N° d'ordre :

REPUBLIQUE ALGERIENNE DEMOCRATIQUE & POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR & DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE DJILLALI LIABES
FACULTE DES SCIENCES EXACTES
SIDI BEL ABBES

THESE DE DOCTORAT DE 3ème CYCLE

Présentée par M^lle^ **ANANI Djihed**

Domaine : Mathématiques Informatique
Filière : Informatique
Intitulé de la formation : Réseaux et sécurité de l'information

Intitulée

Partage de secret cryptographique appliqué aux images numériques

Soutenue le *18/01/2017*
Devant le jury composé de :

Président : **Dr. GAFOUR Abdelkader (MCA, UDL-SBA)**

Examinateurs : **Dr. KESKES Nabil (MCA, ESI-SBA)**
Dr. BOUKLI HACENE Sofiane (MCA, UDL-SBA)
Pr. BELALEM Ghalem (Pr, Université d'Oran 1)

Directeur de thèse : **Pr. FARAOUN Kamel Mohamed (Pr, UDL-SBA)**

Année universitaire *2016/2017*

À mes parents ...

Remerciements

Cette thèse est le fruit de trois ans de travail dont lequel n'aurait pas abouti sans la contribution de mon directeur de thèse monsieur «Faraoun Kamel Mohamed » dont je lui suis très reconnaissante, et que je tiens à remercier sincèrement de m'avoir bénéficié de son savoir, ses compétences scientifiques, pour ses encouragements, ses conseils et sa patience tout au long de mes travaux de thèse.
Je remercie également messieurs les membres du jury, de l'honneur qu'ils m'ont fait en acceptant de siéger à mon jury de doctorat.
Mes reconnaissances vont aussi à tous mes enseignants du département d'informatique de l'université de Sidi Bel Abbès qui m'ont mené vers la voie de la recherche.
Enfin, un grand merci à mes parents, mes frères et ma sœur pour leurs encouragements, leur grande compréhension et pour leur inconditionnel soutien le long de la thèse.
Merci à tous ceux qui m'ont moralement ou pratiquement soutenue et dont ces lignes ont accidentellement oublié de mentionner.

ملخص

في بيئة متعددة المستخدمين، سياسة تقاسم المفاتيح في أمان هي واحدة من وحدات الأمن الرئيسية، ويمكن أن تكون أهم مصدر ضعف وحدات النظام المختلفة. التقاسم المشفر للمفاتيح بين العديد من المستخدمين يعد إشكالية مختلفة تتطلب أكثر من التبادل البسيط للمفاتيح بين كيانين، أو المواثقة المباشرة لمستخدم واحد من طرف الخادم. منذ أول مخطط الذي إقترحه شامير، تم إقتراح عدد قليل من المخططات المماثلة، نظراً لصعوبة المشكلة، و متطلبات السلامة المستلزمة.
في إطار هذه الأطروحة، قمنا بهذف إقتراح و إقرار مخطط جديد (بما في ذلك مخطط عتبة (t, n)) و بروتوكولات باستخدام أدوات جديدة. لهذا إقترحنا مخططين مختلفين لتقاسم الأسرار بطريقة مشفرة في مجال الوسائط المتعددة و الصور الرقمية على وجه الخصوص؛ المخطط الأول يعتمد على الخلايا التلقائية و معالج بالتالي على الجانب السلبي الكبير لوحدة إتساق الأجزاءt، و التي هي مشتركة بين جميع المخططات التي تم إقتراحها و القائمة على إستعمال الخلايا التلقائية، المخطط المقترح الثاني أتاح إلى علاج نوع آخر من المشاكل و المتعلق بمثالية الأجزاء على عكس طرق تقاسم السر الكلاسيكية، و ذلك بالإستعانة بوسائل أخرى أكثر فعالية المتعمدة على إستخدام أنظمة المعادلات الخطية المعرفة على $GF(2^8)$؛ (ترد تفاصيل المخططين فيما يلي)

الكلمات المفتاحية: عتبة مخططات التقاسم المشفر للأسرار، الخلايا التلقائية، أنظمة المعادلات الخطية، المجموعة المنتهية $GF(2^8)$

Résumé

Dans un environnement multiutilisateur, la politique de partage de clé(s) sécurisé est l'une des principaux modules de sécurité, et peut être la source de vulnérabilité la plus importante parmi les différents modules du système. Un partage de clé(s) cryptographique entre plusieurs utilisateurs constitue une problématique différente et beaucoup plus exigeante que le simple échange de clé(s) entre deux entités, ou l'authentification directe d'un seul utilisateur par un serveur. Depuis le premier schéma proposé par Shamir, un faible nombre de schémas similaires ont été proposé, vu la difficulté du problème, et l'exigence sécuritaire très importante qui sont requises.

Dans le cadre de cette thèse, nous avons visé à proposer et valider de nouveaux schémas (notamment les schémas d'ordre (t,n)) et protocoles en utilisant de nouveaux outils. Pour cela nous avons proposé deux schémas différents de partage de secret(s) cryptographique dans le domaine du multimédia et des images numériques en particulier ; le premier schéma reposant sur les automates cellulaires et répondant ainsi à un inconvénient majeur celui de la t-consistance des parts, et qui est commun à tous les schémas se basant sur les automates cellulaires qui ont déjà été proposé, le deuxième schéma proposé a permis de traiter un autre type de problème celui de l'idéalité des parts contrairement aux méthodes classiques de partage de secret, et ceci en exploitant d'autres méthodes plus performantes visant à utiliser les systèmes d'équations linéaires dans le corps de Rijndael ; (les deux schémas sont détaillés dans ce qui suit).

Mots clés : Les schémas de partage de secret(s) cryptographique à seuil, Les automates cellulaires, Les systèmes d'équations linéaires, Les corps finis de Rijndael

Abstract

In a multiuser environment, secure key(s) sharing policy is one of the main security modules, and can be the most important source of vulnerability among the various system modules. A cryptographic key(s) sharing between multiple users is a different issue and much more demanding than simple key exchange between two entities, or through authentication to a single user by a server. Since the first scheme proposed by Shamir, a small number of similar schemes have been proposed, given the difficulty of the problem, and the very important safety requirement that is required.
Within the framework of this thesis, we aimed to propose and validate new designs (including order patterns (t, n)) and protocols using new tools. For this we have proposed two different secret sharing schemes in the field of multimedia and digital images in particular ; the first scheme based on cellular automata and thus addressing a major drawback that the t-consistency of shares, and which is common to all the patterns based on cellular automata that have already been proposed, the second proposed scheme has allowed to treat another type of problem that ideality shares unlike traditional secret sharing methods, and this by exploiting other more effective methods to use linear equations systems in the Rijndael field ; (Both schemes are detailed in the following).

Keywords : A cryptographic threshold secret sharing scheme, Cellular automata, Systems of linear equation, Rijndael finite field

Table des matières

Table des figures

Liste des tableaux

Liste des Abréviations

AC Automate cellulaire

ACL Automate cellulaire linéaire

ACM Automate cellulaire à mémoire

ACML Automate cellulaire à mémoire linéaire

ACR Automate cellulaire réversible

CRT Le théorème des restes chinois

GF(q) Corps de Galois de cardinal q

Introduction générale

La cryptologie, définie littéralement comme la science du secret, tire son origine à la nécessité de protéger des informations, qui sert à réguler et civiliser l'ère du partage et d'échanges d'information dans laquelle nous entrons.
Afin qu'elle soit significative, elle doit être effectuée de manière unique un peu comme la sécurité d'une serrure qui repose sur le fait que seules les personnes autorisées à ouvrir cette serrure sont ceux ayant la clé correspondante qui est unique à chaque serrure. Donc il est clair que l'unicité est certainement une préoccupation majeure pour les systèmes cryptographiques, le hasard est d'une importance égale d'où la génération de nombres pseudo-aléatoires, c'est pourquoi l'unicité et le hasard restent des composantes essentielles à tout système cryptographique. De leurs caractéristiques, on peut très bien choisir les automates cellulaires pour la génération de nombres pseudo-aléatoires, les automates cellulaires restent un moyen sûr utilisé en cryptographie depuis longtemps (après leur utilisation par Wolfram [Wolfram 1985] en 1986).
Aussi afin de mieux renforcer la sécurité d'un système cryptographique, on peut choisir de ne pas laisser la clé à la portée d'une seule personne mais plutôt de la partager entre plusieurs, l'union d'un ensemble de personnes qualifié entrainera la reconstruction de cette clé secrète, c'est ce qu'on appelle les schémas de partage de secret (Secret Sharing Schemes).
Les schémas de partage de secret sont des procédures cryptographiques introduites en 1979 indépendamment par Shamir [Shamir 1979] et Blakley [Blakley 1979], l'idée est de partager un secret entre un ensemble de participants de telle sorte qu'un nombre de ces participants qualifiés pourront reconstruire le secret en mettant en commun leurs parts attribuées du secret, alors qu'aucun autre ensemble non qualifié ne pourra le faire.
Récemment, plusieurs schémas de partage de secret basés sur les automates cellulaires ont été développés.

Problématique :

Nous traitons dans ce mémoire le problème de partage de secret à seuil, un problème récent en cryptographie consistant à sécuriser une donnée secrète en la partageant entre plusieurs personnes, de telle sorte que la reconstruction de cette donnée ne se fera que avec la collaboration de certaines personnes d'un ensemble qualifié. Etant donné un secret « S », la personne de confiance partage ce secret entre un ensemble de n

participants en offrant à chaque participant « i » ($1 \leq i \leq n$) une part du secret « S », pour un schéma (t,n) à seuil t, la collaboration d'au moins « t » ($t \leq n$) participants parmi les « n » est nécessaire pour reconstruire le secret « S ».
Nous nous sommes intéressés au départ aux automates cellulaires à mémoire linéaire pour résoudre le problème, et après étude des différents travaux réalisés dans le domaine nous nous sommes aperçus que les solutions de partage de secret proposées exploitant ce type d'automate cellulaire s'avèrent inutiles pour des applications réelles suite à leur inconvénient majeur commun qui n'offre pas la possibilité à tous les groupes de « t » participants de reconstruire le secret, mais seulement à ceux ayant des parts consécutives du secret !
Puis nous nous sommes intéressés au cas de partage multi-secret, en cherchant à proposer une autre solution plus performante indépendante aux automates cellulaires, où nous avons proposé un schéma de partage multi-secret en utilisant les systèmes d'équations linéaires surdéterminés définis sur un corps de Rijndael, et dans laquelle il s'est avéré que les parts obtenues par les participants du groupe sont bien optimales contrairement aux solutions travaillant avec un modulo premier !

Motivations et contributions :

Suite à la consultation de l'état de l'art dans ce domaine, nous avons proposé deux schémas de partage de secret cryptographique apportant chacun d'eux respectivement une réponse à un problème donné en apportant des améliorations aux autres schémas qui ont été proposés.
La première solution que nous avons proposé repose sur l'utilisation des automates cellulaires à mémoire linéaire pour partager une image secrète selon un schéma à seuil (t,n) répondant au problème posé précédemment où seulement les participants détenant des parts consécutives du secret pouvaient reconstruire le secret alors que les autres ne le pouvaient pas ; pour faire face à ce problème nous avons choisi de faire appel à une matrice qu'on appellera « matrice d'affectation des configurations » (les configurations de l'automate cellulaire), et où au lieu d'attribuer une seule configuration à chaque participant, nous affectons un ensemble déterminé de configurations à chaque participant i ($1 \leq i \leq n$) de sorte que tout ensemble de t participants aie un ensemble de configurations avec des numéros d'ordre consécutifs qui lui permettra de reconstruire le secret de départ.
Indépendamment des automates cellulaires, La deuxième solution que nous avons proposé permet de traiter le problème de partage multi-secret où on partage « m » images secrètes selon un schéma à seuil (t,n) en exploitant les systèmes d'équations surdéterminés définis sur un corps

de Rijndael, cette deuxième solution a permis d'apporter de meilleurs résultats de performance par rapport aux méthodes classiques rendant ainsi le schéma de partage multi-secret proposé optimal.
Le choix d'en prendre les secrets comme étant des images est fait uniquement à cause des caractéristiques spécifiques qui en sortent de ces dernières.

Plan du mémoire :

Ce mémoire est organisé en trois parties, dont voici ci-dessous la structure donnée :

Partie I : Nous introduisons dans la première partie une description des schémas de partage de secret pour cela :

- Nous abordons quelques notions et caractéristiques des schémas de partage de secret notamment les schémas à seuil, les critères de qualité de ces schémas, ainsi que leurs domaines d'application. Nous décrivons par la suite les schémas de partage de secret classiques qui englobent les premières solutions de partage de secret, puis nous aborderons l'état de l'art des différents schémas proposés exploitant ces solutions classiques.

Partie II : Nous décrirons dans la deuxième partie la manière dans laquelle les automates cellulaires sont utilisés pour le partage de secret.

- Le deuxième chapitre portera sur l'étude des automates cellulaires, où nous présenterons ces derniers par des concepts de base, ainsi nous aborderons la manière dans laquelle les automates cellulaires sont utilisés en cryptographie.
- Dans le chapitre trois, une étude sur l'état de l'art se fera et qui concernera les différents schémas de partage de secret exploitant cette fois-ci les automates cellulaires.

Partie III : Enfin dans la dernière partie, nous verrons les deux solutions qui ont été proposées pour le problème du partage de secret.

- Nous présenterons dans le quatrième chapitre, le premier schéma de partage de secret proposé qui exploite les automates cellulaires.
- Nous décrirons dans le cinquième chapitre, le deuxième schéma de partage de secret proposé basé sur l'utilisation des systèmes d'équations linéaires surdéterminés dans des corps de Rijndael ; Pour

cela nous ferons une initiation sur les systèmes d′équations linéaires et sur les corps de Galois.

Nous conclurons ce mémoire, par voir les apports des travaux réalisés ainsi qu′un ensemble de perspectives futures.

Partie I

Introduction au partage de secret cryptographique

Chapitre 1

Problématique et définitions du partage de secret cryptographique

Sommaire

1.1 Introduction :

Dans certaines situations quotidiennes, il peut être plus prudent de ne pas confier un secret à la volonté d'une seule personne, mais plutôt de le partager entre plusieurs. Le partage de secret (Secret Sharing) est une technique qui consiste à répartir un secret (ou plusieurs dans le cas de partage multi-secret) entre un ensemble de participants, en offrant à chacun d'eux une part du secret (information secrète), en cas de besoin le secret pourra être reconstruit en assemblant les informations détenues par chaque participant ayant rapport avec le secret. On peut trouver ce genre de partage par exemple dans de nombreuses banques où l'intervention du directeur de la banque avec un des deux sous-directeurs par exemple est nécessaire. La cryptographie force alors l'intervention de plusieurs personnes en faisant comme hypothèse qu'il n'y aura pas trop de personnes malintentionnées qui se coalisent pour attaquer le système. Dans ce chapitre, on explique dans un premier temps la problématique de partage de secret cryptographique, avec les définitions de base relatives ainsi que les domaines d'application. Nous présentons ensuite les principaux schémas de partage de secret. Finalement, une présentation détaillée de quelques solutions proposées est fournie.

1.2 Le partage de secret, et partage à seuil :

Un exemple tiré de [Liu 1968] permet d'appréhender rapidement la problématique liée au partage de secret (Secret Sharing) :

" *Eleven scientists are working on a secret project. They wish to lock up the document in a cabinet so that the cabinet can be opened if and only if six or more of the scientists are present.*
What is the smallest number of locks needed ? What is the smallest numbers of keys to the locks each scientist must carry ? "

Qu'on peut traduire en français par :

" *Onze scientifiques travaillent sur un projet secret. Ils veulent enfermer le document dans une armoire de sorte que l'armoire peut être ouverte si et seulement si six ou plus des scientifiques sont présents.*
Quel est le plus petit nombre de serrures nécessaires ? Quel est le plus petit nombre de clés de serrures que chaque scientifique doit retenir ? "

Si cette situation n'est pas très réaliste, elle correspond néanmoins à des cas concrets : une entité souhaite partager un secret entre plusieurs dépositaires de sorte que seules certaines coalitions soient autorisées à

accéder à ce secret.

1.2.1 Schéma de partage de secret :

Dans un schéma de partage de secret, un secret (pouvant représenter une clé, un code, du texte, une image, $\cdots$) est réparti entre plusieurs participants organisés en une structure d'accès recensant tous les groupes pouvant accéder au secret.
L'objectif est de fournir une information propre à chaque participant de sorte que seul un groupe dit qualifié de participants puisse reconstruire le secret. Pour ce faire, un schéma de partage de secret entre n participants est défini à partir de deux phases appelées phase de partage et phase de reconstruction. Plus précisément, on donne la définition suivante :

Définition 1. *(Schéma de partage de secret)[Renner 2014]. Soient $S, S_1, \cdots, S_n$ des ensembles non vides. Un schéma de partage de secret entre n participants est défini par :*

- *Une phase de partage probabiliste, notée Partage(.), satisfaisant :*

$$\forall s \in \mathbb{S}, v_s = Partage(s) \text{ avec } v_s \in \mathbb{E}_s, \tag{1.1}$$

où v_s, appelé vecteur de partage, est constitué des n parts : $c_i \in \mathbb{S}_i$ dont chacune est supposée détenue par un participant i, pour $i \in \{1, \cdots, n\}$. De plus, l'ensemble $E_s \subseteq S_1 \times \cdots \times S_n$ désigne l'ensemble des vecteurs de partage pouvant être obtenu en appliquant la procédure de partage à un secret $s \in \mathbb{S}$.

