. Dans le premier cas, cela impose qu'on n'utilise plus la primitive et qu'on ne la recommande plus. De plus, les attaques (comme la construction) permettent

d'apprendre ce qu'il ne faut plus faire et donc d'éviter des faiblesses en vérifiant qu'elles ne s'appliquent pas à d'autres chiffrements. Dans le deuxième cas, la cryptanalyse réalisée peut demander un trop grand temps de calcul ou attaquer moins de tours dans la primitive ou une variante de celle-ci. La différence entre le coût de l'attaque et un coût atteignable et/ou entre la variante attaquée et le système d'origine nous garantit (ou non) une confiance dans la primitive considérée. Quantifier précisément la marge de sécurité que l'on a sur un chiffrement est un problème non trivial.

En revanche, ce qui est sûr, c'est que moins de cryptanalyse induit moins de confiance et une plus faible marge de sécurité. Il convient donc de rester critique et continuer à défendre la nécessité de la cryptanalyse qui reste à ce jour une base nécessaire à la cryptographie.

### Vers de nouveaux compromis performance-sécurité

Aujourd'hui, plusieurs branches de la cryptographie s'intéressent à construire des chiffrements et protocoles offrant d'autres fonctionnalités que la confidentialité, l'authenticité et l'intégrité des messages transmis. On peut citer par exemple l'essor du chiffrement complètement homomorphe [Gen09] qui permet théoriquement de réaliser des calculs sur des données chiffrées sans les révéler. Dans ce contexte, et afin d'obtenir un chiffrement complètement homomorphe ayant des performances acceptables, l'utilisation d'une technique hybride est privilégiée, demandant des chiffrements symétriques dont les métriques de coût sont différentes des cas classiques. De plus l'essor en cryptographie symétrique de la cryptographie dite « à bas coût » a vu un grand nombre de nouveaux chiffrements proposés ces dernières années.

Ainsi est apparu récemment un grand nombre de chiffrements cherchant à minimiser différentes métriques de coût, liées à leur implémentation pour des environnements matériels ou logiciels ou bien à une utilisation comprise dans un protocole plus avancé de cryptographie.

Or, il y a un compromis entre le coût d'un algorithme et la sécurité qu'il offre. On essaye donc de trouver cette limite. Cependant il ne faut probablement pas que l'envie de réduire le coût affecte trop la sécurité ou du moins formuler un avertissement clair sur la marge de sécurité que l'on offre. Pour aider à garantir un niveau de sécurité suffisant, je pense qu'il faut tout simplement faire de la cryptanalyse pour éviter en amont des problèmes qui pourraient être désastreux.

### Un enjeu majeur pour la société

Comme pour tout domaine, il faut se demander quel est notre impact et l'intérêt de notre science dans la société. Les données personnelles et l'information ont une valeur monétaire. L'explosion de la quantité de données exposées aujourd'hui nécessite de les protéger. En effet, on ne compte plus le nombre de cyberattaques et il semble difficile de s'en prémunir dans un cas pratique d'utilisation. La cryptographie (et donc la cryptanalyse) permettent d'identifier exactement où les problèmes peuvent survenir, mais surtout cette science permet

d'offrir à chacun la capacité de protéger ses informations. On oublie peut-être un peu trop souvent l'impact que peuvent avoir en pratique les failles de sécurité : au delà du simple vol de données, elles peuvent paralyser des hôpitaux ou des universités. Mais elles ont aussi un coût économique : ces failles coûtent cher à éviter et elles coûtent très cher une fois qu'elles sont exploitées. Par ailleurs, protéger ses informations est absolument nécessaire aujourd'hui pour éviter des pressions, par exemple sur des lanceurs d'alerte dont l'utilité n'est pas à démontrer. L'expression et la circulation d'une information libre est garante d'un bon fonctionnement de la démocratie. La cryptographie permet d'assurer que chaque citoyen puisse communiquer librement sans être jugé ou inquiété pour l'avoir fait. Préserver la vie privée est un droit humain comme l'explique l'article 8 de la Convention Européenne des Droits de l'Homme. De plus, la censure et le contrôle de l'information par un petit nombre de personnes est, cela va sans dire, un énorme problème sociétal et met directement à mal la démocratie.

Il y a quelques années, je n'aurais peut-être pas écrit cela, considérant que le droit à la protection de la vie privée était acquis. Mais, des prises de positions et événements récents en France, en Europe et dans le monde me poussent aujourd'hui à insister sur cet aspect.

Bien que la cryptographie ne réponde pas à l'ensemble du problème qui est bien plus complexe, elle sert de brique de base dans notre monde aujourd'hui numérisé et permet d'identifier les points de faiblesse en matière de sécurité. Plus précisément, nous aurons toujours besoin de confiance dans des entités physiques, mais l'utilisation de systèmes cryptographiques éprouvés avec une implémentation vérifiée et vérifiable permet de supprimer tout un ensemble de failles techniques et de faire reposer la confiance sur la science. C'est dans ce contexte que s'inscrit ce travail et ce document : nous verrons des éléments de cryptanalyse permettant d'en apprendre un peu plus sur ce qui fait la sécurité dans une construction cryptographique.

### Perspectives générales

Mes travaux de cryptanalyse s'inscrivent dans le contexte suivant : je suis passionné par les attaques qui exploitent une représentation polynomiale pour décrire une partie des chiffrements. En effet, toute transformation peut se décrire à l'aide d'une représentation polynomiale. En cryptographie, les polynômes définissant les transformations sont *a priori* publics mais il est impossible de les calculer sauf pour un nombre restreint de tours. Par ailleurs, même lorsque nous connaissons tout ou partie des polynômes définissant les transformations cryptographiques nous ne savons pas toujours comment les exploiter. Enfin, il manque à ce jour des algorithmes permettant d'obtenir les critères cryptographiques usuels des primitives même en ayant une description polynomiale.

Je suis intimement persuadé qu'il est possible d'améliorer les attaques existantes en prenant en compte plus d'informations dans les représentations polynomiales, notamment en identifiant des dépendances plus précises et non nécessairement linéaires dans les différentes composantes des chiffrements.

### Structure du document

Dans ce contexte, le premier chapitre présentera trois cryptanalyses sur deux chiffrements et une fonction de hachage. Ces cryptanalyses exploitent principalement une représentation polynomiale qui est dans chaque cas entièrement calculable. Ensuite, le deuxième chapitre détaillera certaines attaques algébriques et intégrales avec une application en tête : le chiffrement par bloc `Pyjamask`. Enfin, le troisième et dernier chapitre abordera des cryptanalyses génériques sur deux constructions avec des modèles de sécurité différents, et ouvrira le débat sur des constructions un peu plus récentes.

Après un bref chapitre de conclusion décrivant certaines perspectives ouvertes par le travail présenté ici, je présenterai brièvement les activités scientifiques que j'ai réalisées jusque là, que ce soit dans la recherche (publications, comités de programme, projets de recherche), dans l'enseignement universitaire (thèses encadrées, cours et pédagogies, projets d'enseignement), dans la médiation scientifique ou dans l'engagement organisationnel de l'université.

# Table des matières

# Chapitre 1

# Cryptanalyse dédiée : morceaux choisis

Dans ce chapitre, nous motivons au travers d'exemples l'intérêt d'une analyse incessante et approfondie des algorithmes cryptographiques que nous utilisons. En effet, la sécurité des procédés cryptographiques repose sur des hypothèses que devraient vérifier des constructions de base appelées primitives. Typiquement, dans le cas des chiffrements par bloc, les modes opératoires qui permettent de chiffrer des messages de longueur arbitraire peuvent être prouvés sûrs, à condition que les chiffrement par blocs sous-jacents se comportent de manière idéale [1]. Actuellement, c'est essentiellement la cryptanalyse qui fournit des arguments tendant à montrer que ces propriétés idéales sont atteintes.

Étant passionné de cryptanalyse, j'ai participé à plusieurs travaux dédiés sur divers chiffrements, notamment sur le chiffrement FLIP [DLR16], sur les registres à décalages à rétroaction linéaire filtrés [CR16], sur les attaques par invariant [BCLR17], sur Ketje [FNR18], MORUS [AEL+18] ou encore sur la fonction de hachage Troika [BFR24]. Aujourd'hui, je cherche à dégager des techniques de cryptanalyse qui exploitent une représentation polynomiale que l'on aurait pour un faible nombre de tours, principalement en ayant des dépendances et des relations que l'on peut calculer exactement entre les différentes informations qu'un attaquant peut obtenir. Je cherche à sortir des attaques exploitant des propriétés statistiques, en exploitant des informations que l'on peut obtenir avec probabilité 1. C'est la raison pour laquelle dans ce chapitre je choisis de présenter les cryptanalyses réalisées sur `GEA` [BDL+21], `Subterranean` [DMMR20] et Keccak [HNR21].

Ainsi, ce chapitre fait état de trois travaux d'analyse de sécurité et met en évidence des améliorations d'attaques déjà existantes. Appliquées à certains chiffrements, elles prévoient ainsi en amont et de manière théorique de potentielles attaques dévastatrices. Avec une vision un peu plus positive, nous montrons en

1. *i.e.* comme un ensemble de permutations choisies au hasard selon une distribution uniforme.

fait que la majorité des constructions sont plutôt sûres, les attaques coûtant souvent cher en pratique, ce qui augmente la confiance dans les algorithmes cryptographiques.

## 1.1 Cryptanalyse de GEA-1

Dans cette section, nous décrivons une attaque dévastatrice sur le chiffrement à flot `GEA-1`, un standard utilisé dans la communication par paquets dans la téléphonie mobile. Cette attaque a été publiée à EUROCRYPT en 2021 [BDL+21] et a été réalisée en collaboration avec Christof Beierle, Patrick Derbez, Gregor Leander, Gaëtan Leurent, Håvard Raddum, David Rupprecht et Lukas Stennes.

### 1.1.1 Contexte

Le GPRS (General Packet Radio Service) est un prolongement de la norme GSM (Global System for Mobile communications) améliorant les performances de cette dernière. La technologie GPRS a été largement déployée autour des années 2000 et est encore utilisée de nos jours notamment en tant que technologie de substitution des réseaux 4G ou 5G. Afin de communiquer de manière sécurisée, des chiffrements y sont utilisés. Dans le cas du chiffrement GPRS, deux chiffrements à flot sont utilisés : `GEA-1` et `GEA-2`. Ces algorithmes ont été conçus par le groupe ETSI Security Algorithms Group of Experts (SAGE) en 1998. Un rapport technique qui peut être trouvé dans [Bro01] explique que des algorithmes devaient être exportables en prenant en compte les différentes restrictions étatiques en vigueur à cette époque.

Dans le cas de la France, ces régulations étaient plutôt strictes. En particulier, à la fin des années 90, il était nécessaire d'avoir l'accord du Premier Ministre pour l'utilisation, l'importation et l'exportation de moyens cryptographiques forts [Koo13]. Plus précisément, les décrets 98-206 [2] et 98-207 [3] permettaient des exceptions à cet accord du Premier Ministre, pour les cas d'usage de la cryptographie civile, et seulement pour des algorithmes dont le coût de recouvrement de la clef ne dépassait pas $2^{40}$ essais.

### 1.1.2 Spécifications

#### Structure de `GEA`

En 2011, Nohl et Melette ont montré qu'il était facile d'écouter le trafic des communications GPRS [NM11], démontrant la nécessité d'activer les algorithmes cryptograhiques. De plus, Nohl et Melette ont donné des descriptions (incomplètes) des chiffrements propriétaires `GEA-1` et `GEA-2` obtenues par des techniques de rétro-ingénierie. Principalement `GEA-1` est un chiffrement à flot d'un état interne de 96 bits séparé en trois registres $A$, $B$ et $C$ de tailles respectives 31, 32 et 33 bits et utilise une clef de 64 bits. Une fonction non-linéaire est

2. https://www.legifrance.gouv.fr/jorf/id/JORFTEXT000000753702
3. https://www.legifrance.gouv.fr/jorf/id/JORFTEXT000000753703

utilisée pour initialiser les registres internes. La fonction de mise à jour est une fonction linéaire. Plus précisément il s'agit de registres à décalage à rétroaction linéaire (LFSR). Enfin, la fonction de filtrage est de degré multivarié 4.

`GEA-2` possède la même structure que `GEA-1` mais a un état interne plus grand, formé par l'ajout d'un quatrième LFSR noté $D$. La fonction de filtrage peut être vue comme l'addition bit à bit de trois (ou quatre pour `GEA-2`) fonctions identiques, prenant en entrée 7 bits indépendants et situés dans des positions dont le choix dans chacun des registres ne suit pas de schéma particulier. La sortie de la fonction de filtrage donne un bit de suite chiffrante. Une représentation visuelle de `GEA-1` et de `GEA-2` est donnée par la figure 1.1.

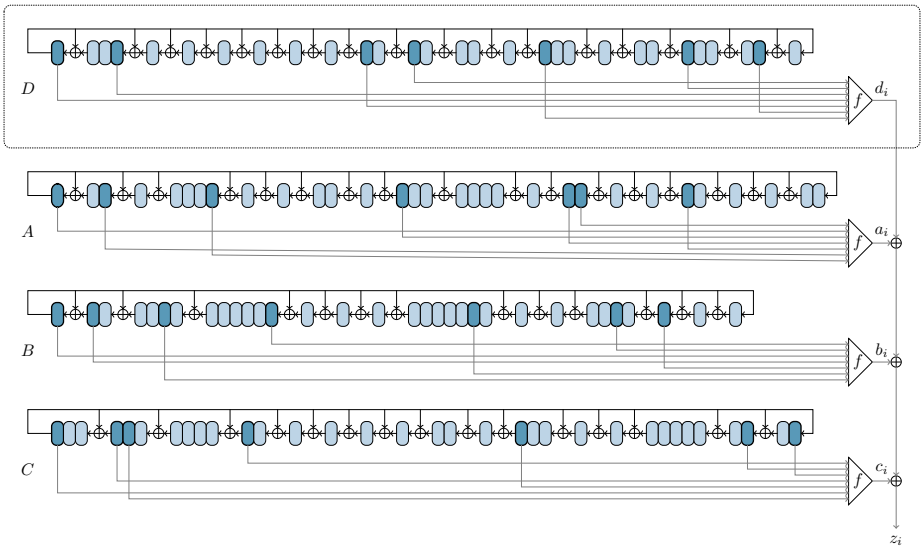


FIGURE 1.1 – Génération de la suite chiffrante pour `GEA-1` et `GEA-2`. Le registre $D$ n'est présent que pour `GEA-2`.

Dans leur présentation, Nohl et Melette affirment que `GEA-1` est vulnérable aux attaques algébriques, principalement du fait de la linéarité de la fonction de mise à jour. Cependant les détails de cette attaque ne sont pas publics.

### Initialisation

Tout chiffrement doit avoir un caractère non-déterministe, dans le sens où deux chiffrements du même message avec la même clef ne doivent pas produire le même message chiffré. Dans un chiffrement symétrique, authentifié ou non, ceci est permis par l'utilisation d'un vecteur d'initialisation ($IV$) ou d'un nonce ($N$). Ce sont des valeurs publiques et, comme la deuxième terminologie l'explicite, ces valeurs ne doivent pas être utilisées deux fois avec la même clef secrète (par exemple un compteur peut être utilisé pour différencier des sessions différentes). Pour les chiffrements à flot, il faut donc définir une procédure qui associe à la clef

secrète et au vecteur d'initialisation l'état initial du (des) registre(s), point de départ de la génération de suite chiffrante. Comme le vecteur d'initialisation est public (et parfois peut être choisi par l'adversaire), il faut que cette procédure soit aussi cryptographiquement sûre, dans un modèle où l'attaquant peut choisir plusieurs vecteurs d'initialisation différents.

**`GEA-1`.** Dans le cas de `GEA-1`, la procédure d'initialisation consiste en l'itération d'un NFSR (Non-linear Feedback Shift Register) de taille 64 bits noté $S$. Le registre $S$ est initialisé à 0, puis il est mis à jour 97 fois en additionnant un à un à la case 63 les 32 bits du vecteur d'initialisation $IV$, un bit *dir* (pour la direction de la communication) et les 64 bits de la clef $K$. Le registre est ensuite mis à jour 128 fois. L'initialisation du registre est décrite à la figure 1.2. La même fonction $f$ que celle de la génération de suite chiffrante est utilisée comme fonction de mise à jour de ce registre.

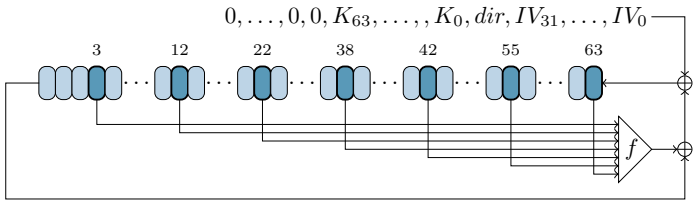


FIGURE 1.2 – Initialisation du registre $S$ de `GEA-1`.

Cette procédure produit donc une suite de 64 bits (secrets) que nous notons $s = s_0, \ldots, s_{63}$ (ce sont les valeurs contenues dans le registre $S$). Ceux-ci doivent être utilisés pour initialiser les registres $A$, $B$ et $C$ qui seront ensuite combinés et mis à jour pour générer la suite chiffrante. Pour cela, tous les registres sont initialisés à zéro et chaque registre est mis à jour 64 fois, où un bit particulier de $s$ est additionné dans le registre avant d'appliquer la fonction de mise à jour. Cependant, les mêmes bits de $s$ ne sont pas ajoutés aux mêmes moments dans les trois registres. Plus précisément, $s$ est décalé cycliquement de 16 positions vers la gauche pour le registre $B$ et de 32 positions pour le registre $C$ : la suite de bits insérée dans le registre $A$ est $s_0, s_1, \ldots, s_{63}$ ; la suite de bits insérée dans le registre $B$ est $s_{16}, s_{17}, \ldots, s_{63}, s_0, \ldots, s_{15}$ ; la suite de bits insérée dans le registre $C$ est $s_{32}, s_{33}, \ldots, s_{63}, s_0, \ldots, s_{31}$.

**`GEA-2`.** L'initialisation de `GEA-2` suit la même stratégie et réemploie les composants de `GEA-1`. Cependant, `GEA-2` utilise un registre $W$ de taille 97 bits (au lieu d'un registre $S$ de taille 64 bits) avec la même fonction de mise à jour (les positions déterminant les entrées de la fonction étant naturellement différentes). La clef, le vecteur d'initialisation et le bit *dir* sont aussi présents et ajoutés dans le registre $W$ de la même manière en réalisant 97 mises à jour pour insérer ces valeurs, puis, le registre $W$ est mis à jour 194 fois.

Enfin, l'insertion des valeurs de ce registre se fait de la même manière que pour `GEA-1`, mais avec des décalages cycliques différents. Principalement, le registre $D$ commence par le bit 0, quand les registres $A$, $B$ et $C$ commencent par les bits 16, 33 et 51 respectivement.

### 1.1.3 Principes de construction

Avant d'expliquer l'idée de l'attaque, nous pouvons passer en revue les différents éléments présents dans ce chiffrement, car plusieurs ont été conçus en ayant en tête un certain nombre d'attaques possibles.

Tout d'abord, la fonction de filtrage $f$ est équilibrée et a une non-linéarité élevée. Elle peut être décomposée en deux fonctions courbes de six variables. La fonction de filtrage est aussi de degré quatre et d'immunité algébrique quatre.

Les positions des entrées de la fonction $f$, que ce soit pour l'initialisation comme pour la génération de suite chiffrante ne suivent pas de schéma particulier permettant *a priori* de lier l'information entre différents registres et ce qui se produit à différents instants.

L'utilisation de registres linéaires de petite taille permet d'avoir une conception très efficace pour implémenter l'algorithme en matériel. Ces registres sont aussi utilisés en mode Galois et non en mode Fibonacci.

Le fait d'additionner la sortie des trois ou des quatre registres pour former la suite chiffrante permet d'éviter complètement les attaques par corrélation [Sie85]. L'utilisation d'au moins trois registres permet aussi d'éviter une attaque de type *meet-in-the-middle* [4].

L'initialisation hautement non-linéaire réalisée à l'aide des registres dédiés semble être suffisamment solide pour mélanger correctement le vecteur d'initialisation (public) et la clef (secrète). De plus, le choix d'initialiser les registres $A$, $B$, $C$ (ou $D$ pour `GEA-2`) avec un décalage différent pour chacun est important pour éviter les relations simples entre les entrées des fonctions de filtrage.

### 1.1.4 Attaques sur `GEA-1` et `GEA-2`

L'utilisation de registres à décalage et à rétroaction linéaire ainsi que l'initialisation induit deux propriétés. D'un côté, la fonction qui, à la valeur des registres $S$ ou $W$ (après avoir injecté les clef et le vecteur d'initialisation) associe la valeur initiale des registres $A$, $B$, $C$ et/ou $D$ est linéaire. D'un autre côté, la sortie de la suite chiffrante est de degré multivarié constant (4, le degré de la fonction de filtrage) en les états initiaux des registres dédiés à l'initialisation.

#### `GEA-1`

On note $s \in \mathbb{F}_2^{64}$ la valeur (secrète) contenue dans le registre $S$ après la phase d'initialisation. Comme le reste de la procédure d'initialisation est linéaire et que les registres $A$, $B$ et $C$ sont de tailles respectives 31, 32 et 33, il existe trois

4. i.e., trouver $(x, y) \in X \times Y$ tel que $f(x) + g(y) = 0$ pour deux fonctions $f$ et $g$.

matrices notées respectivement $M_A \in \mathbb{F}_2^{31\times 64}$, $M_B \in \mathbb{F}_2^{32\times 64}$ et $M_C \in \mathbb{F}_2^{33\times 64}$ telles que

$$\begin{aligned}\alpha &= M_A s \ ,\\ \beta &= M_B s \ ,\\ \gamma &= M_C s \ ,\end{aligned}$$

où $\alpha$, $\beta$ et $\gamma$ sont les états internes des trois registres après initialisation.

Les matrices $M_A$, $M_B$ et $M_C$ sont toutes de rang plein, permettant d'assurer que le nombre d'états initiaux possibles pour les registres $A$, $B$ et $C$ est respectivement $2^{31}$, $2^{32}$ et $2^{33}$. Cependant, en regardant le nombre d'états possibles conjointement pour une paire de registres un phénomène étrange apparaît.

En notant $T_{AC} \coloneqq \ker(M_A) \cap \ker(M_C)$ et $U_B \coloneqq \ker(M_B)$, on observe :

1. $\dim(T_{AC}) = 24$ et $\dim(U_B) = 32$ ,
2. $U_B \cap T_{AC} = \{0\}$ .

Par conséquent, nous pouvons garantir que le nombre d'états possibles pour les registres $A$ et $C$ conjointement est $2^{64-24} = 2^{40}$.

Il s'en suit alors très rapidement une attaque en $2^{40}$ opérations sur `GEA-1` en appliquant une stratégie de type diviser pour mieux régner. D'après ce qui précède, on peut décomposer $\mathbb{F}_2^{64}$ en la somme directe de trois espaces vectoriels $U_B$, $T_{AC}$ et $V$ où $V$ est de dimension 8. Ainsi les valeurs $\alpha$, $\beta$ et $\gamma$ peuvent s'écrire

$$\begin{aligned}\beta &= M_B(u+t+v) = M_B(t+v)\\ \alpha &= M_A(u+t+v) = M_A(u+v)\\ \gamma &= M_C(u+t+v) = M_C(u+v) \ ,\end{aligned}$$

où $u \in U_B$, $t \in T_{AC}$ et $v \in V$.

Soit $n$ un entier naturel. On pré-calcule dans un premier temps les valeurs de sortie $(b_i)_{0\leq i\leq n}$ (voir la figure 1.1) pour chaque valeur initiale possible $\beta$ pour le registre $B$ (ce qui coûte $2^{32}$) et on sauvegarde ces valeurs dans une table triée par la valeur de sortie $(b_i)_{0\leq i\leq n}$. Ainsi, en observant une partie de la suite chiffrante $(z_i)_{0\leq i\leq n}$, il est possible de parcourir les $2^{40}$ états possibles pour les registres $A$ et $C$ conjointement, calculer pour chacun de ces états la valeur correspondante $(a_i + c_i + z_i)_{0\leq i\leq n}$ et chercher dans la liste triée cette même valeur. Si cette valeur est absente, cela signifie que nous n'avons pas la bonne valeur initiale dans les registres $A$ et $C$. Sinon, nous retrouvons l'état initial de chaque registre. Il faut cependant prendre un $n$ suffisamment grand pour éviter les faux positifs. Plus précisément $n$ doit être supérieur à l'entropie qui est ici de 64. Enfin, le recouvrement de ces états permet de retrouver la clef maître.

#### `GEA-2`

Dans la version améliorée de `GEA-1`, il y a quatre registres et nous n'observons pas du tout le phénomène ayant permis l'attaque décrite ci-dessus. Cependant,

le degré de l'expression algébrique de la suite chiffrante en fonction du secret est toujours constant. Ainsi, en connaissant suffisamment de bits de la suite chiffrante, il serait possible de retrouver le système par simple linéarisation (voir section 2.1.1). En revanche, les sessions dans `GEA-2` sont limitées à 1600 octets, ce qui rend le système sous-déterminé si l'on considère tous les monômes comme des variables indépendantes.

Cependant rien n'est perdu pour autant et l'on peut toujours trouver des techniques plus élaborées permettant de retrouver l'état initial. Tout d'abord, le nombre de monômes présents dans l'expression algébrique est limité à

$$1 + \sum_{i=1}^{4} \left[ \binom{29}{i} + \binom{31}{i} + \binom{32}{i} + \binom{33}{i} \right] = 152682\,.$$

Ceci est dû simplement à la structure de `GEA-2`. Ce nombre dépasse $1600 \times 8$, mais en faisant des hypothèses sur certains bits des registres, on peut construire (en fonction de ces hypothèses) des équations avec moins de monômes. La technique ci-dessus est extrêmement classique et parfois dévastatrice comme pour A5/1 [Gol97, And94] ou bien pour le chiffrement FLIP [DLR16]. Cette technique est appelée *Guess-and-Determine*, ou bien l'approche hybride [BFP09]. Seule, elle ne permet pas d'obtenir une attaque en moins de $2^{64}$ opérations.

Par contre, il est tout à fait possible de combiner cette technique avec une technique de type diviser pour mieux régner. Plus précisément, il est possible de faire des hypothèses sur les registres $A$ et $D$, permettant de dériver des équations linéaires dont l'expression est indépendante de la valeur contenue dans $A$ et $D$. Plus formellement, sous l'hypothèse réalisée, on calcule un ensemble de masques sur $8 \times 1600$ bits[5], de telle sorte que le produit scalaire de chacun des masques avec $a + d$ soit nul où $a + d$ désigne la suite générée par les deux registres $A$ et $D$ (voir la figure 1.1). Ainsi, il ne reste plus qu'une dépendance entre les registres $B$ et $C$, dont les valeurs peuvent être retrouvées en faisant une fusion de deux listes de tailles respectives $2^{32}$ et $2^{33}$. Cette attaque peut être améliorée, par exemple en pré-calculant les pivots de Gauss nécessaires, ce qui conduit à une attaque en $2^{45.1}$ applications de `GEA-2`.

### 1.1.5 Synthèse et perspectives

Il est clairement contre-productif d'utiliser des chiffrements propriétaires (comme pour A5/1) avec des spécifications non-publiques, rendant le travail de l'analyse de sécurité plus difficile. Si les chiffrements ne sont pas publics, on ne peut pas savoir si la cryptographie que l'on utilise tous les jours n'est pas vulnérable à des attaques, indiquant la nécessité de suivre l'un des grands principes de la cryptographie émis par Auguste Kerckhoffs en 1883 [Ker83]. Il s'agit ici d'une question de confiance : si la spécification reste secrète, cela signifie qu'il faut avoir confiance en une analyse réalisée par un petit nombre de personnes. D'ailleurs il est illusoire de considérer que la spécification d'un algorithme cryptographique reste privée *ad vitam æternam* [LPSS24].

5. taille des sessions dans `GEA-2`.

La propriété permettant l'attaque sur `GEA-1` est extrêmement spécifique. En effet, pour deux matrices arbitraires de rang plein, la probabilité que la dimension de l'intersection de leurs noyaux soit grande est extrêmement faible. Ceci est facile à démontrer avec des matrices aléatoires, mais des expériences réalisées montrent que cette probabilmité reste tout aussi faible si ces matrices sont dérivées d'une initialisation « à la `GEA` », c'est-à-dire avec des LFSRs en mode Galois et des décalages cycliques de la valeur secrète rajoutée bit à bit dans chacun des registres.

Le coût mémoire de l'attaque sur `GEA-1` est relativement élevé et peut être amélioré en calculant la table indépendamment pour chaque $v \in V$, sans changer le coût de l'attaque. Cette amélioration a été proposée à EUROCRYPT en 2022 dans [AD22].

Enfin, en exploitant du pré-calcul et des techniques plus récentes, nous montrons qu'on arrive à améliorer (parfois à la marge) les attaques. En effet, selon l'adage devenu célèbre en cryptographie, les attaques ne font que s'améliorer.

`GEA-1` et `GEA-2` sont des chiffrements un peu anciens, conçus à une époque où 64 bits de clef étaient suffisants pour se prémunir de la recherche exhaustive. Il convient donc de souligner que, même sans nos attaques, il ne faut plus utiliser de chiffrements travaillant sur d'aussi petites tailles. D'ailleurs, à CRYPTO 2024 dans [ACC+24], il est montré qu'un attaquant passif peut déchiffrer la majorité des communications GSM en un temps très rapide.

Il reste d'autres algorithmes conçus à l'époque de `GEA-1` pour lesquels il y a de grandes chances de trouver d'autres attaques, pas forcément aussi dramatiques que celle que nous avons réalisée sur `GEA-1`. En effet, des techniques de cryptanalyse nouvelles ont émergées depuis la conception de ces chiffrements. Cela a par exemple été réalisé sur HALFLOOP dans [DDLS22, LRS23].

## 1.2 Subterranean 2.0

En 2019 a commencé la compétition du NIST [6] (National Institute of Standards and Technology) sur la cryptographie dite « à bas coût », recevant en mars 2024 un ensemble de 56 soumissions d'algorithmes cryptographiques symétriques exploitant diverses stratégies de conception. J'ai participé à cette compétition en analysant différents chiffrements (dont en particulier un, `Pyjamask`, dont nous décrirons l'attaque au chapitre 2), mais aussi en proposant le chiffrement `Subterranean` et en analysant sa sécurité. C'est l'objet de cette section, qui décrira les briques de base et les idées derrière la conception et l'analyse de ce chiffrement. La suite d'algorithmes appelée `Subterranean 2.0` (permettant de réaliser du hachage et du chiffrement authentifié) a été conçue avec Joan Daemen, Pedro Maat Costa Massolino et Alireza Mehrdad. La description peut se trouver sur le site du NIST, mais aussi dans un article paru dans le journal *IACR Transactions on Symmetric Cryptology* [DMMR20].

L'objectif visé par la conception de `Subterranean` est d'avoir une implémentation matérielle extrêmement efficace, ce qui a été réussi [TMC+21]. Mal-

6. https://csrc.nist.gov/Projects/Lightweight-Cryptography

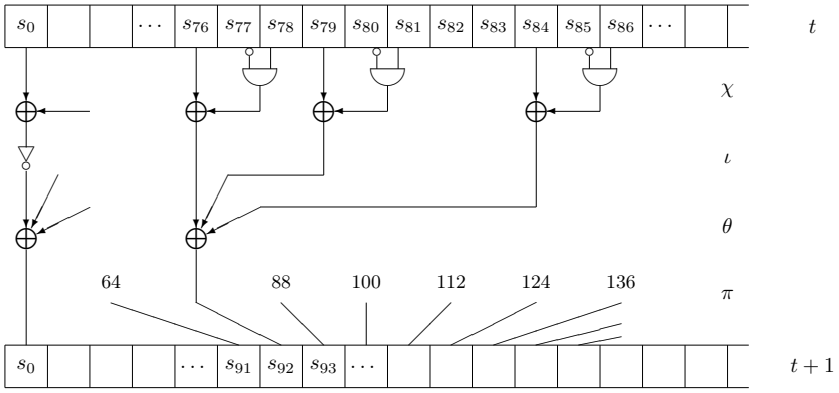


FIGURE 1.3 – Fonction de tour de `Subterranean`.

heureusement, le peu d'analyse de sécurité qu'a reçu `Subterranean` ne lui a pas permis d'aller plus loin que le deuxième tour de sélection de la compétition du NIST. Pour l'instant le chiffrement `Subterranean` n'a pas été cassé, malgré quelques analyses intéressantes que nous détaillons plus loin

### 1.2.1 Description

`Subterranean` est une primitive cryptographique que l'on peut facilement utiliser pour faire du hachage ou un chiffrement à flot. La construction originale est aussi vieille que moi : sa description originelle date de 1993 [CDGP93] et est aussi présentée dans la thèse de Joan Daemen [Dae95]. Dans cette construction revisitée, nous avons supprimé le *buffer* utilisé et modifié les modes de chiffrement et de hachage en utilisant des constructions de type duplex [BDPV12a, BDPV12b] et éponge [BDPV07, BDPV11].

#### La fonction de tour

La fonction de tour, notée $R$, de `Subterranean` opère sur un état de 257 bits et consiste en la composition de quatre fonctions notées $\pi$, $\theta$, $\iota$ and $\chi$ :

$$R = \pi \circ \theta \circ \iota \circ \chi \,.$$

En notant $s$ l'état interne de `Subterranean` et $s_i$ les bits de l'état pour $i$ allant de 0 à 256, les quatre fonctions peuvent être décrites très simplement par les expressions suivantes où chaque indice est naturellement considéré modulo 257. Une description visuelle de la fonction de tour est présentée à la figure 1.3. Pour tout $i$ allant de 0 à 256,

$$\begin{array}{lcll} \chi & : & s_i & \leftarrow & s_i + (s_{i+1}+1)s_{i+2}\,, \\ \iota & : & s_i & \leftarrow & s_i + \delta_i\,, \\ \theta & : & s_i & \leftarrow & s_i + s_{i+3} + s_{i+8}\,, \\ \pi & : & s_i & \leftarrow & s_{12i}\,, \end{array}$$

où $\delta_i = 1$ si $i = 0$ et 0 sinon.

### Absorption

Comme nous réalisons un objet de type duplex (voir section 3.1.1), il est nécessaire de pouvoir injecter de l'information dans l'état interne (typiquement un nonce ou les blocs du message à authentifier). Dans le cas de `Subterranean`, la quantité d'information absorbée peut être de 32 bits dans une utilisation avec une clef secrète ou de 8 bits sans clef (fonction de hachage). Ceci est nécessaire afin de garantir au moins 112 bits de sécurité. Pour absorber un bloc de message $\sigma$ où $|\sigma| \leq 32$, il faut d'abord appliquer la fonction de tour $R$, puis injecter dans l'état interne la suite $x = \sigma||1||0^{32-|\sigma|}$ de taille 33 bits (le *padding* utilisé est $10^*$) de la manière suivante. Pour tout $j$ allant de 0 à 32,

$$s_{12^{4j}} \leftarrow s_{12^{4j}} + x_j \,.$$

### Extraction

Pour compléter l'objet duplex et obtenir une primitive correspondant à une fonction de hachage, à une fonction à taille de sortie variable (XOF - *eXtendable Output Function*) ou à un chiffrement authentifié, il est aussi nécessaire de pouvoir extraire de l'information de l'état interne. Dans notre cas, nous avons choisi d'extraire 32 bits d'information en utilisant une technique similaire à celle utilisée dans le chiffrement Trivium [De 06] : nous sommons deux bits de l'état bien choisis pour générer la suite chiffrante. Plus précisément, la sortie $z$ de l'extraction est construite comme suit. Pour tout $j$ allant de 0 à 31,

$$z \leftarrow z||(s_{12^{4j}} + s_{-12^{4j}}) \,.$$

### Le mode duplex

La construction duplex est décrite dans [BDPV12a, BDPV12b] et sera succinctement décrite à la section 3.1. Dans le cas du chiffrement authentifié, nous avons choisi d'initialiser l'état à 0 et d'absorber pas à pas les 128 bits de clef, puis les 128 bits du nonce, en utilisant plusieurs appels au mode duplex, produisant ainsi l'état initial. Huit applications de la fonction de tour $R$ sont réalisées à blanc, c'est-à-dire sans injecter ni extraire d'information. Ensuite, le mode duplex peut être utilisé, en extrayant et absorbant successivement de l'information, pour chiffrer ou déchiffrer. Enfin huit autres applications de la fonction de tour $R$ sont réalisées avant de produire l'empreinte [7] $T$ de longueur 128 bits, au moyen de quatre extractions successives, entrecoupées de l'application de la fonction de tour.

7. Le Tag

#### La construction en éponge

Nous ne décrirons pas la construction en éponge [BDPV07, BDPV11] permettant de construire une fonction de hachage ou une fonction à taille de sortie variable (XOF). Dans le cas de `Subterranean`, nous avons choisi d'absorber le message à condenser sous forme de blocs de 8 bits mais aussi de réaliser deux applications de la fonction de tour $R$ après chaque absorption. Huit tours à blanc sont aussi réalisés entre la phase d'absorption et la phase d'extraction qui, elle, extrait toujours 32 bits.

### 1.2.2 Arguments de conception

#### La fonction $\chi$

Seule l'application $\chi : \mathbb{F}_2^n \to \mathbb{F}_2^n$ est non linéaire. Celle-ci est utilisée aujourd'hui dans un grand nombre de primitives cryptographiques, dont la plus connue est probablement la permutation Keccak-$p$ [BDPV14] utilisée dans le standard des fonctions de hachage `SHA-3` [SHA15]. Cette fonction n'est une permutation que pour $n$ impair ($n = 3$ pour Xoodoo [DHVV18] et $n = 5$ pour Keccak-$p$). Dans notre cas, on a $n = 257$. L'avantage du faible degré de la fonction de tour est qu'il est plus facile d'implémenter des contre-mesures aux attaques par canaux cachés (principalement la DPA [KJJ99]) en utilisant des schémas de masquage [GP99, CJRR99]. En effet, le coût des techniques de masquage [ISW03] augmente largement avec le nombre de portes logiques `AND` du circuit.

De plus, le fait de choisir une fonction opérant sur l'entièreté de l'état (et non sur plusieurs sous-parties en parallèle comme c'est le cas usuellement avec les boîtes-S) permet d'assurer que la fonction inverse de $\chi$ est de haut degré multivarié : dans notre cas, le degré de $\chi^{-1}$ est 128. Il est à noter que la formule exacte (sous forme polynomiale) de cette fonction a été déterminée récemment [LSMI22] et que l'ordre du sous-groupe des permutations engendré par $\chi$ a aussi récemment été trouvé et démontré dans [KK24] tout en dégageant des propriétés algébriques de $\chi$. D'autres travaux récents [SD24a, SD24b] ont aussi exhibé des propriétés intéressantes dont je ne parlerais pas car cela sort du cadre de ce document.

#### La fonction $\iota$

Il s'agit tout simplement de l'addition d'une constante de tour présente uniquement pour « casser » la symétrie induite par toutes les autres fonctions. La fonction $\iota$ modifie uniquement le bit de l'état à la position 0.

#### La fonction $\theta$

Cette opération permet une diffusion dans l'état tout en étant très creuse. Le choix des positions (+0, +3 et +8) permet d'assurer qu'une différence à l'entrée

de $\theta$ avec deux bits actifs formera nécessairement 4 bits actifs en sortie [8].

De plus, si l'on identifie l'état interne $s$ à un polynôme $s(X)$ à coefficients dans $\mathbb{F}_2$ modulo $1 + X^{257}$, alors l'application de $\theta$ peut être vue comme la multiplication par $1+X^{249}+X^{254}$ dans l'anneau $\mathbb{F}_2[X]/(1+X^{257})$. Ceci permet de montrer que l'inverse de $\theta$ est de poids de Hamming 127 ce qui révèle une grande diffusion dans le sens inverse. Nous pouvons aussi montrer que l'ordre de $\theta$ est $2^{16} - 1$, qui est maximal dans $\left(\mathbb{F}_2[X]/(1 + X^{257})\right)^{\times}$.

#### La fonction $\pi$

Cette fonction est un croisement de fils dans l'algorithme, rendant celui-ci bien plus efficace pour une implémentation matérielle mais très mauvais dans une implémentation logicielle. Vue différemment, cette fonction place côte-à-côte des bits qui sont éloignés de 12 positions. Ceci permet de garantir que chaque bit de l'état après deux applications de la fonction de tour dépend de 81 bits de l'état en entrée [9], assurant une grande diffusion, combinée naturellement avec $\chi$ et $\theta$. De plus, l'ordre de 12 est un générateur de $(\mathbb{Z}/257\mathbb{Z})^{\times}$. Ceci, combiné avec les propriétés algébriques de $\theta$, permet de calculer l'ordre de la couche linéaire dans `Subterranean` qui est 256.

### 1.2.3 Analyse de sécurité

Comme tout chiffrement, il n'est pas suffisant de justifier ses choix par des arguments élégants. Il est nécessaire et impératif de regarder le chiffrement en profondeur, avec une casquette de cryptanalyste, en cherchant maintenant à casser le chiffrement, pour ne pas réussir, mais toujours en espérant y arriver. Nous mentionnerons aussi les autres cryptanalyses que `Subterranean` a reçues qui sont naturellement nécessaires pour obtenir un niveau de confiance satisfaisant.

#### Sur le nombre de tours

Une manière d'attaquer un chiffrement authentifié de type duplex consiste à observer des liens imprévus entre l'initialisation et les premiers blocs de suite chiffrante. On peut tenter d'utiliser des valeurs de nonce $N$ ayant une certaine différence et exploiter des propriétés différentielles, trouver des biais en sortie de l'initialisation, ou encore réaliser des attaques de type cube. Ainsi, un choix de huit tours à blanc de degré algébrique 2 permet *a priori* de se prémunir contre ces attaques, même en prenant en compte des bornes plus fines sur le degré [BCD11]. De même, huit tours sont aussi nécessaires avant de produire l'empreinte $T$. Le choix des huit tours est conforté par l'analyse détaillée réalisée en 2019 dans [LIM19].

8. Par exemple, un choix de positions +0, +3 et +6 n'est pas bon car si $s_0$ et $s_3$ sont deux seuls bits actifs, alors seulement $s_{251}$ et $s_3$ seront actifs en sortie de $\theta$

9. Après un tour, la dépendance est de 9 bits. On peut montrer qu'après deux tours, la dépendance est de $9 \times 9 = 81$ bits.

De manière plus générale, nous avons besoin de permutations plus solides cryptographiquement lorsqu'elles sont utilisées dans la phase d'initialisation et de finalisation que lorsqu'elles sont utilisées dans la partie du mode duplex en chiffrement ou en déchiffrement. Cela est dû principalement au faible contrôle qu'un attaquant a sur les couples entrées/sorties du mode duplex, qui ressemblent plus à des attaques sur des chiffrements à flot. Ceci est logique puisque le mode peut être vu comme un chiffrement à flot, à la différence que l'attaquant peut contrôler (dans une attaque à clair choisi) ce qui est inséré dans le mode duplex. Nous verrons plus tard à la section 3.1 qu'il peut être bien plus intéressant d'attaquer ce mode en déchiffrement qu'en chiffrement.

#### Positions des bits pour absorption/extraction

Les bits qu'un attaquant peut contrôler (par exemple dans la phase d'initialisation) ou connaître (par exemple dans la phase de chiffrement) se situent justement aux positions engendrées par $12^4 = 176 \mod 257$. Le choix de $12^4$ n'est pas anodin et est lié au 12, le générateur de $(\mathbb{Z}/257\mathbb{Z})^*$ utilisé dans la définition de $\pi$. En effet, nous souhaitons éviter qu'il y ait un lien entre des blocs de suite chiffrante consécutifs. En prenant un peu de recul, et en oblitérant les applications $\theta$ et $\chi$, on se rend compte que toutes les positions prises par $176^j$ pour $j$ allant de 0 à 63 seront envoyées par l'application $\pi$ aux positions $150 \times 176^j$, *i.e.* un décalé du sous-groupe multiplicatif engendré par 176. Ce sous-groupe étant d'ordre 64, il a donc quatre décalés possibles. Maintenant, en prenant en compte $\chi$ et $\theta$ et en observant que chacune de ces opérations prend linéairement en compte le bit à la même position, on peut penser qu'il sera très difficile de trouver des biais linéaires en ne prenant en compte que quatre blocs consécutifs de la suite chiffrante.

Cet argument manque de preuve, et d'ailleurs il a été montré plus tard en 2021 dans un article [STSH21] paru dans *Designs, Codes and Cryptography*, qu'il existait des biais pour quatre blocs consécutifs de suite chiffrante, dûs aux termes non-linéaires et à des manières différentes d'approximer le `AND` de deux bits.

#### Analyse différentielle

Tout d'abord, il est à noter qu'une analyse différentielle de la seule fonction de tour de `Subterranean` ne donnera *a priori* pas lieu à des attaques, car il faudrait que ces différentielles concordent avec les positions des bits extraits en sortie du mode duplex. Nous pourrions penser à une stratégie similaire à celle employée sur RadioGatún par Thomas Fuhr et Thomas Peyrin dans [FP09], où des propriétés différentielles sont utilisées avec un algorithme permettant de choisir les valeurs en entrée de manière adaptative. Cependant, pour réussir à faire cela, il faut un grand nombre de degrés de liberté dans le choix des mots en entrée, ce qui semble difficile à réaliser car `Subterranean` absorbe seulement 8 bits par application de la fonction de tour sur un état de 257 bits.

Pour une utilisation en mode chiffrement authentifié de `Subterranean`, nous

devons cependant évaluer les propriétés différentielles de la fonction de tour et de ses itérations, notamment pour garantir que le nombre de tours à blanc (8) est suffisant pour ne pas pouvoir empêcher de produire des empreintes valides.

Cependant, une telle analyse de chemins différentiels n'est pas chose aisée, principalement car nous avons un choix de conception qui ne permet pas de garantir théoriquement la non-existence de chemins différentiels de poids faible pour un certain nombre de tours comme peut le faire la *wide trail strategy* [DR20]. Mais tout n'est pas perdu pour autant, et pour des constructions dites non-alignées [BDKV21], nous pouvons réaliser des analyses algorithmiques permettant de prouver la non-existence de chemins différentiels de poids faible. Ces algorithmes utilisent des stratégies de type *branch and bound* en exploitant des propriétés spécifiques de la couche linéaire ou de la boîte-$S$. Dans le cas de `Subterranean`, il est possible d'utiliser des algorithmes du type de ceux développés pour Keccak [MDV17] et Xoodoo [DHVV18]. En particulier il est possible de montrer qu'il n'existe pas de chemins différentiels de poids supérieur à 98 pour 8 tours de `Subterranean`.

Plusieurs analyses ont ensuite suivi la publication de `Subterranean`, notamment une réalisée en 2021 et publiée à ToSC [STSH21] dans un contexte de mauvaise utilisation du nonce dans le cas différentiel. Une analyse plus fine et plus générale sur les propriétés différentielles de la fonction $\chi$ a ensuite été présentée dans [MMD23].

### Recouvrement d'état

`Subterranean` possède des propriétés similaires au chiffrement authentifié Ketje [BDP+16] dans le sens où la permutation choisie pendant la phase de (dé)chiffrement du mode duplex semble faible cryptographiquement (faible degré algébrique, diffusion non complète de l'état). Or, en 2018, nous avons montré avec Thomas Fuhr et María Naya Plasencia [FNR18] qu'il est possible de retrouver l'état interne de la construction Ketje, principalement en utilisant le fait que la partie externe de l'état couvre entièrement plusieurs boîtes-$S$ de taille 5, ce qui permet d'inverser entièrement l'information et de la propager dans le sens direct et indirect. En combinant cela avec un découpage en deux parties de l'état interne (ceci étant réalisable de par la structure de Ketje), et en utilisant une fusion de listes et le faible degré algébrique de la fonction de tour, il est possible de retrouver l'état interne. Ketje a donc ensuite dû être modifié et extraire l'information de l'état en ne prenant qu'un bit par boîte-$S$.

Comme le taux[10] dans `Subterranean` est de 32 bits, la proportion entre le taux et la taille de l'état est aussi semblable. Il est donc nécessaire de vérifier si une attaque similaire pourrait s'appliquer. Cependant, il n'y a qu'une seule boîte-$S$ dans `Subterranean` et les positions de l'état interne ne sont pas côte à côte, ne permettant pas d'inverser même localement la boîte-$S$. C'est d'ailleurs la raison pour laquelle nous avons choisi des positions de bits insérés et extraits éloignées les unes des autres. Ceci est particulièrement important dans un cas

10. Le taux est la quantité d'information absorbée ou extraite dans le mode duplex. Il est appelé *rate* en anglais.

d'utilisation de la fonction $\chi$, car la connaissance de bits consécutifs en sortie permet d'inverser partiellement ladite fonction.

Enfin l'ajout « à la Trivium » qui consiste à sommer deux bits de l'état pour produire la suite chiffrante rend ce type de cryptanalyse difficile. Nous revendiquons que, même si la quantité d'information extraite par application du mode duplex est la même que dans Ketje, celle-ci est plus difficile à exploiter dans le cas de `Subterranean`.

Utiliser une permutation avec un faible nombre de tours dans le mode duplex n'est *a priori* pas un problème en soi. Mais, il faut être particulièrement précautionneux sur les emplacements choisis définissant l'état externe. Plus particulièrement, il faut qu'il soit impossible (ou extrêmement coûteux) de lier les informations obtenues en observant des blocs successifs de suite chiffrante et il ne faut surtout pas que la connaissance de blocs successifs de suite chiffrante permette de diminuer le nombre d'états internes possibles. Nous l'avons montré de manière cruciale avec Christina Boura et Rachelle Heim Boissier [BBR22] par une attaque dévastatrice sur le chiffrement Panther [BSL21].

### Collisions

Comme nous avons choisi d'effectuer huit tours à blanc de `Subterranean` entre la phase d'absorption et la phase d'essorage pour la construction en éponge, nous supposons qu'il est plus difficile de construire des collisions sur la sortie de la fonction de hachage que d'essayer de construire une collision interne, dont la complexité est en $2^{c/2}$ où $c$ est la capacité. Dans le cas de `Subterranean`, $c = 257 - 8 - 1 = 248$ (le 1 venant du bit de *padding*). Ainsi, il existe une attaque permettant de construire une collision en $2^{124}$ ce qui dépasse notre revendication de sécurité de 112 bits.

La collision interne sur la construction en éponge fonctionne comme suit. Il faut considérer plusieurs messages aléatoires, de manière à trouver deux messages produisant le même état interne, ce qui est tout à fait réalisable sans mémoire avec des algorithmes de recherche de cycles. Une fois ces messages trouvés, il faut les concaténer respectivement avec deux blocs de $r$ bits dont la différence compense celle qui figure dans l'état externe, engendrant ainsi une collision complète sur l'état entier et donc sur la sortie.

Cependant, dans un contexte où la fonction de tour est très simple, il est possible d'écrire complètement les équations que doivent satisfaire, dans une attaque de ce type, les valeurs des blocs de message précédents et de l'état interne un instant plus tôt. En fixant alors certaines valeurs pour lesdits messages, il est possible d'aboutir plus souvent que dans le cas aléatoire à une collision sur l'état interne. Dans le cas de `Subterranean`, si nous n'avions fait qu'un tour dans la construction en éponge entre deux absorptions, nous aurions eu de cette manière une attaque en $2^{116}$. Considérant que cette valeur est trop proche de $2^{112}$, nous avons souhaité conserver une marge de sécurité en effectuant deux tours au lieu de un, ce qui nous conduit à penser qu'il est difficile d'appliquer cette technique. Nous laissons le soin à quiconque aurait envie d'analyser la construction d'essayer d'appliquer cette technique pour deux tours, car les polynômes

sont encore calculables en machine.

## 1.3 Collisions internes sur les « petits » Keccak

Keccak [BDPV14] est une fonction de hachage qui a gagné en 2012 la compétition `SHA-3` [SHA15] et a été choisie comme standard. Cette construction se décline en quatre variantes, ayant un état interne de taille respectivement 1600, 800, 400 et 200 bits. Il faut noter que la dernière variante ne doit pas être utilisée pour du hachage mais peut cependant être employée *a priori* dans un environnement restreint en mode duplex, comme proposé dans [KY10]. De plus, l'intérêt de ces versions est qu'elles permettent de réfléchir à des attaques, bien que seules les variantes avec un état interne de 1600 bits aient été standardisées. De plus, les auteurs de Keccak ont motivé les cryptanalyses sur toutes les variantes via le Crunchy Contest [BDPV]. Dans cette section, je présente une attaque sur les deux petites versions de Keccak où la permutation interne est réduite à 2 tours. Ce travail a été réalisé avec Rachelle Heim Boissier et a été publié à ToSC [HNR21].

### 1.3.1 Préliminaires sur Keccak

#### Précédentes analyses

La première attaque sur Keccak est due à M. Naya-Plasencia, A. Röck et W. Meier en 2011 [NRM11] et utilise des techniques de cryptanalyse différentielle, permettant d'attaquer 2 tours. Puis, I. Dinur, O. Dunkelman et A. Shamir [DDS12] ont réalisé une attaque en collision sur 4 tours. Par la suite, Keccak a fait l'objet d'un grand nombre de cryptanalyses en collision, où les méthodes utilisées sont des analyses des différentielles internes [DDS14] et des techniques de linéarisation [QSLG17, SLG17]. Dans le cas des attaques en préimage, l'utilisation de structures linéaires introduites par J. Guo, M. Liu et L. Song en 2016 [GLS16] est privilégiée. Un grand nombre de travaux ont ensuite amélioré ces techniques [LSLW17, KRA18, LS19, KMS18, Raj19]. D'autres stratégies d'attaque essayent d'attaquer Keccak en utilisant son faible degré via des attaques de type cube ou *zero-sum*. Les premières analyses des propriétés algébriques ont été réalisées par J. Aumasson, D. Khovratovich et W. Meier en 2009 [AK09, AM09] puis par A. Canteaut et C. Boura en 2010 dans [BC11, BC10] et largement améliorées dans plusieurs articles. Cependant les propriétés distinguantes de ce type ne permettent pas de construire des collisions. Je présenterais plus en détail les attaques de ce type au chapitre 2.

Cependant, toutes ces attaques s'appliquent à des versions réduites en nombre de tour du standard `SHA-3`, c'est-à-dire où l'état interne est de 800 ou 1600 bits. En effet, les attaques permettant de dépasser quatre tours (et même de les atteindre) ne sont applicables que sur ces « grandes » variantes, principalement car le rapport $r/b$ est grand, où $r$ est le taux et $b$ la taille de l'état. Le fait que ce rapport soit grand implique qu'un attaquant a plus de contrôle sur l'état (en proportion d'information) que sur une petite variante où la capacité $c$ est

plus grande en proportion [11]. Or, les analyses réalisées ont besoin d'un grand ensemble de messages à contrôler pour que les propriétés exploitées puissent apparaître avec grande probabilité. On parle du nombre de « degrés de liberté » nécessités par les attaques. De plus, les attaques en collision essayent plutôt de réaliser des collisions sur la sortie et non sur la partie interne. Je vais montrer qu'au contraire, il vaut mieux essayer de chercher une collision interne [12].

### Description de Keccak

La fonction de hachage `SHA-3` est aujourd'hui une des constructions les plus connues, à la fois par sa victoire de la compétition du NIST, mais aussi parce qu'il s'agit d'une des premières applications de la construction en éponge. Pour des raisons de clarté, nous décrivons rapidement les briques de base de Keccak.

Keccak utilise la construction en éponge et une permutation qui opère sur un état de taille $25 \times 2^\ell$ bits vu comme un état tri-dimensionnel $A[5, 5, \omega]$ d'éléments de $\mathbb{F}_2$ où $\omega = 2^\ell$. Ainsi, nous parlerons de tranche, plan, feuille, ligne, colonne et tube représentés dans la figure 1.4. Plus précisément, une tranche (en orange dans la figure 1.4) est un ensemble de 25 bits avec une coordonnée $z$ constante, un plan est défini par tous les bits positionnés par une coordonnée $y$ donnée et une feuille est définie par une coordonnée $x$. Enfin, une ligne (5 bits) est montrée en cyan à la figure 1.4, une colonne (5 bits) est en rouge et un tube ($\omega$ bits) est en bleu.

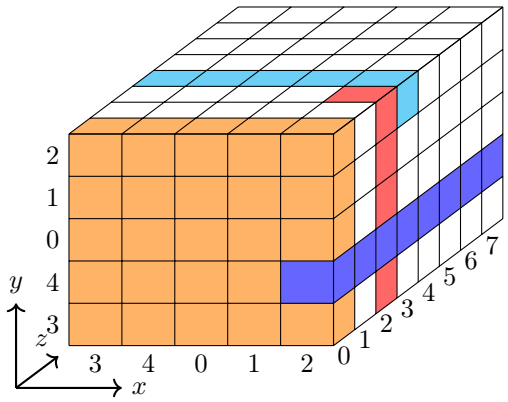


Figure 1.4 – État interne de Keccak.

Chaque tour de la permutation Keccak-$p$ se décompose en 5 fonctions, $\theta$, $\rho$, $\pi$, $\chi$ et $\iota$. Dans tout ce qui suit, les opérations sur les positions des coordonnées $x$, $y$, $z$ sont réalisées modulo 5, 5 et $\omega$ respectivement. Pour tout $0 \leq x, y < 5$ et $0 \leq z < \omega$,

11. La capacité est généralement égale au double du niveau de sécurité, qui lui est fixé aujourd'hui à 128 ou 256.

12. Une collision interne se transforme très facilement en collision complète sur la sortie, et ce peu importe la taille de celle-ci, en compensant le dernier bloc de message.

– $\theta$ additionne à chaque bit de l'état la parité de deux colonnes de l'état :

$$\theta(A)[x,y,z] = A[x,y,z] \oplus \sum_{i=0}^{4}(A[x-1,i,z] \oplus A[x-1,i,z-1])$$

– $\rho$ décale cycliquement chaque tube par une constante :

$$\rho(A)[x,y,z] = A[x,y,z+c(x,y)]$$

– $\pi$ est une transposition des tubes :

$$\pi(A)[x,y,z] = A[x+3y,x,z]$$

– $\chi$ est la seule fonction non-linéaire, de deux degré 2, comme définie dans la section précédente :

$$\chi(A)[x,y,z] = A[x,y,z] \oplus (\neg A[x+1,y,z]) \wedge A[x+2,y,z]$$

– $\iota$ est l'addition d'une constante de tour, dont la valeur dépend du tour considéré : $\iota$ ajoute au tube $A[0][0][*]$ une valeur $i_j$, où $i_j$ dépend de la taille de l'état considéré et $j$ est l'indice du tour.

Enfin, la fonction de tour de Keccak-$p$, notée $R$ est définie comme suit.

$$R = \iota(-,i_j) \circ \chi \circ \pi \circ \rho \circ \theta$$

Le nombre de tours est au moins 12 et est naturellement d'autant plus grand que la variante de Keccak utilisée opère sur un état plus grand.

### 1.3.2 Idée générale

Dans le cas où le taux est petit dans une construction en éponge, il est nécessaire, pendant la phase d'extraction, d'extraire plusieurs blocs et donc d'appliquer plusieurs fois la permutation afin de garantir un niveau de sécurité suffisant (ce qui n'est pas le cas dans les « grandes » versions). Ainsi, il semble bien plus difficile d'attaquer les petites variantes en cherchant une collision sur la sortie, car on ne contrôle facilement que le premier bloc de sortie. Une des différences dans le reste de l'état a de grandes chances de ne pas produire de collision sur les blocs suivants. C'est la raison pour laquelle nous essayons de construire des collisions internes, *i.e.* des collisions sur une sous-partie de l'état de taille $c$ bits où $c$ est la capacité. Une première attaque proposée au concours Crunchy Contest [BDPV] visait les petites variantes quand la permutation utilisée était réduite à 1 tour. Ici, nous construisons des collisions quand la permutation est réduite à 2 tours, et ce de manière moins coûteuse que l'algorithme générique de complexité $2^{c/2}$ comme expliqué dans la section précédente.

L'idée consiste simplement à générer plusieurs chemins en insérant des débuts de messages différents, qui donneront un ensemble d'états différents auxquels nous associerons deux blocs de messages bien choisis (donc $4r$ bits d'information). L'attaque peut se décrire simplement comme suit.

1. Produire un état $s$ aléatoire en utilisant une chaîne suffisamment longue de message ;
2. choisir $r$ bits du bloc de message $m$ tel que $v = f(m||\widehat{s})$ appartient à un sous-ensemble strict $X$ de $\mathbb{F}_2^c$, où $\widehat{s}$ est l'état interne de $s$ et $f$ est la permutation KECCAK-$p$ tronquée à l'espace de l'état interne et naturellement réduite à 2 tours ;
3. il se peut que la valeur calculée précédemment n'appartienne pas à $X$ : sauvegarder cette valeur uniquement dans le cas où $v \in X$ ;
4. continuer jusqu'à trouver une collision.

Le jeu consiste donc à :

1. définir un sous-ensemble $X$ suffisamment petit ;
2. pour lequel il est « facile » de choisir un $m$ tel que la probabilité que $f(m||\widehat{s}) \in X$ soit élevée.

Ainsi, si on produit $\sqrt{|X|}$ éléments, on trouvera avec une grande probabilité une collision. La complexité de l'attaque dépendra donc : du coût du calcul des blocs de message $m$, de la probabilité de réussir à tomber dans le sous-ensemble, et naturellement de la taille de $X$.

### 1.3.3 Propriétés nécessaires

La réalisation de cette attaque repose sur des propriétés spécifiques des briques élémentaires de la permutation KECCAK-$p$ que je passe en revue ci-dessous.

#### Inversion partielle

Chaque opération de KECCAK permet la diffusion ou la confusion dans une sous-partie de l'état. Plus précisément, la fonction $\rho$ consiste simplement en une translation cyclique à l'intérieur de chaque tube de l'état, $\chi$ permet la confusion dans une ligne de l'état et $\pi$ est une réorganisation des tubes de l'état de telle sorte que deux tubes de l'état qui appartiennent au même plan ne se retrouvent ni dans le même plan ni dans la même feuille. Enfin, $\iota$ est l'addition d'une constante de tour et ne jouera pas de rôle dans notre attaque et seule $\theta$ permet une grande diffusion. Plus précisément, les propriétés suivantes permettent de montrer que l'on peut partiellement inverser la dernière fonction de tour (sauf $\theta$) pour se ramener à une recherche de collision sur une fonction de degré multivarié 2.

**Propriété 1.3.1** *Soient $A, A' \in \mathbb{F}_2^b$. $\rho$ est une permutation des tubes. En particulier pour tout $0 \leq i, j < 5$, $\rho(A)[i, j, *] = \rho(A')[i, j, *]$ si et seulement si $A[i, j, *] = A'[i, j, *]$.*

De même, on a un phénomène similaire pour la permutation $\chi$ qui opère indépendamment sur chaque ligne.

**Propriété 1.3.2** *Soit $A, A' \in \mathbb{F}_2^b$. Pour tout $0 \leq j < 5$, $\iota \circ \chi(A)[*, j, *] = \iota \circ \chi(A')[*, j, *]$ si et seulement si $A[*, j, *] = A'[*, j, *]$.*

Enfin, l'application $\pi$ réorganise les tubes de telle sorte que nous avons la propriété suivante.

**Propriété 1.3.3** *Soient $A, A' \in \mathbb{F}_2^b$ tels que $\pi(A) = A'$. Soient $0 \leq y < 5$, $0 \leq z < \omega$ tels que $A'[*, y, z]$ est une ligne de l'état. Alors chaque paire de bits de $\pi^{-1}(A'[*, y, z])$ se situe dans deux colonnes différentes de la tranche $A[*, *, z]$.*

Ces propriétés permettent directement de gagner un tour et de remarquer que, si la partie de l'état interne de $c$ bits, sur laquelle on cherche à réaliser une collision consiste en des plans complets, alors nous pouvons montrer qu'obtenir une collision sur cette partie est équivalent à trouver une collision sur ce que l'on a appelé l'état interne alternatif, qui est l'antécédent par $\iota \circ \chi \circ \pi \circ \rho$ de l'état interne usuel. Ceci est formalisé par la proposition suivante.

**Proposition 1.3.1** *Soient $A, A' \in \mathbb{F}_2^b$. On suppose que $5\omega$ divise $c$. Alors obtenir une collision sur l'état interne alternatif de $A_\theta$ et $A'_\theta$ est équivalent à obtenir une collision sur l'état interne de $A$ et $A'$ où $A_\theta$ et $A'_\theta$ sont les antécédents de $A$ et $A'$ par l'application $\iota \circ \chi \circ \pi \circ \rho$.*

Afin d'aider à la compréhension, une représentation visuelle de cette observation est donnée à la figure 1.5. Ainsi, pour construire une collision, nous devons choisir des messages sur un plan complet, de manière à forcer cet état interne alternatif à être dans un sous-ensemble strict de $\mathbb{F}_2^c$. La transformation entre l'état au moment où le message est injecté et l'endroit où l'on cherche à obtenir une collision reste cependant non-linéaire mais est de degré plus faible, uniquement 2 : cette transformation est la fonction de tour composée avec $\theta$.