- *Une phase de reconstruction déterministe, notée Reconstruction(.), vérifiant :*

$$\forall v_s = (c_1, \cdots, c_n) \in \mathbb{E}_s, s = Reconstruction((c_i)_{i \in \mathbb{Q}}) \text{ avec } s \in \mathbb{S}, \tag{1.2}$$

où Q est un groupe qualifié de participants. En particulier, le nombre minimum de parts quelconques d'un vecteur de partage permettant de reconstituer le secret s est nommé paramètre de reconstruction, qu'on notera r.

Un tel schéma de partage de secret est associé à un ensemble E défini tel que :

$$E = \{(s, v_s) : s \in \mathbb{S}, v_s \in \mathbb{E}_s \subseteq S_1 \times \cdots \times S_n\}. \tag{1.3}$$

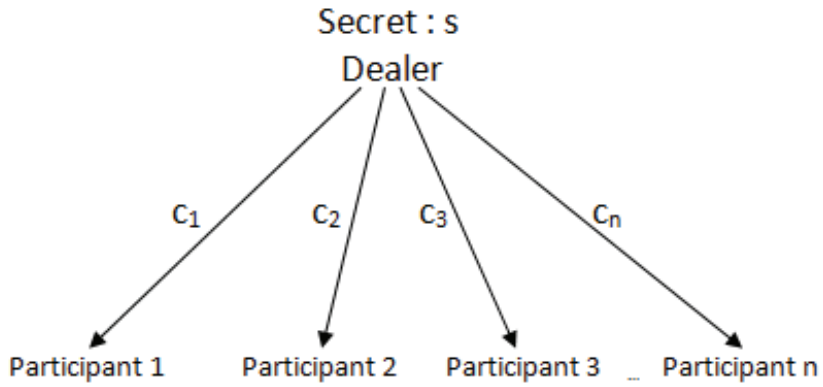


FIGURE 1.1 – Partage de secret.

1.2.2 Problématique :

Le problème de partage de secret à seuil « t » peut être énoncé comme suite :
Ayant un secret S, comment le diviser en n composantes, de telle manière que S soit reconstructible à partir de n′importe quel sous-ensemble de t participants, t≤n. Avec la précision que toute connaissance de moins de t parts ne permet pas d′avoir la moindre information sur le secret S.
En autre, l′objectif est de fournir une information à chaque participant tel que [Kaced 2012] :

1. Un sous-groupe qualifié d′au moins t participants doit pouvoir reconstituer le secret (intégrité),
2. Tout autre sous-groupe de moins de t participant ne peut obtenir aucune information sur le secret (confidentialité).

1.2.3 Schéma à seuil :

Les schémas (t,n) de partage de secret à seuil t (avec t ≤ n), sont des procédures cryptographiques (cryptographie à seuil) permettant à l′initiateur de partager le secret entre un groupe de n participants de tel sorte que tout sous-groupe qualifié d′au moins t participants parmi n pourra reconstruire le secret qui a été partagé en mettant en commun leurs parts attribuées du secret, et en les envoyant soit à l′initiateur du schéma de partage ou à une autre personne de confiance (qui peut être une personne parmi les t), qui va s′en charger de récupérer les parts des t participants et donc de reconstruire le secret de départ, cependant aucun sous-groupe inferieur à t participants ne pourra être autorisé de le faire (un ensemble de participants de moindre cardinalité n′ait aucune

information dans le sens de Shannon [Shannon 1949] à propos du secret), pour plus de précision, les définitions suivantes sont données :

Définition 2. *(Paramètre de sécurité)[Renner 2014]. Le paramètre de sécurité d'un schéma de partage de secret, noté ps ($ps = t - 1$), indique le nombre maximum de parts d'un vecteur de partage pouvant être connu sans qu'aucune information sur le secret partagé ne soit révélée.*

Définition 3. *(Schéma à seuil)[Renner 2014]. Considérons un schéma de partage de secret de paramètre de reconstruction r et de sécurité ps. On dit qu'un schéma de partage de secret est à seuil lorsque $r = ps + 1$.*

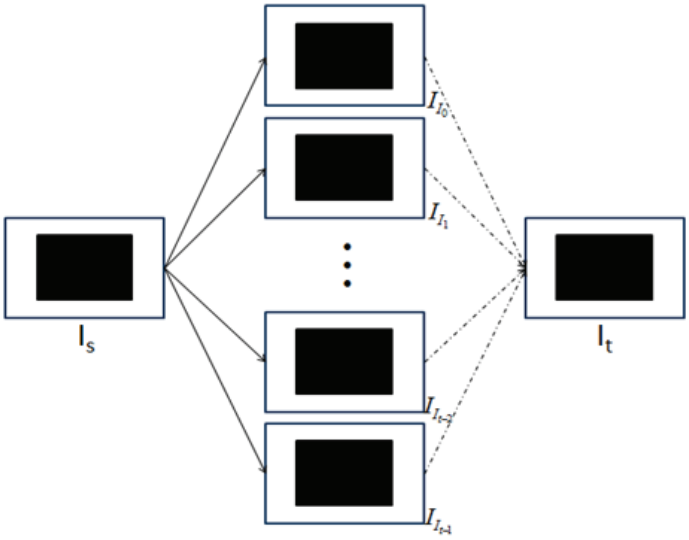


FIGURE 1.2 – Principe de partage d'image secrète.

Le principe en est le même pour un secret donné sous forme d'image I_s, la collaboration d'un certain nombre de participants (selon le seuil choisi), et après avoir réuni leurs parts du secret, permettra de reconstruire l'image I_t représentante le secret du départ.

✤ **Remarques :**

- **On peut partager un secret comme on peut partager plusieurs (voir m secrets), ce type de protocole cryptographique est utilisé de la même manière et sans plus d'information que dans le cas où un seul secret est à protéger c'est ce qu'on appelle un schéma de partage multi-secret à seuil (m,t,n), dont les premiers schémas ont été proposés en 1994 par He et Dawson [He 1994].**
- **Il est possible de donner plus d'importance à certains utilisateurs en leur confiant plusieurs parts du secret.**

1.3 Caractéristiques des schémas de partage de secret :

On rappelle que dans les protocoles partagés, afin de casser le cryptosystème l'adversaire doit obtenir les t parts du secret (on a une t-consistance des parts). On parle alors de sécurité inconditionnelle, si l'intégrité mentionnée dans l'objectif ci-dessus (dans la sous-section 1.2.2) est atteinte, car la confidentialité est garantie même si l'adversaire est doté d'une puissance de calcul illimitée [Kaced 2012].

Définition 4. *(Une t-consistance des parts)[Benaloh 1986]. Un ensemble de n parts $s_1, s_2, \cdots, s_n$ est t-consistant si tout sous-ensemble contenant t parts définit le même secret.*

Définition 5. *(Une forte t-consistance des parts)[Harn 2010]. Un ensemble de n parts $s_1, s_2, \cdots, s_n$ est fort t-consistant si :*

a) Tout sous-ensemble contenant t parts ou plus définit le même secret.

b) Tout t-1 ou moins de parts ne peut définir le même secret.

Si la connaissance de moins de t parts n'apporte aucune information sur le secret, le schéma de partage est dit parfait, et le taux d'information obtenu pourra être calculé par :

$$Taux\ d'information = \frac{taille\ en\ bits\ du\ secret}{taille\ en\ bits\ d'une\ part} \tag{1.4}$$

Si le taux d'information est égal à 1, c'est-à-dire que si la taille d'une part est identique à la taille du secret à partager, le schéma est dit idéal.

Définition 6. *(Idéal)[Renner 2014]. On dit qu'un schéma de partage de secret associé à l'ensemble $E = \{(s, v_s) : s \in \mathbb{S}, v_s \in \mathbb{E}_s \subseteq S_1 \times \cdots \times S_n\}$ est idéal si :*

$$la\ taille\ S = la\ taille\ S_1 = \cdots = la\ taille\ S_n. \tag{1.5}$$

Selon l'aptitude du partage de secret on peut avoir trois classes comme suite [Sandhya Sarma 2013] :

- ***Partage de secret proactif :*** *en 1991, Ostrovsky et Yung [Ostrovsky 1991] ont proposé la sécurité proactive. Ce concept a été appliqué sur le partage de secret par Hezberg et al. en 1995 [Herzberg 1995]. Dans cette méthode, on ne considère pas les anciennes parts sauf les nouvelles ce qui permet de mettre à jour périodiquement les parts du secret.*
- ***Partage de secret dynamique :*** *dans un tel partage, on a La possibilité de changer la structure d'accès. Le concessionnaire a la capacité de changer une structure d'accès particulière sur un ensemble donné et/ou permettre aux participants de reconstituer les différents secrets (dans des instants de temps différents).*
- ***Partage de secret avec Veto :*** *C'est la capacité de bloquer la reconstruction. Dans ce type de partage un ensemble qualifié peut empêcher tout autre ensemble de participants à reconstruire la clé secrète.*

1.4 Les domaines d'application :

Les schémas de partage de secret à seuil trouvent plusieurs applications dans les systèmes quotidiens, on peut citer par exemple ce qui suit :

- Le recouvrement de clés ou le stockage des clés privées dans des crypto-systèmes à clés publiques : en partageant la clé entre t personnes ou en la plaçant dans t endroits au lieu de la confier à une seule personne ou dans un seul endroit, pour que la trahison de moins de t personnes ne permet pas de reconstituer la clé privée, et que la destruction de quelques endroits de stockage n'entraîne pas forcément la perte de la clé.
- La protection des coffres forts des banques : nous pouvons citer l'exemple d'une banque à trois responsables, d'où on estime qu'un responsable ne peut ouvrir à lui seul le coffre mais qu'il faut deux des trois responsables pour ouvrir le coffre ensemble notamment si le troisième est indisponible, ceci afin d'éviter toute corruption, ce qui représente un schéma (2,3) de partage de secret à seuil 2.
- Les décisions militaires, et dans le déclenchement des armes exceptionnelles : par exemple au début des années 1990, la Russie à mis en œuvre un système de partage de secret à seuil, où pour utiliser l'arme nucléaire, il fallait la participation de deux personnes parmi le président, le ministre de la défense et le chef des armées, ce qui mène aussi à un schéma à seuil (2,3).
- Le vote électronique (E-voting) : où après avoir récupéré tous les résultats du niveau local pour connaitre le résultat régional, et tous les résultats du niveau régional pour trouver le résultat du vote au niveau national, la machine comptable chargée de transmettre le résultat final nécessite une clé qui doit être récupérée par les personnes concernées afin de découvrir le résultat (le secret) dans le but d'empêcher à toute personne de découvrir le résultat du vote avant les autres [Fouque 2001].
- La loterie électronique : en supposant que la loterie est la personne de confiance (dont le processus est surveillé par un juge arbitre), qui génère des nombres aléatoires (selon le nombre de participants), pour que chaque participant calcule sa valeur du secret selon une fonction bien définie, et l'envoie à la loterie qui de son coté désigne le ticket gagnant (le secret) selon la fonction de calcule utilisée [Fouque 2001].
- Le calcul distribué ou calcul multi-parties : qui concerne l'étude générale des calculs sécurisés entre plusieurs entités en décrivant des protocoles nécessitant des interactions intensives entre les parties.
- Les ventes aux enchères en ligne sécurisés.

— Dans le stockage d'information distribuée.

— ...

1.5 Construction d'un schéma de partage de secret :

Dans cette partie, nous décrivons dans un premier temps la méthode naïve de partage de secret ainsi que son inconvénient, puis nous aborderons les premiers schémas de partage de secret apparus de Shamir [Shamir 1979], Blackley [Blakley 1979], ainsi que de Asmuth et Blum [Asmuth 1983].

1.5.1 La méthode naïve

Supposant que la clé soit : « PASSWORD »
La méthode la plus simple peut être de découper cette clé en autant de morceaux que de personnes doivent la partager. Ainsi, si quatre personnes doivent se réunir pour récupérer la clé, on donnera à chaque personne un morceau de la clé et la position de ce morceau :

1. La personne n°1 obtiendra « PA—— ».
2. La personne n°2 obtiendra « –SS—- ».
3. La personne n°3 obtiendra « —-WO– ».
4. La personne n°3 obtiendra « ——RD ».

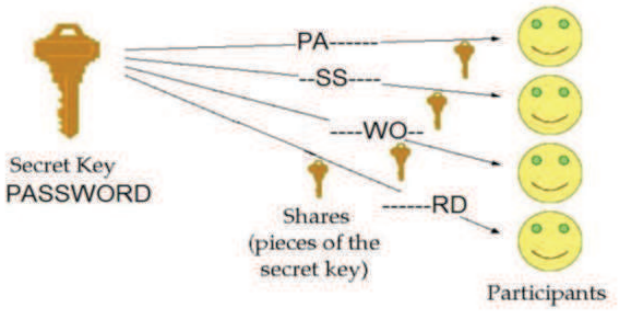


FIGURE 1.3 – Partage de secret naïf.

✤ L'inconvénient : Cependant, cette méthode n'est pas sûre. En effet, ceux qui possèdent un morceau de la clé ont beaucoup plus de chances que les autres de retrouver la clé : sur une clé de 8 caractères, ils n'ont que 6 caractères à trouver (puisqu'ils connaissent déjà leurs deux propres caractères), alors que ceux qui ne possède pas un morceau de la clé doivent trouver les 8 caractères. Si la clé

ne contient que des lettres de l'alphabet, pour essayer toutes les possibilités, trouver 8 caractères demande $26^8 = 208\ 827\ 064\ 567$ essais ; trouver 6 caractères ne demande que $26^6 = 308\ 915\ 776$ essais, c'est-à-dire quand même $208\ 518\ 148\ 791$ fois de moins ! Pire encore, si la personne n°3 persuade ou force la personne n°2 de lui dire sa partie de clef, elle connaîtra alors 4 caractères sur 8 et n'aura donc que $26^4 = 456\ 976$ essais à faire, au maximum, pour découvrir la clé complète. Une machine le fait très rapidement.

1.5.2 Le partage de secret de Shamir

En 1979, Adi Shamir (l'un des inventeurs de RSA) [Shamir 1979] a proposé un schéma de partage de secret à seuil reposant sur l'interpolation polynomial basé sur le lemme suivant :

Lemma 1.5.1. *Soient $l \leq t$ deux entiers positifs, p un nombre premier et $(a_i, b_i) \in \mathbb{Z}/pZ, 1 \leq i \leq l$ des couples tels que les a_i sont distincts et non nuls. Alors il y a exactement p^{t-l} polynômes $f \in \mathbb{Z}/pZ[X]$ de degré au plus $t-1$ tel que $f(a_i) = b_i$.*

La formule d'interpolation de Lagrange permet de calculer un polynôme f passant par un ensemble de points donnés $f_0, f_1, \cdots, f_{t-1}$ comme suite :

$$f(X) = f_0 + f_1 X + \cdots + f_{t-1} X^{t-1} \tag{1.6}$$

Soit F un corps fini et $F[X]$ l'anneau des polynômes à coefficients dans le corps F et à valeurs dans F. On représente par $F_{n-1}[X]$ l'ensemble des polynômes de $F[X]$ de degré $n-1$.

Theorem 1.5.2. *Il existe un seul polynôme $f \in \mathbb{F}_{n-1}[X]$ prenant les valeurs données dans $F : b_1, \cdots, b_n$ aux points deux à deux distincts $a_1, \cdots, a_n$ de F, tel que $f(a_i) = b_i$ pour $i = 1, \cdots, n$.*

Démonstration. Pour prouver l'unicité, on considère deux polynômes f_1 et f_2 de degré au plus $n-1$ qui décrivent la suite de points $\{(a_i, b_i) : i \in \{1, \cdots, n\}\}$. On dit qu'un polynôme p décrit une suite $\{(a_i, b_i) : i \in \{1, \cdots, n\}\}$ si $b_i = p(a_i)$ pour tout $i \in \{1, \cdots, n\}$. Dans ce cas, le polynôme $f_1 - f_2$ est de degré $n-1$ et s'annule en n valeurs. Ce polynôme est donc le polynôme nul et par conséquent $f_1 = f_2$.
Pour prouver l'existence, construisons un polynôme f de degré $n-1$ vérifiant $f(a_i) = b_i$ pour tout $i \in \{1, \cdots, n\}$. Considérons pour tout i, le polynôme $f_i(X) = \prod_{j \neq i} \frac{X - a_j}{a_i - a_j}$ de $F_{(n-1)}[X]$. Il prend la valeur 1 en a_i et 0 en a_j pour tout $j \neq i$. Alors, $f(X) = \sum_{i=1}^{n} b_i f_i(X)$ appartient à $F_{n-1}[X]$ et pour tout $i = 1, \cdots, n, f(a_i) = b_i$. La formule

$$f(X) = \sum_{i=1}^{n} b_i \frac{\prod_{j \neq i}(X - a_j)}{\prod_{j \neq i}(a_i - a_j)}, \tag{1.7}$$

s'appelle la formule d'interpolation de Lagrange. Ainsi, avec n valeurs, on peut reconstruire un polynôme de degré $n-1$ et déterminer la valeur de ce polynôme en n'importe quel point. □

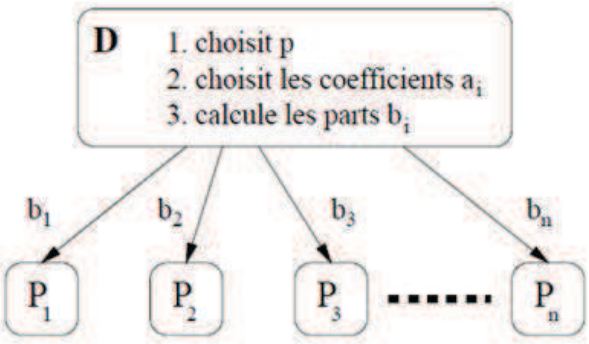


FIGURE 1.4 – Partage de secret à la Shamir.