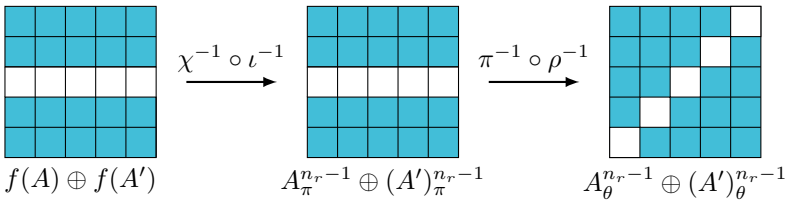


FIGURE 1.5 – Illustration de la proposition 1.3.1.

### Gérer l'application $\theta$

L'application $\theta$ permet d'assurer une grande diffusion au cours de la construction. La formule de son inverse est complexe si on l'exprime bit à bit. Dans ce que nous souhaitons faire, nous ne contrôlons que quelques bits de l'état avant $\chi$ et nous avons peu de degrés de liberté pour contrôler un grand ensemble de bits qui seraient tous pris en considération par l'application $\theta$. Il convient donc de trouver une meilleure stratégie.

Chaque bit de l'état en sortie de l'application $\theta$ dépend exactement de 11 bits en entrée (lui-même et 10 bits de colonnes proches). En revanche, si l'on considère non pas des bits indépendamment mais des relations entre plusieurs bits de l'état, on obtient une très faible diffusion. Plus précisément, on a l'effet suivant.

**Propriété 1.3.4** *La somme de deux bits localisés dans la même colonne après $\theta$ est égale à la somme des bits localisés aux mêmes endroits avant $\theta$.*

Une implication directe est le théorème suivant qui, combiné avec des techniques de linéarisation vont nous permettre de forcer l'état interne alternatif à être dans un ensemble de taille réduite.

**Théorème 1.3.1** *Soient $A, A' \in \mathbb{F}_2^b$ deux états qui collisionnent dans l'état interne. Soit $V$ la somme des deux états après l'application de la première couche non-linéaire $\chi$. Alors $V$ doit être constant sur $\frac{c}{5\omega}$ bits de chaque colonne.*

### 1.3.4 Attaque en collisions

Nous rappelons maintenant que nous cherchons des messages $m, m'$ dans $\mathbb{F}_2^r$ où $r$ est le taux, tels que, à la donnée de $s$ et $s'$, on obtienne $f(m||s) = f(m'||s')$ où $f$ est la permutation de Keccak-$p$ réduite à deux tours et tronquée aux $c$ derniers bits de l'état. Les propriétés précédentes peuvent alors être exploitées.

#### Vision algébrique des collisions

Choisir ces messages $m, m'$ en fonction de $s, s'$ peut être vu comme la résolution de $c$ équations de la forme suivante.

$$\begin{cases} f_0(m_0, \dots, m_{r-1}, s_0, \dots, s_{c-1}) = f_0(m'_0, \dots, m'_{r-1}, s'_0, \dots, s'_{c-1}) \\ f_1(m_0, \dots, m_{r-1}, s_0, \dots, s_{c-1}) = f_1(m'_0, \dots, m'_{r-1}, s'_0, \dots, s'_{c-1}) \\ \dots \\ f_{c-1}(m_0, \dots, m_{r-1}, s_0, \dots, s_{c-1}) = f_{c-1}(m'_0, \dots, m'_{r-1}, s'_0, \dots, s'_{c-1}) \end{cases} \quad (\mathcal{S})$$

où les $f_i$ pour $i$ allant de 0 à $c-1$ sont les fonctions composantes de $f$.

La valeur du taux $r$ étant petite, nous ne pourrons pas satisfaire entièrement ces équations. Cependant, le fait d'être constant sur un suffisamment grand nombre de bits de chaque colonne de l'état (voir théorème 1.3.1) est équivalent à satisfaire des combinaisons linéaires d'équations dans le système ($\mathcal{S}$).

#### Linéarisation

Les techniques de linéarisation, sont largement utilisées pour analyser Keccak et peuvent être trouvées dans les articles suivants [DLWQ17, SGSL18, SLG17,FNR18,LSLW17,KRA18,KMS18,GLL+20,LS19,GLS16,Raj19,QSLG17] (liste probablement non-exhaustive). Les propriétés exploitées sont aussi relativement simples et proviennent de la fonction $\chi$.

Tout d'abord, $\chi$ est la fonction identité sur chaque bit avec une probabilité de 0.75.

Ensuite et surtout, si un seul bit en entrée de Keccak est connu, alors l'expression de deux bits en sortie est linéaire en les autres bits de l'entrée. L'idée générale consiste donc souvent à fixer certains bits puis à dériver des équations linéaires dans les bits en sortie de $\chi$, facilitant ainsi les équations à résoudre. Cela s'applique donc très naturellement dans les techniques de pré-images, notamment quand le taux est bien plus grand que le niveau de sécurité [13] : on inverse partiellement $\chi$ comme nous l'avons montré précédemment, puis, les équations à satisfaire deviennent linéaires en l'entrée située un tour avant si certains bits de l'entrée sont fixés. Le fait de fixer certaines valeurs en entrée permet donc de simplifier le système à résoudre pour trouver une pré-image.

Cependant, nous n'avons pas souhaité utiliser ces techniques pour la recherche de pré-images dans le cas des « petites » variantes, mais pour trouver des collisions. En effet, quand nous considérons les « petites » variantes de Keccak pour trouver une pré-image, il faut d'abord être capable d'inverser plusieurs fois la permutation Keccak-$p$ (car le taux est plus petit que le niveau de sécurité). Ainsi, trouver une pré-image avec ces techniques semble moins adapté dans notre contexte.

De plus, dans les petites versions de Keccak, un attaquant a bien moins de contrôle sur l'état (en proportion de la taille de l'état) que dans le cas des grandes versions. Par exemple, dans le cas des attaques par linéarisation, fixer quelques bits du message à des valeurs consomme déjà beaucoup trop de degrés de liberté, ne laissant à l'attaquant que peu de latitude pour construire une pré-image.

Pour toutes ces raisons, nous nous sommes donc intéressés à utiliser ces techniques de linéarisation dans un contexte d'attaque en collision interne. Plus précisément, nous fixons d'abord un certain nombre de bits de l'état interne avant la première application $\chi$. Ceci fixe donc des relations linéaires entre les bits de l'avant-dernier message notés $(m_i)_{0\leq i<r}$ et les bits de l'état notés $(s_i)_{0\leq i<c}$ produits par les premiers blocs de messages choisis aléatoirement.

La valeur de certains bits avant $\chi$ étant fixée, l'expression des bits après $\chi$ devient donc linéaire en les $(m_i)_{0\leq i<r}$ et les $(s_i)_{0\leq i<c}$. Autrement dit, des combinaisons linéaires du système $(\mathcal{S})$ deviennent linéaires.

Enfin, la valeur de certaines colonnes avant $\chi$ ne doit pas être choisie arbitrairement. Comme la fonction $\chi$ peut être approchée (pour des bits pris dans une ligne différente) par la fonction identité, certaines valeurs de colonnes (toutes à 0 ou toutes à 1) ont plus de chances d'assurer le caractère constant des colonnes après $\chi$, qui est un critère nécessaire pour avoir une collision interne.

### Résultats

Tout un travail minutieux reste cependant à accomplir, car nous sommes extrêmement limités dans le nombre de valeurs de colonnes que l'on peut fixer, ce qui ne nous permet pas de satisfaire les équations du système en suffisamment grand nombre. Fixer des valeurs augmente la probabilité que d'autres équations

13. C'est le cas pour les deux grandes versions.

du système soient satisfaites mais consomme des degrés de liberté. Les meilleurs choix permettent de trouver une collision (quand la permutation est réduite à deux tours) sur KECCAK$[200, c = 160]$ en $2^{69} \times 2^4 \times 2 = 2^{74}$ appels à la permutation KECCAK-$p$, sur KECCAK$[200, c = 128]$ en $2^{48} \times 2^5 \times 2 = 2^{54}$ et sur KECCAK$[400, c = 256]$ en $2^{96} \times 2^7 \times 2 = 2^{104}$.

**Perspectives**

Les travaux sur cryptanalyse de KECCAK sont naturellement toujours d'actualité. De nombreuses pistes doivent encore être explorées : que ce soit des améliorations pratiques d'attaques en pré-image sur peu de tours [LHY21, HLY21] ; ou bien en utilisant des techniques de résolution de systèmes multivariés de degré deux au lieu de systèmes linéaires [WWF+21] ; ou encore en exploitant, comme nous l'avons fait, plusieurs messages pour forcer l'état interne à appartenir à un sous-ensemble particulier, mais aussi en connectant avec des solveurs SAT [HBYDM22].

Un effort de cryptanalyse est encore nécessaire pour analyser les petites variantes qui ont toutefois été peu regardées, à la fois en pré-image que ce soit de manière générique ou en cryptanalyse dédiée en se comparant à la borne suivante donnée dans [LM22]

$$\min\left\{\max\left\{2^{n-r'}, 2^{c/2}\right\}, 2^n\right\} .$$

## 1.4 Conclusion

J'ai présenté dans ce chapitre des cryptanalyses diverses qui, en un certain sens, exploitent une représentation polynomiale. L'attaque sur `GEA-1` et `GEA-2` est possible du fait de la divisibilité entre polynômes bien choisis définis par l'initialisation [BFL21]. La cryptanalyse de `Subterranean` repose sur la possibilité de sauvegarder en mémoire et de manipuler le polynôme multivarié complet, ainsi que sur son caractère cyclique qui permet de travailler dans l'anneau des polynômes $\mathbb{F}_2[X]/(1 + X^{257})$. Enfin, la cryptanalyse sur KECCAK pourrait être améliorée si nous savions satisfaire plus facilement un ensemble d'équations non-linéaires, sans tirer uniquement parti du caractère très creux des équations qui rend la linéarisation possible. Je suis intimement persuadé qu'il reste des pistes d'améliorations des attaques en regardant la forme exacte des polynômes, que ce soit pour des fonctions de hachage ou du chiffrement, afin de connecter les distingueurs et le recouvrement de clef dans un chiffrement par bloc par exemple.

# Chapitre 2

# Attaques exploitant la représentation polynomiale

Nous venons de voir divers exemples de cryptanalyses qui tirent parti, d'une manière ou d'une autre, d'une représentation polynomiale de certains composants du système étudié.

Dans ce chapitre, nous nous intéresserons donc un peu plus en détail à ce que l'on peut réaliser aujourd'hui en exploitant la représentation polynomiale des primitives cryptographiques. Les classes d'attaques que nous allons considérer sont les attaques algébriques et les attaques intégrales. Une première section décrira les bases de ces cryptanalyses et une deuxième section en donnera une illustration avec la cryptanalyse de `Pyjamask` [GJK+20] un candidat à la compétition pour la standardisation des chiffrements symétriques dits à bas coût lancée en 2019 par le NIST. Cette cryptanalyse a été réalisée avec Christoph Dobraunig et Jan Schoone et publiée à ToSC [DRS20]. Enfin, une troisième section sera consacrée à des réflexions et des mises en lumière cherchant quel(s) point(s) améliorer dans ce contexte d'attaque.

## 2.1 Quelques types d'attaques

La terminologie « attaque algébrique » est souvent utilisée pour désigner toutes les techniques exploitant la résolution de polynômes (non)linéaires. En revanche, les attaques exploitant les différentielles d'ordre supérieur possèdent plusieurs noms : attaque différentielle d'ordre supérieur [Lai94], par saturation intégrale [KW02], attaque par cube [DS09] ou *zero-sum* [AM09]. Devant faire un choix, nous parlerons toujours ici d'attaques intégrales.

### 2.1.1 Attaques algébriques

#### Linéarisation

Le principe des attaques algébriques consiste à exploiter généralement des équations non-linéaires (généralement de faible degré) entre bits de clef. La première technique très connue est la linéarisation [1] où la stratégie pour résoudre un système non-linéaire avec des équations de degré au plus $d \in \mathbb{N}$ en $n$ variables binaires secrètes, les bits de la clef, *i.e.* $k = (k_0, \ldots, k_{n-1}) \in \mathbb{F}_2^n$, consiste tout simplement à considérer chaque monôme $k^u$ comme une nouvelle variable indépendante des autres où $u$ est un élément de $\mathbb{F}_2^n$ de poids de Hamming inférieur à $d$. On récolte alors au moins

$$D = \sum_{i=1}^{d} \binom{n}{i}$$

équations. Ainsi, avec grande probabilité, le système peut être résolu en $\mathcal{O}\left(D^3\right)$ opérations, ce qui permet de retrouver la valeur du secret.

Le degré multivarié est donc un critère important à prendre en compte pour ne pas être vulnérable à ce type d'attaque.

#### Immunité algébrique

Cependant, toute équation Booléenne de degré plus grand que $\frac{n}{2}$ peut se transformer en une équation de degré inférieur à $\frac{n}{2}$. En effet, les attaques de Courtois et Meier en 2003 [CM03, Cou03] ont montré que l'on peut transformer une équation $f(k) = 0$ ou $f(k) = 1$ de haut degré en une équation de plus petit degré. En effet, pour toute fonction booléenne $f$ à $n$ variables, il existe toujours deux fonctions booléennes $g$ et $h$ à $n$ variables de degré multivarié inférieur à $\frac{n}{2}$ telles que $g \cdot f = h$.

Exploiter ainsi l'idéal des annulateurs d'une fonction permet de diminuer le degré des équations au moins jusqu'à $\frac{n}{2}$ et ce pour toute fonction booléenne $f$, permettant alors de linéariser de manière moins coûteuse non pas le système initial mais des équations de degrés moins élevés dérivées du système d'origine. Pour plus de détails sur l'immunité algébrique, je renvoie au livre de Claude Carlet [Car20].

Les autres techniques plus générales de résolution de systèmes non-linéaires utilisent le calcul de bases de Gröbner [CLO15] à l'aide d'algorithmes de type (F4 ou F5) [Fau99] et FGLM [FGLM93]. Cependant, ces techniques étant génériques, elles sortent du cadre de ce chapitre. D'ailleurs analyser leur complexité effective n'est pas chose aisée. Ici, nous nous intéresserons plutôt à des méthodes pour lesquelles il est possible d'évaluer en détail et de manière effective la complexité et de garantir, pour des systèmes bien spécifiques, que l'attaque fonctionne [2].

1. Dans ce contexte, la technique est différente de celle utilisée dans Keccak au chapitre précédent.

2. Typiquement on supposera toujours que l'inversion d'un système linéaire se fait en $n^3$ opérations et non en $n^\omega$.

En effet, nous ne pouvons pas considérer que les systèmes que l'on observe en cryptographie sont aléatoires. Il y a très probablement pour certains d'entre eux, des techniques en amont à adapter afin de transformer le système, pour se ramener à un contexte où le rang de la matrice à résoudre est plus faible, ou bien d'utiliser des approches hybrides ou de type *guess and determine* qui sont en fait plus adaptées [Gol97, And94, BFP09, DLR16, CDM+18].

### 2.1.2 Attaques intégrales

En 1994, Lai [Lai94] publie un article intitulé *Higher Order Derivatives and Differential Cryptanalysis* et conclut par la phrase : « Pour tout $i$ petit, la dérivée $i$-ième d'une fonction cryptographique doit prendre chaque valeur de manière uniforme ». De cette phrase découle en grande partie le critère utilisé ensuite dans un grand nombre d'articles pour monter des attaques dites intégrales, selon la terminologie introduite en 2002 dans [KW02].

#### Principe

Le principe des attaques intégrales est relativement simple. On se place dans un contexte à clairs choisis, et on considère des clairs dans un certain espace affine noté $V + c \subset \mathbb{F}_2^n$ où $V$ est un espace linéaire de dimension $\ell$. On cherche alors à tirer parti d'une propriété de la somme

$$\sum_{x \in V+c} F_K(x) \tag{2.1}$$

où $F_K$ est n'importe quelle fonction dépendante de la clef $K$ définie par une partie du chiffrement.

#### Applications

Ce type d'attaque a été pour la première fois employé sur le chiffrement Square [DKR97] et peut être appliqué sur l'AES [3]. L'idée consiste à saturer un ou plusieurs octet(s) [4] et laisser les autres constants. Alors la somme des valeurs obtenues à la sortie du troisième tour vaut toujours 0, et ce indépendamment de la valeur de la clef. Des tours avant et après peuvent être rajoutés dans cette attaque afin de retrouver la clef. Il faut pour cela sélectionner un ensemble de valeurs ayant la bonne propriété (saturation d'un octet) après le premier tour et pouvoir inverser partiellement le dernier tour en fonction des chiffrés obtenus. Ceci est possible en réalisant des hypothèses sur une partie de la clef secrète, la diffusion n'étant pas complète sur un tour. Les candidats testés ne vérifiant pas la propriété peuvent alors être éliminés.

3. C'est un excellent exercice pour des travaux dirigés de cryptographie, sans avoir à parler de cryptanalyse différentielle.

4. Faire en sorte qu'un octet bien choisi prenne toutes les valeurs, et ce indépendemment de la valeur de la clef.

**Premiers critères**

Dans les chiffrements par bloc de type SPN [5], on peut donc toujours gagner un tour en amont de ce type d'attaque. En effet, le premier étage non-linéaire est une concaténation de boîtes-$S$ bijectives. Ainsi, l'image d'un espace affine en entrée du chiffrement de l'étage non-linéaire défini par la saturation de certaines boîtes-$S$ sera toujours un espace affine en sortie de cet étage. En utilisant le fait que la structure du chiffrement est alignée comme pour l'AES, ou en réalisant une analyse approfondie du chiffrement, on peut aussi réussir à gagner plusieurs tours en entrée.

Considérons maintenant le distingueur à proprement parler. Pour une fonction booléenne $f$ donnée de degré multivarié $d$, on sait que sa dérivée en n'importe quel point est de degré au plus $d-1$. La somme (2.1) correspond à l'évaluation d'une fonction dérivée $\ell$ fois. Notre chiffrement doit donc, au moins, être de grand degré. Ceci semble réalisable avec un faible nombre de tours, le degré augmentant *a priori* de manière exponentielle avec le nombre de tours. Cependant, Christina Boura, Anne Canteaut et Christophe De Cannière ont montré en 2011 [BCD11] que lorsque la la fonction non-linéaire itérée est formée en concaténant plusieurs petites boîtes-$S$, l'augmentation du degré ralentit à l'approche de sa valeur maximale ($n-1$ pour des applications bijectives quand $n$ est la taille de l'état).

Enfin, il y a l'étape de recouvrement de clef. Comme dans la plupart des cas et comme nous l'avons vu plus haut, l'existence d'un distingueur sur un certain nombre de tours $r$ se transforme en recouvrement de clef sur $r' > r$ tours, par exemple en inversant partiellement les valeurs de chiffré au moyen d'une hypothèse de clef.

Il y a donc trois étapes distinctes dans une attaque intégrale, comme nous l'illustrerons à la section 2.2 sur un cas d'application. Il existe dans la littérature un très grand nombre d'attaques intégrales, par exemple sur Misty [SL09] ou sur Simon [KSI16, FSW17] pour n'en citer que deux.

### 2.1.3 Attaques par cube

En 2009, Itait Dinur et Adi Shamir ont proposé les attaques dites « par cube » [DS09] qui utilisent sensiblement la même propriété mais permettent de réaliser une attaque (qui marche particulièrement bien sur la phase d'initialisation des chiffrements à flot) sans avoir à connaître l'expression algébrique du polynôme multivarié défini par ledit chiffrement. L'idée est la suivante : on note $p(k,x)$ le polynôme multivarié prenant en entrée la clef secrète $k \in \mathbb{F}_2^\kappa$ et une valeur publique $x \in \mathbb{F}_2^n$ (le vecteur d'initialisation ou le nonce) défini par le chiffrement considéré. Alors pour tout $u \in \mathbb{F}_2^n$, le polynôme $p$ peut s'écrire sous la forme

$$p(x,k) = q(x,k)x^u + r(x,k)$$

où $\deg(q) \leq \deg(p) - \mathsf{w}_h(u)$ où $\mathsf{w}_h(u)$ est le poids de Hamming de $u$ et où $r(x,k)$ ne contient naturellement pas le monôme $x^u$. Si, de plus $q(x,k)$ ne dé-

5. *Substitution Permutation Networks*

pend que de $k$, alors en sommant sur les bons espaces affines (tous les décalés de $\{x, x \prec u\}$), on retrouve par la transformée de Möbius les valeurs de $q$ évaluées en la clef. Si $q$ est de degré 1, une phase de pré-calcul du polynôme réalisée sur plusieurs clefs permet de déterminer $q$ et une phase « online » fournit alors des équations linéaires en la clef. Cette idée a ensuite été appliquée à plusieurs chiffrements, notamment `MD-6` et Trivium [ADMS09, FV14], puis à Grain [DS11] et à Keccak [DMP+15] pour ne citer qu'eux.

### 2.1.4 Division Property

Il est clair que le degré seul d'un chiffrement n'est pas un critère suffisant. En effet, si on compose une fonction formée d'une concaténation de boîtes-$S$ par une fonction de degré $d$, on obtient facilement une somme nulle comme nous l'avons précédemment observé pour des espaces affines biens choisis de dimension $d+1$, alors que le degré de cette composition de fonctions est bien plus grand que $d$. Cela implique plusieurs observations :

— Les directions dans lesquelles on dérive la fonction permettent de faire diminuer le degré plus rapidement ;
— Dans l'expression algébrique de la composition du chiffrement par une application linéaire, Il y a certains monômes dont aucun multiple n'apparaît.

En 2015, Yosuke Todo a proposé une méthode [Tod15b] permettant justement de chercher des monômes spécifiques présents ou absents dans la forme algébrique normale d'une fonction, mais surtout de capturer ces propriétés dans un contexte de chiffrement itératif. Cette technique a conduit à une attaque sur Misty-1 [Tod15a]. Depuis, des améliorations de cette technique ont été proposées comme par exemple dans [BC16, TM16]. Comme la *division property* permet de voir le problème sous forme de chemins décrivant si un monôme apparaît (ou non) dans la forme algébrique normale d'une fonction, il est aussi possible d'utiliser des modèles MILP permettant de chercher ces propriétés de manière automatique [TIHM17].

## 2.2 Cryptanalyse de `Pyjamask`

Pendant la compétition du NIST pour la standardisation d'algorithmes cryptographiques à bas coût, `Pyjamask`, un chiffrement par bloc conçu pour être facile à masquer afin de résister aux attaques par canaux cachés a été proposé [GJK+20]. Avec Christoph Dobraunig et Jan Schoone, nous avons réalisé une attaque intégrale sur ce chiffrement [DRS20], dont la description sera simple et courte, car elle suit le schéma général que nous venons d'évoquer.

### 2.2.1 Description (succinte) de `Pyjamask`

`Pyjamask` est un chiffrement par bloc, qui se décline en deux variantes : une opérant sur 96 bits et une opérant sur 128 bits. Pour des raisons de clarté,

et parce que l'attaque complète ne fonctionne que sur `Pyjamask-96`, nous ne décrirons que celle-là.

Le nombre de tours de `Pyjamask-96` est 14. La fonction de tour est la composition de 3 opérations : l'addition de clef, la couche de boîtes-$S$ et une application linéaire $L$.

— L'addition de clef est l'addition bit à bit d'une clef de tour. Chaque clef de tour est l'image par une application linéaire de la clef maître, à laquelle on ajoute une constante de tour.
— Les boîtes-$S$ opèrent indépendamment sur 3 bits et sont quadratiques : leur inverse l'est donc aussi [6].
— L'application linéaire $L$ est vue comme la concaténation de trois applications linéaires opérant sur 32 bits. Ces applications linéaires sont définies par des matrices circulantes.

### 2.2.2 Cryptanalyse

#### Le distingueur

Le faible degré de la fonction de tour de `Pyjamask-96` et de la fonction de tour inverse (2 dans les deux cas), combiné avec la borne sur le degré [BCD11] qui provient principalement du fait que `Pyjamask-96` opère sur 96 bits [7] permet directement de dire que pour toute clef $K$ et pour tout espace affine $\mathcal{V}$ de dimension 94,

$$\sum_{x \in \mathcal{V}} \mathtt{Pyj}_K^{10}(x)$$

est une constante. Ici $\mathtt{Pyj}_K^{10}()$ est l'application de 10 tours de `Pyjamask-96`. La même propriété apparaît pour 10 tours de la fonction de tour inverse.

#### Choix des espaces affines

Comme cette propriété fonctionne pour n'importe quel espace affine, nous allons utiliser l'astuce classique : choisir en entrée des espaces affines particuliers qui sont transformés, par l'application d'un tour en d'autres espaces affines.

Afin d'utiliser le distingueur présenté ci-dessus, nous avons besoin d'espaces de dimension 94. Pour cela, nous saturons en entrée $96/3 - 1 = 31$ boîtes-$S$ et nous utilisons en entrée de la dernière boîte-$S$ un espace de dimension un. Plus précisément, on définit $\mathcal{V} = V_0 \bigoplus U$ où $U = \{x \in \mathbb{F}_2^{96}, x_0 = x_{32} = x_{64} = 0\}$, où $V_0 = \{0, v_0\}$ et $v_0$ peut être n'importe quel vecteur non-nul mais dont les composantes sont nulles aux indices non divisibles par 32. Comme une seule boîte-$S$ n'est pas saturée complètement, nous pouvons montrer le théorème suivant qui nous permet de gagner un tour.

6. Toute fonction vectorielle sur $n$ bits bijective est de degré multivarié inférieur à $n$.
7. C'est la raison pour laquelle l'attaque ne fonctionne pas sur tous les tours de `Pyjamask-128`.

**Théorème 2.2.1** *Notons $S$ l'application non-linéaire de* `Pyjamask-96`*. Soit $F$ n'importe quelle fonction Booléenne de degré inférieur à 94 et soit $\mathcal{V}$ un espace de dimension 94 comme défini plus haut. Alors la valeur*

$$\sum_{v \in \mathcal{V}} (F \circ S)(x + k + v)$$

*ne dépend que de 3 bits de $x$ et de 3 bits de $k$.*

### Où regarder

L'idée de l'attaque est de récupérer des équations en les bits de clef. Nous allons utiliser la technique usuelle qui consiste à exprimer un petit nombre de bits qui, après 11 tours du chiffrement, somment à une valeur constante, au moyen de leur expression en fonction des chiffrés, obtenue en inversant partiellement les derniers tours. Pour limiter la diffusion, nous nous concentrerons sur 3 bits en sortie d'une même boîte-$S$.

D'après le théorème 2.2.1, pour chaque valeur de $x$ non nulle, nous pouvons « théoriquement » pré-calculer l'expression de la somme en les 3 bits de clef sur trois bits de sortie bien choisis. La variable $x$ appartient à un espace de dimension 3 et doit être non-nulle [8]. Donc, on obtient $(2^3 - 1) \times 3 = 21$ équations en 3 bits de clefs. Le nombre de monômes que l'on peut former avec 3 variables est 7. Ainsi, nous pouvons pré-calculer exactement $21 - 7 = 14$ équations qui sont indépendantes de la clef et qui doivent être vérifiées après 11 tours du chiffrement. Nous pouvons aussi rajouter la couche linéaire dans le distingueur en travaillant avec une clef équivalente. Cependant pour pouvoir faire cela, nous avons besoin de réaliser l'attaque dans le sens du déchiffrement, puisque la couche linéaire est appliquée après la couche non-linéaire dans `Pyjamask`.

Enfin, la technique présentée fonctionne en considérant 2 valeurs en entrée pour une boîte-$S$ et le choix de la boîte-$S$ en question est arbitraire. Nous pouvons donc obtenir $14 \times 32 = 448$ équations en utilisant la valeur des trois mêmes bits en sortie.

### Récupérer des équations

Maintenant que nous avons un critère distinguant, nous pouvons passer à la phase de recouvrement de clef. La propriété distinguante peut être exprimée comme 448 équations ayant la forme suivante.

$$\sum_i \sum_j f_i \left( \mathtt{Pyj}_K^{-11.5}(C_j) \right) = 0 \,,$$

où les $f_i$ sont les fonctions composantes correspondant aux trois bits bien sélectionnés et où $j$ parcourt les espaces affines définis précédemment, de dimension 94 et $i$ parcourt un sous-ensemble de $\{0, 1, 2\}$.

8. On peut remarquer que si l'on prenait toutes les valeurs de $x$, alors la somme de toutes les équations obtenues aura toujours la valeur zéro car le chiffrement est une permutation. Il faut donc enlever une valeur de $x$ pour éviter d'avoir une équation qui ne nous donne pas plus d'information. Nous choisissons donc d'enlever arbitrairement la valeur de $x$ nulle.

Comme nous nous plaçons dans un contexte où l'attaquant observe toutes les paires de clair et de chiffré, les 448 équations ci-dessus peuvent s'exprimer aussi en fonction des message clairs, *i.e.*

$$\sum_i \sum_j g_i(K, P_j) = 0 \tag{2.2}$$

où les $g_i$ sont des fonctions composantes de 2.5 tours de `Pyjamask-96`[9]. Ceci permet de construire des équations en la clef $K$ de faible degré.

### Calculer les équations

Une façon de récupérer de l'information sur la clef consiste à faire des hypothèses sur la clef $K$ permettant de calculer les $g_i(K, P_j)$, vérifier si les équations (2.2) sont satisfaites et filtrer ainsi les possibilités. Pour faire cela, il faut que les fonctions $g_i$ ne dépendent que de peu de bits de clef. Nous devons par ailleurs calculer ces fonctions pour tous les $P_j$ qui sont au nombre de $2^{96}$. Dans le cas de `Pyjamask-96`, la diffusion de la couche linéaire ne permet pas de faire cela. Il faut donc trouver une autre façon, qui coûte moins cher, pour calculer l'expression des équations (2.2).

En fait, une analyse plus fine couche par couche du chiffrement permet de montrer que le nombre de monômes présents dans l'expression des $g_i$ est bien plus faible qu'attendu. Ainsi, au lieu de calculer les $g_i(K, P_j)$ pour tout $P_j$ et pour un sous-ensemble de clefs $\mathcal{K}$, nous calculons la forme polynomiale en la clef $K$ directement, dont nous pouvons borner précisément son nombre de monômes. Il est aussi possible de borner le nombre de monômes à évaluer (noté $N_e$ dans la table 2.1), permettant encore de diminuer la complexité de l'attaque.

### Résoudre les équations

Une fois que l'on a évalué les $g_i$ pour tous les clairs possibles, nous disposons de 448 équations non-linéaires en la clef. Nous ne pouvons pas connaître avec précision la forme des polynômes que l'on obtiendrait et le nombre de monômes dans ces polynômes, correspondant à $N_s$ dans la table 2.1, est bien plus grand que 448.

En revanche, nous pouvons facilement lister tous les monômes qui peuvent apparaître dans l'expression algébrique. En utilisant de plus les propriétés quadratiques de la fonction de tour, nous pouvons regarder le système sous un autre angle, et non pas résoudre les équations en la clef de tour, mais considérer l'expression du système en fonction d'une clef non-linéairement équivalente à celle d'origine. Ceci permet encore de réduire la taille du système que l'on a à résoudre, en termes de poids des équations.

Enfin, une stratégie de type *guess-and-determine* [HR00, EJ03] est possible. En déterminant précisément quels monômes peuvent apparaître, certaines hypothèses sur les bits de clef permettent de se ramener toujours à un système

9. Nous montrons ici l'attaque sur les 14 tours de `Pyjamask-96`.

| Tours | $N_m$ | $N_e$ | $N_s$ | $N_k$ |
|---|---|---|---|---|
| 1.5 | 648 | 571 | 569 | 56 |
| 2.5 | 7 642 713 | 3 910 569 | 3 829 480 | 154 |

TABLE 2.1 – Nombre maximal $N_m$ de monômes dans l'expression algébrique de `Pyjamask` ; $N_e$ est le nombre maximal de monômes à évaluer, $N_s$ le nombre maximal de monômes dans le système final et $N_k$ le nombre effectif de bits de clef pris en compte dans l'expression.

avec moins de 448 monômes grâce auquel il est possible de filtrer une classe de clefs secrètes et de scanner l'espace des clefs en un temps plus court que la recherche exhaustive.