Le schéma proposé par Shamir est décomposé en trois phases : (1) la phase d'initialisation, (2) la phase de génération des parts ou bien la phase de partage, et (3) la dernière phase qui est la phase de reconstruction du secret partagé, le schéma est décrit comme suite :

a. **La phase d'initialisation : Le distributeur D détermine un nombre premier p** $(p > n+1)$ **et code le secret comme un élément s de** Z_p**. D choisit n éléments distincts non nuls** a_i **de** Z_p **pour** $1 \leq i \leq n$**, qu'il donne à chaque participant** P_i **(identifiant public).**

b. **La phase de partage : D veut partager le secret** $s \in \mathbb{Z}_p$ **entre n participants. Il choisit secrètement (au hasard)** $t-1$ **éléments indépendants dans** Z_p **notés** $f_1, f_2, \cdots, f_{t-1}$**. Pour tout i,** $1 \leq i \leq n$**, D calcule** $b_i = f(a_i)$ **où le polynôme** f **est défini par :**

$$f(x) = s + \sum_{j=1}^{t-1} f_j x^j \ mod\ p \tag{1.8}$$

puis transmet secrètement à chaque participant P_i **la part** b_i**.**

c. **La phase de reconstruction : En mettant en commun les** t **parts de chaque participant autorisé à reconstruire le secret, un polynôme unique de degré** $t-1$ **pourra être reconstruit à partir de** t **points distincts (les t parts) à l'aide de la formule d'interpolation de Lagrange comme suite :**

$$f(x) = \sum_{i=1}^{t} b_i \frac{\prod_{j \neq i}(x - a_j)}{\prod_{j \neq i}(a_i - a_j)}[p] \tag{1.9}$$

Pour obtenir la valeur de f en 0, la formule devient :

$$f(0) = \sum_{i=1}^{t} b_i \frac{\prod_{j \neq i}(a_j)}{\prod_{j \neq i}(a_i - a_j)}[p] \tag{1.10}$$

Supposons que les participants $P_{i_1}, \cdots, P_{i_t}$ souhaitent reconstruire le secret. Ils savent que $b_{i_j} = f(a_{i_j})$ pour $1 \leq j \leq t$ où $f \in \mathbb{F}[X]$ est le polynôme secret choisi par D. Le polynôme f est de degré au plus $t-1$, $f(X) = f_0 + f_1 X + \cdots + f_{t-1}X^{t-1}$ où $f_1, \cdots, f_{t-1} \in \mathbb{F}$ sont inconnus et $f_0 = s$. Comme $b_{i_j} = f(a_{i_j})$ pour $1 \leq j \leq t$, on obtient t équations linéaires aux t inconnues $f_0, \cdots, f_{t-1}$. Si les équations sont indépendantes, il y a une solution unique et on obtient la clé s. Il est important que le système des t équations linéaires admet une unique solution. On peut écrire le système d'équations linéaires sous la forme :

$$\begin{cases} f_0 + f_1 a_{i_1} + \cdots + f_{t-1} a_{i_1}^{t-1} = b_{i_1} \\ f_0 + f_1 a_{i_2} + \cdots + f_{t-1} a_{i_2}^{t-1} = b_{i_2} \\ \qquad\qquad\qquad\qquad\qquad\qquad \vdots \\ f_0 + f_1 a_{i_t} + \cdots + f_{t-1} a_{i_t}^{t-1} = b_{i_t} \end{cases} \tag{1.11}$$

Ou sous forme matricielle $A.F = B$, où La matrice A est une matrice de Vandermonde (voir ci-dessous).

$$\begin{pmatrix} 1 & a_{i_1} & a_{i_1}^2 & \cdots & a_{i_1}^{t-1} \\ 1 & a_{i_2} & a_{i_2}^2 & \cdots & a_{i_2}^{t-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & a_{i_t} & a_{i_t}^2 & \cdots & a_{i_t}^{t-1} \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_{t-1} \end{pmatrix} = \begin{pmatrix} b_{i_1} \\ b_{i_2} \\ \vdots \\ b_{i_t} \end{pmatrix} \tag{1.12}$$

Ci-dessous, un exemple est donné afin d'illustrer l'idée de base du schéma de Shamir :
Supposons que notre secret est **1234** ($S = 1234$). Nous tenons à partager le secret en **6** parties ($n = 6$), où une réunion quelconque de **3** parties ($t = 3$) suffit pour reconstruire le secret. Au hasard, on obtient **2** numéros : **166**, **94** ($f_0 = 166; f_1 = 94$). Le polynôme pour produire les clés est donc :

$$f(x) = 1234 + 166x + 94x^2 \tag{1.13}$$

Nous avons construit **6** points à l'aide du polynôme :

$$(1, 1494); (2, 1942); (3, 2578); (4, 3402); (5, 4414); (6, 5614) \tag{1.14}$$

Nous donnons à chaque participant **i** ($1 \leq i \leq 6$) un point différent (à la fois x et $f(x)$ qui ne sont que a_i et b_i).
Afin de reconstituer le secret, **3** points seront suffisants. Par exemple

$$(a_0, b_0) = (2, 1942); (a_1, b_1) = (4, 3402); (a_2, b_2) = (5, 4414)$$

Le polynôme de Lagrange associé s'écrit : $f(x) = \sum_{j=0}^{2} b_j l_j(x)$, où les l_j sont les polynômes de base de Lagrange :

$$\begin{aligned} l_0 =& \frac{x-a_1}{a_0-a_1}\frac{x-a_2}{a_0-a_2} = \frac{x-4}{2-4}\frac{x-5}{2-5} = \frac{1}{6}x^2 - \frac{3}{2}x + \frac{10}{3} \\ l_1 =& \frac{x-a_0}{a_1-a_0}\frac{x-a_2}{a_1-a_2} = \frac{x-2}{4-2}\frac{x-5}{4-5} = -\frac{1}{2}x^2 + \frac{7}{2}x - 5 \\ l_2 =& \frac{x-a_0}{a_2-a_0}\frac{x-a_1}{a_2-a_1} = \frac{x-2}{5-2}\frac{x-4}{5-4} = \frac{1}{3}x^2 - 2x + \frac{8}{3} \end{aligned} \tag{1.15}$$

Par conséquent :

$$\begin{aligned} f(x) =& 1942(\frac{1}{6}x^2 - \frac{3}{2}x + \frac{10}{3}) + 3402(-\frac{1}{2}x^2 + \frac{7}{2}x - 5) + 4414(\frac{1}{3}x^2 - 2x + \frac{8}{3}) \\ =& 1234 + 166x + 94x^2 \end{aligned} \tag{1.16}$$

Rappelons que le secret est le premier coefficient, ce qui signifie que $S = 1234$.

Aussi, on peut utiliser la géométrie élémentaire pour définir le schéma de Shamir, la figure 1.5 montre un exemple de ceci, où on a un schéma à seuil (2,3), dans lequel on partage un secret entre 3 participants en offrant à chacun d'eux une part donnée du secret, et où 2 participant parmi les 3 suffisent à reconstruire le secret en réunissant leurs parts attribuées du secret (puisque 2 point suffisent à définir une droite), le secret est donné par le point d'intersection de la droite publique et la droite secret définie par les points représentant les parts des participants.

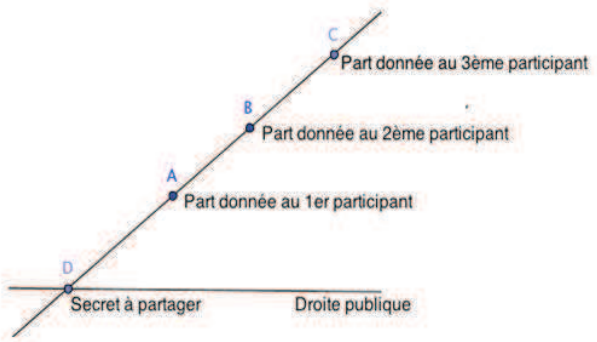


FIGURE 1.5 – Schéma de Shamir utilisant la géométrie.

1.5.3 Le partage de secret de Blakley

En même année de 1979, George Blakley [Blakley 1979] a proposé de son coté et indépendamment à Shamir [Shamir 1979] un autre schéma de partage de secret à seuil basé sur la géométrie des hyperplans sur

les corps finis, où un système linéaire à n équations et t inconnues est constitué, et dont la seule solution est la donnée secrète. Un hyperplan dans un espace t-dimensionnel à coordonnées dans un corps F peut être décrit par une équation de la forme suivante :

$$a_1x_1 + a_2x_2 + \cdots + a_tx_t = y \tag{1.17}$$

Le secret est représenté par un point d'un espace t-dimensionnel, et les n parts secrètes par des hyperplans affines passant par ce point.

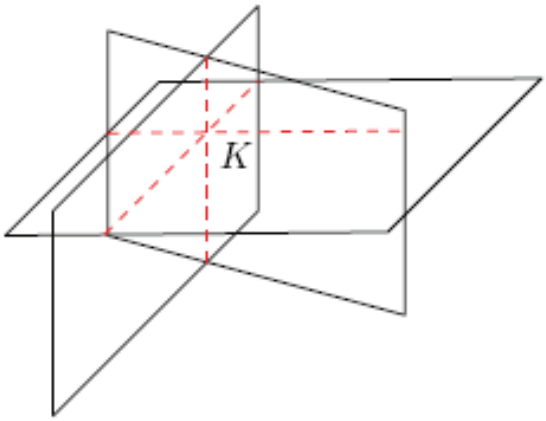


FIGURE 1.6 – Partage de secret de Blakley.

Les figures 1.6 et 1.7 montrent un exemple du schéma de partage de secret de Blakley, où chaque partage de secret est un plan, deux partages se croisent en une ligne d'intersection dont le point représentant le secret lui appartient, et l'intersection avec le troisième partage permet de définir le secret (le point d'intersection entre les trois plans).

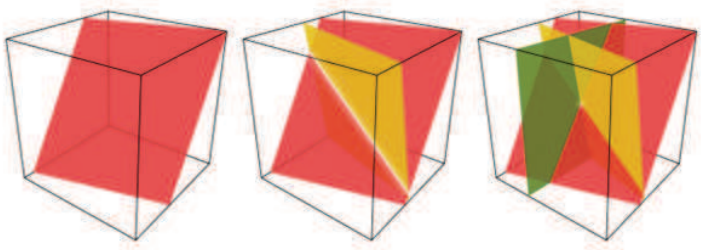


FIGURE 1.7 – Exemple d'un schèma de partage de secret à seuil selon Blakley.

Dans le cas précédent, 3 parts sont nécessaires pour reconstruire le secret (schéma à seuil 3), on peut se trouver dans un cas où seulement deux parts sont nécessaires (schéma à seuil 2) pour reconstruire le secret (représenté par un point dans le plan). L'intersection de deux parts où chacune est représentée par une ligne passant par ce point donne le secret,$\cdots$, plusieurs d'autres schémas existent selon le seuil pris.
Le schéma de Blakley n'est pas parfait, puisque toute personne possédant une part du secret sait que le secret est un point de son hyperplan ; Il est aussi moins efficace que celui de Shamir, car les parts sont t fois plus grandes que le secret, contrairement à l'approche de Shamir qui préserve la taille du secret original.

1.5.4 Le partage de secret de Asmuth-Bloom et Mignotte

En 1983, Asmuth-Bloom [Asmuth 1983] et Mignotte [Mignotte 1982] ont proposé un autre schéma de partage de secret à seuil qui utilise le théorème des restes chinois (CRT), dans cette approche les parts sont générées en réduisant le secret modulo un ensemble de nombres premiers $m_1, m_2, \cdots, m_n$, quand à la reconstruction, elle peut être réalisée en résolvant essentiellement le système de t congruences utilisant CRT. Dans cette approche, le secret S est considéré comme un entier grand. Afin de générer les différentes parts, le distributeur opère comme suite :

1. Choisit un ensemble de nombres premiers $m_0 < m_1 < \cdots < m_n$, de telle sorte que l'équation suivante soit vérifiée :

$$\prod_{i=1}^{t} m_i > m_0^2 \prod_{i=1}^{t-1} m_{n-i+1} \tag{1.18}$$

2. Soit M définie par $M = \prod_{i=1}^{t} m_i$, le distributeur génère un nombre positif aléatoire A et calcule $y = S + A * m_0$, avec $0 \leq y \leq M$.
3. La part du $i^{\text{ème}}$ utilisateur est calculée par $S_i = y \bmod m_i$, pour $1 \leq i \leq n$.

Quand un ensemble C d'aux moins t participants décide de reconstruire le secret, le distributeur ou la personne de confiance de l'ensemble C effectue les étapes suivantes :

1. La valeur $M_C = \prod_{i \in \mathbb{C}} m_i$ est calculée en premier, avec un ensemble de t valeurs différentes $M_{C/\{i\}} = \prod_{j \in \mathbb{C}, j \neq i} m_j$ et leurs inverses multiplicatifs correspondant $M_{C/\{i\}}^{-1}$ dans le corps Z_{m_i} (i.e. $M_{C/\{i\}}.M_{C/\{i\}}^{-1} \equiv 1(mod\ m_i)$) ;
2. Pour un participant i, le distributeur calcule la valeur $u_i = S_i.M_{C/\{i\}}.M_{C/\{i\}}^{-1}(\text{mod}M_C)$;
3. Le distributeur calcule après $y = (\sum_{i \in \mathbb{C}} u_i)(mod\ M_C)$, et déduit le secret S en calculant $S = y\ mod\ m_0$.

L'ensemble $\{m_0, m_1, \cdots, m_n\}$ est considéré comme une information publique qui doit être acceptée par quiconque sans compromettre la sécurité du schéma (i.e. en révélant aucune information utile sur le secret partagé).
Ci-dessous, un exemple du CRT est donné :
Pour $n = 3$, supposons que $m_1 = 7$, $m_2 = 11$ et $m_3 = 13$. On a donc $M = 1001$.
On calcule $M_1 = 143, M_2 = 91, M_3 = 77$. D'après l'algorithme d'Euclide étendu, on a $M_1^{-1}\ mod\ m_1 = 5, M_2^{-1}\ mod\ m_2 = 4$ et $M_3^{-1}\ mod\ m_3 = 12$.
Par exemple, pour le système d'équations :

$$\begin{cases} x\ mod\ 7 = 5 \\ x\ mod\ 11 = 3 \\ x\ mod\ 13 = 10 \end{cases} \tag{1.19}$$

La formule donne pour solution :

$$\begin{aligned} x =& (5 \cdot 143 \cdot 5 + 3 \cdot 91 \cdot 4 + 10 \cdot 77 \cdot 12)\ mod\ 1001 \\ x =& 13\ 907\ mod\ 1001 \\ x =& 894 \end{aligned} \tag{1.20}$$

On peut facilement vérifier cette solution en introduisant x dans le système d'équations.

- Remarque : Les trois approches proposées respectivement par Shamir, Blakley et par Asmuth-Bloom / Mignotte sont inconditionnellement sécurisées, et offrent un niveau de sécurité élevé avec une moyenne de calcul acceptable lors d'un partage de secret de petite taille et de longueur fixe. Alors que dans le cas d'un partage de secret de taille grande plusieurs problèmes se posent. Notamment, pour le partage d'images numériques qui nécessite des schémas adaptés en raison de leurs caractéristiques particulières, tels que la capacité de données volumineuse et la forte corrélation entre les pixels.

1.6 Propriétés additionnelles des schémas de partage de secret :

On peut ajouter des fonctionnalités aux schémas de partage de secret comme :

1.6.1 La vérifiabilité :

Lorsqu'une situation de partage de secret se présente, il faut pouvoir permettre aux participants du groupe de vérifier que le secret a bien été distribué de façon cohérente, et de vérifier la validité des parts en

s'assurant qu'elles proviennent bien de l'initiateur (dans le partage de secret standard l'initiateur est supposé être honnête), un tel schéma répondant aux deux problèmes est appelé schéma de partage de secret vérifiable, ce type de schéma a été introduit par Chor et al dans [Chor 1985].

1.6.2 La détection de tricheur :

Dans un schéma de partage de secret à seuil, si une personne parmi celles désirants reconstruire le secret, triche et fournit une mauvaise part au lieu de donner la bonne part qui lui a été attribuée, cette personne aura la possibilité de s'intégrer au secret ainsi de découvrir les parts des autres participants, aussi la valeur obtenue par le groupe ne représentera pas le secret, donc on doit être en mesure de détecter le tricheur qui empêche le décodage du secret ; en 1990, Brickell et Stinson [Brickell 1990] ont proposé une solution afin de détecter les tricheurs dans un schéma à seuil, et cela en distribuant de l'information supplémentaire aux participants.