### 2.2.3 Réflexions et améliorations

Dans ce travail, nous avons illustré plusieurs points qui peuvent sembler immédiats à première vue mais qui, je pense, soulèvent quelques problématiques, que ce soit du point de vue de la cryptanalyse ou de la conception. Les remarques ci-dessous serviront (un peu) de base à la section suivante.

#### Recouvrement de clef

Alors que la grande diffusion réalisée par la couche linéaire dans `Pyjamask` servait *a priori* à se prémunir d'un recouvrement de clef par simple diffusion incomplète, nous montrons qu'une analyse approfondie de la forme polynomiale permet de résoudre le système plus efficacement. Cette analyse nous permet aussi de gagner des tours dans la partie du recouvrement de clef.

Par ailleurs, nous montrons aussi qu'il est possible de gagner un tour sur le distingueur lui-même en le rendant dépendant en la clef. Autrement dit, on exprime une dépendance entre les polynômes en la clef qui correspondent aux coefficients devant chaque monôme de degré 94 en les variables publiques.

Enfin, il est aussi possible d'améliorer la complexité du recouvrement de clef en utilisant une transformée de Fourier rapide (FFT), comme proposé dans [TA14]. La complexité de ce recouvrement de clef dépend toujours du nombre de bits de clef et de messages clairs ou chiffrés (selon le sens de l'attaque) intervenant dans les fonctions $g_i$ qui permettent de filtrer les clefs. Cela a notamment été réalisé sur `Pyjamask` en 2022 [CHWW22b], conduisant à une amélioration de l'attaque, notamment sans avoir besoin d'utiliser les $2^{96}$ paires clairs/chiffrés mais $2^{95}$. En revanche la complexité en temps totale de l'attaque utilisant la FFT (au lieu de la résolution du système) dépasse le coût de notre attaque.

### Coût et recherche du distingueur

À la suite de notre travail, les auteurs de [CHWW22b] ont utilisé des modèles MILP et la division property afin de trouver d'autres distingueurs intégraux. Cependant les distingueurs trouvés ont la même forme que ceux identifiés « à la main ».

Le principal problème des attaques intégrales reste donc le coût en données du distingueur, qui nécessite souvent un très grand nombre de couples clairs/chiffrés choisis. Nous pouvons donc réellement nous poser la question de la pertinence des attaques intégrales dans un contexte pratique d'utilisation du chiffrement. En effet, si l'attaque nécessite presque tout le dictionnaire, *i.e.* tous les couples clairs/chiffrés du chiffrement par bloc, alors il est probable que le chiffrement doive faire face à des problèmes plus criants, dûs par exemple au mode utilisé si celui-ci ne garantit rien au delà de la borne des anniversaires [Leu24], ou tout simplement à des compromis temps-mémoire et des pré-calculs [Hel80] (que nous réalisons aussi dans notre attaque et en grand nombre).

### Points positifs

Cependant, montrer que l'on arrive à « casser » un chiffrement par bloc, dans le sens où nous pouvons retrouver la clef en un temps plus court que la recherche exhaustive est toujours intéressant pour comprendre jusqu'où on peut aller. Les attaques s'améliorant toujours, il est bon de commencer par chercher ce qui n'est pas réaliste. Ici, une analyse plus approfondie conduit réellement à une attaque plus efficace et on remarque que l'on arrive souvent à gagner de la sorte quelques tours par rapport à une analyse rudimentaire ne prenant en compte que le distingueur usuel et le critère du degré.

Ainsi, on montre que le critère du degré n'est évidemment pas suffisant pour se prémunir de ces attaques et qu'il est bon de rajouter quelques tours de marge de sécurité. Il en va de même des attaques algébriques : le degré n'est qu'un critère nécessaire et non suffisant, puisqu'en étudiant plus précisément la forme des polynômes que l'on obtient et des représentations équivalentes, il est généralement possible d'établir une stratégie (certes simple) plus efficace que les algorithmes génériques, et ce, même si nous sommes limités en nombre d'équations.

### Problèmes soulevés

Avant d'ouvrir un peu plus le débat, je souhaiterais pointer du doigt quelques problématiques liées à ce type d'attaque.

Tout d'abord, nous n'avons aucune garantie que les 448 équations que l'on obtient sont linéairement indépendantes. En effet, l'argument usuel est que les $P_j$ correspondant aux chiffrés que l'attaquant observe suivent une distribution uniforme (sans remise). Ceci n'est évidemment pas le cas, et rien ne nous assure que le critère que l'on utilise pour filtrer les mauvaises clefs les écarte effectivement : peut-être la propriété est-elle également valide dans le sens inverse, pour

toutes les clefs ou une grande partie. Bien que cela semble très peu vraisemblable, nous n'avons pas de garantie. Par exemple, dans l'attaque sur `Pyjamask`, le décalage $x$ doit être non-nul, car nous nous sommes rendus compte que cette équation était nécessairement linéairement dépendante des autres et n'apportait pas plus d'information, car elle provenait simplement de la propriété de bijectivité. On peut donc légitimement se poser la question suivante : Quelle quantité d'information apporte réellement le critère distinguant d'une attaque intégrale et sur les équations que l'on récupère ?

Par ailleurs, et comme nous l'avons mentionné brièvement plus haut, dans l'immense majorité des applications, un chiffrement par bloc doit être utilisé au sein d'un mode opératoire. L'étude des modes permet de prouver qu'un schéma de chiffrement est sûr si le chiffrement par bloc utilisé l'est. Mais il faut peut-être se poser la question dans l'autre sens : est-ce qu'une attaque sur un chiffrement par bloc s'adapte de manière pertinente quand ce dernier est utilisé comme primitive dans un mode prouvé sûr ? Dans le cas de `Pyjamask`, on répond par la négative à cette question : `Pyjamask-96-AEAD` utilise le mode `OCB3` [KR14] et dans ce mode, des masques dépendant de la clef et du nonce sont utilisés avant et après chaque chiffrement. Plus précisément, une valeur $O_0(K, N)$ est calculée, dépendant non-linéairement de la clef et du nonce, et une autre valeur dépendant uniquement de la clef $E_K(0)$ est utilisée pour dériver une base $(L_i)_{0\leq i<96}$ de $\mathbb{F}_2^{96}$. Alors chaque masque est de la forme

$$O_0(K, N) + \sum_{i=0}^{95} a_i L_i$$

où les $a_i$ sont connus et dépendent du nombre d'itérations du chiffrement. Ainsi, il est possible de choisir en entrée ou en sortie du chiffrement des espaces affines, mais sans pouvoir contrôler ni leur base ni leur décalage. De plus, l'ajout de ces masques dépendants de la clef doit être considéré dans notre scénario de résolution de système comme une nouvelle clef indépendante augmentant ainsi le nombre de monômes dans l'expression algébrique que l'on considère. Et enfin, la quantité de données chiffrées avec la même clef dans Pyjamask-96-AEAD est naturellement limitée à $2^{48}$ blocs, ce qui ne permet d'attaquer que 7 tours au lieu des 14 en pratique.

Tous ces problèmes ne sont pas spécifiques à `Pyjamask` et la section suivante aborde, à la fois plus en détail mais de manière plus générique, les pistes de recherche que nous pourrions réaliser dans ce contexte pour analyser peut-être plus finement la sécurité de nos schémas de chiffrement.

## 2.3 Élements de réflexion

Dans cette section nous réexaminons brièvement les attaques dites intégrales ainsi que les attaques algébriques et leur application conjointe. Ce qui suit reste à ce jour des pistes de réflexion quant aux améliorations de ce type d'attaques, notamment pour comprendre si elles pourraient être dévastatrices dans un cas pratique d'utilisation.

### 2.3.1 Critères de résistance

Comme nous l'avons vu précédemment, il faut clairement affiner les critères de résistance car le degré n'est évidemment pas un critère suffisant mais bien uniquement nécessaire.

#### Sur la division property

Depuis 2015, beaucoup de travaux ont été entrepris dans cette direction en utilisant et en affinant la *division property*, et cette technique a permis d'améliorer beaucoup d'attaques. Je citerais ici le travail de Phil Hebborn, Gregor Leander et Aleksei Udovenko [HLU23] qui passe en revue à la fois les avancées et les problèmes que nous ne pouvons pas encore résoudre avec cette vision. Dans cet article, les auteurs pointent deux problématiques liées à la *division property* :

1. « *Conventional* division property may only exhibit a monomial such that *all its multiples* are missing in the ANF. In addition, the technique is imperfect, meaning that it does not guarantee finding a distinguisher even if it exists ; inexistence of a distinguisher can never be proven either. However, application of conventional division property is feasible for nearly all used ciphers. »
2. « Perfect variants of division property, on the other hand, can be used to compute a chosen ANF coefficient exactly. On practice, however, it is feasible only in some use-cases and often requires specific optimizations and fine-tuning of utilized solvers / optimization software. »

Tout d'abord, le contexte dans lequel on se place généralement dans ce type d'attaque est l'étude des monômes maximaux (dans l'ordre partiel des monômes), afin de plus facilement déterminer si oui ou non ceux-ci apparaissent dans la forme algébrique normale de la primitive et ainsi monter une attaque. Les stratégies permettent de tracer tour après tour l'apparition de monômes maximaux (et donc potentiellement de grand degré). On ne déduit donc que peu d'information sur les monômes de bas degré et, si on voulait le faire avec la *division property* exacte, alors cela coûterait trop cher, car il faudrait calculer l'ensemble des chemins possibles et l'espace de recherche pour les monômes deviendrait beaucoup trop grand. Ceci est cependant possible dans certains cas, notamment pour des chiffrements par bloc opérant sur 64 bits [DL22].

Ainsi, je pense qu'il faut trouver une variante de ces propriétés qui soit peut-être moins fine mais dont la description est accessible par un ordinateur, ce qui fournirait d'autres informations sur la représentation polynomiale, en visant plutôt les monômes de bas de degré et potentiellement leur dépendance en la clef.

#### Sur le caractère creux de la représentation

Pour cette première direction je renvoie à la thèse de Margot Funk [Fun24] qui a développé un outil où la caractéristique étudiée est le nombre de monômes

présents par degré dans la forme algébrique normale. Cette représentation peut être sauvegardée en machine et nous pouvons :

1. estimer le nombre de monômes présents dans l'ANF, en prenant en compte l'annulation de monômes (mais sans savoir quels monômes s'annulent exactement) ;
2. si cette estimation est biaisée, alors on peut distinguer la primitive peut-être avec moins de données qu'en utilisant un seul monôme.

Dans ce travail, plusieurs effets ont été observés et nous montrons que des recherches plus approfondies sont possibles :

1. La proportion de monômes de degré inférieur au degré maximal est très faible ;
2. les tests expérimentaux montrent que la dépendance en la clef renforce le caractère creux des polynômes (il y a moins de monômes en pratique que ce à quoi on s'attend).

Tout ceci me pousse à chercher, au lieu des monômes maximaux qui apparaissent ou non dans la forme algébrique normale d'une fonction impliquée dans une primitive, des monômes qui seraient de degré plus petit. Pour ces monômes de degré plus petit, on ne pourra pas s'attendre à ce que le coefficient soit constant, mais le polynôme en la clef ainsi récupéré pourrait être combiné à d'autres (récupérés de la même manière) afin par exemple, d'obtenir des relations linéaires ou biaisées. Plus généralement, garder une information dépendante en la clef déduite des différentielles d'ordre supérieur, mais en dérivant selon un nombre de directions limité. Évidemment, ceci n'est probablement pas chose aisée, puisque les polynômes en la clef devant des monômes deviennent de plus en plus compliqués quand le degré des monômes diminue. En général, les attaques intégrales permettent de récupérer des équations simples en la clef : elles consistent, en utilisant beaucoup de données, à dériver plusieurs fois la fonction cryptographique afin de récupérer des équations plus simples (de plus petit degré). Par conséquent, utiliser moins de données pour se rapprocher d'une attaque pratique produira nécessairement des équations plus complexes. Ceci ne veut pas dire pour autant que toutes les combinaisons de ces équations le sont.

Enfin, nous recherchons souvent des équations vérifiées avec probabilité un. Même sans considérer la dépendance en la clef, nous pouvons aussi analyser le biais des équations obtenues par les différentielles d'ordre supérieur. Ceci a été observé dans [ADMS09, HPTY23], laissant penser qu'il reste encore beaucoup de critères à trouver dans les attaques intégrales, que ce soit en exploitant des biais ou en générant des équations.

### Arguments prouvés

Un des problèmes des attaques intégrales est que la représentation de la primitive peut changer l'analyse de sécurité, par exemple si l'on prend une boîte-$S$ équivalente linéairement à celle utilisée. Plusieurs travaux se sont intéressés

à ce problème [LDF20, DF20] permettant dans le même temps d'améliorer les attaques existantes.

Une autre manière de se prémunir de ce type d'attaque pour un chiffrement consiste aussi à essayer de prouver sa résistance (sous certaines hypothèses). Une première étape dans cette direction a été réalisée dans [HLLT21], où les auteurs arrivent à montrer, sous une hypothèse d'indépendance entre les clefs de tour, que pour tout ensemble non-trivial $\mathcal{X} \subset \mathbb{F}_2^n$ et tout $\beta$ non nul, l'expression

$$\sum_{x \in \mathcal{X}} \langle \beta, E_k(x) \rangle$$

dépend de la clef (pour un certain nombre de tours de chiffrement). Ceci est obtenu en considérant les coefficients de la forme algébrique normale des monômes de plus haut degré et en déterminant l'existence ou non de chemins de propagation en utilisant la *division property*. Les techniques utilisées permettent de montrer que les polynômes en la clef devant chaque monôme sont linéairement indépendants.

Bien que ce résultat soit intéressant, il ne prouve pas réellement la résistance aux attaques intégrales. Tout d'abord car, comme on l'a dit plus haut, on aimerait pouvoir considérer la dépendance en la clef dans le distingueur : par exemple la somme ou le produit de deux polynômes en la clef obtenus pourrait tout à fait se simplifier en un polynôme à peu de variables ou un polynôme linéaire. Nous pourrions aussi exploiter des propriétés de polynômes biaisés. Enfin, et peut-être de manière plus évidente : dès que les clefs de tour ne sont plus indépendantes, le résultat devient nécessairement faux, ce qui se produit quand la taille de l'entrée est égale à la taille de la clef ou bien que l'on utilise un chiffrement par blocs paramétrable.

En revanche, une telle preuve apporte peut-être des garanties de résistance quand la quantité de données accessibles à l'attaquant est limité : il faut pour cela que l'ensemble des polynômes pouvant être récupérés vérifie les critères suivants.

- Les polynômes sont tous linéairement indépendants ;
- Il y a peu de combinaisons linéaires de ces polynômes qui sont biaisées ;
- L'idéal engendré par cet ensemble de polynômes n'est pas engendré par un polynôme simple (par exemple linéaire ou très creux) ;
- Et probablement d'autres critères sont à déterminer.

Il semble peut-être illusoire d'essayer de prouver la résistance aux attaques intégrales. En l'état il faudrait probablement atteindre plus de maturité dans ce domaine pour définir plus formellement ce qui est faisable dans ce contexte par un attaquant.

### 2.3.2 Améliorations des attaques

#### Comment améliorer la première partie de l'attaque

Comme nous l'avons vu, une stratégie classique consiste à exploiter la structure des chiffrements afin d'obtenir un espace affine après quelques tours du

chiffrement. Dans un contexte de chiffrements suivant une stratégie de conception non-alignée, ceci est plus ardu. Mais, pour ces stratégies de conception qui utilisent parfois des fonctions de tour de bas degré, on peut réellement calculer le polynôme $p(x, k)$ qui définit plusieurs tours du chiffrement. Alors on peut naturellement poser la question suivante.

Quel nombre minimal d'hypothèses sur la clef (c'est-à-dire identifier un ensemble de clefs faibles $\mathcal{W}_k \subsetneq \mathcal{K}$) pouvons-nous faire et quelle(s) restriction(s) sur les entrées (c'est-à-dire identifier un sous-ensemble $\mathcal{X}$) afin que

$$\{p(x, k), x \in \mathcal{X}, k \in \mathcal{W}_k\}$$

soit un espace affine ?

Ce qui est perturbant ici c'est que l'on a une représentation polynomiale sur un nombre faible de tours qui est manipulable en machine, mais que l'on n'a pas d'algorithme (même heuristique) permettant de répondre à cette question. Peut-être que la représentation polynomiale n'est pas l'outil approprié et qu'il faut représenter nos fonctions différemment mais nous n'avons pas beaucoup d'autres représentations que celle-ci et la représentation itérative donnée par le chiffrement. En effet, cette question du minimum d'hypothèses à réaliser sur la clef apparaît déjà, mais dans un contexte différent où on cherche à minimiser le nombre d'hypothèses à réaliser sur la clef pour pouvoir calculer des composantes d'une boîte-$S$ [BCFG+21, Bro23]. Les algorithmes utilisés dans ce contexte ne fonctionnent cependant plus dès que le nombre de variables est grand.

### Amélioration du recouvrement de clef

De la même manière, il n'est pas aisé de déterminer la meilleure stratégie pour le recouvrement de clef, notamment quand la dépendance en la clef est complète, c'est-à-dire lorsque chaque bit ou combinaison de bits obtenu(s) par le distingueur dépend de tous les bits de clef. Il est aussi difficile de comparer les techniques de calculs partiels [FKL+01] ou de transformée de Fourier rapide [TA14]. Des éléments de réponse sont déjà là, notamment dans des modélisations permettant d'automatiser la recherche de distingueur et de recouvrement de clef [HSE23, HGSE24] au moyen de programmation par contrainte.

En revanche, si le degré des équations est petit et que l'on peut stocker les équations obtenues, alors il est possible de faire mieux en utilisant des techniques algébriques simples. Des améliorations des techniques employées jusqu'ici sont donc possibles, notamment sur une fonction de faible degré mais avec beaucoup de variables, en analysant plutôt le polynôme et en trouvant un algorithme qui ferait des hypothèses sur la clef plus intelligentes pour calculer un polynôme, et décrire plus simplement, le support d'un polynôme $p(k)$ donné [10]. Il faudrait pour cela un algorithme qui, à la donnée d'un polynôme, permet de décrire ces espaces différemment, peut-être par exemple en utilisant les premières pistes évoquées dans [BCFG+21, Bro23].

10. L'espace antécédent de 0 et son complémentaire.

### 2.3.3 Et pour une autre représentation ?

Jusqu'à maintenant, nous n'avons utilisé que la représentation multivariée des polynômes. Cependant, rien ne nous dit qu'en changeant la représentation, on n'obtienne pas des polynômes bien plus creux en représentation univariée. Même s'il y a peu de chances que cela arrive, rien ne nous prouve que ce n'est pas le cas. Cependant, chercher cette représentation et ne serait-ce qu'en connaître une information même partielle est très complexe, même pour un petit nombre de tours, et même pour des fonctions pour lesquelles cette représentation est naturelle [BCP23].

En effet, depuis quelques temps, sont apparues des constructions symétriques (fonctions de hachage et/ou chiffrements) définies au moyen d'une arithmétique non pas de $\mathbb{F}_2$, mais sur des grands corps. On peut par exemple citer MiMC [AGR+16], Poseidon [GKR+21, GKS23], Ciminion [DGGK21] ou Anemoi [BBC+22, BBC+23]. Il s'avère que les attaques algébriques sur ces types de chiffrement semblent mieux fonctionner [ABM24, Bar23, ACG+19, EGL+20, BCP23]. Cependant le caractère résistant aux attaques intégrales est encore relativement compris. Des premières observations existent par exemple sur les saturations de sous-groupes multiplicatifs [BCD+20] ou encore sur la *division property* [CHWW22a] mais je suis intimement convaincu que beaucoup de travail reste à faire dans cette direction.

## 2.4 Conclusion et perspectives

Les cryptanalyses intégrales permettent de simplifier drastiquement les équations en la clef que l'on obtient sans pour autant prendre en compte toute l'information que l'on a à notre disposition puisque nous exploitons uniquement la somme des valeurs considérées. Un compromis est probablement possible où nous chercherions à extraire de l'information d'un ensemble de polynômes compliqués à exploiter. Il reste donc à définir quelles sont les informations exploitables, et comment les chercher de manière efficace. Peut-être qu'une partie de la réponse peut être obtenue en limitant le nombre de données à la borne des anniversaires, afin de se focaliser sur la distribution de la sortie pour des espaces affines en entrée de dimension réduite.

Par ailleurs, il nous manque des algorithmes (même heuristiques) travaillant directement sur la représentation polynomiale pour trouver des distingueurs coûtant moins cher en prenant en compte la dépendance en la clef ainsi que pour trouver des stratégies de recouvrement optimales.

Enfin, il reste beaucoup de travail pour comprendre comment les attaques qui opéraient sur $\mathbb{F}_2$ s'appliquent quand on change l'arithmétique avec laquelle on travaille.

# Chapitre 3

# Point de vue plus générique

Les chapitres précédents décrivaient des cryptanalyses dédiées, c'est-à-dire des analyses de sécurité où l'on cherche à exploiter les détails de la représentation, notamment algébrique, des composants de la primitive. Dans ce troisième et dernier chapitre, nous verrons plutôt des cryptanalyses ou arguments de sécurité plus génériques sur différentes constructions cryptographiques. Les travaux présentés ici ne prennent donc plus en compte les détails des composants internes.

D'abord, la première section présente une cryptanalyse générique sur le mode duplex. Ensuite, nous étudierons deux constructions de fonctions de compression paramétrées par une clef et comparerons leur niveau de sécurité. Enfin, nous évoquerons les perspectives ouvertes par des propositions plus récentes de fonctions faiblement pseudo-aléatoires (WPRF) dont les définitions de sécurité sont amoindries, conduisant à des principes de construction bien plus simples mais dont la sécurité reste un problème peut-être plus ouvert.

## 3.1 Attaque générique sur le mode duplex

Cette section décrit succinctement un travail réalisé avec Henri Gilbert, Rachelle Heim Boissier et Louiza Khati [GHKR23]. Pour plus de détails, je renvoie à cet article ou bien à la thèse de Rachelle Heim Boissier [Hei24].

### 3.1.1 Le mode duplex

Une des méthodes pour réaliser un chiffrement authentifié avec potentiellement des données associées (AEAD) est la construction de type duplex, comme SpongeWrap ou MonkeyDuplex [BDPV12a, BDPV12b]. Le mode duplex est décrit assez simplement à la figure 3.1.

Une phase d'initialisation prend en entrée la clef secrète et le nonce (et potentiellement plusieurs blocs de données associées), et produit en utilisant une permutation $P_{init}$ un état (secret) de taille $b$ bits. Ensuite, et avec la même

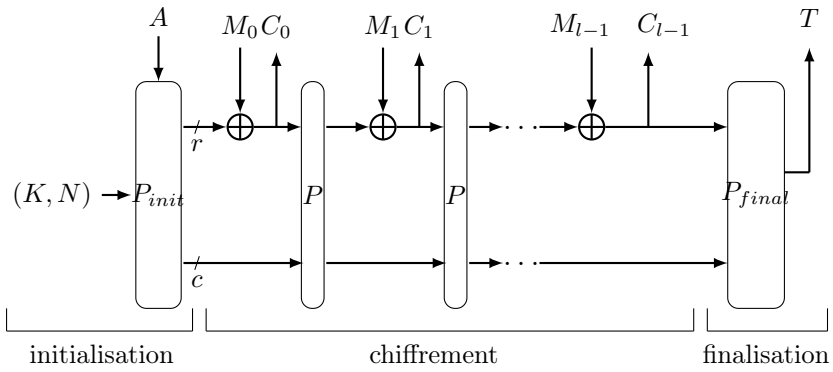


FIGURE 3.1 – Le mode duplex.

terminologie que celle nous avons utilisée au premier chapitre, cet état est séparé en deux : une partie de $r$ bits appelée état externe et l'autre partie de taille $c = b - r$ bits est l'état interne. Les valeurs successives prises par les $r$ bits de l'état externe sont alors utilisées comme suite chiffrante. Une fois que tout le message à chiffrer et à authentifier a été absorbé, une phase de finalisation, déterministe en l'état final, est appliquée pour produire une empreinte $T$ en tronquant la sortie de la permutation $P_{final}$.

### 3.1.2 Sécurité prouvée du mode duplex

Les premières preuves de sécurité du mode duplex ne permettaient pas de démontrer sa sécurité au delà de la borne des anniversaires en la capacité $c$. Plus précisément, le mode était prouvé sûr uniquement jusqu'à la borne

$$\mathcal{T} \leq \min\left\{2^{\frac{c}{2}}, 2^{\kappa}\right\}$$

où $\kappa$ est la taille de la clef, $c$ la capacité et tant que le nombre de requêtes en déchiffrement est significativement plus petit que $2^{\tau}$ où $\tau$ est la taille de l'empreinte $T$. Ensuite, en 2014 [JLM14] puis en 2019 [JLM+19] des preuves de sécurité sur le mode duplex ont montré que si le nombre de requêtes en déchiffrement était limité, alors la sécurité du mode était garantie au delà de la borne des anniversaires. Plus précisément, la borne simplifiée est la suivante :

$$\mathcal{T} \leq \min\left\{2^{\frac{b}{2}}, \frac{2^c}{\alpha}, \frac{2^c}{\sigma_d}, 2^{\kappa}\right\} \text{ avec } q_d \leq 2^{\tau}$$

où $\sigma_d$ est le nombre d'applications de la permutation employée dans le mode duplex nécessitées par les requêtes en déchiffrement de l'adversaire et où $q_d$ est le nombre d'essais de falsification par l'adversaire.

Ainsi, sans limiter les termes $\sigma_d$ et $q_d$, on ne peut supposer que le mode duplex assure une sécurité au delà de la borne des anniversaires. Avant notre

travail, la seule attaque générique [JLM+19] sur le mode duplex atteignait uniquement la complexité $2^c/\alpha$ où $\alpha$ est une petite constante.

### 3.1.3 Principe de l'attaque

L'objectif de l'attaque est de trouver un couple message chiffré-empreinte valide sans connaître la clef secrète. Notre attaque utilise principalement deux briques élémentaires : des requêtes en déchiffrement et des fonctions dites exceptionnelles.

Plus précisément, on remarque d'abord que fixer les blocs de messages chiffrés dans le déchiffrement du mode duplex permet de contrôler exactement $r$ bits de l'état interne à chaque itération. Ainsi, le déchiffrement du duplex peut être vu comme l'itération d'une fonction de la forme $P(C||\cdot)$ tronquée aux $c$ bits définis par l'état interne où $P$ est la permutation utilisée dans la phase de (dé)chiffrement du duplex. Nous avons donc accès à un ensemble de fonctions dont on va supposer que chacune est choisie de manière uniforme parmi toutes les fonctions de $\mathbb{F}_2^c$ dans $\mathbb{F}_2^c$.

Ensuite, nous pré-calculons, pour un grand nombre $(B)$ de ces fonctions notées $(f_i)_{0\leq i<B}$ (c'est-à-dire pour plusieurs valeurs de chiffrés $C_i$) la taille de la période ultime [1] des suites $(u_n^i)_{n\geq 0}$ définies par la relation $u_{n+1}^i = f_i(u_n^i)$ pour tout $i$ compris entre 0 et $(B-1)$ et tout entier $n$ positif et pour des valeurs initiales $u_0^i$ prises aléatoirement selon la distribution uniforme dans $\mathbb{F}_2^c$.

Dès que l'on a trouvé un certain $i_0$, *i.e.* un certain bloc de chiffré $C_{i_0}$, compris entre 0 et $(B-1)$ tel que la période de la suite correspondante est suffisamment petite (notée ici $p_0$) on arrête la phase de pré-calculs. On note alors $E$ l'ensemble des valeurs prises dans cette période. On utilise ensuite toutes ces valeurs pour attaquer le mode duplex en réalisant des requêtes de déchiffrement de la forme suivante. Soit $\ell$ un entier naturel. On réalise les requêtes en déchiffrement suivantes : $\left(C_{i_0}^{\ell}||T_j\right)_{0\leq j\leq p_0-1}$ où $C_{i_0}^{\ell}$ est la concaténation de $\ell$ blocs de chiffrés de même valeur $C_{i_0}$ et les $T_j$ pour $j$ allant de 0 à $p_0-1$ sont toutes les valeurs possibles de l'empreinte lorsque l'état interne juste avant le processus de finalisation a sa valeur dans l'ensemble $E$. Avec grande probabilité l'un de ces couples message chiffré-empreinte est valide.

#### Analyse de complexité

Maintenant que nous avons décrit le principe de l'attaque, il est nécessaire d'en évaluer le coût. Celui-ci dépend du nombre $B$ de fonctions $(f_i)_{0\leq i<B}$ que nous devons analyser en pré-calcul ainsi que du coût de leur analyse, afin d'obtenir au moins une suite de période de petite taille $|E|$. Il y a donc naturellement un compromis : une période de plus petite taille nécessite moins de requêtes en déchiffrement pour produire un couple chiffré-empreinte valide mais augmente le nombre de fonctions à analyser en pré-calcul pour en trouver une qui engendre

1. Comme nous sommes dans un espace de taille fini, toutes ces suites sont ultimement périodiques.

une suite avec une telle période. La longueur des chiffrés ($\ell$) utilisée doit aussi être prise en compte.

Afin de trouver ce compromis, il faut se plonger dans l'analyse des graphes de fonctions. Pour tout domaine fini $D$, chaque fonction de $D$ dans lui-même peut se représenter sous forme d'un graphe orienté $G = (D, A)$ où chaque noeud est étiqueté par un élément de $D$ et où l'ensemble des arêtes $A$ est défini par l'ensemble des couples antécédents-images de la fonction, *i.e.* $A = \{(x, y), f(x) = y\}$. Pour toute fonction, le graphe qui lui est associé peut être décomposé en composantes connexes disjointes, chacune ayant un unique cycle et où chaque noeud de chaque cycle est la racine d'un arbre.

Exprimé dans ces termes, nous cherchons une valeur de chiffré $C_{i_0}$ telle que le graphe de la fonction $f_{i_0} : \mathbb{F}_2^c \to \mathbb{F}_2^c$ qui lui est associée ait un petit cycle. Cependant, avoir simplement un petit cycle dans le graphe n'est pas suffisant car il faut aussi pouvoir détecter ce cycle et tomber dedans avec une probabilité non négligeable en itérant la fonction et en prenant un point de départ aléatoire. Par ailleurs, il faut aussi s'assurer qu'un point de départ pris aléatoirement n'est pas à une trop grande distance du cycle, car cela impliquerait dans l'attaque une longueur de chiffrés ($\ell$) trop grande. Par conséquent, nous nous intéressons à chercher des fonctions dont le cycle de la plus grande composante connexe du graphe associé est petit et nous cherchons aussi à estimer la distance au cycle [2] pour un point aléatoire.

Ainsi nous définirons deux types de composantes connexes différentes. Dans ce qui suit, $n$ désigne la taille du domaine de définition, *i.e.* $2^c$ dans le cas de notre attaque sur le mode duplex.

**Définition 3.1.1 ($\nu$-composante)** *Soit $0 < \nu < \frac{1}{2}$. Une $\nu$-composante est une composante d'un graphe de fonction dont le cycle est de taille au plus $n^{\frac{1}{2}-\nu}$.*

**Définition 3.1.2 ($(s, \nu)$-composante)** *Soit $0 < \nu < \frac{1}{2}$, $0 < s < 1$. Une $(s, \nu)$-composante est une $\nu$-composante dont la taille est plus grande ou égale à $ns$.*

L'analyse combinatoire des propriétés des graphes de fonction est très fournie : nous pouvons citer par exemple [Har60, Moo70, DeL88, FO90, FS09]. Parmi ces études, les principaux résultats que l'on utilisera sont les suivants.

**Moyenne de la taille du cycle et de la taille de la queue [FO90].** Pour un point pris aléatoirement dans le graphe d'une fonction tirée elle aussi aléatoirement, Flajolet et Odlyzko ont montré qu'en moyenne et asymptotiquement, la taille du cycle et la taille de la queue valent

$$\sqrt{\frac{\pi n}{8}}.$$

2. appelée plus loin la taille de la queue.