1.7 État de l'art :

Diverses techniques ont été proposées dans la littérature pour le partage sécurisé de données, mais les travaux liés au partage des images secrètes ne sont pas nombreux. Le partage d'images secrètes reste un cas particulier qui impose des contraintes supplémentaires à prendre en considération. Nous décrirons dans ce qui suit quelques solutions existantes qui exploitent les schémas classiques de partage de secret, en l'occurrence celui de Shamir [Shamir 1979], de Blakely [Blakley 1979] et de Asmuth et Bloom [Asmuth 1983] pour résoudre le problème de partage des images numériques.

1.7.1 Approche de C. Chen et W. Fu, 2008 [Chen 2008] :

Cette approche est basée sur la géométrie de Blackley [Blakley 1979], où les auteurs ont proposé un schéma à seuil (k,n) pour partager une image secrète, l'algorithme de la solution est donné comme suite :

a. Phase de partage : dans cette phase l'image secrète est partitionnée en ensembles de k pixels, pour chaque ensemble de k pixels : un point x de k-dimension est formé $x = (x_1, x_2, \cdots, x_k)$ de telle sorte à sélectionner aléatoirement n ensembles de solution différente $(a_1, a_2, \cdots, a_k, b)$ satisfaisant l'équation $a_1x_1 + a_2x_2 + \cdots + a_kx_k = b$, chaque solution est stockée avec le même nombre de bits dans une part d'image, et ceci jusqu'à avoir traité tous les ensembles de k pixels de l'image secrète.

b. Phase de reconstruction : un ensemble de k+1 paramètres est extrait de chaque part d'image appartenant à un participant parmi les

k désirant reconstruire le secret, à partir de ces paramètres, les k hyperplans sont construits et leur intersection donne un point de k-dimension qui va être stocké dans l'image, l'opération est répétée jusqu'à avoir traité tous les ensembles de k+1 paramètres des k parts d'image, ce qui permettra de reconstruire l'image secrète totalement.

1.7.2 Approche de Harn et Lin, 2010 [Harn 2010] :

Dans cette approche, les auteurs ont proposé un schéma (n,t,n) de partage multi-secret à seuil t, où le nombre de secret à partager dépend du nombre de participants (n) dans le groupe, le schéma proposé repose sur la proposition de Pedersen [Pedersen 1991], l'algorithme de la solution est donné ci-dessous :

a. Phase de génération du secret maitre : Chaque participant P_i, joue le rôle du « dealer », et sélectionne aléatoirement un sous-polynôme $f_i(x)$ de degré $t-1$, où le sous-secret $S_i = f_i(0)$, le secret maitre S est calculé par la somme des sous-secrets S_i de chaque participant ($S = \sum_{i=1}^{n} S_i$).

b. Phase de génération des parts maitres :
   - Suivant la procédure de Shamir [Shamir 1979], chaque participant P_i calcule les sous-parts, $S_{(t,n)}(f_i(x)) = (s_{i,1}, s_{i,2}, \cdots, s_{i,n})$.
   - Chaque participant P_i transmet secrètement la sous-part $s_{i,j}$ aux autres participants P_j, pour $j = 1, 2, \cdots, n$ et $j \neq i$.
   - Chaque participant P_i calcule sa part maitre par $m_i = \sum_{j=1}^{n} s_{j,i}$.

c. Phase de reconstruction du secret maitre : comme le schéma est à seuil t, tout sous-ensemble de t participants $\{i_1, i_2, \cdots, i_t\}$ parmi les n peut reconstruire le secret maitre, en réunissant leurs parts maitres $(m_{i_1}, m_{i_2}, \cdots, m_{i_t})$, et ceci en utilisant la formule d'interpolation de Lagrange, le secret maitre S est obtenu par $S = F(0) = \sum_{i=1}^{n} S_i$.

1.7.3 Approche de Y. Liu et al, 2012 [Liu 2012] :

Y. Liu et al, ont été inspiré par la solution précédente, et ont proposé un schéma de partage multi-secret (n,t,n), le schéma est décomposé en trois phase :

a. Phase de génération des secrets :
   - Chaque participant P_i sélectionne un sous-polynôme secret aléatoire $f_i(x)$ de degré $t-1$, avec le sous-secret $S_i = f_i(0)$.
   - Les participants se réunissent pour sélectionner un ensemble de n-uplet de vecteurs e_i, pour $i = 1, \cdots, (n-t+1)$, où $e_i = (e_{i,1}, e_{i,2}, \cdots, e_{i,n})$, doit vérifier que tous $(n-t+1)$-uplet d'éléments dans $e_l (l \in [1, n-t+1])$ doivent être linéairement indépendants.

- Les $(n-t+1)$ secrets maitres sont calculés par $M_l = \sum_{i=1}^{n} e_{l,i} S_i$, $l = 1, 2, \cdots, (n-t+1)$

b. Phase de génération des parts maitres :

- Chaque participant P_i calcule les sous-parts $S_{t,n}(f_i(x)) = (s_{i,1}, s_{i,2}, \cdots, s_{i,n})$.
- Chaque participant P_i envoie secrètement une sous-part $s_{i,j}$ aux autres P_j, où $j \neq i$.
- Pour chaque secret maitre M_l, P_i calcule la part maitre à partir des n sous-parts : $m_{i,l} = \sum_{j=1}^{n} e_{l,j} s_{j,i}$.

c. Phase de reconstruction des secrets maitres : chaque t participants combinent leurs parts maitres $(m_{i_1,l}, m_{i_2,l}, \cdots, m_{i_t,l})$ pour reconstruire les $(n-t+1)$ secrets $M_l, l = 1, 2, \cdots, (n-t+1)$,en utilisant la formule d'interpolation de Lagrange.

1.7.4 Approche de Subba Rao Y V et C. Bhagvati, 2014 [Bhagvati 2014] :

Le schéma de partage (m,t,n) proposé dans leur solution exploite la solution de Asmuth et Bloom [Asmuth 1983], et utilise le théorème des restes chinois (CRT) pour partager leurs secrets comme dans ce qui suit :

a. Phase d'initialisation : « le dealer » utilise z matrices binaires A_i, pour chaque groupe de participants $G_i, i = 1, \cdots, z$, chacune d'ordre $n_i \times r_i$, où n représente le nombre de participants, et r les parties de la clé $K, k = m_{i,j} : 1 \leq i \leq z \ et \ 1 \leq j \leq r$, les éléments $a_{i,j}$ de la matrice A sont à 1, si la $j^{\text{ème}}$ partie de la clé est utilisé par le participant i ; 0 sinon.

b. Partage de secret : le secret S_i, est choisi de 1 à M_i, où $M_i = \sum_{k=1}^{r_i} m_k$ pour chaque groupe G_i ; « le dealer » utilise ensuite CRT pour calculer et diffuser X qui satisfait les congruences $X = S_i (mod\ M_i)$.

c. Reconstruction du secret : chaque t_i membres d'un groupe G_i, peuvent reconstruire leurs secrets S_i avec les éléments de la clé k_i en utilisant CRT.

1.7.5 Approche de A. Cheraghi, 2014 [Cheraghi 2014] :

A. Cheraghi a proposé deux schémas de partage multi-secret à seuil, dont voici ci-dessous la description de chacun d'eux :

✤ Schéma 1 : son premier schéma est décrit comme suite :

a. Phase d'initialisation :

- « Le dealer » choisit indépendamment t éléments secrets $a_0, a_1, \cdots, a_{t-1}$.

- Calcule les m secrets $S_i = f(i)$, $i = 1, 2, \cdots, m$, où $f(x) = \sum_{j=0}^{t-1} a_j x^j \bmod q$.
- Choisit n éléments différents publiques x_i, $(x_i \geq m)$, et attribue chaque valeur x_i à un participant P_i, $i = 1, 2, \cdots, n$.

b. Phase de distribution :

- « Le dealer » calcule $y_i = f(x_i)$,et donne secrètement la part y_i à chaque participant P_i, $i = 1, 2, \cdots, n$.

c. Phase de reconstruction : les m secrets sont reconstruits avec la formule d'interpolation de Lagrange après résolution du système de t équations linéaires fourni par les t participants.

✤ Schéma 2 : le deuxième schéma repose sur l'utilisation d'un chiffrement en blocs (chiffrement à rétroaction) opérant sur le mode CFB (Cipher FeedBack), où les éléments sont donnés dans un corps fini de Galois GF(q), avec q un grand nombre premier ; la figure 1.8 montre son mode de fonctionnement :

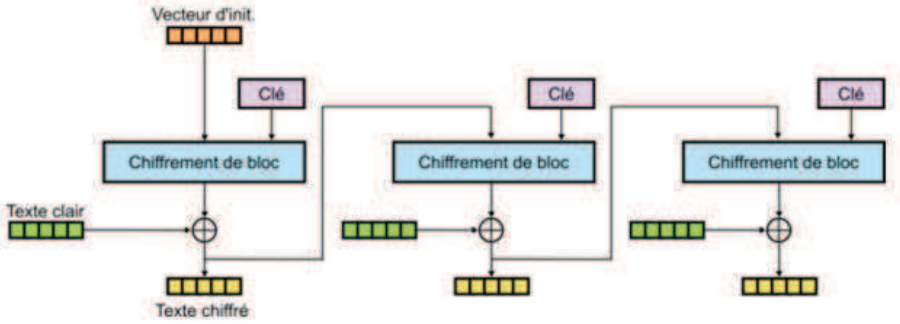


FIGURE 1.8 – Mode de chiffrement CFB.

Le schéma proposé est composé de trois phases comme suite :

a. Phase d'initialisation : « le dealer » génère et publie m cryptogrammes $c_1, c_2, \cdots, c_m$, et cela en utilisant le mode CFB à partir des deux valeurs privées représentants le vecteur d'initialisation $c_0 = VI$ et la clé k ainsi que des m secrets (correspondants au texte clair dans la figure 1.8).

b. Phase de distribution :

- « Le dealer » choisit $t-2$ éléments $a_2, \cdots, a_{t-1}$.
- Calcule $y_i = f(x_i)$, Pour $i = 1, \cdots, n$, où $f(x) = c_0 + kx + \sum_{j=2}^{t-1} a_j x^j \ modq$.
- Distribue la part (x_i, y_i) pour chaque participant $i = 1, \cdots, n$.

c. Phase de reconstruction : un sous-groupe de t participants peut reconstruire les secrets, et ceci en calculant f(x) en utilisant la formule d'interpolation de Lagrange qui permet de donner les deux valeurs secrètes $c_0 = f(0)$ et k comme coefficients de x dans f(x), et donc de reconstruire les m secrets en inversant le mode CFB à partir de $(c_0, c_1, \cdots, c_m, k)$.

1.8 La problématique :

Quelques solutions de partage de secret ont été décrites parmi plusieurs dans ce chapitre, les solutions vues précédemment sont basées sur différents principes mathématiques, dont l'interpolation polynomiale utilisée par Shamir dans [Shamir 1979], l'intersection d'hyperplans de Blakley dans [Blakley 1979] et le théorème des restes Chinois proposée par Asmuth et Bloom dans [Asmuth 1983]. Ces approches sont inconditionnellement sécurisées, et fournissent un niveau de sécurité élevé avec une moyenne de calcul acceptable lors de partage de secret de petite taille.
Cependant, les images numériques étant largement utilisées, partagées et transmises dans les applications de nos jours, la nécessité de techniques de partage efficaces et adaptées devient cruciale, car les schémas existants s'avèrent difficiles à adapter, ce qui fait que lors de partage de secret de taille grande (le cas des images), plusieurs problèmes se posent notamment si on utilise les schémas classiques mentionnés précédemment, dont la complexité de calcule qu'offrent ces derniers.

1.9 Conclusion :

La transmission secrète des données est une tâche importante pour les préserver des menaces probables lors de la transmission. Le partage de secret reste l'une des méthodes efficaces approprié à la transmission sécurisée de données. La motivation de partage de secret provient de la notion de gestion de clé sécurisée. Dans certaines situations, il y a habituellement une clé secrète qui donne accès à de nombreux fichiers importants. Si une telle clé est perdue (par exemple, la personne qui connaît la clé devient indisponible, ou l'ensemble de l'ordinateur qui stocke la clé est détruit, ...), tous les fichiers importants deviennent inaccessibles. L'idée de base dans le partage de secret est de diviser la clé secrète en morceaux et de distribuer ces morceaux à des personnes différentes dans le groupe de sorte que certains sous-ensembles de personnes peuvent se réunir pour récupérer la clé.
Une recherche bibliographique a été présentée couvrant quelques solutions de partage de secret proposées se basant sur les schémas classiques [Shamir 1979, Blakley 1979, Asmuth 1983] ; néanmoins la difficulté

d′application de ces schémas lors de partage d′images secrètes reste présente pour cause des caractéristiques spécifiques de ces dernières notamment la taille, ainsi de la complexité de calcule qu′offrent les schémas ; dans le but de répondre aux exigences des schémas proposés lors de partage d′images secrètes, une approche à été réalisée dans le dernier chapitre (Chapitre 5), pour faire face aux exigences de sécurité des schémas de partage d′images secrètes.

Partie II

Les automates cellulaires et le partage de secret cryptographique

Chapitre 2

Les Automates Cellulaires

Sommaire

2.1 Introduction :

La problématique majeure dans l'étude des systèmes complexes consiste à comprendre comment un ensemble d'objets interagissant selon des règles locales déterminées peut engendrer un comportement global complexe, difficile à comprendre au simple vu des règles locales.
Parmi les nombreux modèles de systèmes complexes existants, on a les automates cellulaires, ces derniers sont exploités dans de nombreux domaines de recherche (en physique, mathématique, biologie, informatique, $\cdots$). Les automates cellulaires constituent en une grille régulière d'objets identiques appelés « cellules » possédant chacune un « état » choisi parmi un ensemble fini, et qui peut évoluer de manière déterministe et synchrone au cours du temps, fournissant ainsi un modèle remarquable par sa simplicité de définition, et sa complexité de production de certains comportements qui sont difficiles à prévoir dans lesquels le problème transversale issue des systèmes complexes est étudié, et où on considère les automates cellulaires comme un outil très puissant de modélisation de tel système.

2.2 Historique :

L'histoire des automates cellulaires remonte aux années quarante aux deux scientifiques Stanislaw Ulam [Ulam 1952] et John Von Neumann [Von Neumann 1966], qui cherchaient à explorer les manières dont lesquelles une structure artificielle pourrait se comporter comme un être vivant et construire des structures très complexes à partir de règles extrêmement simples.
Stanislaw Ulam étudiait la croissance des cristaux au Laboratoire national de Los Alamos, en la modélisant sur une grille. En parallèle, John Von Neumann, collègue d'Ulam à Los Alamos, s'intéressait à la théorie des automates autoréplicateurs et travaillait à la conception d'une machine autoréplicatrice « le kinématon » dans le but de chercher à décrire un système (un automate) capable de s'autoproduire en produisant des répliques de lui-même, Le risque est important quand on traite de l'autoreproduction de considérer des phénomènes « triviaux » qui sont fréquents (Triangle de Pascal (Figure 2.4), l'automate de Fredkin (Figure 2.16), $\cdots$), afin d'éviter cet écueil Von Neumann a pris le parti de ne considérer que des reproducteurs universels, capables de réaliser toute construction dont la description leur est fournie avec divers instruments d'interprétation et de fabrication pour mieux la compliquer.
Von Neumann rencontrait des difficultés à expliciter son modèle initial d'un robot qui se copierait tout seul à partir d'un ensemble de pièces détachées. C'est pourquoi Ulam lui suggéra de s'inspirer de ses travaux

et d'utiliser ce qu'il appelait « les espaces cellulaires » (cellular spaces) pour construire sa machine autoréplicatrice. Il pouvait ainsi s'affranchir des conditions physiques réelles pour travailler dans un univers extrêmement simplifié pourtant apte à engendrer une haute complexité. Von Neuman a ainsi réussi à concevoir sa machine virtuelle à l'aide d'un automate cellulaire autoréplicateur (200 000 cellules, 29 états) contenant un copieur universel, une description de lui-même et une machine de Turing pour la supervision. De cette conception, seul est resté le terme d'automate cellulaire.

2.3 Qu'est ce qu'un Automate Cellulaire ?

Un automate cellulaire (AC) est un objet mathématique, étudié aussi en informatique théorique, il consiste en une grille régulière de cases appelées « cellules » pouvant prendre dans un temps discret plusieurs états choisis parmi un ensemble fini contenant le plus souvent deux états $\{0, 1\}$ (« mort » ou « vivant ») évoluant par étapes (génération après génération) au cours du temps de manière discrète selon des règles très simples et imitant d'une certaine manière les capacités autoreproductrices des êtres vivants. Les états des cellules sont mis à jour de façon synchrone en utilisant la règle de transition locale qui définit l'état de chaque nouvelle cellule en fonction de son ancien état ainsi que de l'état des cellules voisines correspondantes. À chaque nouvelle unité de temps, les mêmes règles sont appliquées simultanément à toutes les cellules de la grille, produisant une nouvelle génération/configuration de cellules dépendante entièrement de la génération/configuration précédente. Formellement, on a la définition suivante :

Définition 7. *(Automate Cellulaire). Un automate cellulaire A est un quadruplet $A\_(d, \sigma, \delta, V)$ où :*

- *$d \in \mathbb{N}$ est la dimension de l'automate (le graphe sur lequel se trouvent les cellules est alors Z^d) ;*
- *σ est un ensemble fini dont les éléments sont appelés états de l'automate ;*
- *$V \subset Z^d$ est un ensemble fini de vecteurs appelé voisinage. On considère toujours que 0 appartient à V ;*
- *$\delta : \sigma^V \to \phi$ est la fonction de transition locale de l'automate.*

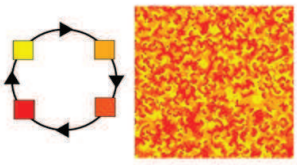


FIGURE 2.1 – Exemple d'un AC à quatre états.

Si la même règle est utilisée pour toutes les cellules alors l'AC résultant est nommé uniforme (homogénéité). Sinon, si une règle de transition différente est utilisée à chaque fois que la cellule change, l'AC est appelé non-uniforme (hétérogénéité).

2.4 Les caractéristiques d'un Automate Cellulaire :

Un AC se caractérise par la différence étonnante entre la simplicité des règles de départ et la complexité des résultats obtenus grâce à lui. Les quatre composantes qui peuvent décrire un AC sont comme suite :

2.4.1 La dimension :

On peut distinguer plusieurs sortes d'AC selon la dimension de ce dernier (avec aucune limite), le plus généralement la dimension est de 1 pour un AC élémentaire, ou 2 pour un AC de type Life, passé 3 dimensions on doit recouvrir à des projections sur R3 ou R2 pour visualiser correctement l'hyper-matrice.

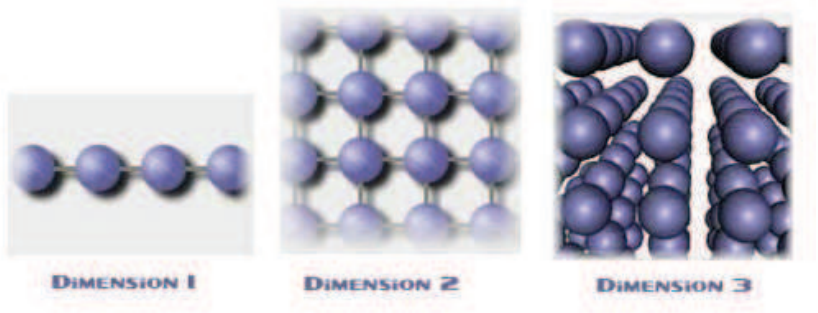


FIGURE 2.2 – Dimensions 1, 2 et 3 d'un AC.

2.4.2 Le voisinage :

Le voisinage d'une cellule définit l'ensemble des cellules qui auront une influence sur la cellule étudiée. En pratique, le voisinage est souvent limité à la cellule cible et aux cellules adjacentes.

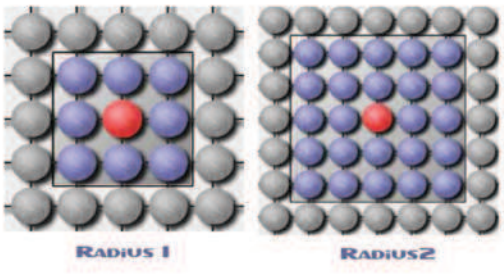


FIGURE 2.3 – Rayons de voisinage d'une cellule.

La géométrie des cellules est également étroitement liée à son voisinage. Une cellule peut être un hyper-cube (fréquemment), mais également un autre hyper-volume ou désorienté (blind neighbourhood) : un voisinage « aveugle » dont lequel chaque cellule lui appartenant joue un rôle identique.
La cellule cible dont on étudie le voisinage a connaissance des différents états des cellules voisines, mais ne connait en aucun cas la correspondance de telle cellule à un tel état.
La plupart des ACs définissent sans le préciser des voisinages « aveugles ».

2.4.3 L'espace d'états :

L'espace d'état d'un AC correspond à l'ensemble des états que peut prendre une cellule. Le plus souvent limité à 2, il n'y a aucune limite théorique. Pour exemple, Von Neumann a étudié mathématiquement un automate à 29 états. Pratiquement, ces états sont représentés par des couleurs, qui permettent de suivre l'évolution de l'automate.
Lors de modélisation du système, les états des cellules correspondent à des états physiques locaux. Par exemple, dans le Jeu de la Vie, une cellule est soit « morte » soit « vivante », mais on pourrait très bien imaginer des états transitoires de dégénérescence d'une cellule, en augmentant le nombre d'états de l'automate.

2.4.4 La fonction de transition :

C'est l'ensemble des règles qui permettent de déterminer le nouvel état d'une cellule en fonction de son état précédent et des états précédents

de l'ensemble du voisinage. Pour un automate à n états avec un voisinage de k cellules, il peut y avoir n^k configurations de voisinage différentes :

- pour $n = 2$ et $k = 3$, $n^k = 8$ voisinages différents (AC élémentaire de wolfram) ;
- pour $n = 2$ et $k = 9$, $n^k = 512$ voisinages différents (jeu de la vie).

De plus, la fonction de transition est créée en associant à chaque voisinage un état de sortie. Il y a donc potentiellement n^{n^k} fonctions de transition pour un automate cellulaire à n états ayant un voisinage de k cellules.
La plupart du temps, ces règles ne sont pas explicitées, mais résumées. Elles sont synthétisées sous la forme de méta-règles (règles du jeu de la vie).

2.5 Les propriétés des automates cellulaires :

Ci-dessous quelques propriétés remarquables d'automate cellulaire dont :

2.5.1 La reproduction :

Le but de Von Neumann, à l'origine, lorsqu'il a conçu le principe des automates cellulaires, était de concevoir un mécanisme copiant la vie dans le sens où il pourrait se reproduire lui même.
Ainsi certains automates cellulaires sont capables de produire des copies d'eux-mêmes, propriété particulièrement intéressante lorsqu'il s'agit de modéliser la vie d'êtres vivants. Ce point de vue de la reproduction permet de classer les automates cellulaires en deux grandes catégories :

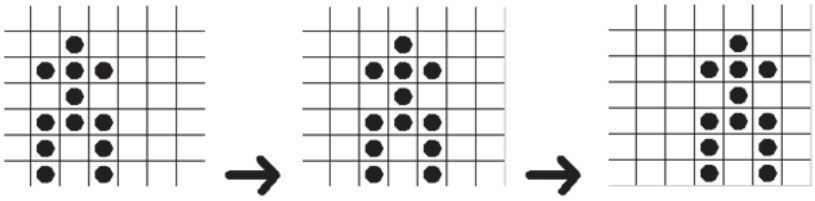
- Les automates cellulaires actifs, c'est à dire auto-reproducteurs.
- Les automates cellulaires passifs.

Les automates cellulaires actifs sont auto-reproducteurs dans le sens où ils contiennent une sous-configuration qui se comporte en « copieur universel » dirigeant activement la réplication via une fonction de transition assurant la réplication.
Les automates cellulaires passifs sont ainsi nommés car leur reproduction est provoquée par la règle de transition et non pas par les caractéristiques de la configuration initiale. Le Jeu de la Vie est un exemple d'automate cellulaire passif.

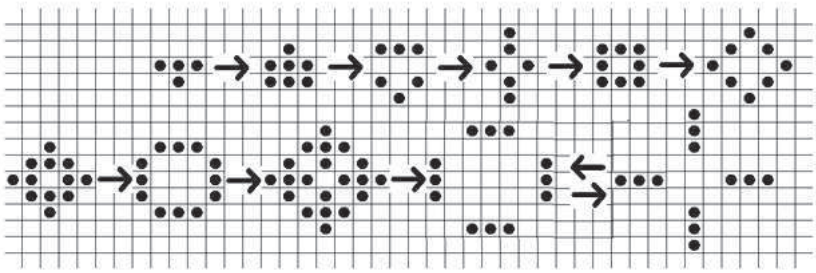
2.5.2 L'inversibilité :

On appelle automate inverse l'automate qui permet de revenir aux états précédents.

Un exemple simple d'automate inverse est celui de l'automate déplacement Est. Chaque case peut avoir deux états, 0 et 1 (vide ou plein). L'automate regarde l'état de la case voisine Ouest, s'en souvient et agit en le prenant pour nouvel état de la case. Un réseau d'automates Déplacement Est sur un plan a pour effet, d'une génération à l'autre, de déplacer d'une case vers l'Est le motif initial. Son inverse est l'automate Déplacement Ouest.



Cet automate possède deux états : 0 et 1, représentés l'un par une case blanche, l'autre par une case noire. D'une génération à l'autre, chaque automate du réseau regarde l'état de son voisin Ouest et le prend pour lui-même. Le résultat est que le dessin se déplace vers l'Est.
On a démontré que tous les automates n'ont pas nécessairement un inverse. Pour démontrer qu'un automate n'a pas d'inverse il suffit de trouver deux configurations différentes qui aboutissent à la même configuration.



Cet automate, utilisant les règles du jeu de la vie, commence par un pentamino et évolue en 11 étapes. Au bout de la onzième l'étape 12 est la même que la dixième. On a donc deux configurations différentes qui donnent la même : l'automate de Conway (qui est celui du jeu de la vie) n'est pas inversible.
La question qui se pose alors est, lorsqu'on a construit un automate, de savoir si celui-ci est inversible. Et bien cette question est un problème indécidable.

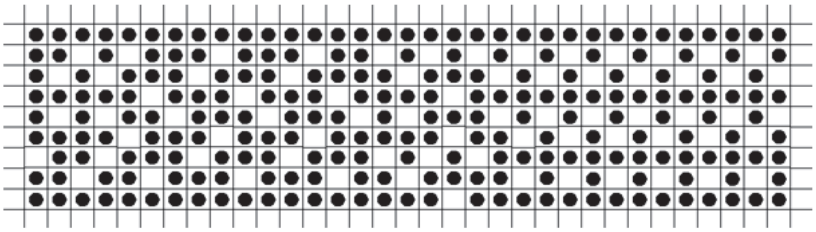
2.5.3 L'indécidabilité :

L'une des caractéristiques importantes des automates cellulaires est le caractère d'indécidabilité qui touche le nombre de leurs propriétés. Ainsi,

déterminer si un automate cellulaire possède un inverse est indécidable : il ne sera jamais possible d'écrire un programme prenant en paramètres un automate quelconque et pouvant décider si oui ou non cet automate possède un inverse.
De la même façon, « l'avenir » d'un automate est indécidable. On n'a pas de méthode générale permettant de déterminer si un automate ne va pas s'éteindre au bout d'un certain nombre de générations ou s'il va se stabiliser

2.5.4 Les Jardins d'Eden :

Un Jardin d'Eden est une configuration qui ne possède pas d'antécédent. Cela ne se produit bien entendu pas avec les automates inversibles. De même, par exemple, une configuration Jardin d'Eden ne peut être un attracteur car elle ne peut apparaître que comme première configuration d'une suite de configurations. L'aspect remarquable de ce concept est que la question de l'existence de Jardins d'Eden est indécidable. Cela a été démontré par J. Kari en 1990.
Un autre résultat intéressant avait déjà été trouvé en 1962 par E. Moore et J. Myhill : un automate possède des Jardins d'Eden si et seulement si deux configurations finies donnent le même résultat. Cette précision pour souligner le fait que cette question a su tenir en haleine les informaticiens, et peut-être plus généralement, les logiciens pendant longtemps.



Cette configuration n'a pas de prédécesseur : c'est un jardin d'Eden.

2.5.5 Les attracteurs :

Les attracteurs des automates sont des configurations qui reviennent indéfiniment. Étant des cas particuliers d'automates, ils permettent des études intéressantes sur les automates et la démonstration de propriétés. Ainsi, J. Kari a démontré que toute propriété de l'ensemble limite (l'ensemble des attracteurs) est vraie pour certains automates et fausses pour d'autres est indécidable. Ce résultat est finalement le plus extraordinaire de tous, car il nous montre que nous ne saurons jamais rien à l'infini des réseaux d'automates cellulaires.

2.6 Les automates cellulaires unidimensionnels :

Un automate cellulaire unidimensionnel est un espace cellulaire formé par un tableau d'une seule dimension 1D (d=1), représentant une configuration du réseau à un instant t ($C^{(t)}$). Une configuration ($C^{(t)}$) d'un AC est constituée de plusieurs cellules, ayant chacune un état $s_i^{(t)}$, l'état de chaque cellule « i » à l'instant t+1 dépend de l'état de la cellule « i » et des cellules voisines à l'instant t, les voisins sont une sélection spécifique des cellules relativement choisies par rapport à la position d'une cellule donnée qui peut être définie pour chaque cellule en utilisant un rayon de voisinage « r », cela donnera « 2r+1 » voisins différents, y compris la cellule elle-même.
Formellement, si nous définissons l'état d'une cellule « i » à l'instant « t » avec s_i^t, son état au temps « t+1 » dépendra seulement sur les états des voisins correspondants à l'instant t, en appliquant une règle de transition qui définit la manière dont les états sont mis à jour. Si le rayon de voisinage est r, et si seulement deux états cellulaires sont définis (0 ou 1), la longueur de chaque règle de transition est alors 2^{2r+1} bit, et le nombre de règles possibles est égal à $2^{2^{2r+1}}$. La règle de transition d'un AC binaire unidimensionnel est généralement codée en utilisant la représentation binaire de la valeur entière correspondante, et les différentes configurations de l'AC sont représentées par des blocs binaires où les chiffres 0 et 1 sont représentés par des figures (le plus souvent des rectangles blancs et noirs), par exemple pour un réseau $\{0,0,0,1,0,0,0\}$ à un instant t on a le dessin suivant qui lui correspond :

L'évolution de l'AC permet de générer de nouvelle configuration à chaque nouvel instant, la séquence $E^{(k)} = \{C^{(t)}\}_{0 \leq t \leq k}$ est appelée l'évolution d'ordre k de l'AC et la superposition de chaque nouvelle configuration avec la précédente constitue ce qu'on appelle un diagramme d'espace de temps.

FIGURE 2.4 – Exemple d'automate à une dimension (Triangle de Pascal).

Une classe simple parmi les classes des automates cellulaires unidimensionnels est celle d'automate cellulaire élémentaire décrite dans la sous-section suivante.

2.6.1 Les automates cellulaires élémentaires :

Un automate cellulaire élémentaire (ACE) est un automate cellulaire unidimensionnel à deux états de cellules $\{0, 1\}$ dont le rayon de voisinage $r = 1$ détermine l'ensemble de voisinage V à trois cellules ($V = 2^{2(1)+1}$), le voisinage est constitué de la cellule concernée et des deux cellules voisines (une à droite et l'autre à gauche). En conséquence, l'évolution d'un automate cellulaire élémentaire peut être complètement décrite par une table indiquant l'état d'une cellule donnée s_i dans la prochaine génération (au temps t+1) sur la base de la valeur de la cellule s_{i-1} se trouvant à sa gauche, la valeur elle-même de la cellule s_i, et la valeur de la cellule de droite s_{i+1} au temps précédent t. Dans un tel automate, on peut avoir 2^V c'est-à-dire $2^3 = 8$ combinaisons binaires possibles de trois états de cellules $\{s_{i-1}, s_i, s_{i+1}\}$ ainsi que $2^8 = 256$ règles possibles $R \in \{0, 255\}$.

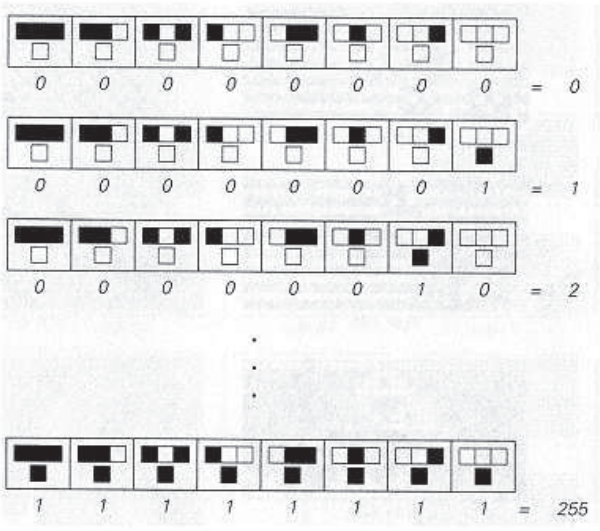


FIGURE 2.5 – Les 256 Règles de transition possibles d'un automate cellulaire élémentaire.

La règle choisie est représentée sous forme binaire dans la colonne correspondante à l'état de la cellule s_i à l'instant t+1 ($s_i^{(t+1)}$). Le tableau qui suit montre un exemple de la règle 30 représentée en binaire par $(00011110)_2$:

$s_{i-1}^{(t)}$	$s_i^{(t)}$	$s_{i+1}^{(t)}$	$s_i^{(t+1)}$
1	1	1	0
1	1	0	0
1	0	1	0
1	0	0	1
0	1	1	1
0	1	0	1
0	0	1	1
0	0	0	0

Tableau 2.1 – La règle de transition 30.