**Probabilité d'appartenir à une $\nu$-composante [Har60].** Pour un point pris aléatoirement dans le graphe d'une fonction tirée aléatoirement, la probabilité $p_\nu$ que la taille du cycle qui est connecté à ce point soit plus petite que $n^{\frac{1}{2}-\nu}$ vaut

$$p_\nu = \frac{\sqrt{2\pi}}{2n^\nu} + \mathcal{O}\left(\frac{1}{n^{2\nu}}\right).$$

**Probabilité qu'un graphe de fonction ait une $(s,\nu)$-composante [DeL88].** Pour une fonction prise aléatoirement, la probabilité $p_{s,\nu}$ que son graphe associé ait une $(s,\nu)$-composante peut être estimée par

$$p_{s,\nu} = \sqrt{\frac{2(1-s)}{\pi s}} n^{-\nu} \left[1 + \mathcal{O}\left(r_n(s)\right)\right]$$

où $r_n(s) = s^{-2}n^{-\frac{1}{2}-3\nu} + s^{\frac{-1}{2}}n^{-\nu} + n^{-\frac{1}{3}}$. Ainsi le terme de gauche de cette expression fournit une bonne estimation de cette probabilité quand la taille du domaine considéré est grande.

**Probabilité que l'image par $f^\ell$ d'un point aléatoire appartienne au cycle de la composante [Har60].** En plus d'estimer la probabilité que le cycle soit de petite taille, Harris donne dans son article une estimation asymptotique de la fonction de densité du nombre de successeurs d'un point. Ceci permet d'affirmer que la probabilité $p_\ell$ que $f^{\ell-1}(x)$ soit dans le cycle (pour $f$ et $x$ choisis aléatoirement) est minorée par

$$1 - \exp\frac{\ell^2}{2n}.$$

À partir de ces résultats, nous pouvons maintenant dériver les meilleurs choix pour notre attaque.

### Phase de pré-calculs

Dans cette phase de l'attaque, il faut analyser chaque fonction et trouver la taille de la pré-période et de la période, et ce pour plusieurs points pris aléatoirement. L'analyse des fonctions, pour chaque point peut se paralléliser, mais l'analyse d'une fonction pour un point se fait de manière séquentielle en utilisant par exemple l'algorithme de Brent [Bre80]. Sans rentrer dans les détails, la complexité de l'algorithme recherchant une fonction $f_i$ dont le graphe ait une $(s,\nu)$-composante peut s'estimer par la formule suivante.

$$\frac{1}{p_{s,\nu}}\left(3\sqrt{\frac{\pi n}{8}} + p_\nu\left(3\sqrt{\frac{\pi n}{8}} + n^{\frac{1}{2}-\nu}\right)\omega\right),$$

où $\omega$ est un grand entier fixé à l'avance correspondant au nombre de points aléatoires dont on doit tester l'appartenance ou non à la composante connexe ayant un petit cycle. En effet, il serait bien trop coûteux ($2^c$) de former le graphe

complet de chaque fonction pour garantir que la taille de la grande composante est effectivement grande. Pour s'assurer avec grande probabilité que la taille de la grande composante atteint effectivement $ns$, nous adoptons une stratégie heuristique qui considère des points aléatoires et calcule la proportion de ces points dans la composante. Ainsi, pour des choix appropriés de $\omega$ et $\nu = \frac{1}{4}$, nous pouvons garantir qu'avec une complexité bornée par $6n^{3/4}$, la phase de pré-calculs trouve une fonction $f_i$ dont le graphe possède une $(0.73, \frac{1}{4})$-composante [3], et ce avec une probabilité de succès supérieure à 0.99.

#### Requêtes en déchiffrement

Comme dit précédemment, lorsque nous avons identifié un bloc de chiffré $C_{i_0}$ définissant une fonction $f_{i_0}$ dont le graphe possède une grande composante reliée à un petit cycle, il faut soumettre à l'oracle de déchiffrement des requêtes de la forme

$$\left(C_{i_0}^{\ell} || T_j\right)_{0 \leq j \leq p_0 - 1} .$$

Il se peut que l'état initial se trouve en dehors de la grande composante connexe de la fonction. Pour pallier cela, il faut soumettre à l'oracle de déchiffrement des nonces différents. En notant $m$ le nombre de nonces différents, on peut montrer que la complexité de cette étape est majorée par

$$n^{\frac{1}{2} - \nu} m \ell ,$$

quand la probabilité de succès est égale à

$$1 - (1 - p_\ell s)^m .$$

### 3.1.4 Résultats, problématiques et travaux futurs

#### Résultats

Toujours sans aller dans les détails, de bons choix de paramètres permettent d'assurer une probabilité de succès supérieure à 0.95, en ayant une complexité asymptotique totale de l'attaque en

$$21 n^{\frac{3}{4}} .$$

Ceci donne alors une attaque sur le mode duplex en $\mathcal{O}\left(2^{3c/4}\right)$ où $c$ est la capacité. Cette complexité résulte d'un équilibre entre la phase de pré-calculs et la phase des requêtes en déchiffrement : il est à noter que si l'on passe plus de temps sur la phase de pré-calculs, alors il est possible de ramener le coût de l'attaque aussi proche que possible de $2^{c/2}$, rendant donc très peu probable l'existence d'une preuve de sécurité du mode duplex au delà de cette valeur [4].

3. La grande composante regroupe environ 3/4 des points et son cycle est de taille $n^{1/4}$.

4. On doit toujours prendre en compte la phase de pré-calculs. Cependant cette partie est indépendante du secret.

Des calculs sur de petites valeurs de capacité permettent de vérifier les heuristiques réalisées dans cette étude. Par souci de concision je renvoie le lecteur ou la lectrice à l'article correspondant ou bien à la thèse de Rachelle Heim Boissier [Hei24].

### Applications de l'attaque

L'attaque s'applique à tous les modes de type duplex mais nécessite que l'attaquant puisse réellement contrôler la valeur de l'état externe : l'attaque ne peut fonctionner en mode chiffrement. On peut aussi remarquer que la longueur des chiffrés nécessaire à utiliser dans l'attaque est très grande ($2^{c/2}$). Dès que l'on limite la quantité de données utilisées avec la même clef ou la longueur des messages, cela semble complètement entraver l'attaque dans le sens où l'entropie de l'état interne final ne peut être réduite drastiquement.

Ainsi, des rafraîchissements de session réguliers empêcheraient cette attaque. Un autre manière d'empêcher cette attaque est aussi de réutiliser la clef secrète dans le processus de finalisation, comme c'est le cas dans les modes utilisés pour Ascon [DEMS21] ou `NORX` [AJN16], ce qui rend impossible pour un attaquant de dériver l'empreinte en fonction de l'état final obtenu à l'issue du chiffrement.

Une autre stratégie qui permet de résister à ce type d'attaque consiste à utiliser une fonction de rétroaction empêchant un attaquant de contrôler la valeur en entrée de chaque application de la permutation, que ce soit en chiffrement ou en déchiffrement. Ceci est utilisé par exemple dans Beetle [CDNY18] ou dans SPARKLE [BBC+20].

### Suites et perspectives

À la suite de ce travail l'attaque a été améliorée dans [BHLS24] amenant une complexité asymptotique en $\mathcal{O}(2^{2c/3})$. Ces travaux sont aussi décrits dans la thèse de Rachelle Heim Boissier [Hei24] et utilisent des fonctions exceptionnelles [5] au sens que nous venons de décrire, mais aussi imbriquées. Par ailleurs, ces techniques s'adaptent aussi aux attaques sur les combinaisons de fonctions de hachage, permettant de gagner encore en complexité asymptotique.

Il reste peut-être d'autres caractéristiques des graphes de fonctions qui pourraient être exploitées différemment. Par exemple, on pourrait penser à des fonctions dont le cycle de la grande composante n'est pas forcément petit, mais dont la profondeur des arbres qui y aboutissent est, elle, petite. Dans une telle situation, la longueur des messages considérés dans l'attaque pourrait être réduite, au prix d'une augmentation du nombre d'empreintes à essayer et de la complexité de la phase de pré-calculs.

Dans l'estimation de la complexité, il est supposé (et vérifié expérimentalement) que la distribution de la distance des points au cycle est sensiblement la même quand on ne s'intéresse qu'aux cycles de petite taille. Pourtant ces deux événements ne sont pas indépendants mais il manque à ce jour une analyse plus approfondie et théorique de cette dépendance.

5. Des fonctions dont le graphe a une grande composante reliée à un petit cycle.

Enfin, il serait aussi intéressant de comprendre comment un tel phénomène d'existence de fonctions exceptionnelles se traduit lorsque l'on fixe la permutation utilisée. On pourrait par exemple se poser la question : comment construire une permutation $P$ telle que toutes les fonctions dérivées de la permutation n'aient pas de petit cycle relié à la grande composante connexe ?

## 3.2 Comparaison de fonctions de compression

D'un point de vue très haut niveau, la cryptographie a besoin de primitives qui étendent de l'information ou qui compressent de l'information, c'est-à-dire de fonctions de compression et de fonctions extensibles. Sans rentrer dans trop de formalisme, les fonctions de compression opèrent sur une entrée de taille potentiellement arbitraire et produisent une sortie de taille fixe (suffisamment grande pour atteindre un niveau de sécurité suffisant). Les fonctions extensibles, elles, opèrent sur des entrées de taille fixe (toujours suffisamment grande pour garantir un niveau de sécurité suffisant) et produisent des sorties de taille variable. Dans cette section nous nous focalisons sur les fonctions de compression et nous comparons la sécurité offerte par deux stratégies différentes visant à compresser de l'information de manière sécurisée : la stratégie sérielle et la stratégie parallèle. Les résultats qui suivent ont été publiés avec Joan Daemen et Jonathan Fuchs à CRYPTO [FRD23].

### 3.2.1 Préliminaires

Une fonction de compression est une famille de fonctions paramétrées par une clef secrète $K$ qui opèrent sur des messages de taille variable et qui produit une sortie de taille fixe. Pour qu'une fonction de compression soit sécurisée, la famille de fonctions doit être indistinguable de l'ensemble des fonctions ayant les mêmes domaines d'entrée et de sortie. Il y a plusieurs manières de construire des fonctions de compression sécurisées. Une première manière est d'utiliser une fonction de hachage protégée construite de la manière suivante : considérons une fonction de compression $F_K$ paramétrée par une clef $K$, ainsi qu'une autre fonction (par exemple un chiffrement par bloc $E_K$) opérant sur des données de taille identique à celle de la sortie de $F_K$. Alors la fonction de hachage protégée est définie par $E_{K'}(F_K(m))$. Si l'on suppose que $E_K$ est une permutation pseudo-aléatoire, alors la fonction de hachage protégée est une fonction pseudo-aléatoire s'il est difficile de produire des collisions sur $F_K$. Une autre manière de construire un MAC est d'utiliser la construction de Wegman-Carter(-Shoup) [WC81, Sho96] : avec une permutation pseudo-aléatoire $E_K$, un nonce $N$ et une fonction de compression paramétrée par une clef $F_K$, l'empreinte est calculée avec la formule $T = E_{K'}(N) + F_K(m)$.

Dans la première construction, la probabilité d'obtenir une collision sans connaître la clef pour deux messages est bornée par la notion d'$\varepsilon$-universalité, quand pour la deuxième construction la probabilité de générer un triplet $(m, N, T)$ valide est bornée grâce à la notion d'$\varepsilon$-$\Delta$-universalité [Sti95].

Dans un cas un peu plus pratique, *i.e.* quand il s'agit d'instancier les constructions, il y a plusieurs choix. On peut vouloir construire un MAC dont la sécurité est prouvée si le chiffrement par bloc sous-jacent est une permutation pseudo-aléatoire, c'est-à-dire où la fonction $F_K$ est construite à partir d'un chiffrement par bloc. C'est le cas de CBC-MAC [BKR94], CMAC [IK03], PMAC [BR02], Protected Counter Sums [Ber99] et LightMAC [LPTY16]. Des approches utilisant des fonctions de hachage comme HMAC [BCK96] peuvent aussi fonctionner en faisant reposer la sécurité sur la fonction de hachage.

Une autre stratégie peut aussi consister à utiliser des multiplications et additions où les propriétés d'$\varepsilon$-universalité et de $\varepsilon$-$\Delta$-universalité peuvent être dérivées avec des arguments plus mathématiques comme pour le mode Galois Compteur (GCM) [MV06].

Enfin, une dernière stratégie peut être d'utiliser des permutations publiques comme pour la phase de compression de Farfalle [BDH+17] ou de Pelican-MAC [DR05b]. Dans cette stratégie qui utilise des permutations publiques, on distingue deux conceptions différentes : la stratégie sérielle et la stratégie parallèle. Des analyses sur la stratégie sérielle ont déjà été menées dans [DR05a, DR10,DM19] mais supposent que la permutation choisie soit tirée aléatoirement.

Le but de ce travail est donc d'analyser la sécurité de la dernière stratégie sans considérer que la permutation publique est aléatoire mais bien fixée en comparant les constructions sérielle et parallèle, tout en considérant que les clefs utilisées pour masquer les messages sont toutes indépendantes et tirées selon la distribution uniforme. Plus précisément, nous montrons que la sécurité est entièrement déterminée par les propriétés différentielles de la permutation lorsque la sortie de la fonction de compression est protégée.

### 3.2.2 Définitions

Dans tout ce qui suit, nous nous placerons dans un groupe fini noté $G$ qui sera à la fois le domaine de sortie des fonctions de compression paramétrées et l'entrée des permutations publiques.

#### Universalité

Rappelons d'abord les notions d'$\varepsilon$-universalité et d' $\varepsilon$-$\Delta$-universalité [Sti95], où la probabilité est calculée sur l'ensemble des clefs possibles avec la distribution uniforme.

**Définition 3.2.1 ($\varepsilon$-universalité)** *Une fonction de compression $F_K$ paramétrée par une clef est $\varepsilon$-universelle si pour toute paire de messages $m$ et $m'$ différents,*

$$\Pr[F_K(m) = F_K(m')] \leq \varepsilon\,.$$

**Définition 3.2.2 ($\varepsilon$-$\Delta$-universalité)** *Une fonction de compression $F_K$ paramétrée par une clef est dite $\varepsilon$-$\Delta$-universelle si pour toute paire de messages $m$ et $m'$ différents et pour tout $\Delta \in G$ où $G$ est l'espace image de $F$,*

$$\Pr[F_K(m) - F_K(m') = \Delta] \leq \varepsilon\,.$$

**Différentielles**

Le but du travail étant de quantifier la sécurité des constructions, nous redéfinissons ici des quantités pertinentes. Le lecteur ou la lectrice averti.e remarquera que les quantités suivantes sont déterministes une fois que la permutation est fixée, mais s'utilisent en pratique de manière probabiliste, l'entrée de chaque application de la permutation étant uniformément distribuée.

**Définition 3.2.3 (Probabilité différentielle)** *Soit $f$ une fonction de $G$ dans $G$. La probabilité d'une différentielle $(a, b) \in G \times G$ notée $\mathsf{DP}_f(a, b)$ est la quantité*

$$\mathsf{DP}_f(a, b) = \frac{\#\{x \in G | f(x + a) - f(x) = b\}}{\#G}$$

**Définition 3.2.4 (Probabilité différentielle maximale)** *Soit $f$ une fonction de $G$ dans $G$. La probabilité différentielle maximale notée ici $\mathsf{MDP}_f$ de $f$ est la quantité*

$$\mathsf{MDP}_f = \max_{a \neq 0, b} \mathsf{DP}_f(a, b)\,.$$

**Définition 3.2.5 (Probabilité différentielle en norme 2 maximale)** *Soit $f$ une fonction de $G$ dans $G$. La probabilité différentielle en norme 2 de $f$ notée $\mathsf{MNDP}_f$ est le maximum du carré de la norme euclidienne du vecteur $\mathsf{DP}(a, \cdot)$.*

$$\mathsf{MNDP}_f = \max_{a \neq 0} \sum_b \mathsf{DP}_f^2(a, b)\,.$$

### 3.2.3 Constructions sérielle et parallèle

**La construction sérielle**

La construction sérielle fonctionne comme suit et est décrite à la figure 3.2. Le message est décomposé en blocs d'éléments de $G$. Chaque bloc de message est ajouté à un bloc de clef, puis une permutation $f : G \to G$ est appliquée. Le résultat est ajouté à la deuxième clef et au deuxième bloc de message, et la permutation est appliquée à nouveau. Le processus est itéré jusqu'à ce que le dernier bloc de message soit absorbé et la permutation est alors appliquée une dernière fois.

Dans ces conditions, nous montrons que le niveau de sécurité de la construction sérielle est entièrement déterminé par la probabilité différentielle maximale de la permutation. Plus formellement, on a le théorème suivant.

**Théorème 3.2.1 (Universalité de la construction sérielle)** *La construction sérielle utilisant la permutation $f : G \to G$ est $\mathsf{MDP}_f$-universelle et $\mathsf{MDP}_f$-$\Delta$-universelle.*

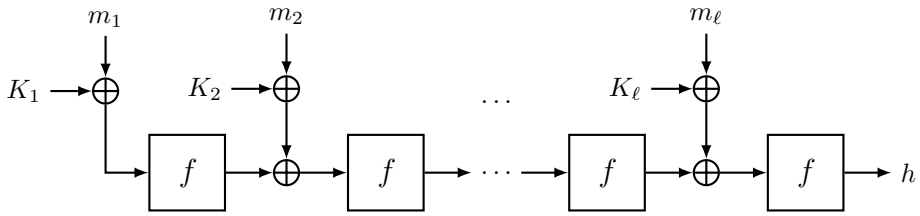


FIGURE 3.2 – Construction sérielle.

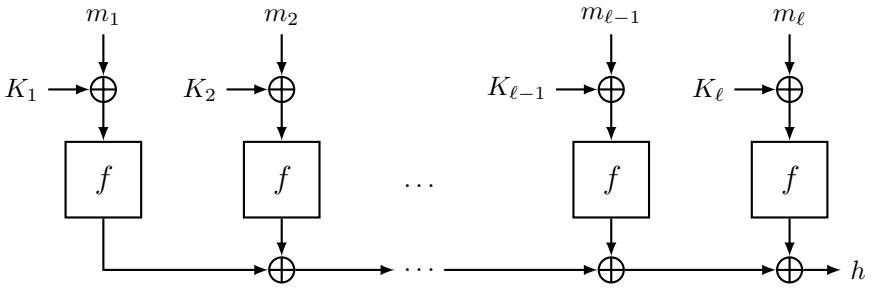


FIGURE 3.3 – Construction parallèle.

**La construction parallèle**

La construction parallèle est, elle, décrite à la figure 3.3. Le message est décomposé en blocs d'éléments de $G$. Chaque bloc de message est ajouté à un bloc de clef, puis une permutation $f : G \to G$ est appliquée à chacun de ces blocs. La sortie de la fonction de compression est tout simplement la somme dans $G$ des sorties de toutes les applications. Alors, le niveau de sécurité de la construction parallèle est déterminée par la probabilité différentielle maximale en norme euclidienne.

**Théorème 3.2.2 (Universalité de la construction parallèle)** *La construction parallèle utilisant une permutation $f : G \to G$ est $\mathsf{MDP}_f$-$\Delta$-universelle et $\mathsf{MNDP}_f$-universelle.*

### 3.2.4 Implications

Alors que les deux constructions semblent apporter le même niveau de sécurité pour une taille de sortie donnée et pour une permutation prise au hasard, ce n'est pas le cas dès que l'on fixe la permutation.

**Argument de sécurité**

L'apport de nos résultats est de déterminer le niveau de sécurité effectif pour une permutation donnée (par exemple publique) sans l'argument qui fonctionne pour une permutation tirée aléatoirement selon la distribution uniforme. En effet, le cadre où l'on considère que les permutations sont tirées aléatoirement peut être considéré comme irréaliste puisqu'en pratique, nous fixons les permutations utilisées.

Si l'on travaille sur des tailles d'état raisonnables, de telle sorte que l'attaque classique en $\sqrt{\#G}$ soit impraticable, on montre que ce sont les seules propriétés différentielles de la permutation utilisée qui définissent la sécurité. Alors, on voit que la construction parallèle offre une meilleure sécurité que la construction sérielle puisque l'on a naturellement, pour toute fonction $f$ que

$$\mathsf{MDP}_f \geq \mathsf{MNDP}_f \,,$$

avec égalité si et seulement si une ligne de la table des différences de la permutation ne prend que deux valeurs $(0, \delta)$ et que $\delta = \#G \times \mathsf{MDP}_f$, ce qui est hautement improbable.

**Évaluation pratique des primitives**

Maintenant que l'on a un critère précis de sécurité, il faut l'évaluer en pratique et ce n'est pas chose aisée. La probabilité différentielle maximale (MDP) peut être estimée de plusieurs manières mais cela reste un problème complexe.

On peut déterminer tous les chemins différentiels de petit poids avec des techniques par recherche d'arbre comme établi dans [MDV17, DMM21, EMMD22, MMGD22, MDV23, BFR24] ou bien avoir des arguments plus théoriques utilisant la *wide-trail strategy* [DR20].

Cependant aucune technique ne permet d'examiner tous les chemins différentiels et on ne sait pas pour autant s'il peut y avoir des effets de regroupement des chemins de poids faibles ni si les transitions peuvent être vues comme une chaîne de Markov, *i.e.* si la différence instant $t$ est entièrement déterminée par celle à l'itération précédente.

Même si calculer précisément le MDP ou le MNDP d'une permutation fixée est difficile, et fait l'objet de multiples travaux de recherche, nos résultats fournissent néanmoins une première estimation du niveau de sécurité pour les constructions sérielle et parallèle au moyen des propriétés différentielles de la permutation.

### 3.2.5 Synthèse

En protégeant la construction par un chiffrement par bloc, on peut s'autoriser à relaxer le modèle de sécurité dans lequel on se place. Ici l'attaquant n'observe rien sauf s'il y a une collision. Ce modèle de sécurité permet de limiter drastiquement le pouvoir de l'attaquant et ici, il ne peut utiliser que les propriétés différentielles de la permutation choisie.

Il faut cependant être capable de garantir que les clefs utilisées sont uniformes et indépendantes (ce qui n'est jamais le cas en pratique) et évidemment combiner la sortie de la construction par une autre construction bien plus forte cryptographiquement.

On se ramène finalement au seul problème d'estimer les propriétés différentielles d'une permutation publique, ce qui dans un contexte itératif est encore un sujet complexe et ouvert mais avec de récentes avancées [BR22], contrairement au cas des fonctions dont on dispose d'une expression polynomiale (par exemple la fonction inverse) pour lesquelles on peut calculer exactement les propriétés différentielles.

## 3.3 Cryptanalyse de fonctions faiblement pseudo-aléatoires

Dans tout ce que nous avons vu jusqu'à maintenant, nous pouvons remarquer que selon les contextes d'utilisation, le niveau de sécurité nécessaire pour garantir la sécurité des constructions varie. Dans l'attaque générique sur le mode duplex, si on limite dans le protocole le nombre de tentatives de falsification ou la quantité de données chiffrées avec la même clef, alors l'attaque ne fonctionne plus. La section précédente montre aussi que l'on n'a probablement pas besoin d'une permutation extrêmement solide cryptographiquement pour compresser l'information si l'on délègue la sécurité à un Générateur Pseudo-Aléatoire pour générer des clefs et si l'on protège la sortie avec une seule application d'une primitive cryptographiquement sûre. De même, nous n'avons pas besoin des mêmes arguments de sécurité dans le mode duplex pour la phase d'initialisation et de finalisation que pour la phase de chiffrement/déchiffrement. La même observation s'applique aux chiffrements à flot.

Afin de gagner en performance, il est possible, selon les cas d'utilisation de se contenter de propriétés de sécurité plus faibles à certains endroits du chiffrement sans théoriquement changer la sécurité de la construction complète. Dans cette section, nous passerons brièvement en revue de nouvelles constructions qui garantissent un niveau de sécurité suffisant si la primitive utilisée est sécurisée, mais dans un modèle bien plus faible. Cette section doit être vue comme un aperçu des prochaines constructions que nous souhaitons analyser.

### 3.3.1 Fonctions (faiblement) pseudo-aléatoires

D'un point de vue assez théorique, certaines recherches s'intéressent à déterminer à quel point on peut réduire la complexité de circuit (profondeur multiplicative par exemple et nombre de portes logiques) réalisant une Fonction Pseudo-Aléatoire qui reste sécurisée. On sait qu'il n'est pas possible d'obtenir des PRFs dans la classe de complexité $\mathtt{ACC}^0[p]$ [CIKK16], *i.e.* les circuits ayant une taille polynomiale en la taille de l'entrée, une profondeur constante mais sans borner le nombre de portes logiques utilisées classiques `AND`, `OR` et `NOT` mais aussi en pouvant utiliser des portes réalisant la réduction modulo $p$. De même on

sait qu'il n'est pas possible d'avoir des fonctions pseudo-aléatoires faibles dans la classe $\mathtt{AC}^0$ [LMN89].

Les fonctions pseudo-aléatoires ont pour but de borner l'avantage d'un adversaire qui vise à distinguer une famille de fonctions paramétrées par une clef (la fonction pseudo-aléatoire) et une famille de fonctions aléatoires ayant les mêmes domaines de définition.

Dans le contexte d'une fonction pseudo-aléatoire classique, l'adversaire peut choisir les entrées envoyées à l'oracle pour pouvoir distinguer les fonctions. Il peut même les choisir de manière adaptative. En revanche, dans le cas des fonctions faiblement pseudo-aléatoires, l'adversaire est réduit à des couples entrée-sortie qu'il ne choisit pas : plus précisément les entrées correspondant aux couples entrée-sortie que l'adversaire observe sont limitées et supposées suivre la distribution uniforme dans l'espace de définition de la fonction. De manière formelle, une fonction faiblement pseudo-aléatoire peut être définie comme suit.

**Définition 3.3.1 (Fonction faiblement pseudo-aléatoire)** *Soit $\lambda$ un paramètre de sécurité. Une fonction $\mathcal{F}_\lambda : \mathcal{X}_\lambda \times \mathcal{K}_\lambda \to \mathcal{Y}_\lambda$ de domaine $\mathcal{X}_\lambda$, d'espace de clefs $\mathcal{K}_\lambda$ et d'espace de sortie $\mathcal{Y}_\lambda$ est une $(\ell, t, \varepsilon)$-fonction faiblement pseudo-aléatoire si pour tout adversaire réalisant $t(\lambda)$ calcul, son avantage pour distinguer les distributions*

$$\{(x_i, \mathcal{F}_\lambda(x_i, k))\}_{i\in[\ell]} \text{ et } \{(x_i, y_i)\}_{i\in[\ell]}$$

*où $k \leftarrow_\$ \mathcal{K}_\lambda$, $x_i \leftarrow_\$ \mathcal{X}_\lambda$ et $y_i \leftarrow_\$ \mathcal{Y}_\lambda$, est borné par $\varepsilon$.*

Dans des termes plus concrets, cela signifie que le modèle d'attaquant est un modèle à clair connu et non à clair choisi. Dans ces conditions, un grand nombre d'attaques ne fonctionnent plus dès que la taille de l'entrée est grande, notamment les attaques différentielles. En effet, si l'on prend une taille d'entrée assez grande et qu'on limite la quantité de données connues, on peut montrer qu'il est trop rare qu'un nombre suffisant de paires d'entrées satisfassent une différence fixée à l'avance. Ainsi, pour gagner en efficacité dans certains contextes d'application (notamment pour des protocoles de cryptographie avancée comme du calcul multi-partite ou bien du chiffrement complètement homomorphe), une direction de recherche actuelle vise à construire des fonctions faiblement pseudo-aléatoires ayant une faible complexité d'implémentation, .

### 3.3.2 Quelques propositions

Le but de cette section n'est pas de faire un tour exhaustif des fonctions faiblement pseudo-aléatoires ou de nouvelles constructions symétriques pour des applications avancées, mais nous choisissons volontairement de n'en présenter que trois dont nous discuterons après le niveau de confiance. J'aimerais dans un premier temps m'intéresser à l'analyse de sécurité de ces constructions.

### Crypto Dark Matter

En 2018, dans leur article *Exploring Crypto Dark Matter* à TCC [BIP+18], les auteurs proposent une construction de fonctions faiblement pseudo-aléatoires qui consiste simplement à appliquer à l'entrée une application linéaire définie par la clef secrète dans un anneau modulo un certain entier $q$ (2), puis d'envoyer ce vecteur résultant en faisant tout simplement un produit scalaire du résultat mais dans un autre anneau modulo $p$ (3). Plus précisément, la clef $K$ est une matrice à coefficients binaires ($q = 2$) et de taille $n \times m$. Ainsi, à une entrée $x \in \mathbb{F}_2^n$, on calcule $y = K \cdot x$ qui est un vecteur binaire à $m$ coordonnées. La sortie est alors

$$\sum_{i=0}^{m-1} y_i \mod 3$$

lorsque $p = 3$.

### VDLPN

En 2020, des auteurs ont proposé une construction de fonctions faiblement pseudo-aléatoires [BCG+20] dont l'expression algébrique est extrêmement simple et peut être décrite par la formule suivante.

$$F_K(x) = \bigoplus_{i=1}^{D} \bigoplus_{j=1}^{w} \bigwedge_{k=1}^{i} (x_{j,k} \oplus K_{i,j,k})$$

Autrement dit, cette construction ajoute plusieurs fonctions triangulaires indépendantes, où les fonctions triangulaires sont des fonctions utilisées par exemple dans le chiffrement FLIP [MJSC16]. Par conséquent, cette expression est très structurée et naturellement cette fonction ne peut être une fonction pseudo-aléatoire [6] mais est conjecturée comme une fonction faiblement pseudo-aléatoire.

### Rasta

En cryptographie symétrique, même si le formalisme n'est pas le même, sont apparues récemment des stratégies de conception différentes de celles utilisées usuellement cherchant à s'adapter aux nouvelles applications. Rasta [DEG+18] est un chiffrement à flot dont le but est d'être performant dans le contexte de protocoles de calcul multipartite, dans des preuves à divulgation nulle de connaissance ou dans un chiffrement hybride combiné avec un chiffrement complètement homomorphe. Il s'inscrit dans une longue lignée de travaux réalisés ces dernières années, ayant produit des primitives qui peuvent opérer sur des corps de caractéristique impaire ou sur des grands corps, mais dont les opérations utilisées sont compatibles avec les protocoles ci-dessus.

6. La fonction est directement cassée par simple analyse différentielle.

Une des premières constructions dans ce contexte est le chiffrement LowMC [ARS+15] qui est un chiffrement par bloc avec des couches non-linéaires incomplètes et dont le modèle de sécurité est relaxé : LowMC demande une sécurité avec uniquement une seule paire clair-chiffré. LowMC a reçu un très grand nombre de cryptanalyses, notamment en utilisant des techniques algébriques qui s'avèrent être extrêmement efficaces [DEM16, DLMW15, RST18, BBDV20, LIM21, BBVY21, ZLL24, LSW+22, LMSI22, SCW23]. En 2016, il est observé que l'utilisation du chiffrement à flot semble plus adaptée dans ce contexte [CCF+16, CCF+18]. Ont ainsi été proposés Kreyvium puis FLIP, un autre chiffrement à flot dont la stratégie de conception a la particularité de déléguer la diffusion à un aléa public [MJSC16] mais qui pose quelques problèmes de sécurité [DLR16] nécessitant de modifier la taille des paramètres.

C'est dans cette direction que les auteurs de Rasta ont souhaité aller. Rasta délégue aussi une partie de la sécurité à un aléa public qui définit la structure du chiffrement : au lieu d'utiliser des permutations de bits prises aléatoirement et publiques, les auteurs tirent des couches linéaires aléatoires permettant d'obtenir une meilleure diffusion en moyenne. Plus formellement, un nonce $N$ et un compteur $ctr$ sont utilisés en entrée de SHAKE-256 [SHA15] générant ainsi un grand nombre de matrices bijectives opérant sur $n$ bits notées $A_{j,N,ctr}$ où $j$ est l'indice du tour. Alors chaque bloc de suite chiffrante est obtenu en alternant la composition de ces applications linéaires avec la fonction $\chi : \mathbb{F}_2^n \to \mathbb{F}_2^n$ dont nous avons largement parlé précédemment où $n$ est impair et est au dessus du niveau de sécurité visé. La suite chiffrante $Z$ est alors définie par la formule suivante.

$$Z = (A_{r,N,ctr} \circ \chi \circ A_{r-1,N,ctr} \circ \chi \circ \cdots \chi \circ A_{0,N,ctr}(K)) \oplus K$$

Enfin, suite à ces propositions, des constructions ont été proposées en utilisant des stratégies différentes comme MiMC [AGR+16] ou GMiMC [AGP+19], Jarvis [AD18], HADES [GLR+20], Poseidon [GKK+19], Vision [AAB+20], Rescue [AAB+20], Ciminion [DGGK21] ou encore Anemoi [BBC+23] pour n'en citer que quelques uns.