Si une configuration initiale à un instant t est donnée comme suite : $\{\cdots, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, \cdots\}$, l'évolution de l'AC avec la règle 30 décrite dans le tableau précédent donnera le diagramme d'espace de temps suivant :

FIGURE 2.6 – Diagramme d'espace de temps d'un AC utilisant la règle 30.

Les 256 automates cellulaires élémentaires de Wolfram (de la règle 0 à la règle 255) correspondants à la configuration initiale $\{\cdots, 0, 0, 0, 1, 0, 0, 0, \cdots\}$ sont donnés en annexe de ce document.

2.6.2 La classification de Wolfram :

Les règles choisies conduisent à un état statique, périodique, chaotique ou intermédiaire (un état "complexe", l'état de la vie biologique). En 1983 Stephen Wolfram s'est intéressé aux automates « légaux » (qui d'une part éliminent toute cellule dont le voisinage est vide, et d'autre part sont symétriques.), ces automates étaient unidimensionnels, à deux états avec un voisinage de deux, il a donc réalisé son étude systématique sur 32 automates légaux et a réussi à montré que, de nombreux automates

cellulaires (peut-être tous) s'intègrent dans quatre classes principales :

* **Classe 1 – état limite homogène (indépendant état initial).**

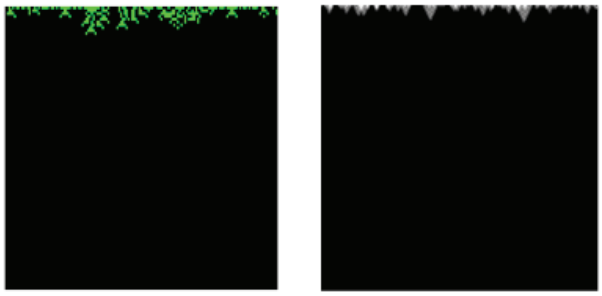


FIGURE 2.7 – Exemple d'automate de classe 1 (règle 36, règle 160).

En partant d'un état initial, après un certain nombre d'itérations, l'évolution de l'AC tend vers un état homogène dans lequel les cellules sont représentées par les mêmes valeurs. Toute information sur l'état initial du system sera détruite, et la prédiction de l'évolution est donc évidente. Nous pouvons citer à titre d'exemple les règles numéro $0, 32, 36, 160, \cdots$

* **Classe 2 – état limite simple ou périodique.**

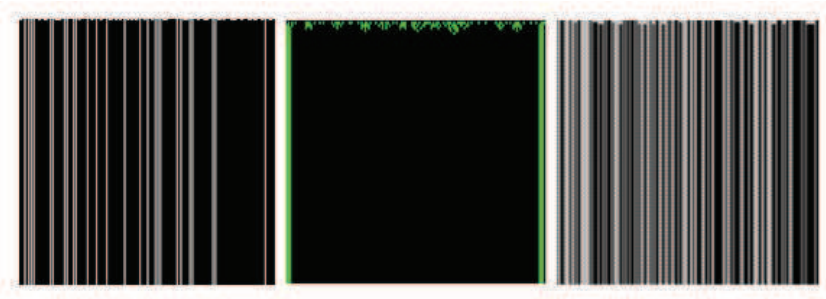


FIGURE 2.8 – Exemple d'automate de classe 2 (règles 4, 40 et 108).

En partant d'un état initial, après un certain nombre d'itérations, l'évolution de l'AC conduit à un ensemble séparé de structures simples stables ou périodiques. La modification d'une cellule de l'état initial n'affectera qu'une région finie de son entourage, et la prédiction de l'évolution reste faisable. Nous pouvons citer à titre d'exemple les règles numéro $4, 40, 108, \cdots$

* **Classe 3 – chaotique.**

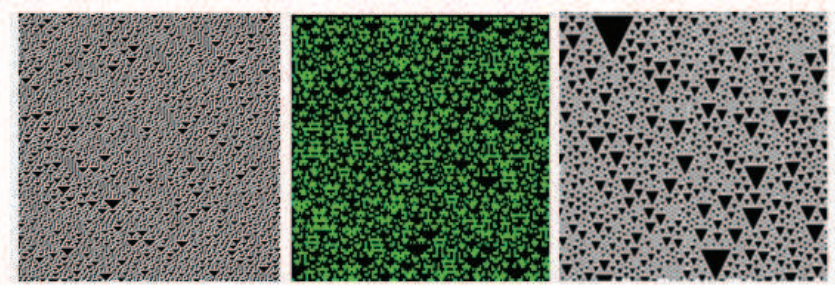


FIGURE 2.9 – Exemple d'automate de classe 3 (règles 30, 18 et 126).

Dans cette classe, au cours de l'évolution de l'AC et contrairement à ceux de la classe 2, les cellules propagent les informations à vitesse constante, après un certain nombre d'itérations en partant de l'état initial, l'évolution de l'AC conduit à un motif chaotique caractérisé par des attracteurs étranges et des structures périodiques. La prédiction de l'évolution n'est possible qu'à partir d'un nombre infini d'états initiaux. Nous pouvons citer à titre d'exemple les règles numéro $22, 18, 30, 126, \cdots$

* **Classe 4 – Les automates de cette classe évoluent vers des configurations globales complexes.**

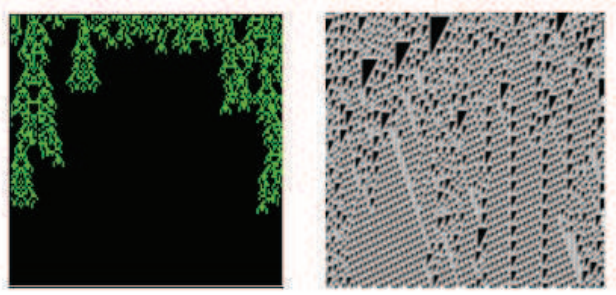


FIGURE 2.10 – Exemple d'automate de classe 4 (règle 20, règle 110).

Dans la classe 4, l'évolution de l'AC conduit à un état mort avec l'apparition de structures complexes stables ou périodiques. Quand au degré de la non-prédiction, il est encore plus élevé que celui de la classe 3. Le jeu de la vie [Gardner 1970] est un parfait exemple.

Nous remarquons que les patterns obtenus par certains automates ressemblent aux motifs exhibés par certains coquillages (comme le montre la figure 2.11).

FIGURE 2.11 – Les motifs de certains coquillages.

2.7 Les automates cellulaires bidimensionnels :

Un automate cellulaire bidimensionnel est un espace cellulaire formé par un tableau à deux dimensions 2D ($d = 2$) de $r \times c$ objets identiques appelés cellules chacune représentée par un état $s_{i,j}$ à un instant donné t formant ce qu'on appel une configuration $C^{(t)}$ du réseau, comme suite :

$$C^{(t)} = \begin{pmatrix} s_{1,1}^{(t)} & s_{1,2}^{(t)} & \cdots & s_{1,c}^{(t)} \\ s_{2,1}^{(t)} & s_{2,2}^{(t)} & \cdots & s_{2,c}^{(t)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ s_{r,1}^{(t)} & s_{r,2}^{(t)} & \cdots & s_{r,c}^{(t)} \end{pmatrix} \tag{2.1}$$

L'AC évolue de façon déterministe dans un temps discret changeant les états de toutes les cellules selon une fonction de transition locale, f. L'état d'une cellule $s_{i,j}$ est mis à jour à chaque instant selon l'ensemble de voisinage $V \subset Z * Z$ au temps précédent $s_{i,j}^{(t+1)} = f(V_{i,j}^{(t)}), 1 \leq i \leq r, 1 \leq j \leq c$. Parmi les voisinages on a :

- **Von Neuman : considère les seuls voisins Nord, Sud, Est, Ouest ainsi que la cellule correspondante (5 cellules) $\begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}$, avec $2^5 = 32$ numéros de règles possibles.**
- **Moore : voisinage complet (9 cellules) $\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$, avec $2^9 = 512$ numéros de règles possibles.**
- **Moore-Von Neumann : voisinage élargi qui comprend le voisinage de Moore auquel on ajoute les 4 voisins les plus proche de Von Neuman.**
- **Moore étendu : où on étend la distance de voisinage au delà de un.**

→ **Voisinage en croix : comporte les cellules disposées en diagonale en plus de la cellule elle-même (5 cellules)** $\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$ **,** $2^5 = 32$ **numéros de règles possibles.**

→ **Toom : un voisinage constitué de la cellule elle-même, de sa voisine du dessus et de sa voisine de droite.**

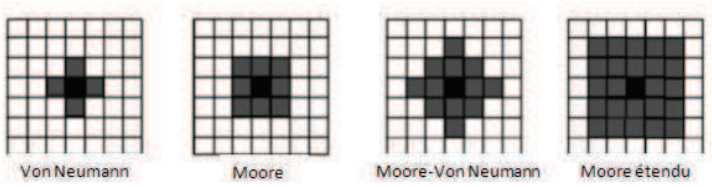


FIGURE 2.12 – Exemples de voisinage à 2 dimensions.

2.7.1 Le Jeu de la vie :

Le Jeu de la Vie [Gardner 1970] est une bonne illustration d'un automate cellulaire simple 2D introduit en 1970 par le mathématicien John Horton Conway, un modèle où chaque état conduit mécaniquement à l'état suivant à partir de règles préétablies. Le jeu se déroule sur une grille infinie (mais de longueur et de largeur finies et plus ou moins grandes dans la pratique) dont les cases (les cellules) peuvent prendre deux états distincts : « vivantes » ou « mortes ».

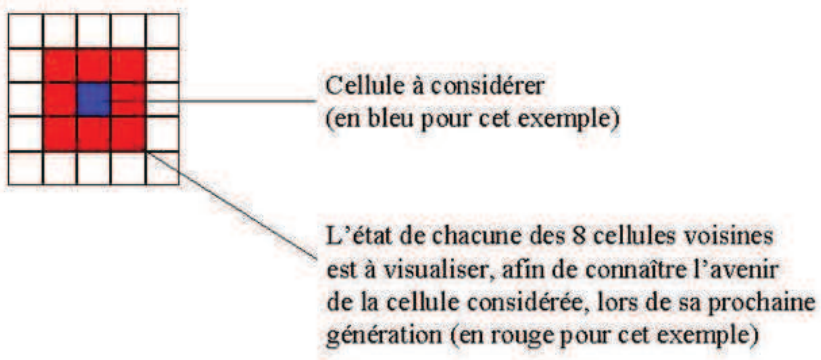


FIGURE 2.13 – Détermination du voisinage.

À chaque étape, l'évolution d'une cellule est entièrement déterminée par l'état de ses huit voisines (chaque cellule observe les huit cellules voisines) et :

☞ **Naît si elle a exactement 3 voisines vivantes ;**

- ☞ **Survit si elle a 2 ou 3 voisines vivantes ;**
- ☞ **Meurt d'étouffement si elle a plus de 3 voisines vivantes ;**
- ☞ **Meurt d'isolement si elle a moins de 2 voisines vivantes.**

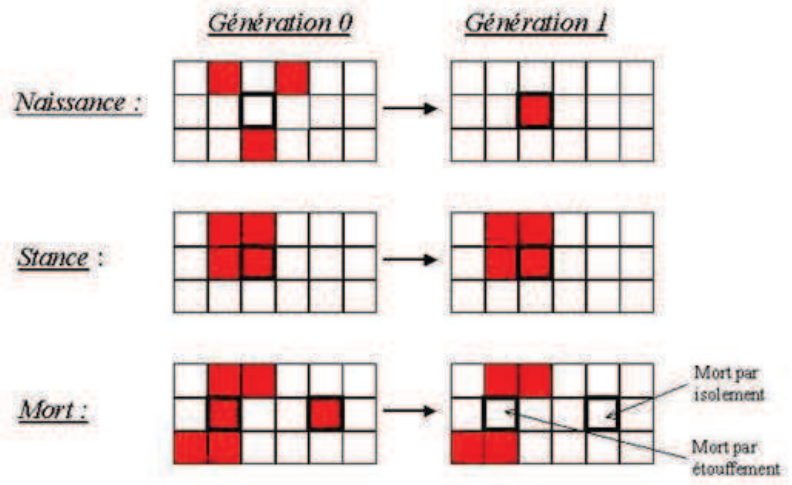


FIGURE 2.14 – Les règles du jeu de la vie.

Comme les ACs, le principal intérêt du jeu de la vie est de faire émerger des phénomènes complexes à partir de règles simples, on peut distinguer plusieurs structures dans l'univers du jeu, et qui peuvent être classées en différentes « espèces » (les structures stables, les graines de pulsar, les pulsars, les structures périodiques ou oscillateurs, les vaisseaux, les planeurs, les canons, les mangeurs, ⋯).

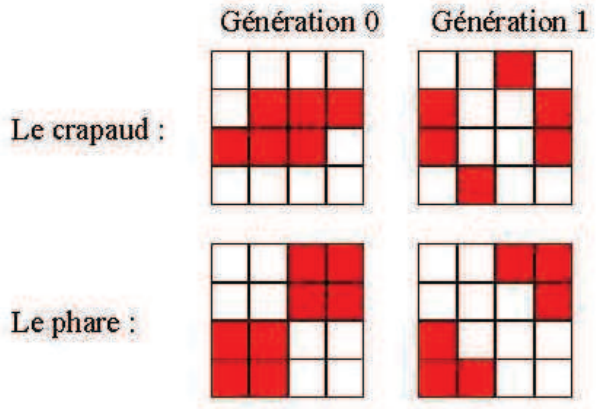


FIGURE 2.15 – Exemple de structure périodique (oscillateurs).

2.7.2 L′automate de Fredkin :

En 1961, Ed-ward Fredkin a inventé un automate cellulaire dont la loi d′évolution est une des plus simples qui puisse être et dont le comportement global, tout en étant tout le contraire d′un phénomène émergeant (comme l′est le jeu de la vie), révèle une propriété apparemment inattendue : décupler de façon périodique n′importe quel motif.
L'automate de Fredkin qui utilise un voisinage de Moore est basé sur la parité du voisinage. C'est un automate de type sommatif, c'est-à-dire que l'état des cellules dépend du nombre de voisins actifs, indépendamment de leur position. En l'occurrence, il n'y a reproduction que si la valeur de voisinage est impaire. Cet automate a la propriété remarquable de reproduire toute configuration de base en neuf exemplaires. La règle de Fredkin est généralisable à plus de deux dimensions.
Dans l′automate de Fredkin, les règles sont les suivantes :

- Si le voisinage d′une cellule (les huit cellules adjacentes) contient un nombre impair de cellules actives, celle-ci sera active à la génération suivante.
- Dans le cas contraire elle sera inactive.

La figure 2.16 montre un exemple de l′automate de Fredkin qui utilise un voisinage de Moore :

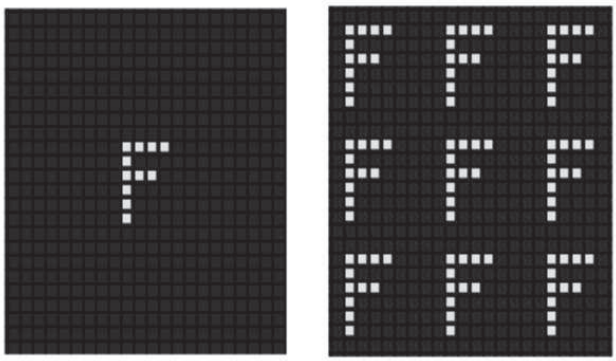


FIGURE 2.16 – Fredkin générations 0 et 8.

Les automates cellulaires basés sur des règles du type "Jeu de la Vie" (automate en grille, deux états, voisinage de contact direct) peuvent prendre des comportements très divers. Ils se révèlent, suivant les cas soit stables, soit un nombre supérieur d′états et pour n′importe quel voisinage chaotique, explosif, et même émergeant (jeu de la vie). Ils peuvent aussi se montrer ordonnés et reproductifs (compteur de parité) illustrant une fois de plus les rapports complexes qu′il peut y avoir entre le comportement d′un système multi-agent et les lois élémentaires le régissant :

la moindre modification dans ces lois peut changer du tout au tout ce comportement. Mais contrairement au jeu de la vie (dont on ne peut anticiper ni expliquer rationnellement le comportement), cette propriété du compteur de parité s'explique parfaitement : il s'agit en fait d'un simple phénomène arithmétique qui peut être généralisé avec un nombre supérieur d'états et pour n'importe quel voisinage.
Brian's Brain par exemple, présenté par Brian Silverman en 1984 a utilisé trois états vie/fantôme/mort pour engendrer une grande diversité de planeurs complexes au sein de configurations graphiques étonnantes.

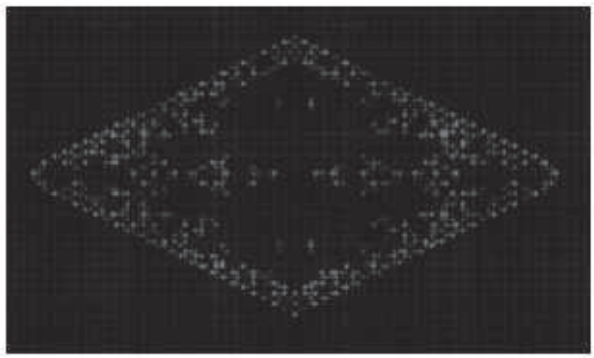


FIGURE 2.17 – Brian's Brain.