### 3.3.3 Et la sécurité dans tout ça ?

Tout d'abord ces constructions ainsi que celles que je n'ai pas citées sont assez récentes. Il faut du temps aux attaques et aux cryptanalystes afin de gagner en confiance. Je dirais même qu'il faut potentiellement encore plus de temps, car il n'y a, à ma connaissance, pas de réduction efficace entre le niveau de sécurité à clair choisi et à clair connu. En cryptographie symétrique, on se place toujours dans le modèle de l'attaquant le plus fort, garantissant par la même occasion la sécurité dans un modèle plus faible. Évaluer la sécurité dans un modèle plus faible est donc quelque chose d'inhabituel, qui nécessite naturellement plus de maturité.

### Crypto Dark Matter et son utilisation

Une première analyse de la construction modulo $p$ - modulo $q$ a été effectuée dans [CCKK21] permettant de gagner un certain avantage mais sans pour autant casser la construction d'un point de vue théorique. Cependant c'est une première étape vers la définition des tailles qu'il faudrait utiliser pour atteindre un niveau de sécurité suffisant.

Par ailleurs, cette construction a vu son intérêt grandir dans un contexte différent, celui du rafraîchissement de clef [DMMS21]. Dans cet article, les auteurs proposent d'utiliser la construction Crypto Dark Matter pour générer des clefs de sessions différentes à partir d'une clef maître dans un but de résistance aux attaques par canaux cachés. L'idée est d'avoir un schéma de rafraîchissement de clef qui soit facile à masquer. Dans cet article, les auteurs montrent déjà que tous les choix de $p$ dans la construction ne sont pas équivalents, et qu'il faut aussi des tailles suffisamment grandes pour les entrées-sorties. Nous avons ensuite montré qu'un des modèles était nécessairement cassé dans ce contexte dans [HMM+23], mais laissant encore ouverte la question de savoir si utiliser une telle construction est correcte du point de vue du cryptanalyste. Par ailleurs, ceci soulève également la question de savoir s'il est possible de cacher un générateur d'aléa dans une implémentation matérielle.

Enfin, Crypto Dark Matter est aussi proposé pour être utilisé en combinaison avec TFHE [ADDG24].

Une analyse de sécurité a ensuite largement amélioré la première attaque proposée et montre aussi qu'il faut choisir des paramètres plus grands [JMN23]. Il est à noter qu'il n'y a pas encore eu beaucoup d'analyses sur cette construction et qu'il y a donc probablement beaucoup à améliorer dans les attaques. C'est un travail auquel j'aimerais consacrer du temps dans les prochaines années.

### Sur VDLPN

La construction étant tellement simple algébriquement, je pense qu'il est tout à fait possible d'améliorer significativement les attaques génériques. C'est d'ailleurs un travail que j'ai entamé récemment au sein de l'équipe crypto du laboratoire de mathématiques de Versailles. Avec mes étudiant.e.s et Balthazar Bauer, nous avons déjà observé des propriétés inhabituelles de cette fonction. En effet, cette fonction est conçue pour générer deux suites pseudo-aléatoires corrélées entre elles lorsque deux entrées de la fonction le sont aussi. Il me semble peu probable de réaliser cela aussi simplement et avec une description aussi simple.

En revanche, analyser de telles constructions change radicalement la manière de penser. En particulier, cela exclut complètement la très puissante cryptanalyse différentielle et laisse à notre disposition uniquement les outils de cryptanalyse linéaire et algébrique.

L'idée derrière cette construction est que les monômes de haut degré sont là pour résister aux attaques algébriques quand les monômes de bas degré assurent l'absence de biais linéaires. Il faut donc des stratégies différentes, notamment en

prenant en compte la dépendance en la clef dans les monômes de bas degré afin d'avoir une idée plus précise du système à résoudre : il n'y a pas d'éléments qui nous permettent de dire si le système induit par la fonction, pour une certaine classe d'entrées choisies conjointement a, par exemple, un rang plus petit qu'attendu. Reste cependant à déterminer quand ces phénomènes arrivent et avec quelle probabilité.

### Sur Rasta

L'idée générale consiste dans la plupart des cas à déléguer l'aléa de manière publique mais partagée et d'utiliser cet aléa pour le chiffrement. Cela signifie qu'on transfère la sécurité à plusieurs étapes du protocole et donc il faut bien faire attention à ce que cet aléa soit géré correctement dans l'implémentation.

Je suis relativement perplexe quant à cette stratégie, car ça signifie que le cryptanalyste a, en un sens, beaucoup plus d'instances de chiffrement à regarder : si la structure du chiffrement elle-même change à chaque application, où donner de la tête pour pouvoir garantir la sécurité ? L'espace à considérer est extrêmement grand, donc l'aléa doit être introduit à un endroit où on peut s'assurer que toutes les instances du chiffrement vont se comporter de la même manière. Ce n'est par exemple pas le cas dans Rasta où pour garantir la sécurité, il faut regarder toutes les matrices inversibles de taille $n$ où $n$ est la taille des blocs, et analyser l'ensemble de toutes les propriétés. Il y a peu de chances qu'un bloc n'ait pas une propriété étrange mais il est très difficile de le trouver, la taille de l'espace de recherche considéré étant beaucoup trop grande. À ce jour seules les attaques algébriques fonctionnent avec une inversion du dernier tour en exploitant une fonction implicite de $\chi$ [LSMI21].

Pour améliorer les performances de Rasta, une variante, Dasta, a ensuite été proposée en 2020 où les matrices aléatoires utilisées dans la couche linéaire du chiffrement sont dans un ensemble plus réduit [HL20]. Toutefois, on manque à mon sens du recul sur cette stratégie de conception et notamment d'une analyse combinatoire permettant d'identifier d'éventuelles matrices faibles et leur probabilité d'apparition. Changer radicalement la stratégie de conception est très intéressant intellectuellement, mais je ne me prononcerais pas quant à la sécurité offerte par ce chiffrement.

Enfin, un grand nombre des nouvelles constructions que nous avons citées précédemment se sont révélées attaquables, principalement par des techniques algébriques appliquées à une modélisation du système d'équations, réussissant à garantir une résolution plus simple qu'attendu [ACG+19, BCD+20, EGL+20, LAW+23, LMØM23, ZWY+23, Bar23, BBL+24, LMM24, LKSM24]. En fait, opérer sur de gros corps nécessite tout de même une description qui permette d'assurer certaines propriétés cryptographiques et donc d'avoir dans ces constructions un minimum de structure pour garantir la sécurité face à des attaques classiques. En revanche, l'expression très simple des fonctions de tour dans le corps conduit à des comportements inattendus du système polynomial. Ces constructions ont relancé le sujet des cryptanalyses algébriques. Il faut analyser leur résistance aux attaques intégrales mais en tirant parti de la représentation utilisée par des

polynômes avec peu de variables mais sur un grand corps.

## 3.4 Perspectives

Il reste encore des écarts entre les preuves de sécurité et les attaques génériques que l'on peut combler. Par ailleurs, il reste aussi probablement pléthore de propriétés à chercher sur les fonctions ou les permutations utilisées qui permettraient d'améliorer les attaques. Mais il ne suffit pas de dégager les propriétés permettant d'accélérer les attaques génériques, il faut aussi que celles-ci soient identifiables en pratique et qu'elles apparaissent avec une probabilité non négligeable, ce qui demande des analyses combinatoires détaillées.

Dans certains cas, il est aussi possible de définir plus précisément les capacités de l'attaquant lorsqu'on se place dans un modèle de sécurité différent, permettant d'arguer la sécurité sous des hypothèses peut-être plus réalistes mais que nous ne pouvons pas toujours garantir entièrement. Il faut donc continuer à analyser les primitives. Les preuves de sécurité réalisées dans des modèles réduits permettent alors de mieux cibler les failles, ce qui conduit les cryptanalystes à se concentrer sur des propriétés qui ont du sens relativement à l'utilisation des primitives.

Peut-être un vent nouveau souffle-t-il sur la cryptographie, apportant quantité de constructions dont les revendications de sécurité sont moindres mais qui visent des applications de cryptographie avancées, où les performances sont bien plus critiques que dans un contexte classique. Je n'ai présenté ici que trois constructions que je trouve intéressantes et que j'aimerais regarder en détail avec une casquette de cryptanalyste.

Ces constructions étant récentes, on ne peut à ce jour avoir un niveau de confiance élevé dans leur utilisation. Par ailleurs, ces constructions doivent être combinées à un protocole bien défini et ne sont donc pas à utiliser dans un contexte classique. Par ailleurs, l'utilisation d'aléa dans ces constructions, que ce soit dans les entrées ou dans la description du chiffrement me semble relativement risquée, à moins de garantir par une analyse fine et poussée qu'aucun effet problématique n'apparaît avec une probabilité non-négligeable. Il nous manque donc des analyses fines et détaillées ainsi qu'une compréhension de ce que l'on peut faire dans différents contextes, que ce soit en changeant le modèle de sécurité ou la stratégie de conception.

Par ailleurs, ces constructions comme leurs modèles de sécurité affaiblis poussent le cryptanalyste à changer de vision. Cela amènera sûrement des idées d'attaques qui pourraient être appliquées éventuellement à des constructions plus classiques, enrichissant ainsi la boîte à outils du cryptanalyste.

# Chapitre 4

# Conclusions et perspectives

La cryptanalyse est un domaine de recherche passionnant avec encore de nombreuses pistes à explorer. C'est aussi une discipline qui nécessite de garder en permanence un regard critique sur l'ensemble des constructions cryptographiques, ce qui à mon sens représente la quintessence de la science. En effet, la cryptanalyse incarne par définition ce scepticisme, composant intrinsèque et indispensable du domaine de la cryptographie. La cryptanalyse est en totale synergie avec la cryptographie, nous permettant de concevoir des chiffrements de plus en plus sécurisés.

Il faut continuer à améliorer sans cesse les attaques, analyser les chiffrements avec un mélange de visions classiques et nouvelles sur l'ensemble des constructions proposées. Pour gagner en confiance, il ne faut pas cesser d'attaquer les constructions connues et ne pas oublier qu'il n'y a à ce jour aucune preuve complète permettant de garantir que les hypothèses sur lesquelles elles reposent sont satisfaites.

Parmi les techniques étudiées dans ce document, les attaques intégrales soulèvent diverses questions qui restent ouvertes. Ainsi, utiliser de plus petits espaces affines permettrait, en prenant en compte les dépendances en la clef, d'améliorer la complexité du distingueur et donc de l'attaque. La difficulté réside cependant dans la taille de l'espace de recherche et dans la capacité en mémoire de nos ordinateurs.

Plus généralement, il nous manque encore des algorithmes de recherche de propriétés cryptographiques exactes (ou même approchées) permettant d'appréhender plusieurs tours d'une fonction itérée, même quand la représentation complète est manipulable. Ce sont des points sur lesquels des travaux doivent être menés.

Aujourd'hui, nous nous rendons compte que certains critères de résistance sont peut-être « trop forts ». Par exemple nous n'avons pas forcément besoin que les chiffrements par bloc soient des permutation pseudo-aléatoires dans tous les contextes. En effet, la plupart des modes opératoires simples ne sont sûrs

qu'en deçà de la borne des anniversaires et, pléthore d'attaques ont besoin d'une énorme quantité de données ce qui est bien plus coûteux que certaines attaques génériques. On pourrait donc aujourd'hui exiger « moins » à la primitive, ce qui permettrait de proposer des chiffrements moins coûteux.

En revanche, il faut faire extrêmement attention à ce que l'on fait et définir quelle marge de sécurité est acceptable n'est pas chose aisée. Pour cela, on en revient encore et toujours à la cryptanalyse, qu'il faut continuer de pousser notamment dans des modèles de sécurité plus faibles et en s'intéressant en particulier aux attaques utilisant une quantité de données raisonnable. Proposer des constructions sécurisées dans des modèles plus faibles est extrêmement intéressant intellectuellement, à la fois pour réfléchir à de nouvelles techniques de cryptanalyse adaptées à ces nouveaux contextes mais aussi pour la conception dans la mesure où des modèles de sécurité plus faibles pourraient permettre d'identifier les rôles joués par les différentes composantes d'une primitive dans sa sécurité.

La plupart des problèmes de sécurité ne venant aujourd'hui pas de la cryptographie mais d'implémentations défectueuses, d'utilisations inappropriées ou de problèmes de sécurité physiques, il serait dommage d'ajouter des failles à celles déjà existantes. Il est donc extrêmement important de garder une marge de sécurité importante, surtout si l'on implémente les algorithmes en matériel. Il faut que la recherche permanente de meilleures performances reste dans les limites du raisonnable pour ne pas porter atteinte à la sécurité de manière incontrôlée, et surtout utiliser avec précaution les primitives qui résulteraient de ces travaux de recherche.

Peut-être devons-nous attendre un peu, car changer les modèles de sécurité peut aussi changer complètement l'analyse, qui ne correspond plus à ce que l'on peut faire dans un contexte plus standard. Il faut redoubler de vigilance et surtout...

...continuer à faire de la cryptanalyse.

# Annexe A

# Récapitulatif des travaux futurs

Pour terminer ce document, je remets ici une liste succincte des directions et des différents projets que je compte mener dans les prochaines années.

**Création de contenu d'enseignement.** Principalement la création de vidéos courtes sur la cryptographie et la création d'une grande base de données d'exercices de cryptographie.

**Évolution des pratiques de l'enseignement.** En particulier dans le cours d'Applications Web et Sécurité, puisque les outils automatisés sont en train de changer les manières de programmer. Je souhaite aussi réfléchir à adapter les contenus pour motiver le plus possible les élèves dans les cours plus théoriques.

**Cryptanalyse des WPRFs.** C'est un travail que j'avais commencé avec Maé Miachon Lemeulle en stage et que je poursuis avec Balthazar Bauer et Mariana Moll de Alba.

**Algorithmes travaillant sur la représentation polynomiale.** Recherche de techniques permettant de donner des stratégies d'hypothèses de clef pour améliorer le recouvrement de clefs dans les attaques intégrales. De même je souhaiterais avoir des outils permettant de lier l'information entre les différents polynômes en la clef pour monter des attaques intégrales en utilisant plus d'informations dépendantes de la clef.

**Utilisation de la représentation univariée en cryptanalyse.** Réussir à exploiter la représentation univariée en cryptanalyse, notamment dans le cas des attaques intégrales. De même, je souhaite comprendre plus en détail les transformations dans cette représentation, sujet que Gaël Chopin commence à aborder dans sa thèse.

**Amélioration des cryptanalyses.** En se focalisant notamment sur la réduction de la quantité de données nécessaires.

# Annexe B

# Bilan des activités

Ces quelques pages détaillent l'ensemble des mes activités scientifiques, incluant mes activités de recherche, d'enseignement, de médiation scientifique et d'organisation de la vie universitaire.

## Carrière professionnelle

**2012 - 2015 : Formation d'ingénieur en informatique à Télécom ParisTech.** Parcours Cryptographie et Théorie de l'Information, Mathématiques, Complexité et Recherche Opérationnelle.

**2014 - 2015 : Master Parisien pour la Recherche en Informatique à Paris Diderot (Paris Cité).** Cryptographie, Algorithmes et Combinatoire. Mémoire de master : les représentations équivalentes d'un LFSR et leur impact en cryptanalyse. Sous la direction d'Anne Canteaut.

**2015 - 2018 : Doctorat en Informatique de Sorbonne Université.** Thèse préparée au centre Inria de Paris-Rocquencourt dans l'équipe-projet SECRET. *Mathématiques discrètes appliquées à la cryptographie symétrique. Directrice* : Anne Canteaut. Rapporteurs : Joan Daemen et Henri Gilbert. Jury : Pierre-Alain Fouque, Sihem Mesnager, María Naya-Plasencia, Sondre Rønjom, Antoine Joux. **Prix de la meilleure thèse de l'EDITE 2018.**

**2018 - 2019 : PostDoc à l'Université de Radboud.** Réalisée à Nimègue (Pays-Bas) dans le département Digital Security, sous la direction de Joan Daemen.

**depuis 2019 : Maître de Conférences à UVSQ.** Université de Versailles Saint-Quentin en Yvelines, Université Paris-Saclay, Laboratoire de Mathématiques de Versailles, CNRS (UMR 8100).

## B.1 Responsabilités scientifiques

### B.1.1 Projets de recherche

Je suis actif dans trois projets de recherche financés par l'Agence Nationale de la Recherche (ANR) :

**ANR SWAP.** Sboxes for Symmetric-Key Primitives. Projet ANR-21-CE39-0012. Début du projet : janvier 2022. Durée du projet : 48 mois. L'objectif de ce projet est de concevoir et d'analyser les boîtes-$S$, composants essentiels des primitives cryptographiques dans le but d'aider à la conception et à l'analyse des chiffrements adaptés à des applications spécifiques comme les schémas de masquage, le chiffrement complètement homomorphe ou le calcul multipartite. Le projet est coordonné par Christina Boura et a pour partenaires le LITIS (Rouen), l'IMATH (Toulon), CryptoExperts (Paris), Inria de Paris, l'ANSSI/SGDSN (Paris) et le LMV (Versailles). Je suis le responsable local du projet depuis septembre 2024.

**ANR OREO.** Modélisation MILP pour la Cryptographie Symétrique. Projet ANR-22-CE39-0015. Début du projet : janvier 2023. Durée du projet : 48 mois. Le but du projet consiste à concevoir de meilleurs modèles MILP pour la cryptanalyse dans le but d'améliorer les attaques existantes par des recherches automatisées. Le projet est coordonné par Patrick Derbez et a pour partenaires l'IRISA (Rennes), le LORIA (Nancy) et le LMV (Versailles). Je suis le responsable local du projet depuis septembre 2024.

**PEPR Cybersecurity - Projet Cryptanalyse.** Programme de Recherche France 2030. Début du projet : décembre 2023. Durée : 5 ans. Dans ce grand projet qui fédère l'ensemble de la communauté nationale travaillant en cryptanalyse symétrique et asymétrique nous étudions la résistance des systèmes cryptographiques. Je participe à ce projet sous la coordination de Gaëtan Leurent et Emmanuel Thomé.

### B.1.2 Comités de programme

J'ai participé ou participe aux comités de programme suivants :
— CRYPTO 2024 et 2025 ;
— IACR Transactions on Symmetric Cryptology, de 2020 à 2024 ;
— SAC 2022 et 2023 ;
— AFRICACRYPT 2022 et 2023 ;
— INDOCRYPT 2021 ;
— Journées Codage et Cryptographie 2023.

### B.1.3 Relectures

J'ai été rapporteur d'articles dans les conférences et journaux suivants.

— Theoretical Computer Science (2022) ;
— The Computer Journal (2021) ;
— Discrete Applied Mathematics (2020) ;
— Finite Fields and their Applications (2020 - 2024) ;
— IEEE Information Theory (2019 - 2024) ;
— Designs Codes and Cryptography (2017 - 2024) ;
— « subreviewer » dans les comités de programmes : CRYPTO 2018, ASIACRYPT 2018, ISIT 2019, EUROCRYPT 2019, NutMiC 2019, AFRICACRYPT 2019, EUROCRYPT 2021, CRYPTO 2021, ASIACRYPT 2021 and 2022, EUROCRYPT 2023, CRYPTO 2023.

### B.1.4 Invitations à des séminaires et groupes de travail

Participation aux groupes de travail et séminaires suivants sur invitation :
— Higher Order Derivatives, cubes, algebraic, integral, 2019. présentation invitée aux Journées Codage et Cryptographie, Hendaye, France.
— FrisiaCrypt 2019, Symmetric Cryptography : groupe de travail sur `GEA-1` et présentation de la cryptanalyse de `Pyjamask`.
— Dagstuhl 2020, séminaire 20041, Symmetric Cryptography : groupe de travail sur `GEA-1` et présentation sur les fonctions de compression.
— Dagstuhl 2022, séminaire 22141, Symmetric Cryptography : groupe de travail sur la cryptanalyse intégrale exploitant la représentation univariée.
— Frisiacrypt 2022, Symmetric Cryptography : présentation de l'attaque générique sur le mode duplex et problèmes ouverts sur les fonctions booléennes.
— Dagstuhl 2024, séminaire 24041, Symmetric Cryptography, groupe de travail sur TEA.
— Lorentz 2024, beating Real-Time Crypto : Solutions and Analysis, groupe de travail sur le chiffrement par bloc Saturnin.

### B.1.5 Séminaires invités

Au delà des présentations usuelles que je réalise régulièrement au séminaire du laboratoire, en groupe de travail ou en conférences, j'ai été invité à présenter une partie de mon travail dans les séminaires suivants.

**Algebraic Atacks Revisited, 2018.** Séminaire national Codes et Cryptographie et Algorithmes, Paris, France.

**New directions in attacks against stream ciphers, 2018.** Séminaire de l'École Polytechnique Fédérale de Lausanne, Suisse.

**On the concrete security of Goldreich's Pseudorandom Generator, 2019.** Séminaire CARAMBA, Inria de Nancy, France.

**Invariant attacks : how to choose the round constants, 2019.** Séminaire du LIX, Polytechnique, Saclay, France.

**Finding Collisions using Differentials, 2019.** Séminaire CASYS, Grenoble, France.

**Attacks against Filter Generator, 2019.** SIAM Conference on Applied Algebraic Geometry, Berne, Suisse.

**Algebraic Cryptanalysis of Keccak 2 round, 2019.** Séminaire CWI, online, Amsterdam, Pays-Bas.

**Cryptanalysis of GEA-1 and GEA-2 ciphers, backdoor and proprietary ciphers, 2019.** Séminaire Crypto ENS, Paris, France.

### B.1.6 Engagement universitaire

L'université française regorge de conseils et je m'engage dans la vie universitaire de l'Université de Versailles Saint-Quentin en Yvelines ainsi que dans l'Université Paris-Saclay dans différents conseils.

- Conseil de l'Unité de Formation et de Recherche des Sciences de l'UVSQ (depuis 2020) ;
- Conseil de la Graduate School Informatique et Science du Numérique de l'Université Paris-Saclay (depuis 2023) ;
- Commission Recherche de l'Université Paris-Saclay (depuis 2024) ;
- Comité Social d'Administration de l'UVSQ (depuis 2022).

## B.2 Liste des publications

**Attacks Against Filter Generators Exploiting Monomial Mappings.** *Anne Canteaut et Yann Rotella*, Fast Software Encryption 2016, LNCS 9783, pages 78 à 98, Springer, Heidelberg.
DOI 10.1007/978-3-662-52993-5_5.

**Cryptanalysis of the FLIP Family of Stream Ciphers.** *Sébastien Duval, Virginie Lallemand et Yann Rotella*, CRYPTO 2016, LNCS 9814, pages 457 à 475, Springer, Heidelberg.
DOI 10.1007/978-3-662-53018-4_17.

**Proving Resistance Against Invariant Attacks : How to Choose the Round Constants.** *Christof Beierle, Anne Canteaut, Gregor Leander et Yann Rotella*, CRYPTO 2017, LNCS 10402, pages 647 à 678, Springer, Heidelberg.
DOI 10.1007/978-3-319-63715-0_22.

**Boolean Functions with Restricted Input and their Robustness ; Application to the FLIP Cipher.** *Claude Carlet, Pierrick Méaux et Yann Rotella*, IACR Transactions on Symmetric Cryptology 2017 volume 3, pages 192 à 227,
DOI 10.13154/TOSC.V2017.I3.192-227.

**On the Concrete Security of Goldreich's Pseudorandom Generator.** *Geoffroy Couteau, Aurélien Dupin, Pierrick Méaux, Mélissa Rossi et Yann Rotella*, ASIACRYPT 2018, LNCS 11273, pages 96 à 124, Springer.
DOI 10.1007/978-3-030-03329-3_4.

**Cryptanalysis of MORUS.** *Tomer Ashur, Maria Eichlseder, Martin M. Lauridsen, Gaëtan Leurent, Brice Minaud, Yann Rotella, Yu Sasaki et Benoit Viguier*, ASIACRYPT 2018, LNCS 11273, pages 35 à 64, Springer.
DOI 10.1007/978-3-030-03329-3_2.

**State-Recovery Attacks on Modified Ketje Jr.** *Thomas Fuhr, María Naya-Plasencia et Yann Rotella*, IACR Transactions on Symmetric Cryptology 2018 volume 1, pages 29 à 56,
DOI 10.13154/TOSC.V2018.I1.29-56.

**Algebraic and Higher-Order Differential Cryptanalysis of Pyjamask-96.** *Christoph Dobraunig, Yann Rotella et Jan Schoone*, IACR Transactions on Symmetric Cryptology 2020 volume 1, pages 289-312,
DOI 10.13154/TOSC.V2020.I1.289-312.

**The Subterranean 2.0 Cipher Suite.** *Joan Daemen, Pedro Maat Costa Massolino, Alireza Mehrdad, et Yann Rotella*, IACR Transactions on Symmetric Cryptology 2020 Special Issue, pages 262-294,
DOI 10.13154/TOSC.V2020.IS1.262-294.

**Cryptanalysis of the GPRS Encryption Algorithms GEA-1 and GEA-2.** *Christof Beierle, Patrick Derbez, Gregor Leander, Gaëtan Leurent, Håvard Raddum, Yann Rotella, David Rupprecht et Lukas Stennes*, EUROCRYPT 2021, LNCS 12697, pages 155-183,
DOI 10.1007/978-3-030-77886-6_6.

**Algebraic Collision Attacks on Keccak.** *Rachelle Heim Boissier, Camille Noûs et Yann Rotella*, IACR Transactions on Symmetric Cryptology 2021 volume 1, pages 239 à 268,
DOI 10.46586/TOSC.V2021.I1.239-268.

**Breaking Panther.** *Christina Boura, Rachelle Heim Boissier et Yann Rotella*, AFRICACRYPT 2022, LNCS 13503, pages 176 à 188,
DOI 10.1007/978-3-031-17433-9_8.

**Differential Analysis of the Ternary Hash Function Troika.** *Christina Boura, Margot Funk et Yann Rotella*, Selected Areas in Cryptography 2022, LNCS 13742, pages 96 à 115,
DOI 10.1007/978-3-031-58411-4_5.

**Generic Attack on Duplex-Based AEAD Modes Using Random Function Statistics.** *Henri Gilbert, Rachelle Heim Boissier, Louiza Khati et Yann Rotella*, EUROCRYPT 2023, LNCS 14007, pages 348 à 378,
DOI 10.1007/978-3-031-30634-1_12.

**On the Security of Keyed Hashing Based on Public Permutations.** *Jonathan Fuchs, Yann Rotella et Joan Daemen*, CRYPTO 2023, LNCS 14083, pages 607-627,
DOI 10.1007/978-3-031-38548-3_20.

**Learning with Physical Rounding for Linear and Quadratic Leakage Functions.** *Clément Hoffmann, Pierrick Méaux, Charles Momin, Yann Rotella, François-Xavier Standaert et Balazs Udvarhelyi*, CRYPTO 2023, LNCS 14083, pages 410 à 439,
DOI 10.1007/978-3-031-38548-3_14.

## B.3 Enseignement

Jusqu'à maintenant, je n'ai parlé que d'activités de recherche scientifique, mais l'existence d'un enseignement de qualité est à mon sens tout aussi, voire plus important, que l'avancée dans la recherche scientifique. En effet, nous manquons crucialement de personnes bien formées, capables de sécuriser produits et services informatiques. Je dirais donc quelques mots sur l'enseignement.

### B.3.1 Cours universitaires

Je suis intéressé par à peu près tous les cours qui touchent de près ou de loin à la cryptographie et très favorable à l'existence de parcours proposant une culture scientifique en largeur de ce qui touche à la sécurité. Ceci nécessite dans notre domaine des bases assez solides d'algorithmique, de programmation et de mathématiques afin d'appréhender plus tard aussi bien la sécurité pratique que la cryptographie. Dans ce contexte, je donne des cours à l'UVSQ dans principalement deux masters et la licence d'informatique.

**Licence d'Informatique.** Dans la licence d'informatique de l'UVSQ, je donne ou ai donné des cours en mathématiques discrètes (L2), programmation (L1) et bases de la cryptographie (L3).

**Master SECRETS.** Dans ce master de cybersécurité, j'interviens dans des cours de Mathématiques et de Cryptographie, mais aussi dans un cours d'Applications Web et de Sécurité où je propose une pédagogie par projets (dont l'organisation se rapproche un peu de l'apprentissage par problèmes et par projets).

**Master Algèbre Appliquée.** Dans ce master recherche, je donne des cours de Programmation en C, de Théorie de l'Information et d'Analyse d'Algorithmes. De plus, j'organise le séminaire de cryptographie du Laboratoire de Mathématiques de Versailles en dehors des heures de cours afin que les étudiant.e.s puissent s'intéresser à des sujets actuels de recherche en cryptographie. La quantité de savoirs augmentant mais le nombre d'heures d'enseignement diminuant, je pense que la participation des étudiant.e.s est nécessaire pour garder au maximum ce lien entre la base à enseigner et l'état de l'art actuel.

**Défis de l'enseignement.** Tout d'abord l'enseignement oblige l'enseignant à sans cesse réfléchir à la manière la plus directe et intelligible de transmettre l'information, mais aussi à ce que les points les plus importants soient intégrés sur le long terme. Ce n'est pas chose aisée, cela force en permanence à expliquer les éléments avec des termes et des tournures de phrase différentes, qui permettent d'avoir des visions différentes sur des sujets, même basiques, que l'on peut enseigner. C'est donc un premier défi majeur : l'adaptabilité à son audience qui se situe face à nous et à ses connaissances, qui ne vont pas être les mêmes pour des formations en mathématiques ou en informatique par exemple.

Ensuite, il y a la motivation des étudiant.e.s, surtout dans les premières années de licence. J'en rencontre un grand nombre dont la motivation n'est pas la curiosité et l'apprentissage de principes et méthodes qu'iels ne connaissent pas. C'est une énorme difficulté que de motiver des élèves et de faire vibrer cette curiosité, qui, quand on l'a, fait une réelle différence dans la rapidité et l'efficacité d'apprentissage. Une grande question qui revient souvent dans des cours théoriques de licence est : « à quoi ça sert ? ». Il est souvent difficile de répondre à cette question sans prendre un temps trop long pour le cours théorique en question. Or, l'ordonnancement de nos cours est important : il est nécessaire de construire des briques de base solides en mathématiques et en informatique pour faire de la cryptographie avancée, qui nécessite, elle, un effort intellectuel et un certain niveau d'abstraction. Par conséquent, motiver les élèves à être curieux sur des sujets dont iels ne voient pas l'utilité immédiate est complexe.

Enfin, et de manière peut-être plus récente, se pose la question de la modification de l'enseignement pour prendre en compte l'émergence d'outils automatisés. Aujourd'hui, nous disposons de pléthore d'outils numériques permettant par exemple de produire des morceaux de code réalisant exactement ce que l'on demande aux étudiant.e.s en licence ou bien de répondre à des questions de cours, de manière pas toujours exacte. On a aujourd'hui une grande difficulté à expliquer qu'il est peu utile pour l'apprentissage d'utiliser souvent ces outils

numériques. En effet, je suis convaincu que l'utilisation trop poussée d'outils permettant de répondre automatiquement à ces questions pendant une phase d'apprentissage est dangereux, car cela n'incite pas à réfléchir par soi-même. Peut-être que pour résoudre des exercices simples les outils sont suffisants, mais dès que l'on passe à une réflexivité un peu plus complexe, les outils ne fonctionnent plus et les étudiant.e.s n'ont pas intégré les bases (pour lesquelles les outils fonctionnent). Par ailleurs, la plupart de ces outils lissent l'information et nous n'avons plus nécessairement ces détails qui, sans aucun doute, permettent d'apprendre bien mieux.

Je ne m'étendrais pas plus sur l'enseignement, mais je tenais à exprimer la nécessité de réfléchir en permanence à la manière dont on enseigne, que ce soit pour s'adapter à la diversité de l'audience ou bien aux évolutions sociétales et techniques.

### B.3.2 Thèses encadrées

J'ai co-encadré deux thèses et je co-encadre une thèse actuellement.

**Margot Funk (2021 - 2024).** Algorithmes et outils pour la cryptanalyse des primitives symétriques [Fun24], taux d'encadrement : 20%, directeur de thèse : Louis Goubin (30%), co-encadrement Christina Boura (50%). Thèse soutenue le 14 octobre 2024.

**Rachelle Heim Boissier (2021 - 2024).** Symmetric Cryptanalysis : from Primitives to Modes [Hei24], taux d'encadrement : 50%, directeur de thèse : Henri Gilbert (30%), co-encadrement Christina Boura (20%). Thèse soutenue le 15 octobre 2024.