2.8 Les Automates Cellulaires à Mémoire Linéaires :

Dans le paradigme standard des automates cellulaire, l'état de chaque cellule au temps $t+1$ dépend seulement de son voisinage au temps t. Cependant, il est possible de considérer un AC dans lequel l'état de chaque cellule au temps $t+1$ dépend non seulement de son voisinage au temps t, mais également de son voisinage aux temps $t-1, t-2, \cdots, t-k$, (pour un AC d'ordre k). L'évolution d'un tel type d'AC nécessite la connaissance des k configurations initiales ainsi que des $k-1$ numéros de règles pour définir les fonctions de transition locales linéaire, c'est ce qu'on appelle automate cellulaire à mémoire linéaire (ACML). Ce type d'automate est un cas particulier des automates cellulaires réversibles (ACRs).
Dans un ACR chaque configuration a un prédécesseur unique. Plus précisément, les ACRs sont construits de manière à déterminer l'état de chaque cellule au temps $t+1$ à partir de l'état de toutes les cellules précédentes, le principe en est le même que pour les ACMLs, c'est-à-dire qu'à partir de k configurations consécutives, une nouvelle configuration est générée, de la même manière, l'AC peut être inversé pour récupérer les états initiaux à partir de k configurations consécutives, ce type d'AC a été introduit par Fredkin [Fredkin 1990]. Plusieurs approches ont

été définies pour construire un ACR, on peut citer par exemple dans [Toffoli 1990], où une méthode d'AC de second d'ordre a été introduite en utilisant le fait que l'état d'une cellule au temps t dépend non seulement de son voisinage à l'instant $t-1$, mais aussi sur son état à l'instant $t-2$ (voir figure 2.18). Si pour un AC donné, une configuration à chaque étape de temps t est définie par $C^{(t)}$, alors nous pouvons construire un ACR de second ordre en utilisant l'équation suivante :

$$C^{(t-2)} = F(C^{(t-1)}) \oplus C^{(t)} \tag{2.2}$$

Au lieu d'utiliser une configuration initiale, deux configurations sont utilisées pour évoluer l'ACR du second ordre. Après m itérations sur les deux configurations $C^{(0)}, C^{(1)}$ on peut obtenir deux configurations consécutives $C^{(m)}, C^{(m+1)}$. Lors de l'exécution du même ACR à partir des configurations initiales $C^{(m)}, C^{(m+1)}$, nous retrouverons les deux configurations $C^{(0)}, C^{(1)}$ après exactement m itérations et en utilisant la même règle de transition.

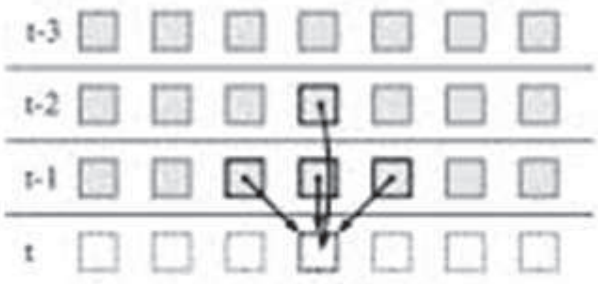


FIGURE 2.18 – Automate cellulaire du second degré.

2.8.1 Les ACMLs unidimensionnels :

La fonction de transition locale de l'ensemble des automates cellulaires unidimensionnels linéaires de taille n, et de voisinage symétrique de rayon r, a la forme suivante :

$$S_i^{(t+1)} = \sum_{j=-r}^{r} \lambda_j s_{i+j}^{(t)} (mod\ 2) \tag{2.3}$$

Où $0 \leq i \leq n-1$, et $\lambda_j \in \{0,1\}$. Comme il y a $2r+1$ cellules dans le voisinage symétrique de rayon r, alors il existe 2^{2r+1} différents ACLs qui vont de 0 à $2^{2r+1}-1$, et chaque ACL est conventionnellement spécifié par un entier décimal w représentant le numéro de la règle par :

$$w = \sum_{j=-r}^{r} \lambda_j 2^{r+j} \tag{2.4}$$

De la même manière que les ACRs, l'état de chaque cellule de l'ACML au temps $t+1$ dépend des états de ses cellules voisines à différentes étapes de temps $t, t-1, t-2, \cdots, t-k$. En particulier, en utilisant les règles de transition linéaire définie par l'équation (2.4), on peut définir un ACML d'ordre k dont la fonction transition locale prend la forme suivante :

$$S_i^{(t+1)} = (f_{w_1}(V_i^{(t)}) + f_{w_2}(V_i^{(t-1)}) + \cdots + f_{w_k}(V_i^{(t-k+1)}))(mod\ 2) \tag{2.5}$$

Où $0 \leq i \leq n-1$, et $w_1, w_2, \cdots, w_k \in \{0, \cdots, 2^{2r+1}-1\}$. Dans ce cas, k configurations initiales sont nécessaires pour faire l'évolution de l'ACML. La proposition suivante décrit comment construire un ACML réversible.

Proposition 1. *Si $f_{w_k}(V_i^{(t-k+1)}) = s_i^{t-k+1}$, alors l'ACML exprimé par :*

$$s_i^{(t+1)} = (f_{w_1}(V_i^{(t)}) + \cdots + f_{w_{k-1}}(V_i^{(t-k+2)}) + s_i^{(t-k+1)})(mod\ 2) \tag{2.6}$$

est réversible et son AC inverse est un autre ACML d'ordre k avec la fonction de transition locale suivante $s_i^{(t+1)} = G(V_i^{(t)}, \cdots, V_i^{(t-k+1)})$, où :

$$G(V_i^{(t)}, \cdots, V_i^{(t-k+1)}) = (-f_{w_{k-1}}(V_i^{(t)}) - \cdots - f_{w_1}(V_i^{(t-k+2)}) + s_i^{(t-k+1)})(mod\ 2) \tag{2.7}$$

Avec : $1 \leq i \leq n-1$ et $w_1, w_2, \cdots, w_{k-1} \in \{0, \cdots, 2^{2r+1}-1\}$.

2.8.2 Les ACMLs bidimensionnels :

Un type particulier des ACs bidimensionnels sont les ACs linéaires dont l'évolution se fait à l'aide d'une fonction de transition locale linéaire définie par une règle ayant comme paramètre l'ensemble de voisinage V d'une cellule s_{ij} à l'instant t, l'état d'une cellule s_{ij} à l'instant $t+1$ est calculé à l'aide de la fonction f comme suite : $s_{ij}^{(t+1)} = f(V_{ij}^{(t)})$, si le voisinage utilisé est celui de Moore on aura :

$$\begin{aligned} V &= \{(-1,-1),(-1,0),(-1,1),(0,-1),(0,0),(0,1),(1,-1),(1,0),(1,1)\} \\ V_{ij} &= \{(i-1,j-1),(i-1,j),(i-1,j+1),(i,j-1),(i,j),(i,j+1),(i+1,j-1), \\ &\quad (i+1,j),(i+1,j+1)\} \end{aligned} \tag{2.8}$$

La fonction de transition locale f est définit par une règle w, $(0 \leq w \leq 511)$, représentée sous forme binaire $(\lambda_{\alpha,\beta} \in Z_2 : \alpha, \beta \in \{-1,0,1\})$ comme suite :

$$\begin{aligned} w =& \lambda_{-1,-1}2^8 + \lambda_{-1,0}2^7 + \lambda_{-1,1}2^6 + \lambda_{0,-1}2^5 + \lambda_{0,0}2^4 + \lambda_{0,1}2^3 + \lambda_{1,-1}2^2 \\ &+ \lambda_{1,0}2^1 + \lambda_{1,1}2^0 \end{aligned} \tag{2.9}$$

La fonction f définie par le numéro de règle w (f_w), et par le voisinage de Moore peut être définie pour tout $1 \leq i \leq r, 1 \leq j \leq c$ par :

$$\begin{aligned} f(V_{ij}^{(t)}) &= \sum_{\alpha,\beta\in\{-1,0,1\}} \lambda_{\alpha,\beta} S_{i+\alpha,j+\beta}^{(t)} (mod\ 2) \\ &= (\lambda_{-1,-1}S_{i-1,j-1}^{(t)} + \lambda_{-1,0}S_{i-1,j}^{(t)} + \lambda_{-1,1}S_{i-1,j+1}^{(t)} + \lambda_{0,-1}S_{i,j-1}^{(t)} + \lambda_{0,0}S_{i,j}^{(t)} \\ &+ \lambda_{0,1}S_{i,j+1}^{(t)} + \lambda_{1,-1}S_{i+1,j-1}^{(t)} + \lambda_{1,0}S_{i+1,j}^{(t)} + \lambda_{1,1}S_{i+1,j+1}^{(t)})(mod\ 2) \end{aligned} \tag{2.10}$$

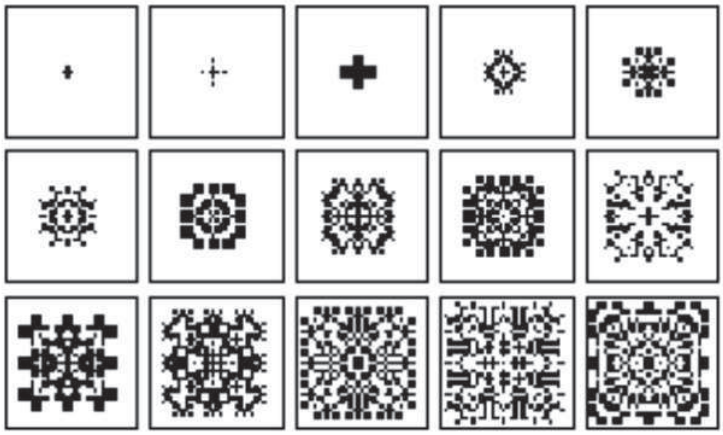


FIGURE 2.19 – Exemple de diagramme d'espace de temps d'un ACML d'ordre 3.

Pour un ACML par exemple d'ordre k, la fonction de transition locale prend la forme $s_{ij}^{(t+1)} = f(V_{ij}^{(t)}, \cdots, V_{ij}^{(t-k+1)})$, pour tout $1 \leq i \leq r, 1 \leq j \leq c$ avec :

$$f(V_{ij}^{(t)}, \cdots, V_{ij}^{(t-k+1)}) = (f_{w_1}(V_{ij}^{(t)}) + \cdots + f_{w_k}(V_{ij}^{(t-k+1)}))(mod\ 2) \tag{2.11}$$

Proposition 2. *Si $f_{w_k}(V_{ij}^{(t-k+1)}) = s_{ij}^{(t-k+1)}$, alors l'ACML exprimé par :*

$$f(V_{ij}^{(t)}, \cdots, V_{ij}^{(t-k+1)}) = (f_{w_1}(V_{ij}^{(t)}) + \cdots + f_{w_{k-1}}(V_{ij}^{(t-k+2)}) + s_{ij}^{(t-k+1)})(mod\ 2) \tag{2.12}$$

est réversible et son AC inverse est un autre ACML d'ordre k avec la fonction de transition locale suivante $s_{ij}^{(t+1)} = G(V_{ij}^{(t)}, \cdots, V_{ij}^{(t-k+1)})$, où :

$$G(V_{ij}^{(t)}, \cdots, V_{ij}^{(t-k+1)}) = (-f_{w_{k-1}}(V_{ij}^{(t)}) - \cdots - f_{w_1}(V_{ij}^{(t-k+2)}) + s_{ij}^{(t-k+1)})(mod\ 2) \tag{2.13}$$

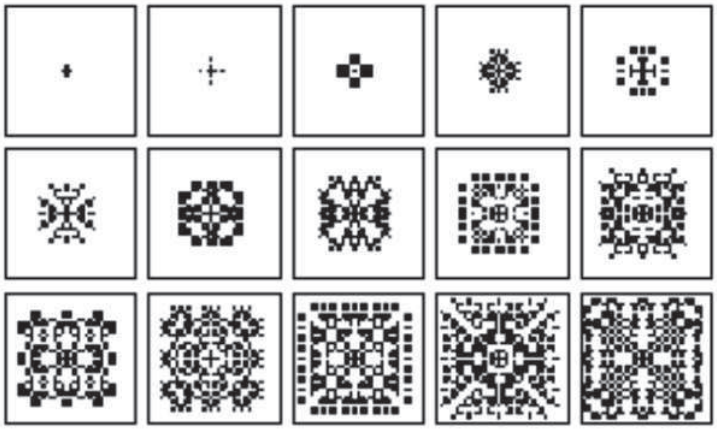


FIGURE 2.20 – Exemple de diagramme d'espace de temps d'un ACML réversible d'ordre 3.

Avec : $1 \leq i \leq r, 1 \leq j \leq c$ *et* $w_1, w_2, \cdots, w_{k-1} \in \{0, \cdots, 511\}$*, la preuve de cette proposition et de la proposition précédente (1) peuvent être trouvées dans Fredkin [Fredkin 1990]..*

2.9 Les automates cellulaires en cryptographie :

Un des concepts fondamentaux des automates cellulaires est celui d'émergence liée à la complexité, les automates cellulaires sont considérés comme un outil très puissant servant à modéliser des systèmes complexes en déterminant les comportements des entités élémentaires de façon locale et à constituer un nouveau paradigme caractérisé principalement par un traitement parallèle de problèmes selon une approche ascendante allant du plus simple au plus complexe ce qui les rendent imprévisibles, efficaces à la génération de nombre pseudo-aléatoires et donc intéressants dans le domaine de la cryptographie.
Le tableau suivant résume les similitudes et les différences entre les automates cellulaires et les algorithmes cryptographiques [BENTOUILA 2013] :

Systèmes d'AC	Crypto-systèmes
Ensemble d'états discrets	Ensemble de valeurs entières (discrètes)
Itérations	Rounds
Etat initial d'évolution	Clé de chiffrement
Distribution aléatoire	Cypher-text aléatoire et confus
Sensibilité aux conditions initiales	Principe de diffusion

Réversibilité difficile sans certains paramètres	Déchiffrement difficile sans clé de chiffrement
Parallélisme implicite	Parallélisme souhaitable
Représentation de données par blocs ou flots	Représentation de données par blocs ou flots

Tableau 2.2 – Comparaison entre les ACs et les algorithmes cryptographiques.

Comme la sécurité des cryptosystèmes est liée à la génération de nombres pseudo-aléatoire et à l'imprévisibilité, plusieurs documents ont suscité un intérêt dans la communauté cryptographique pour l'utilisation d'AC comme primitives dans les chiffrements de flux, le chiffrement par blocs, les fonctions de hachage, les codes d'authentification des messages et le chiffrement à clé publique.

2.9.1 Utilisation d'AC dans le chiffrement symétrique :

La plupart des cryptosystèmes basés sur les automates cellulaires sont à clés symétrique [Seredyński 2003], [Szaban 2006], [Mohamed 2014] et l'automate cellulaire joue le rôle de clé secrète ; En 1986, et sur la base du chiffrement de Vernam Stephen Wolfram fut le premier à avoir utilisé les automates cellulaires comme un générateur de nombre pseudo-aléatoires pouvant générer un flux de données (Keystream). Pour cela, il a proposé d'utiliser un automate cellulaire unidimensionnel exécutant la règle 30 comme moyen de chiffrement primitif. Où il suggère d'initialiser l'état initial de l'AC à l'aide d'une clé, pour avoir une sortie cryptographiquement aléatoire unique à travers le temps. En outre, il suggère que compte tenu de la connaissance d'un flux de sortie, le problème de déduire l'état initial est dans la classe NP (puisque le problème de satisfiabilité booléenne est en classe NP) [Testa 2008], cependant l'attaque Meier et Staffelbach [Meier 1991] a suffi à rendre le système pratiquement insécurisé. Plusieurs travaux concernant ce type de cryptosystème ont été réalisés par la suite que ce soit pour un chiffrement de flux ou par bloc.

2.9.2 Utilisation d'AC dans le chiffrement asymétrique :

Suite aux travaux de Kari [Kari 1992], [Kari 1990] et de Toffoli et Margolus [Toffoli 1990], montrant l'indécidabilité du problème de savoir si un automate cellulaire de dimension au moins égale à deux est inversible, des systèmes cryptographiques à clés publiques basés sur les automates cellulaires ont été proposés [Kari 1992], [Guan 1987]. La confidentialité d'un système cryptographique à clés publiques est basée en général sur la difficulté de trouver la fonction inverse de la fonction de chiffrement. Le calcul de cet inverse peut se faire cependant très facilement si on connaît

une brèche secrète dans la construction de la fonction de chiffrement. En ce qui concerne les cryptosystèmes basés sur les automates cellulaires, la clé publique utilisée pour crypter les messages est un automate cellulaire. Pour déchiffrer le message, il suffit d'appliquer au message crypté l'automate inverse de celui utilisé pour le chiffrement. Or il n'existe pas de procédure connue pour calculer l'inverse d'un automate cellulaire de dimension supérieure ou égale à 2. La brèche secrète utilisée généralement est le fait que l'automate de chiffrement est la composée d'un certain nombre d'automates triviaux que l'on sait inverser facilement. Depuis que les automates cellulaires ont été proposés comme une alternative pour la conception de cryptosystèmes à clés publiques, leur usage n'est pas devenu très populaire. Le problème est sans doute dû au fait que l'on ne sait pas toujours construire une grande famille d'automates cellulaires non triviaux inversibles [TINDO 2006].