**Gaël Chopin.** Fonctions binomiales et leur application à la cryptographie. directrice de thèse : Anne Canteaut. (2024 - aujourd'hui)

### B.3.3 Stages de master

Jusqu'à maintenant j'ai co-encadré ou encadré 5 stages de master 2 (Algèbre Appliquée de Versailles).

**Rachelle Heim Boissier (2020).** Algebraic Collisions on Keccak.

**Margot Funk (2021).** Cryptanalyse de la fonction de hachage Troïka.

**Yann Le Dore (2023).** Amélioration du recouvrement de clef pour certaines boîtes-$S$.

**Gaël Chopin (2024).** Caractérisation des binômes bijectifs.

**Maé Miachon Lemeulle (2024).** Cryptanalyse des WPRFs.

### B.3.4 Médiation scientifique

Tout comme je suis passionné par l'enseignement, je suis très actif dans les activités de médiation scientifique, principalement pour aller présenter la cryptographie dans des lycées, participer à des écoles d'été de mathématiques et réaliser des ateliers à destination du grand public (remise des prix des Olympiades, cours au lycée Condorcet, présentation à l'ESPCI, école d'été à Saint-Germain en Laye...).

Le projet dans lequel je me suis le plus investi est le concours Alkindi [1] que je co-organise depuis 2019 et pour lequel je conçois chaque année des exercices de cryptanalyse pour les tours 1, 2 et 3 du concours qui se déroule de décembre à mai. Je co-organise aussi la finale et propose des exercices. Ce travail nécessite *a minima* une heure par semaine toute l'année. Ce concours s'adresse à des élèves de la 4ème à la 2nde. Chaque édition reçoit approximativement 65 000 participants. L'objectif du concours est de motiver le plus possible les élèves à s'intéresser à des raisonnements élaborés, à faire appel à un raisonnement inductif en y mettant des pointes d'algorithmiques et de mathématiques.

### B.3.5 Projets d'enseignement

Il y a deux grands projets qui me tiennent à coeur aujourd'hui. Le premier vient de commencer : il s'agit de participer avec l'équipe Crypto du Laboratoire de Mathématiques de Versailles à la création de contenu vidéo sur la cryptographie. L'idée est d'avoir un autre support et de faire des vidéos courtes mais expliquant les bases (et peut-être les protocoles avancés) de cryptographie.

Le deuxième projet (un peu plus lointain celui-là) est la création d'une base de données (ou d'un livre) d'exercices de cryptographie, avec des challenges, permettant d'aborder des techniques plus avancées et plus récentes de cryptanalyse.

1. https://concours-alkindi.fr/

# Bibliographie


[AAB+20] Abdelrahaman Aly, Tomer Ashur, Eli Ben-Sasson, Siemen Dhooghe, and Alan Szepieniec. Design of symmetric-key primitives for advanced cryptographic protocols. *IACR Transactions on Symmetric Cryptology*, 2020(3) :1–45, 2020.

[ABM24] Tomer Ashur, Thomas Buschman, and Mohammad Mahzoun. Algebraic cryptanalysis of the HADES design strategy : Application to Poseidon and Poseidon2. In Yannan Li Tianqing Zhu, editor, *ACISP 24 : 29th Australasian Conference on Information Security and Privacy, Part II*, volume 14896 of *Lecture Notes in Computer Science*, pages 225–244, Sydney, NSW, Australia, July 15–17, 2024. Springer, Singapore, Singapore.

[ACC+24] Gildas Avoine, Xavier Carpent, Tristan Claverie, Christophe Devine, and Diane Leblanc-Albarel. Time-memory trade-offs sound the death knell for GPRS and GSM. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology – CRYPTO 2024, Part IV*, volume 14923 of *Lecture Notes in Computer Science*, pages 206–240, Santa Barbara, CA, USA, August 18–22, 2024. Springer, Cham, Switzerland.

[ACG+19] Martin R. Albrecht, Carlos Cid, Lorenzo Grassi, Dmitry Khovratovich, Reinhard Lüftenegger, Christian Rechberger, and Markus Schofnegger. Algebraic cryptanalysis of STARK-friendly designs : Application to MARVELlous and MiMC. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology – ASIACRYPT 2019, Part III*, volume 11923 of *Lecture Notes in Computer Science*, pages 371–397, Kobe, Japan, December 8–12, 2019. Springer, Cham, Switzerland.

[AD18] Tomer Ashur and Siemen Dhooghe. MARVELlous : a STARK-friendly family of cryptographic primitives. Cryptology ePrint Archive, Report 2018/1098, 2018.

[AD22] Dor Amzaleg and Itai Dinur. Refined cryptanalysis of the GPRS ciphers GEA-1 and GEA-2. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology – EUROCRYPT 2022, Part III*, volume 13277 of *Lecture Notes in Computer Science*,

pages 57–85, Trondheim, Norway, May 30 – June 3, 2022. Springer, Cham, Switzerland.

[ADDG24] Martin R. Albrecht, Alex Davidson, Amit Deo, and Daniel Gardham. Crypto dark matter on the torus - oblivious PRFs from shallow PRFs and TFHE. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology – EUROCRYPT 2024, Part VI*, volume 14656 of *Lecture Notes in Computer Science*, pages 447–476, Zurich, Switzerland, May 26–30, 2024. Springer, Cham, Switzerland.

[ADMS09] Jean-Philippe Aumasson, Itai Dinur, Willi Meier, and Adi Shamir. Cube testers and key recovery attacks on reduced-round MD6 and Trivium. In Orr Dunkelman, editor, *Fast Software Encryption – FSE 2009*, volume 5665 of *Lecture Notes in Computer Science*, pages 1–22, Leuven, Belgium, February 22–25, 2009. Springer Berlin Heidelberg, Germany.

[AEL+18] Tomer Ashur, Maria Eichlseder, Martin M. Lauridsen, Gaëtan Leurent, Brice Minaud, Yann Rotella, Yu Sasaki, and Benoît Viguier. Cryptanalysis of MORUS. In Thomas Peyrin and Steven Galbraith, editors, *Advances in Cryptology – ASIACRYPT 2018, Part II*, volume 11273 of *Lecture Notes in Computer Science*, pages 35–64, Brisbane, Queensland, Australia, December 2–6, 2018. Springer, Cham, Switzerland.

[AGP+19] Martin R. Albrecht, Lorenzo Grassi, Léo Perrin, Sebastian Ramacher, Christian Rechberger, Dragos Rotaru, Arnab Roy, and Markus Schofnegger. Feistel structures for MPC, and more. In Kazue Sako, Steve Schneider, and Peter Y. A. Ryan, editors, *ESORICS 2019 : 24th European Symposium on Research in Computer Security, Part II*, volume 11736 of *Lecture Notes in Computer Science*, pages 151–171, Luxembourg, September 23–27, 2019. Springer, Cham, Switzerland.

[AGR+16] Martin R. Albrecht, Lorenzo Grassi, Christian Rechberger, Arnab Roy, and Tyge Tiessen. MiMC : Efficient encryption and cryptographic hashing with minimal multiplicative complexity. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology – ASIACRYPT 2016, Part I*, volume 10031 of *Lecture Notes in Computer Science*, pages 191–219, Hanoi, Vietnam, December 4–8, 2016. Springer Berlin Heidelberg, Germany.

[AJN16] Jean-Philippe Aumasson, Philipp Jovanovic, and Samuel Neves. NORX v3. Submission to the Caesar competition, 2016.

[AK09] Jean-Philippe Aumasson and Dmitry Khovratovich. First analysis of Keccak, 2009. NIST hash forum.

[AM09] Jean-Philippe Aumasson and Willi Meier. Zero-sum distinguishers for reduced Keccak-$f$ and for the core functions of Luffa and Hamsi, 2009. Rump Session of CHES.

[And94] Ross J. Anderson. A5 (was hacking digital phones), 1994. http://yarchive.net/phone/gsmcipher.html (accédé le 10 octobre 2024).

[ARS+15] Martin R. Albrecht, Christian Rechberger, Thomas Schneider, Tyge Tiessen, and Michael Zohner. Ciphers for MPC and FHE. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 430–454, Sofia, Bulgaria, April 26–30, 2015. Springer Berlin Heidelberg, Germany.

[Bar23] Augustin Bariant. A univariate attack against the limited-data instance of Ciminion. Cryptology ePrint Archive, Report 2023/1283, 2023.

[BBC+20] Christof Beierle, Alex Biryukov, Luan Cardoso dos Santos, Johann Großschädl, Léo Perrin, Aleksei Udovenko, Vesselin Velichkov, and Qingju Wang. Lightweight AEAD and hashing using the Sparkle permutation family. *IACR Transactions on Symmetric Cryptology*, 2020(S1) :208–261, 2020.

[BBC+22] Clémence Bouvier, Pierre Briaud, Pyrros Chaidos, Léo Perrin, and Vesselin Velichkov. Anemoi : Exploiting the link between arithmetization-orientation and CCZ-equivalence. Cryptology ePrint Archive, Report 2022/840, 2022.

[BBC+23] Clémence Bouvier, Pierre Briaud, Pyrros Chaidos, Léo Perrin, Robin Salen, Vesselin Velichkov, and Danny Willems. New design techniques for efficient arithmetization-oriented hash functions : Anemoi permutations and Jive compression mode. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology – CRYPTO 2023, Part III*, volume 14083 of *Lecture Notes in Computer Science*, pages 507–539, Santa Barbara, CA, USA, August 20–24, 2023. Springer, Cham, Switzerland.

[BBDV20] Subhadeep Banik, Khashayar Barooti, F. Betül Durak, and Serge Vaudenay. Cryptanalysis of LowMC instances using single plaintext/ciphertext pair. *IACR Transactions on Symmetric Cryptology*, 2020(4) :130–146, 2020.

[BBL+24] Augustin Bariant, Aurélien Boeuf, Axel Lemoine, Irati Manterola Ayala, Morten Øygarden, Léo Perrin, and Håvard Raddum. The algebraic FreeLunch : Efficient Gröbner basis attacks against arithmetization-oriented primitives. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology – CRYPTO 2024, Part IV*, volume 14923 of *Lecture Notes in Computer Science*, pages 139–173, Santa Barbara, CA, USA, August 18–22, 2024. Springer, Cham, Switzerland.

[BBR22] Christina Boura, Rachelle Heim Boissier, and Yann Rotella. Breaking panther. In Lejla Batina and Joan Daemen, editors, *AFRICACRYPT 22 : 13th International Conference on Cryptology in*

*Africa*, volume 2022 of *Lecture Notes in Computer Science*, pages 176–188, Fes, Morocco, July 18–20, 2022. Springer, Cham, Switzerland.

[BBVY21] Subhadeep Banik, Khashayar Barooti, Serge Vaudenay, and Hailun Yan. New attacks on LowMC instances with a single plaintext/ciphertext pair. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2021, Part I*, volume 13090 of *Lecture Notes in Computer Science*, pages 303–331, Singapore, December 6–10, 2021. Springer, Cham, Switzerland.

[BC10] Christina Boura and Anne Canteaut. A zero-sum property for the Keccak-$f$ permutation with 18 rounds. In *IEEE International Symposium on Information Theory, ISIT 2010, June 13-18, 2010, Austin, Texas, USA, Proceedings*, pages 2488–2492. IEEE, 2010.

[BC11] Christina Boura and Anne Canteaut. Zero-sum distinguishers for iterated permutations and application to Keccak-f and Hamsi-256. In Alex Biryukov, Guang Gong, and Douglas R. Stinson, editors, *SAC 2010 : 17th Annual International Workshop on Selected Areas in Cryptography*, volume 6544 of *Lecture Notes in Computer Science*, pages 1–17, Waterloo, Ontario, Canada, August 12–13, 2011. Springer Berlin Heidelberg, Germany.

[BC16] Christina Boura and Anne Canteaut. Another view of the division property. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 654–682, Santa Barbara, CA, USA, August 14–18, 2016. Springer Berlin Heidelberg, Germany.

[BCD11] Christina Boura, Anne Canteaut, and Christophe De Cannière. Higher-order differential properties of Keccak and Luffa. In Antoine Joux, editor, *Fast Software Encryption – FSE 2011*, volume 6733 of *Lecture Notes in Computer Science*, pages 252–269, Lyngby, Denmark, February 13–16, 2011. Springer Berlin Heidelberg, Germany.

[BCD+20] Tim Beyne, Anne Canteaut, Itai Dinur, Maria Eichlseder, Gregor Leander, Gaëtan Leurent, María Naya-Plasencia, Léo Perrin, Yu Sasaki, Yosuke Todo, and Friedrich Wiemer. Out of oddity - new cryptanalytic techniques against symmetric primitives optimized for integrity proof systems. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020, Part III*, volume 12172 of *Lecture Notes in Computer Science*, pages 299–328, Santa Barbara, CA, USA, August 17–21, 2020. Springer, Cham, Switzerland.

[BCFG+21] Marek Broll, Federico Canale, Antonio Flórez-Gutiérrez, Gregor Leander, and María Naya-Plasencia. Generic framework for key-guessing improvements. In Mehdi Tibouchi and Huaxiong Wang,

editors, *Advances in Cryptology – ASIACRYPT 2021, Part I*, volume 13090 of *Lecture Notes in Computer Science*, pages 453–483, Singapore, December 6–10, 2021. Springer, Cham, Switzerland.

[BCG+20] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl. Correlated pseudorandom functions from variable-density LPN. In *61st Annual Symposium on Foundations of Computer Science*, pages 1069–1080, Durham, NC, USA, November 16–19, 2020. IEEE Computer Society Press.

[BCK96] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Keying hash functions for message authentication. In Neal Koblitz, editor, *Advances in Cryptology – CRYPTO'96*, volume 1109 of *Lecture Notes in Computer Science*, pages 1–15, Santa Barbara, CA, USA, August 18–22, 1996. Springer Berlin Heidelberg, Germany.

[BCLR17] Christof Beierle, Anne Canteaut, Gregor Leander, and Yann Rotella. Proving resistance against invariant attacks : How to choose the round constants. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017, Part II*, volume 10402 of *Lecture Notes in Computer Science*, pages 647–678, Santa Barbara, CA, USA, August 20–24, 2017. Springer, Cham, Switzerland.

[BCP23] Clémence Bouvier, Anne Canteaut, and Léo Perrin. On the algebraic degree of iterated power functions. *Designs, Codes and Cryptography*, 91(3) :997–1033, 2023.

[BDH+17] Guido Bertoni, Joan Daemen, Seth Hoffert, Michaël Peeters, Gilles Van Assche, and Ronny Van Keer. Farfalle : parallel permutation-based cryptography. *IACR Transactions on Symmetric Cryptology*, 2017(4) :1–38, 2017.

[BDKV21] Nicolas Bordes, Joan Daemen, Daniël Kuijsters, and Gilles Van Assche. Thinking outside the superbox. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021, Part III*, volume 12827 of *Lecture Notes in Computer Science*, pages 337–367, Virtual Event, August 16–20, 2021. Springer, Cham, Switzerland.

[BDL+21] Christof Beierle, Patrick Derbez, Gregor Leander, Gaëtan Leurent, Håvard Raddum, Yann Rotella, David Rupprecht, and Lukas Stennes. Cryptanalysis of the GPRS encryption algorithms GEA-1 and GEA-2. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021, Part II*, volume 12697 of *Lecture Notes in Computer Science*, pages 155–183, Zagreb, Croatia, October 17–21, 2021. Springer, Cham, Switzerland.

[BDP+16] Guido Bertoni, Joan Daemen, Michaël Peeters, Gilles Van Assche, and Ronnu Van Keer. CAEASAR submission : Ketje v2, September 2016. https://keccak.team/ketje.html.

[BDPV] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Keccak Crunchy Crypto Collision and Pre-image Contest. http://keccak.noeken.org/crunchy_contest.html.

[BDPV07] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Sponge functions, 2007. Ecrypt Hash Workshop, Graz, Austria, https://keccak.team/files/SpongeFunctions.pdf.

[BDPV11] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Cryptographic sponge functions, 2011. https://keccak.team/files/CSF-0.1.pdf.

[BDPV12a] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Duplexing the sponge : Single-pass authenticated encryption and other applications. In Ali Miri and Serge Vaudenay, editors, *SAC 2011 : 18th Annual International Workshop on Selected Areas in Cryptography*, volume 7118 of *Lecture Notes in Computer Science*, pages 320–337, Toronto, Ontario, Canada, August 11–12, 2012. Springer Berlin Heidelberg, Germany.

[BDPV12b] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Permutation-based encryption, authentication and authenticated encryption, July 2012. Directions in Authenticated Ciphers, Stockholm, Sweden, http://www.hyperelliptic.org/djb/diac/record.pdf.

[BDPV14] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. The making of KECCAK. *Cryptologia*, 38(1) :26–60, 2014.

[Ber99] Daniel J. Bernstein. How to stretch random functions : The security of protected counter sums. *Journal of Cryptology*, 12(3) :185–192, June 1999.

[BFL21] Christof Beierle, Patrick Felke, and Gregor Leander. To shift or not to shift : Understanding GEA-1. Cryptology ePrint Archive, Report 2021/829, 2021.

[BFP09] Luk Bettale, Jean-Charles Faugère, and Ludovic Perret. Hybrid approach for solving multivariate systems over finite fields. *J. Math. Cryptol.*, 3(3) :177–197, 2009.

[BFR24] Christina Boura, Margot Funk, and Yann Rotella. Differential analysis of the ternary hash function Troika. In Benjamin Smith and Huapeng Wu, editors, *SAC 2022 : 29th Annual International Workshop on Selected Areas in Cryptography*, volume 13742 of *Lecture Notes in Computer Science*, pages 96–115, Windsor, Canada, August 24-26, 2024. Springer, Cham, Switzerland.

[BHLS24] Xavier Bonnetain, Rachelle Heim Boissier, Gaëtan Leurent, and André Schrottenloher. Improving generic attacks using exceptional functions. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology – CRYPTO 2024, Part IV*, volume 14923

of *Lecture Notes in Computer Science*, pages 105–138, Santa Barbara, CA, USA, August 18–22, 2024. Springer, Cham, Switzerland.

[BIP+18] Dan Boneh, Yuval Ishai, Alain Passelègue, Amit Sahai, and David J. Wu. Exploring Crypto Dark Matter : New simple PRF candidates and their applications. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018 : 16th Theory of Cryptography Conference, Part II*, volume 11240 of *Lecture Notes in Computer Science*, pages 699–729, Panaji, India, November 11–14, 2018. Springer, Cham, Switzerland.

[BKR94] Mihir Bellare, Joe Kilian, and Phillip Rogaway. The security of cipher block chaining. In Yvo Desmedt, editor, *Advances in Cryptology – CRYPTO'94*, volume 839 of *Lecture Notes in Computer Science*, pages 341–358, Santa Barbara, CA, USA, August 21–25, 1994. Springer Berlin Heidelberg, Germany.

[BR02] John Black and Phillip Rogaway. A block-cipher mode of operation for parallelizable message authentication. In Lars R. Knudsen, editor, *Advances in Cryptology – EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 384–397, Amsterdam, The Netherlands, April 28 – May 2, 2002. Springer Berlin Heidelberg, Germany.

[BR22] Tim Beyne and Vincent Rijmen. Differential cryptanalysis in the fixed-key model. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022, Part III*, volume 13509 of *Lecture Notes in Computer Science*, pages 687–716, Santa Barbara, CA, USA, August 15–18, 2022. Springer, Cham, Switzerland.

[Bre80] Richard P. Brent. An improved Monte Carlo factorization algorithm. In *BIT Numerical Mathematics*, page 176–184, Berlin, Heidelberg, 1980.

[Bro01] Charles Brookson. GPRS security, 2001. https://web.archive.org/web/20120914110208/http://www.brookson.com/gsm/gprs.pdf (snapshot of September 14, 2012).

[Bro23] Marek Broll. *Key guessing strategies for Sbox-based ciphers*. PhD thesis, Ruhr University Bochum, Germany, 2023.

[BSL21] K. V. L. Bhargavi, Chungath Srinivasan, and K. V. Lakshmy. Panther : A sponge based lightweight authenticated encryption scheme. In Avishek Adhikari, Ralf Küsters, and Bart Preneel, editors, *Progress in Cryptology - INDOCRYPT 2021 : 22nd International Conference in Cryptology in India*, volume 13143 of *Lecture Notes in Computer Science*, pages 49–70, Jaipur, India, December 12–15, 2021. Springer, Cham, Switzerland.

[Car20] Claude Carlet, editor. *Boolean Functions for Cryptography and Coding Theory*. Cambridge University Press, 2020.

[CCF+16] Anne Canteaut, Sergiu Carpov, Caroline Fontaine, Tancrède Lepoint, María Naya-Plasencia, Pascal Paillier, and Renaud Sirdey. Stream ciphers : A practical solution for efficient homomorphic-ciphertext compression. In Thomas Peyrin, editor, *Fast Software Encryption – FSE 2016*, volume 9783 of *Lecture Notes in Computer Science*, pages 313–333, Bochum, Germany, March 20–23, 2016. Springer Berlin Heidelberg, Germany.

[CCF+18] Anne Canteaut, Sergiu Carpov, Caroline Fontaine, Tancrède Lepoint, María Naya-Plasencia, Pascal Paillier, and Renaud Sirdey. Stream ciphers : A practical solution for efficient homomorphic-ciphertext compression. *Journal of Cryptology*, 31(3) :885–916, July 2018.

[CCKK21] Jung Hee Cheon, Wonhee Cho, Jeong Han Kim, and Jiseung Kim. Adventures in crypto dark matter : Attacks and fixes for weak pseudorandom functions. In Juan Garay, editor, *PKC 2021 : 24th International Conference on Theory and Practice of Public Key Cryptography, Part II*, volume 12711 of *Lecture Notes in Computer Science*, pages 739–760, Virtual Event, May 10–13, 2021. Springer, Cham, Switzerland.

[CDGP93] Luc J. M. Claesen, Joan Daemen, Mark Genoe, and G. Peeters. Subterranean : A 600 mbit/sec cryptographic VLSI chip. In *Proceedings 1993 International Conference on Computer Design : VLSI in Computers & Processors, ICCD '93, Cambridge, MA, USA, October 3-6, 1993*, pages 610–613. IEEE Computer Society, 1993.

[CDM+18] Geoffroy Couteau, Aurélien Dupin, Pierrick Méaux, Mélissa Rossi, and Yann Rotella. On the concrete security of Goldreich's pseudorandom generator. In Thomas Peyrin and Steven Galbraith, editors, *Advances in Cryptology – ASIACRYPT 2018, Part II*, volume 11273 of *Lecture Notes in Computer Science*, pages 96–124, Brisbane, Queensland, Australia, December 2–6, 2018. Springer, Cham, Switzerland.

[CDNY18] Avik Chakraborti, Nilanjan Datta, Mridul Nandi, and Kan Yasuda. Beetle family of lightweight and secure authenticated encryption ciphers. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2018(2) :218–241, 2018.

[CHWW22a] Jiamin Cui, Kai Hu, Meiqin Wang, and Puwen Wei. On the field-based division property : Applications to MiMC, Feistel MiMC and GMiMC. In Shweta Agrawal and Dongdai Lin, editors, *Advances in Cryptology – ASIACRYPT 2022, Part III*, volume 13793 of *Lecture Notes in Computer Science*, pages 241–270, Taipei, Taiwan, December 5–9, 2022. Springer, Cham, Switzerland.

[CHWW22b] Jiamin Cui, Kai Hu, Qingju Wang, and Meiqin Wang. Integral attacks on Pyjamask-96 and round-reduced Pyjamask-128. In Steven D. Galbraith, editor, *Topics in Cryptology – CT-RSA 2022*, volume 13161 of *Lecture Notes in Computer Science*, pages 223–246, Virtual Event, March 1–2, 2022. Springer, Cham, Switzerland.

[CIKK16] Marco L. Carmosino, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Learning algorithms from natural proofs. In Ran Raz, editor, *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, volume 50 of *LIPIcs*, pages 10 :1–10 :24. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

[CJRR99] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In Michael J. Wiener, editor, *Advances in Cryptology – CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 398–412, Santa Barbara, CA, USA, August 15–19, 1999. Springer Berlin Heidelberg, Germany.

[CLO15] David A. Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms*. Undergraduate Texts in Mathematics. Springer Cham, 2015.

[CM03] Nicolas Courtois and Willi Meier. Algebraic attacks on stream ciphers with linear feedback. In Eli Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 345–359, Warsaw, Poland, May 4–8, 2003. Springer Berlin Heidelberg, Germany.

[Cou03] Nicolas Courtois. Fast algebraic attacks on stream ciphers with linear feedback. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 176–194, Santa Barbara, CA, USA, August 17–21, 2003. Springer Berlin Heidelberg, Germany.

[CR16] Anne Canteaut and Yann Rotella. Attacks against filter generators exploiting monomial mappings. In Thomas Peyrin, editor, *Fast Software Encryption – FSE 2016*, volume 9783 of *Lecture Notes in Computer Science*, pages 78–98, Bochum, Germany, March 20–23, 2016. Springer Berlin Heidelberg, Germany.

[Dae95] Joan Daemen. Cipher and hash function design strategies based on linear and differential cryptanalysis, K.U. Leuven, 1995.

[DDLS22] Marcus Dansarie, Patrick Derbez, Gregor Leander, and Lukas Stennes. Breaking HALFLOOP-24. *IACR Transactions on Symmetric Cryptology*, 2022(3) :217–238, 2022.

[DDS12] Itai Dinur, Orr Dunkelman, and Adi Shamir. New attacks on Keccak-224 and Keccak-256. In Anne Canteaut, editor, *Fast*

*Software Encryption – FSE 2012*, volume 7549 of *Lecture Notes in Computer Science*, pages 442–461, Washington, DC, USA, March 19–21, 2012. Springer Berlin Heidelberg, Germany.

[DDS14] Itai Dinur, Orr Dunkelman, and Adi Shamir. Collision attacks on up to 5 rounds of SHA-3 using generalized internal differentials. In Shiho Moriai, editor, *Fast Software Encryption – FSE 2013*, volume 8424 of *Lecture Notes in Computer Science*, pages 219–240, Singapore, March 11–13, 2014. Springer Berlin Heidelberg, Germany.

[De 06] Christophe De Cannière. Trivium : A stream cipher construction inspired by block cipher design principles. In Sokratis K. Katsikas, Javier Lopez, Michael Backes, Stefanos Gritzalis, and Bart Preneel, editors, *ISC 2006 : 9th International Conference on Information Security*, volume 4176 of *Lecture Notes in Computer Science*, pages 171–186, Samos Island, Greece, August 30 – September 2, 2006. Springer Berlin Heidelberg, Germany.

[DEG+18] Christoph Dobraunig, Maria Eichlseder, Lorenzo Grassi, Virginie Lallemand, Gregor Leander, Eik List, Florian Mendel, and Christian Rechberger. Rasta : A cipher with low ANDdepth and few ANDs per bit. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018, Part I*, volume 10991 of *Lecture Notes in Computer Science*, pages 662–692, Santa Barbara, CA, USA, August 19–23, 2018. Springer, Cham, Switzerland.

[DeL88] John M. DeLaurentis. Components and cycles of a random function. In Carl Pomerance, editor, *Advances in Cryptology – CRYPTO'87*, volume 293 of *Lecture Notes in Computer Science*, pages 231–242, Santa Barbara, CA, USA, August 16–20, 1988. Springer Berlin Heidelberg, Germany.

[DEM16] Christoph Dobraunig, Maria Eichlseder, and Florian Mendel. Higher-order cryptanalysis of LowMC. In Soonhak Kwon and Aaram Yun, editors, *ICISC 15 : 18th International Conference on Information Security and Cryptology*, volume 9558 of *Lecture Notes in Computer Science*, pages 87–101, Seoul, Korea, November 25–27, 2016. Springer, Cham, Switzerland.

[DEMS21] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Ascon v1.2 : Lightweight authenticated encryption and hashing. *Journal of Cryptology*, 34(3) :33, July 2021.

[DF20] Patrick Derbez and Pierre-Alain Fouque. Increasing precision of division property. *IACR Transactions on Symmetric Cryptology*, 2020(4) :173–194, 2020.

[DGGK21] Christoph Dobraunig, Lorenzo Grassi, Anna Guinet, and Daniël Kuijsters. Ciminion : Symmetric encryption based on Toffoli-gates over large finite fields. In Anne Canteaut and François-Xavier

Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021, Part II*, volume 12697 of *Lecture Notes in Computer Science*, pages 3–34, Zagreb, Croatia, October 17–21, 2021. Springer, Cham, Switzerland.

[DHVV18] Joan Daemen, Seth Hoffert, Gilles Van Assche, and Ronny Van Keer. The design of Xoodoo and Xoofff. *IACR Transactions on Symmetric Cryptology*, 2018(4) :1–38, 2018.

[DKR97] Joan Daemen, Lars R. Knudsen, and Vincent Rijmen. The block cipher Square. In Eli Biham, editor, *Fast Software Encryption – FSE'97*, volume 1267 of *Lecture Notes in Computer Science*, pages 149–165, Haifa, Israel, January 20–22, 1997. Springer Berlin Heidelberg, Germany.

[DL22] Patrick Derbez and Baptiste Lambin. Fast MILP models for division property. *IACR Transactions on Symmetric Cryptology*, 2022(2) :289–321, 2022.

[DLMW15] Itai Dinur, Yunwen Liu, Willi Meier, and Qingju Wang. Optimized interpolation attacks on LowMC. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology – ASIACRYPT 2015, Part II*, volume 9453 of *Lecture Notes in Computer Science*, pages 535–560, Auckland, New Zealand, November 30 – December 3, 2015. Springer Berlin Heidelberg, Germany.

[DLR16] Sébastien Duval, Virginie Lallemand, and Yann Rotella. Cryptanalysis of the FLIP family of stream ciphers. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 457–475, Santa Barbara, CA, USA, August 14–18, 2016. Springer Berlin Heidelberg, Germany.

[DLWQ17] Xiaoyang Dong, Zheng Li, Xiaoyun Wang, and Ling Qin. Cube-like attack on round-reduced initialization of Ketje Sr. *IACR Transactions on Symmetric Cryptology*, 2017(1) :259–280, 2017.

[DM19] Christoph Dobraunig and Bart Mennink. Security of the suffix keyed sponge. *IACR Transactions on Symmetric Cryptology*, 2019(4) :223–248, 2019.

[DMM21] Joan Daemen, Alireza Mehrdad, and Silvia Mella. Computing the distribution of differentials over the non-linear mapping $\chi$. In Lejla Batina, Stjepan Picek, and Mainack Mondal, editors, *Security, Privacy, and Applied Cryptography Engineering - 11th International Conference, SPACE 2021, Kolkata, India, December 10-13, 2021, Proceedings*, volume 13162 of *Lecture Notes in Computer Science*, pages 3–21. Springer, 2021.

[DMMR20] Joan Daemen, Pedro Maat Costa Massolino, Alireza Mehrdad, and Yann Rotella. The subterranean 2.0 cipher suite. *IACR Transactions on Symmetric Cryptology*, 2020(S1) :262–294, 2020.

[DMMS21] Sébastien Duval, Pierrick Méaux, Charles Momin, and François-Xavier Standaert. Exploring crypto-physical dark matter and learning with physical rounding. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2021(1) :373–401, 2021.

[DMP+15] Itai Dinur, Pawel Morawiecki, Josef Pieprzyk, Marian Srebrny, and Michal Straus. Cube attacks and cube-attack-like cryptanalysis on the round-reduced Keccak sponge function. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 733–761, Sofia, Bulgaria, April 26–30, 2015. Springer Berlin Heidelberg, Germany.

[DR05a] Joan Daemen and Vincent Rijmen. A new MAC construction ALRED and a specific instance ALPHA-MAC. In Henri Gilbert and Helena Handschuh, editors, *Fast Software Encryption – FSE 2005*, volume 3557 of *Lecture Notes in Computer Science*, pages 1–17, Paris, France, February 21–23, 2005. Springer Berlin Heidelberg, Germany.

[DR05b] Joan Daemen and Vincent Rijmen. The Pelican MAC function 2.0. Cryptology ePrint Archive, Report 2005/088, 2005.

[DR10] Joan Daemen and Vincent Rijmen. Refinements of the ALRED construction and MAC security claims. *IET Inf. Secur.*, 4(3) :149–157, 2010.

[DR20] Joan Daemen and Vincent Rijmen. *The Design of Rijndael - The Advanced Encryption Standard (AES), Second Edition.* Information Security and Cryptography. Springer, 2020.

[DRS20] Christoph Dobraunig, Yann Rotella, and Jan Schoone. Algebraic and higher-order differential cryptanalysis of Pyjamask-96. *IACR Transactions on Symmetric Cryptology*, 2020(1) :289–312, 2020.

[DS09] Itai Dinur and Adi Shamir. Cube attacks on tweakable black box polynomials. In Antoine Joux, editor, *Advances in Cryptology – EUROCRYPT 2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 278–299, Cologne, Germany, April 26–30, 2009. Springer Berlin Heidelberg, Germany.

[DS11] Itai Dinur and Adi Shamir. Breaking Grain-128 with dynamic cube attacks. In Antoine Joux, editor, *Fast Software Encryption – FSE 2011*, volume 6733 of *Lecture Notes in Computer Science*, pages 167–187, Lyngby, Denmark, February 13–16, 2011. Springer Berlin Heidelberg, Germany.

[EGL+20] Maria Eichlseder, Lorenzo Grassi, Reinhard Lüftenegger, Morten Øygarden, Christian Rechberger, Markus Schofnegger, and Qingju Wang. An algebraic attack on ciphers with low-degree round functions : Application to full MiMC. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2020, Part I*, volume 12491 of *Lecture Notes in Computer*

*Science*, pages 477–506, Daejeon, South Korea, December 7–11, 2020. Springer, Cham, Switzerland.

[EJ03] Patrik Ekdahl and Thomas Johansson. A new version of the stream cipher SNOW. In Kaisa Nyberg and Howard M. Heys, editors, *SAC 2002 : 9th Annual International Workshop on Selected Areas in Cryptography*, volume 2595 of *Lecture Notes in Computer Science*, pages 47–61, St. John's, Newfoundland, Canada, August 15–16, 2003. Springer Berlin Heidelberg, Germany.

[EMMD22] Solane El Hirch, Silvia Mella, Alireza Mehrdad, and Joan Daemen. Improved differential and linear trail bounds for ASCON. *IACR Transactions on Symmetric Cryptology*, 2022(4) :145–178, 2022.

[Fau99] Jean-Charles Faugére. A new efficient algorithm for computing Gröbner bases (f4). *Journal of Pure and Applied Algebra*, 139(1) :61–88, 1999.

[FGLM93] Jean-Charles Faugère, Patrizia M. Gianni, Daniel Lazard, and Teo Mora. Efficient computation of zero-dimensional Gröbner bases by change of ordering. *Journal of Symbolic Computation*, 16(4) :329–344, 1993.

[FKL+01] Niels Ferguson, John Kelsey, Stefan Lucks, Bruce Schneier, Michael Stay, David Wagner, and Doug Whiting. Improved cryptanalysis of Rijndael. In Bruce Schneier, editor, *Fast Software Encryption – FSE 2000*, volume 1978 of *Lecture Notes in Computer Science*, pages 213–230, New York, NY, USA, April 10–12, 2001. Springer Berlin Heidelberg, Germany.

[FNR18] Thomas Fuhr, María Naya-Plasencia, and Yann Rotella. State-recovery attacks on modified Ketje Jr. *IACR Transactions on Symmetric Cryptology*, 2018(1) :29–56, 2018.

[FO90] Philippe Flajolet and Andrew M. Odlyzko. Random mapping statistics. In Jean-Jacques Quisquater and Joos Vandewalle, editors, *Advances in Cryptology – EUROCRYPT'89*, volume 434 of *Lecture Notes in Computer Science*, pages 329–354, Houthalen, Belgium, April 10–13, 1990. Springer Berlin Heidelberg, Germany.

[FP09] Thomas Fuhr and Thomas Peyrin. Cryptanalysis of RadioGatún. In Orr Dunkelman, editor, *Fast Software Encryption – FSE 2009*, volume 5665 of *Lecture Notes in Computer Science*, pages 122–138, Leuven, Belgium, February 22–25, 2009. Springer Berlin Heidelberg, Germany.

[FRD23] Jonathan Fuchs, Yann Rotella, and Joan Daemen. On the security of keyed hashing based on public permutations. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology – CRYPTO 2023, Part III*, volume 14083 of *Lecture Notes in Computer Science*, pages 607–627, Santa Barbara, CA, USA, August 20–24, 2023. Springer, Cham, Switzerland.

[FS09] Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2009.

[FSW17] Kai Fu, Ling Sun, and Meiqin Wang. New integral attacks on SIMON. *IET Inf. Secur.*, 11(5) :277–286, 2017.

[Fun24] Margot Funk. *Algorithmes et Outils pour la Cryptanalyse des primitives symétriques*. PhD thesis, Université Paris-Saclay, 2024.

[FV14] Pierre-Alain Fouque and Thomas Vannet. Improving key recovery to 784 and 799 rounds of Trivium using optimized cube attacks. In Shiho Moriai, editor, *Fast Software Encryption – FSE 2013*, volume 8424 of *Lecture Notes in Computer Science*, pages 502–517, Singapore, March 11–13, 2014. Springer Berlin Heidelberg, Germany.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st Annual ACM Symposium on Theory of Computing*, pages 169–178, Bethesda, MD, USA, May 31 – June 2, 2009. ACM Press.

[GHKR23] Henri Gilbert, Rachelle Heim Boissier, Louiza Khati, and Yann Rotella. Generic attack on duplex-based AEAD modes using random function statistics. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023, Part IV*, volume 14007 of *Lecture Notes in Computer Science*, pages 348–378, Lyon, France, April 23–27, 2023. Springer, Cham, Switzerland.

[GJK+20] Dahmun Goudarzi, Jeremy Jean, Stefan Kölbl, Thomas Peyrin, Matthieu Rivain, Yu Sasaki, and Siang Meng Sim. Pyjamask : Block cipher and authenticated encryption with highly efficient masked implementation. *IACR Transactions on Symmetric Cryptology*, 2020(S1) :31–59, 2020.

[GKK+19] Lorenzo Grassi, Daniel Kales, Dmitry Khovratovich, Arnab Roy, Christian Rechberger, and Markus Schofnegger. Starkad and Poseidon : New hash functions for zero knowledge proof systems. Cryptology ePrint Archive, Report 2019/458, 2019.

[GKR+21] Lorenzo Grassi, Dmitry Khovratovich, Christian Rechberger, Arnab Roy, and Markus Schofnegger. Poseidon : A new hash function for zero-knowledge proof systems. In Michael Bailey and Rachel Greenstadt, editors, *USENIX Security 2021 : 30th USENIX Security Symposium*, pages 519–535. USENIX Association, August 11–13, 2021.

[GKS23] Lorenzo Grassi, Dmitry Khovratovich, and Markus Schofnegger. Poseidon2 : A faster version of the poseidon hash function. In Nadia El Mrabet, Luca De Feo, and Sylvain Duquesne, editors, *AFRICACRYPT 23 : 14th International Conference on Cryptology in Africa*, volume 14064 of *Lecture Notes in Computer Science*, pages 177–203, Sousse, Tunisia, July 19–21, 2023. Springer, Cham, Switzerland.

[GLL+20] Jian Guo, Guohong Liao, Guozhen Liu, Meicheng Liu, Kexin Qiao, and Ling Song. Practical collision attacks against round-reduced SHA-3. *Journal of Cryptology*, 33(1) :228–270, January 2020.

[GLR+20] Lorenzo Grassi, Reinhard Lüftenegger, Christian Rechberger, Dragos Rotaru, and Markus Schofnegger. On a generalization of substitution-permutation networks : The HADES design strategy. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020, Part II*, volume 12106 of *Lecture Notes in Computer Science*, pages 674–704, Zagreb, Croatia, May 10–14, 2020. Springer, Cham, Switzerland.

[GLS16] Jian Guo, Meicheng Liu, and Ling Song. Linear structures : Applications to cryptanalysis of round-reduced Keccak. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology – ASIACRYPT 2016, Part I*, volume 10031 of *Lecture Notes in Computer Science*, pages 249–274, Hanoi, Vietnam, December 4–8, 2016. Springer Berlin Heidelberg, Germany.

[Gol97] Jovan Dj. Golic. Cryptanalysis of alleged A5 stream cipher. In Walter Fumy, editor, *Advances in Cryptology – EUROCRYPT'97*, volume 1233 of *Lecture Notes in Computer Science*, pages 239–255, Konstanz, Germany, May 11–15, 1997. Springer Berlin Heidelberg, Germany.

[GP99] Louis Goubin and Jacques Patarin. DES and differential power analysis (the "duplication" method). In Çetin Kaya Koç and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES'99*, volume 1717 of *Lecture Notes in Computer Science*, pages 158–172, Worcester, Massachusetts, USA, August 12–13, 1999. Springer Berlin Heidelberg, Germany.

[Har60] Bernard Harris. Probability Distributions Related to Random Mappings. *The Annals of Mathematical Statistics*, 31(4) :1045 – 1062, 1960.

[HBYDM22] Senyang Huang, Orna Agmon Ben-Yehuda, Orr Dunkelman, and Alexander Maximov. Finding collisions against 4-round SHA-3-384 in practical time. *IACR Transactions on Symmetric Cryptology*, 2022(3) :239–270, 2022.

[Hei24] Rachelle Heim Boissier. *Symmetric Cryptanalysis : from Primitives to Modes*. PhD thesis, Université Paris-Saclay, 2024.

[Hel80] Martin E. Hellman. A cryptanalytic time-memory trade-off. *IEEE Trans. Inf. Theory*, 26(4) :401–406, 1980.

[HGSE24] Hosein Hadipour, Simon Gerhalter, Sadegh Sadeghi, and Maria Eichlseder. Improved search for integral, impossible differential and zero-correlation attacks application to Ascon, ForkSKINNY, SKINNY, MANTIS, PRESENT and QARMAv2. *IACR Transactions on Symmetric Cryptology*, 2024(1) :234–325, 2024.

[HL20] Phil Hebborn and Gregor Leander. Dasta – alternative linear layer for Rasta. *IACR Transactions on Symmetric Cryptology*, 2020(3) :46–86, 2020.

[HLLT21] Phil Hebborn, Baptiste Lambin, Gregor Leander, and Yosuke Todo. Strong and tight security guarantees against integral distinguishers. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2021, Part I*, volume 13090 of *Lecture Notes in Computer Science*, pages 362–391, Singapore, December 6–10, 2021. Springer, Cham, Switzerland.

[HLU23] Phil Hebborn, Gregor Leander, and Aleksei Udovenko. Mathematical aspects of division property. *Cryptogr. Commun.*, 15(4) :731–774, 2023.

[HLY21] Le He, Xiaoen Lin, and Hongbo Yu. Improved preimage attacks on 4-round Keccak-224/256. *IACR Transactions on Symmetric Cryptology*, 2021(1) :217–238, 2021.

[HMM+23] Clément Hoffmann, Pierrick Méaux, Charles Momin, Yann Rotella, François-Xavier Standaert, and Balazs Udvarhelyi. Learning with physical rounding for linear and quadratic leakage functions. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology – CRYPTO 2023, Part III*, volume 14083 of *Lecture Notes in Computer Science*, pages 410–439, Santa Barbara, CA, USA, August 20–24, 2023. Springer, Cham, Switzerland.

[HNR21] Rachelle Heim Boissier, Camille Noûs, and Yann Rotella. Algebraic collision attacks on Keccak. *IACR Transactions on Symmetric Cryptology*, 2021(1) :239–268, 2021.

[HPTY23] Kai Hu, Thomas Peyrin, Quan Quan Tan, and Trevor Yap. Revisiting higher-order differential-linear attacks from an algebraic perspective. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology – ASIACRYPT 2023, Part III*, volume 14440 of *Lecture Notes in Computer Science*, pages 405–435, Guangzhou, China, December 4–8, 2023. Springer, Singapore, Singapore.

[HR00] Philip Hawkes and Gregory G. Rose. Exploiting multiples of the connection polynomial in word-oriented stream ciphers. In Tatsuaki Okamoto, editor, *Advances in Cryptology – ASIACRYPT 2000*, volume 1976 of *Lecture Notes in Computer Science*, pages 303–316, Kyoto, Japan, December 3–7, 2000. Springer Berlin Heidelberg, Germany.

[HSE23] Hosein Hadipour, Sadegh Sadeghi, and Maria Eichlseder. Finding the impossible : Automated search for full impossible-differential, zero-correlation, and integral attacks. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023, Part IV*, volume 14007 of *Lecture Notes in Computer Science*, pages 128–157, Lyon, France, April 23–27, 2023. Springer, Cham, Switzerland.

[IK03] Tetsu Iwata and Kaoru Kurosawa. OMAC : One-key CBC MAC. In Thomas Johansson, editor, *Fast Software Encryption – FSE 2003*, volume 2887 of *Lecture Notes in Computer Science*, pages 129–153, Lund, Sweden, February 24–26, 2003. Springer Berlin Heidelberg, Germany.

[ISW03] Yuval Ishai, Amit Sahai, and David Wagner. Private circuits : Securing hardware against probing attacks. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 463–481, Santa Barbara, CA, USA, August 17–21, 2003. Springer Berlin Heidelberg, Germany.

[JLM14] Philipp Jovanovic, Atul Luykx, and Bart Mennink. Beyond $2^{c/2}$ security in sponge-based authenticated encryption modes. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology – ASIACRYPT 2014, Part I*, volume 8873 of *Lecture Notes in Computer Science*, pages 85–104, Kaoshiung, Taiwan, R.O.C., December 7–11, 2014. Springer Berlin Heidelberg, Germany.

[JLM+19] Philipp Jovanovic, Atul Luykx, Bart Mennink, Yu Sasaki, and Kan Yasuda. Beyond conventional security in sponge-based authenticated encryption modes. *Journal of Cryptology*, 32(3) :895–940, July 2019.

[JMN23] Thomas Johansson, Willi Meier, and Vu Nguyen. Differential cryptanalysis of mod-2/mod-3 constructions of binary weak PRFs. In *IEEE International Symposium on Information Theory, ISIT 2023, Taipei, Taiwan, June 25-30, 2023*, pages 477–482. IEEE, 2023.

[Ker83] Auguste Kerckhoffs. La cryptographie militaire. In *Journal des sciences militaires*, volume IX, pages 5–83, 1883.

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael J. Wiener, editor, *Advances in Cryptology – CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397, Santa Barbara, CA, USA, August 15–19, 1999. Springer Berlin Heidelberg, Germany.

[KK24] Björn Kriepke and Gohar M. Kyureghyan. Algebraic structure of the iterates of $\chi$. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology – CRYPTO 2024, Part IV*, volume 14923 of *Lecture Notes in Computer Science*, pages 412–424, Santa Barbara, CA, USA, August 18–22, 2024. Springer, Cham, Switzerland.

[KMS18] Rajendra Kumar, Nikhil Mittal, and Shashank Singh. Cryptanalysis of 2 round Keccak-384. In Debrup Chakraborty and Tetsu Iwata, editors, *Progress in Cryptology - INDOCRYPT 2018 : 19th International Conference in Cryptology in India*, volume 11356 of *Lecture Notes in Computer Science*, pages 120–133, New Delhi, India, December 9–12, 2018. Springer, Cham, Switzerland.

[Koo13] Bert-Jaap Koops. Crypto law survey, 2013. http://www.cryptolaw.org (accédé le 8 octobre 2024).

[KR14] Ted Krovetz and Phillip Rogaway. The OCB authenticated-encryption algorithm, 2014. RFC, 7253 :1-19.

[KRA18] Rajendra Kumar, Mahesh Sreekumar Rajasree, and Hoda AlKhzaimi. Cryptanalysis of 1-round KECCAK. In Antoine Joux, Abderrahmane Nitaj, and Tajjeeddine Rachidi, editors, *AFRICACRYPT 18 : 10th International Conference on Cryptology in Africa*, volume 10831 of *Lecture Notes in Computer Science*, pages 124–137, Marrakesh, Morocco, May 7–9, 2018. Springer, Cham, Switzerland.

[KSI16] Kota Kondo, Yu Sasaki, and Tetsu Iwata. On the design rationale of Simon block cipher : Integral attacks and impossible differential attacks against Simon variants. In Mark Manulis, Ahmad-Reza Sadeghi, and Steve Schneider, editors, *ACNS 16 : 14th International Conference on Applied Cryptography and Network Security*, volume 9696 of *Lecture Notes in Computer Science*, pages 518–536, Guildford, UK, June 19–22, 2016. Springer, Cham, Switzerland.

[KW02] Lars R. Knudsen and David Wagner. Integral cryptanalysis. In Joan Daemen and Vincent Rijmen, editors, *Fast Software Encryption – FSE 2002*, volume 2365 of *Lecture Notes in Computer Science*, pages 112–127, Leuven, Belgium, February 4–6, 2002. Springer Berlin Heidelberg, Germany.

[KY10] Elif Bilge Kavun and Tolga Yalçin. A lightweight implementation of Keccak hash function for radio-frequency identification applications. In Siddika Berna Örs Yalçin, editor, *Radio Frequency Identification : Security and Privacy Issues - 6th International Workshop, RFIDSec 2010, Istanbul, Turkey, June 8-9, 2010, Revised Selected Papers*, volume 6370 of *Lecture Notes in Computer Science*, pages 258–269. Springer, 2010.

[Lai94] Xuejia Lai. Higher order derivatives and differential cryptanalysis. In Richard E. Blahut, Daniel J. Costello, Ueli Maurer, and Thomas Mittelholzer, editors, *Communications and Cryptography : Two Sides of One Tapestry*, pages 227–233, Boston, MA, 1994. Springer US.

[LAW+23] Fukang Liu, Ravi Anand, Libo Wang, Willi Meier, and Takanori Isobe. Coefficient grouping : Breaking Chaghri and more. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023, Part IV*, volume 14007 of *Lecture Notes in Computer Science*, pages 287–317, Lyon, France, April 23–27, 2023. Springer, Cham, Switzerland.

[LDF20] Baptiste Lambin, Patrick Derbez, and Pierre-Alain Fouque. Linearly equivalent S-boxes and the division property. *Designs, Codes and Cryptography*, 88(10) :2207–2231, 2020.

[Leu24] Gaëtan Leurent. *Symmetric Cryptanalysis Beyond Primitives*. 2024. Habiliation à Diriger des Recherches.

[LHY21] Xiaoen Lin, Le He, and Hongbo Yu. Improved preimage attacks on 3-round Keccak-224/256. *IACR Transactions on Symmetric Cryptology*, 2021(3) :84–101, 2021.

[LIM19] Fukang Liu, Takanori Isobe, and Willi Meier. Cube-based cryptanalysis of Subterranean-SAE. *IACR Transactions on Symmetric Cryptology*, 2019(4) :192–222, 2019.

[LIM21] Fukang Liu, Takanori Isobe, and Willi Meier. Cryptanalysis of full LowMC and LowMC-M with algebraic techniques. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021, Part III*, volume 12827 of *Lecture Notes in Computer Science*, pages 368–401, Virtual Event, August 16–20, 2021. Springer, Cham, Switzerland.

[LKSM24] Fukang Liu, Abul Kalam, Santanu Sarkar, and Willi Meier. Algebraic attack on FHE-friendly cipher HERA using multiple collisions. *IACR Transactions on Symmetric Cryptology*, 2024(1) :214–233, 2024.

[LM22] Charlotte Lefevre and Bart Mennink. Tight preimage resistance of the sponge construction. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022, Part IV*, volume 13510 of *Lecture Notes in Computer Science*, pages 185–204, Santa Barbara, CA, USA, August 15–18, 2022. Springer, Cham, Switzerland.

[LMM24] Fukang Liu, Mohammad Mahzoun, and Willi Meier. Modelling ciphers with overdefined systems of quadratic equations : Application to Friday, Vision, RAIN and Biscuit. Cryptology ePrint Archive, Report 2024/786, 2024.

[LMN89] Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, fourier transform, and learnability. In *30th Annual Symposium on Foundations of Computer Science, Research Triangle Park, North Carolina, USA, 30 October - 1 November 1989*, pages 574–579. IEEE Computer Society, 1989.

[LMØM23] Fukang Liu, Mohammad Mahzoun, Morten Øygarden, and Willi Meier. Algebraic attacks on RAIN and AIM using equivalent representations. *IACR Transactions on Symmetric Cryptology*, 2023(4) :166–186, 2023.

[LMSI22] Fukang Liu, Willi Meier, Santanu Sarkar, and Takanori Isobe. New low-memory algebraic attacks on LowMC in the Picnic setting. *IACR Transactions on Symmetric Cryptology*, 2022(3) :102–122, 2022.

[LPSS24] Gregor Leander, Christof Paar, Julian Speith, and Lukas Stennes. HAWKEYE - recovering symmetric cryptography from hardware circuits. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology – CRYPTO 2024, Part IV*, volume 14923 of *Lecture Notes in Computer Science*, pages 340–376, Santa Barbara, CA, USA, August 18–22, 2024. Springer, Cham, Switzerland.

[LPTY16] Atul Luykx, Bart Preneel, Elmar Tischhauser, and Kan Yasuda. A MAC mode for lightweight block ciphers. In Thomas Peyrin, editor, *Fast Software Encryption – FSE 2016*, volume 9783 of *Lecture Notes in Computer Science*, pages 43–59, Bochum, Germany, March 20–23, 2016. Springer Berlin Heidelberg, Germany.

[LRS23] Gregor Leander, Shahram Rasoolzadeh, and Lukas Stennes. Cryptanalysis of HALFLOOP block ciphers destroying HALFLOOP-24. *IACR Transactions on Symmetric Cryptology*, 2023(4) :58–82, 2023.

[LS19] Ting Li and Yao Sun. Preimage attacks on round-reduced Keccak-224/256 via an allocating approach. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019, Part III*, volume 11478 of *Lecture Notes in Computer Science*, pages 556–584, Darmstadt, Germany, May 19–23, 2019. Springer, Cham, Switzerland.

[LSLW17] Ting Li, Yao Sun, Maodong Liao, and Dingkang Wang. Preimage attacks on the round-reduced Keccak with cross-linear structures. *IACR Transactions on Symmetric Cryptology*, 2017(4) :39–57, 2017.

[LSMI21] Fukang Liu, Santanu Sarkar, Willi Meier, and Takanori Isobe. Algebraic attacks on rasta and dasta using low-degree equations. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2021, Part I*, volume 13090 of *Lecture Notes in Computer Science*, pages 214–240, Singapore, December 6–10, 2021. Springer, Cham, Switzerland.

[LSMI22] Fukang Liu, Santanu Sarkar, Willi Meier, and Takanori Isobe. The inverse of $\chi$ and its applications to Rasta-like ciphers. *Journal of Cryptology*, 35(4) :28, October 2022.

[LSW+22] Fukang Liu, Santanu Sarkar, Gaoli Wang, Willi Meier, and Takanori Isobe. Algebraic meet-in-the-middle attack on LowMC. In Shweta Agrawal and Dongdai Lin, editors, *Advances in Cryptology – ASIACRYPT 2022, Part I*, volume 13791 of *Lecture Notes in Computer Science*, pages 225–255, Taipei, Taiwan, December 5–9, 2022. Springer, Cham, Switzerland.

[MDV17] Silvia Mella, Joan Daemen, and Gilles Van Assche. New techniques for trail bounds and application to differential trails in Keccak. *IACR Transactions on Symmetric Cryptology*, 2017(1) :329–357, 2017.

[MDV23] Silvia Mella, Joan Daemen, and Gilles Van Assche. Tighter trail bounds for Xoodoo. *IACR Transactions on Symmetric Cryptology*, 2023(4) :187–214, 2023.

[MJSC16] Pierrick Méaux, Anthony Journault, François-Xavier Standaert, and Claude Carlet. Towards stream ciphers for efficient FHE with low-noise ciphertexts. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016, Part I*, volume 9665 of *Lecture Notes in Computer Science*, pages 311–343, Vienna, Austria, May 8–12, 2016. Springer Berlin Heidelberg, Germany.

[MMD23] Silvia Mella, Alireza Mehrdad, and Joan Daemen. Differential and linear properties of vectorial boolean functions based on chi. *Cryptogr. Commun.*, 15(6) :1087–1116, 2023.

[MMGD22] Alireza Mehrdad, Silvia Mella, Lorenzo Grassi, and Joan Daemen. Differential trail search in cryptographic primitives with big-circle chi : Application to Subterranean. *IACR Transactions on Symmetric Cryptology*, 2022(2) :253–288, 2022.

[Moo70] J. W. Moon. *Counting Labelled Trees*. Canadian Mathematical Congress 1970, William Clowes and Sons, 1970.

[MV06] David A. McGrew and John Viega. The use of Galois message authentication code (GMAC) in ipsec ESP and AH. *RFC*, 4543 :1–14, 2006.

[NM11] K. Nohl and L. Melette. GPRS intercept : Wardriving your country, 2011. http://events.ccc.de/camp/2011/Fahrplan/attachments/1868_110810.SRLabs-Camp-GRPS_Intercept.pdf et https://media.ccc.de/v/cccamp11-4504-gprs_intercept-en#t=1744 (accédé le 8 octobre 2024).

[NRM11] María Naya-Plasencia, Andrea Röck, and Willi Meier. Practical analysis of reduced-round Keccak. In Daniel J. Bernstein and Sanjit Chatterjee, editors, *Progress in Cryptology - INDOCRYPT 2011 : 12th International Conference in Cryptology in India*, volume 7107 of *Lecture Notes in Computer Science*, pages 236–254, Chennai, India, December 11–14, 2011. Springer Berlin Heidelberg, Germany.

[QSLG17] Kexin Qiao, Ling Song, Meicheng Liu, and Jian Guo. New collision attacks on round-reduced Keccak. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017, Part III*, volume 10212 of *Lecture Notes in Computer Science*, pages 216–243, Paris, France, April 30 – May 4, 2017. Springer, Cham, Switzerland.

[Raj19] Mahesh Sreekumar Rajasree. Cryptanalysis of round-reduced KECCAK using non-linear structures. In Feng Hao, Sushmita Ruj, and Sourav Sen Gupta, editors, *Progress in Cryptology - INDOCRYPT 2019 : 20th International Conference in Cryptology*

*in India*, volume 11898 of *Lecture Notes in Computer Science*, pages 175–192, Hyderabad, India, December 15–18, 2019. Springer, Cham, Switzerland.

[RST18] Christian Rechberger, Hadi Soleimany, and Tyge Tiessen. Cryptanalysis of low-data instances of full LowMCv2. *IACR Transactions on Symmetric Cryptology*, 2018(3) :163–181, 2018.

[SCW23] Yimeng Sun, Jiamin Cui, and Meiqin Wang. Improved attacks on LowMC with algebraic techniques. *IACR Transactions on Symmetric Cryptology*, 2023(4) :143–165, 2023.

[SD24a] Jan Schoone and Joan Daemen. Algebraic properties of the maps $\chi_n$. *Designs, Codes and Cryptography*, 92(5) :2341–2365, 2024.

[SD24b] Jan Schoone and Joan Daemen. The state diagram of $\chi$. *Designs, Codes and Cryptography*, 92(5) :1393–1421, 2024.

[SGSL18] Ling Song, Jian Guo, Danping Shi, and San Ling. New MILP modeling : Improved conditional cube attacks on Keccak-based constructions. In Thomas Peyrin and Steven Galbraith, editors, *Advances in Cryptology – ASIACRYPT 2018, Part II*, volume 11273 of *Lecture Notes in Computer Science*, pages 65–95, Brisbane, Queensland, Australia, December 2–6, 2018. Springer, Cham, Switzerland.

[Sha49] Claude E. Shannon. Communication theory of secrecy systems. *Bell Systems Technical Journal*, 28(4) :656–715, 1949.

[SHA15] SHA-3 standard : Permutation-based hash and extendable-output functions. National Institute of Standards and Technology, NIST FIPS PUB 202, U.S. Department of Commerce, August 2015.

[Sho96] Victor Shoup. On fast and provably secure message authentication based on universal hashing. In Neal Koblitz, editor, *Advances in Cryptology – CRYPTO'96*, volume 1109 of *Lecture Notes in Computer Science*, pages 313–328, Santa Barbara, CA, USA, August 18–22, 1996. Springer Berlin Heidelberg, Germany.

[Sie85] Thomas Siegenthaler. Decrypting a class of stream ciphers using ciphertext only. *IEEE Trans. Computers*, 34(1) :81–85, 1985.

[SL09] Xiaorui Sun and Xuejia Lai. Improved integral attacks on MISTY1. In Michael J. Jacobson, Jr., Vincent Rijmen, and Reihaneh Safavi-Naini, editors, *SAC 2009 : 16th Annual International Workshop on Selected Areas in Cryptography*, volume 5867 of *Lecture Notes in Computer Science*, pages 266–280, Calgary, Alberta, Canada, August 13–14, 2009. Springer Berlin Heidelberg, Germany.

[SLG17] Ling Song, Guohong Liao, and Jian Guo. Non-full sbox linearization : Applications to collision attacks on round-reduced Keccak. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017, Part II*, volume 10402 of *Lecture Notes*

*in Computer Science*, pages 428–451, Santa Barbara, CA, USA, August 20–24, 2017. Springer, Cham, Switzerland.

[Sti95] Douglas R. Stinson. On the connections between universal hashing, combinatorial designs and error-correcting codes. *Electron. Colloquium Comput. Complex.*, TR95-052, 1995.

[STSH21] Ling Song, Yi Tu, Danping Shi, and Lei Hu. Security analysis of Subterranean 2.0. *Designs, Codes and Cryptography*, 89(8) :1875–1905, 2021.

[TA14] Yosuke Todo and Kazumaro Aoki. FFT key recovery for integral attack. In Dimitris Gritzalis, Aggelos Kiayias, and Ioannis G. Askoxylakis, editors, *CANS 14 : 13th International Conference on Cryptology and Network Security*, volume 8813 of *Lecture Notes in Computer Science*, pages 64–81, Heraklion, Crete, Greece, October 22–24, 2014. Springer, Cham, Switzerland.

[TIHM17] Yosuke Todo, Takanori Isobe, Yonglin Hao, and Willi Meier. Cube attacks on non-blackbox polynomials based on division property. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017, Part III*, volume 10403 of *Lecture Notes in Computer Science*, pages 250–279, Santa Barbara, CA, USA, August 20–24, 2017. Springer, Cham, Switzerland.

[TM16] Yosuke Todo and Masakatu Morii. Bit-based division property and application to Simon family. In Thomas Peyrin, editor, *Fast Software Encryption – FSE 2016*, volume 9783 of *Lecture Notes in Computer Science*, pages 357–377, Bochum, Germany, March 20–23, 2016. Springer Berlin Heidelberg, Germany.

[TMC+21] Meltem Sönmez Turan, Kerry McKay, Donghoon Chang, Çağdaş Çalık, Lawrence Bassham, Jinkeon Kang, and John Kelsey. NIST IR 8369 status report on the second round of the nist lightweight cryptography standardization process, 2021. https://csrc.nist.gov/pubs/ir/8369/final (accédé le 25 novembre 2024).

[Tod15a] Yosuke Todo. Integral cryptanalysis on full MISTY1. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 413–432, Santa Barbara, CA, USA, August 16–20, 2015. Springer Berlin Heidelberg, Germany.

[Tod15b] Yosuke Todo. Structural evaluation by generalized integral property. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 287–314, Sofia, Bulgaria, April 26–30, 2015. Springer Berlin Heidelberg, Germany.

[WC81] Mark N. Wegman and Larry Carter. New hash functions and their use in authentication and set equality. *Journal of Computer and System Sciences*, 22 :265–279, 1981.

[WWF+21] Congming Wei, Chenhao Wu, Ximing Fu, Xiaoyang Dong, Kai He, Jue Hong, and Xiaoyun Wang. Preimage attacks on 4-round Keccak by solving multivariate quadratic systems. In Jong Hwan Park and Seung-Hyun Seo, editors, *ICISC 21 : 24th International Conference on Information Security and Cryptology*, volume 13218 of *Lecture Notes in Computer Science*, pages 195–216, Seoul, Korea, December 1–3, 2021. Springer, Cham, Switzerland.

[ZLL24] Lulu Zhang, Meicheng Liu, and Dongdai Lin. A three-stage MITM attack on LowMC from a single plaintext-ciphertext pair. In Benjamin Smith and Huapeng Wu, editors, *SAC 2022 : 29th Annual International Workshop on Selected Areas in Cryptography*, volume 13742 of *Lecture Notes in Computer Science*, pages 306–327, Windsor, Canada, August 24-26, 2024. Springer, Cham, Switzerland.

[ZWY+23] Kaiyi Zhang, Qingju Wang, Yu Yu, Chun Guo, and Hongrui Cui. Algebraic attacks on round-reduced Rain and full AIM-III. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology – ASIACRYPT 2023, Part III*, volume 14440 of *Lecture Notes in Computer Science*, pages 285–310, Guangzhou, China, December 4–8, 2023. Springer, Singapore, Singapore.