2.10 Conclusion :

Les automates cellulaires représentent un développement relativement récent de la science moderne, extrêmement vaste et complexe à étudier, de ce fait et par nécessité de sécuriser les systèmes cryptographiques, l′utilisation d′un tel système complexe se caractérisant par l′indécidabilité et l′imprévisibilité, s′avère importante puisque le système est considéré comme l′un des générateurs pseudo-aléatoires le plus efficace et le plus adapté à la sécurité de nombreux crypto-systèmes dépendant de la génération des nombres aléatoires.

Le chapitre suivant est consacré à l′étude concernant l′état de l′art des différentes solutions de partage de secret exploitantes les automates cellulaires.

Chapitre 3

Etat de l′art sur le partage de secret par les ACs

Sommaire

3.1 Introduction :

Depuis l'apparition des premiers schémas de partage de secret faite en 1979 indépendamment par Shamir [Shamir 1979] et Blakley [Blakley 1979], les techniques de partage ne cessent d'évoluer. Ces dernières années, de nouvelles techniques de partage de secret plus performantes de plus en plus sophistiquées ont été apparues reposant sur l'utilisation du nouveau paradigme des automates cellulaires, qui permet d'atteindre des performances plus importantes par rapport aux méthodes classiques de partage de secret, par cause de la complexité linéaire (partage/reconstruction) qu'offrent les automates cellulaires à ces schémas, mais tout de même un problème majeur est lié aux schémas existants de partage de secret basés sur les automates cellulaires.

Nous allons voir dans ce chapitre tout d'abord l'état de l'art concernant les schémas de partage de secret utilisant les automates cellulaires et leurs avantages, puis nous aborderons l'inconvénient ou le problème majeur commun à tous les schémas.

3.2 Les approches de partage de secret utilisant les ACs :

Les schémas de partage de secret ne cessent d'évoluer et de se distinguer entre eux selon :

- ☞ la manière de partager le(s) secret(s) (en utilisant les ACs, la solution de Shamir, Blakely, CRT, $\cdots$),
- ☞ le nombre (un secret ou plusieurs) et le type de secret à partager (texte, image(s), $\cdots$),
- ☞ le seuil choisi,
- ☞ la complexité de partage/reconstruction,
- ☞ ainsi que les aspects d'efficacité et de sécurité qu'incluent certains schémas (propriété de t-consistance ou de forte t-consistance, vérifiabilité du schéma, la résistance aux attaques, $\cdots$).

Dans ce qui suit, nous décrivons quelques solutions de partage de secret exploitant les automates cellulaires réversibles à mémoire linéaire :

3.2.1 Approche de G. Alvarez Maranon et L. Hernandez Encinas, 2003 [Marañón 2003] :

Un schéma à seuil (k,n) a été proposé pour partager une image secrète entre un groupe de n participants en utilisant les ACMLs bidimensionnels, avant d'aborder l'algorithme de leur solution, une description concernant la manière de représentation de l'image secrète en matrice est décrite :

L'image secrète I à partager est définie par c couleurs, $r \times s$ pixels p_{ij} avec $0 \leq i \leq r-1, 0 \leq j \leq s-1$, et considérée comme une matrice appartenant à une configuration de l'AC a un instant t $(C^{(t)})$ de $a \times b$ cellules $a_{ij}^{(t)}$:

1. Si I est une image binaire, i.e. $c = 2$, alors soit $a_{ij}^{(t)} = 0$ si le pixel p_{ij} est noir, ou $a_{ij}^{(t)} = 1$ si le pixel p_{ij} est blanc. Par conséquent, dans ce cas $a = r$ et $b = s$.
2. Si I est une image en niveaux de gris, i.e. $c = 2^8$, et donc le code RGB de chaque pixel p_{ij} peut être défini par 8 bits. Par conséquent, $C^{(t)}$ est une matrice booléenne de $r \times (8.s)$, dans ce cas $a = r$ et $b = 8.s$; une configuration similaire apparait si l'image est définie par 256 couleurs.
3. Finalement, Si I est une image en couleur définie par $c = 2^{24}$ couleurs, alors chaque pixel p_{ij} est donné par 24 bits. Par conséquent, $C^{(t)}$ est une matrice booléenne de $r \times (24.s)$, avec $a = r$ et $b = 24.s$

L'algorithme de cette solution est décomposé en trois phases comme suite :

1. Phase d'initialisation : Dans cette première phase « le dealer » génère un ensemble de $k-1$ nombres aléatoires $w_l, (1 \leq l \leq k-1)$ représentant les numéros de règle de l'AC 2D avec $0 \leq w_l \leq 511$, pour construire la fonction de transition locale $a_{ij}^{(t+1)} = f_{w_1}(V_{ij}^{(t)}) + \cdots + f_{w_{k-1}}(V_{ij}^{(t-k+2)}) + a_{ij}^{(t-k+1)} (mod\ c), 0 \leq i \leq r-1, 0 \leq j \leq s-1$, puis définit les k configurations initiales de cet AC en représentant la matrice de l'image secrète dans la configuration $C^{(0)}$ et à l'aide d'un générateur de nombre pseudo-aléatoire les $k-1$ configurations suivantes sont définies.
2. Phase de partage : Cette deuxième phase consiste à partager les parts produites aux n participants, pour cela « le dealer » choisit un entier m,$(m \geq k)$ et calcule l'évolution d'ordre $(m+n-1)$ de l'AC à partir de l'AC d'ordre k définie précédemment, les n dernières configurations calculées de l'AC représenteront les n parts du secret $S_0 = C^{(m)}, \cdots, S_{n-1} = C^{(m+n-1)}$, partagées aux n participants, en outre chaque participant du groupe recevra également l'ensemble des $k-1$ nombres aléatoires générés en phase 1 afin de s'en passer du « dealer » lors de la reconstruction du secret dans la troisième étape (chaque participant parmi k pourra jouer le rôle du dealer et reconstruire la fonction de transition locale inverse (voir Proposition 2) après collection des $k-1$ autres parts).
3. Phase de reconstruction : Pour reconstruire l'image secrète, tout sous-groupe de k participants parmi n possédant des parts consécutives d'image $S_\alpha = C^{(m+\alpha)}, \cdots, S_{\alpha+k-1} = C^{(m+\alpha+k-1)}, (0 \leq \alpha \leq n-k)$

pourra reconstruire le secret $C^{(0)}$ avec l'évolution inverse d'ordre $m+\alpha+k-1$ de l'AC à partir des k configurations consécutives appartenant aux k participants.

❋ L'avantage de cette solution :

- La taille des images distribuées aux n participants ainsi que de l'image reconstruite est identique à la taille de l'image originale avec aucune perte d'information.
- Le schéma est parfait puisque aucun sous-groupe de moins de k participants ne permet de retrouver l'image secrète.
- La collaboration du « dealer » n'est pas nécessaire pour retrouver l'image originale.

3.2.2 Approche de G. Alvarez al, 2005 [Alvarez 2005] :

Dans cette approche, les auteurs ont proposé un schéma pour partager une image secrète entre un ensemble de n participants, en utilisant ACML-2D, le protocole de partage est décomposé en deux phases comme suite :

1. Phase de partage : « Le dealer » construit un ACML d'ordre n, où les $n-1$ configurations initiales $C^{(0)},\cdots,C^{(n-2)}$ sont représentées par des matrices générées aléatoirement de même taille $r\times s$ que la matrice M représentante l'image secrète, et met $C^{(n-1)}=M$, puis choisit un entier $k\geq max(r,s)/2$, pour calculer l'évolution d'ordre $n+k+1$ à partir des n configurations initiales,

$$\{C^{(n)},C^{(n+1)},\cdots,C^{(n+k+1)},\cdots,C^{(2n+k)}\}, \tag{3.1}$$

et distribuer aux n participants à travers un canal sécurisé les n dernières configurations $C^{(n+k+1)},\cdots,C^{(2n+k)}$.

2. Phase de reconstruction : Pour reconstruire l'image secrète tous les participants combinent leurs parts,

$$\tilde{C}^{(0)}=C^{(2n+k)},\cdots,\tilde{C}^{(n-1)}=C^{(n+k+1)}, \tag{3.2}$$

pour faire l'évolution inverse d'ordre k+2, et donc trouver :

$$M=\tilde{C}^{(n+k+1)}=C^{(n-1)}. \tag{3.3}$$

3.2.3 Approche de A. Martin del Rey et al, 2005 [del Rey 2005] :

Martin del Rey et son équipe ont proposé un schéma de partage de secret à seuil (k,n) basé sur les ACMLs unidimensionnels, le schéma proposé est formé de trois phases citées ci-dessous :

1. Phase d'initialisation : Dans cette première partie « Le dealer » commence par calculer le rayon de voisinage r de l'AC, $0 \leq r \leq \lfloor l/2 \rfloor - 1$, où l représente la taille du secret texte à partager (longueur de bits), pour générer une séquence de $k-1$ numéros de règle $w_i, 1 \leq i \leq k-1$, et $0 \leq w_i \leq 2^{2r+1} - 1$, permettant de définir la fonction de transition locale $a_i^{(t+1)} = f_{w_1}(V_i^{(t)}) + \cdots + f_{w_{k-1}}(V_i^{(t-k+2)}) + a_i^{(t-k+1)} (mod\ 2)$, puis définit un AC d'ordre k en considérant le vecteur représentant le secret à partager comme la configuration initiale $C^{(0)}$ et en calculant le reste des $k-1$ configurations initiales $C^{(1)}, \cdots, C^{(k-1)}$ à l'aide de l'outil cryptographique de générateur de nombre pseudo-aléatoire.
2. La phase de partage : Permet de choisir un nombre aléatoire m,$m \geq k$, et de faire l'évolution d'ordre $n + m - 1$ de l'AC à partir des k configurations initiales $C^{(0)}, \cdots, C^{(k-1)}$, les n dernières configurations $C^{(m)}, \cdots, C^{(n+m-1)}$sont partagées entre les n participants.
3. Phase de reconstruction : Tout sous-groupe de k détenant des parts de configuration avec des numéros d'ordre consécutifs $C^{(m+\alpha)}, \cdots, C^{(m+\alpha+k-1)}, 0 \leq \alpha \leq n-k$ pourront reconstruire le secret $C^{(0)}$ avec l'évolution inverse d'ordre $m+\alpha+k-1$ de l'AC à partir des k configurations $\tilde{C}^{(0)} = C^{(m+\alpha+k-1)}, \cdots, \tilde{C}^{(k-1)} = C^{(m+\alpha)}$.

❄ L'avantage :

- Comme dans [Marañón 2003] et [Marañón 2005], le schéma est parfait, idéal et la taille du secret texte reconstruit est identique à la taille du secret original.

3.2.4 Approche de G. Alvarez Maranon et al, 2005 [Marañón 2005] :

Dans cette approche, un schéma semblable à [Marañón 2003] a été proposé, où il permet de partager une image secrète définie par $a \times b$ pixels p_{ij} avec $0 \leq i \leq a-1, 0 \leq j \leq b-1$ et c couleurs $2, 2^8, 2^{24}$ (comme c'est décrit auparavant) selon un schéma (n, n), le schéma proposé est défini ci-dessous :

1. Phase d'initialisation : « Le dealer » détermine une séquence de $n-1$ nombres aléatoires $0 \leq w_l \leq 511, (1 \leq l \leq n-1)$, et fixe $w_n, (w_n = 16)$, pour définir la fonction de transition locale $a_{ij}^{(t+1)} = \sum_{l=1}^{n} f_{w_l}(V_{ij}^{(t+1-l)}) (mod\ 2)$, puis définit un AC 2D d'ordre n où l'image secrète à partager est considérée comme une configuration initiale $C^{(n-1)}$ de l'AC et les n-1 configurations initiales restante $C^{(0)}, \cdots, C^{(n-2)}$ sont calculées par un générateur de bit aléatoire.
2. Phase de partage : « Le dealer » choisit un entier m,$(m \geq n)$ et calcule l'évolution d'ordre $n + m - 1$ à partir de l'AC d'ordre n défini précédemment. les n dernières parts produites par l'AC

$S_1 = C^{(m)}, \cdots, S_n = C^{(n+m-1)}$ seront distribuées aux n participants dans un triplet contenant aussi le numéro d'ordre du participant i ainsi que le numéro de la règle adéquate w_i.

3. Phase de reconstruction : Pour reconstruire l'image secrète $C^{(n-1)}$ tous les triplets $(i, w_i, S_i = C^{(m+i-1)}), 1 \leq i \leq n$, sont nécessaires pour faire l'évolution inverse d'ordre m de l'AC à partir des configurations $\tilde{C}^{(0)} = C^{(m+n-1)}, \cdots, \tilde{C}^{(n-1)} = C^{(m)}$.

❋ L'avantage de la solution :

- Comme dans [Marañón 2003] Le schéma est parfait, idéal, la taille de l'image reconstruite est identique à la taille de l'image originale, et aucune information n'est révélée de l'image secrète à partir des parts du secret attribuées aux n participants.

3.2.5 Approche de G. Alvarez et al, 2008 [Alvarez 2008] :

G. Alvarez et son équipe ont proposé un schéma à seuil (n, n) basé sur les ACMLs-2D permettant à chaque participant $P_i, 1 \leq i \leq n$, de partager une image secrète S_i avec le reste des participants, les n images secrètes à partager sont de même taille (un padding est nécessaire dans le cas contraire en arrondissant l'image de pixels blancs), aussi de même palette de couleur (« le dealer » considère chaque pixel codé par b bits avec $b \in \{1, 8, 24\}$, en prenant en comte la plus grande palette de couleur de toutes les images secrètes $p = 2^b$), le protocole est comme suite :

1. Phase d'initialisation : « Le dealer » reçoit de chacun des n participants une image secrète, génère n numéros de règle $w_m \in [0, 511], (1 \leq m \leq n)$, afin de définir m fonctions locales de n ACL, f_{w_m}, et construire la fonction de transition locale $s_{ij}^{(t+1)} = (\sum_{m=1}^{n} f_{w_m}(V_{ij}^{(t-m+1)}) + s_{ij}^{(t-n)})(mod\ 2)$ de l'ACML d'ordre n+1 avec les n images secrètes $S_m, 1 \leq m \leq n$, dans les composantes initiales $C^{(m)}$ et une image générée aléatoirement S_0 de même taille et même palette de couleur que les autres images en configuration $C^{(0)}$.
2. La phase de partage : Cette deuxième phase consiste à calculer les parts d'image qui vont être partagées de manière sécurisée aux n participants du groupe pour cela « le dealer » génère aléatoirement un entier $l \geq n + 2$, et calcule l'évolution d'ordre $l + n - 1$ de l'ACML défini dans la phase précédente, les n dernières configurations produites $R_m = C^{(l+m-1)}, 1 \leq m \leq n$, seront distribuées aux n participants selon le triplet (m, w_m, R_m), l'entier l et la dernière configuration de l'AC inverse $R_0 = C^{(l-1)} = \tilde{C}^{(n+1)}$ sont publiés.
3. Phase de reconstruction : Dans cette phase comme le schéma de partage est à seuil (n, n) tous les participants doivent partager leurs

parts pour découvrir les n images secrètes, « le dealer » commence par récupérer les données secrètes et publiques de chaque participants afin de faire l′évolution inverse d′ordre $l-1$ de l′AC à partir des configurations initiales $\tilde{C}^{(1)} = R_n = C^{(l+n-1)}, \cdots, \tilde{C}^{(n)} = R_1 = C^{(l)}, \tilde{C}^{(n+1)} = R_0 = C^{(l-1)}$, les n dernières configurations calculées représentent les n images secrètes reconstruites $\tilde{C}^{(l)} = C^{(n)} = S_n, \tilde{C}^{(l+1)} = C^{(n-1)} = S_{n-1}, \cdots, \tilde{C}^{(n+l+1)} = C^{(1)} = S_1$.

❋ L′avantage :

- Schéma parfait et idéal, dans lequel on peut partager plusieurs images secrètes en couleur tout en gardant leurs caractéristiques lors de la phase de reconstruction.

3.2.6 Approche de R. Dura´n D´iaz et al, 2009 [Hernández Encinas 2009] :

Dans laquelle, un schéma de partage multi-secret a été proposé, où chaque participant partage une image secrète avec le reste des participants, le schéma est décomposé en deux phase :

1. Phase de partage : « le dealer » définit un ACML d′ordre $n+1$, en représentant la configuration initiale $C^{(0)}$ par une séquence de bits pseudo-aléatoire I_0 de même longueur que les images secrètes, et les n configurations suivantes chacune par une image secrète, puis génère un entier $e > n+1$ pour faire l′évolution d′ordre $e-1$ de l′AC à partir des $n+1$ configurations initiales :

$$\{C^{(0)}, \cdots, C^{(n)}, C^{(n+1)}, \cdots, C^{(e)}, \cdots, C^{(n+e-1)}\}, \tag{3.4}$$

   puis distribue les n dernières configurations ainsi que leurs numéros d′ordre respectifs aux n participants, et publie la configuration $C^{(e-1)} = S^{(n)}$.

2. Phase de reconstruction : les n participants pourront reconstruire les n secrets en mettant en commun leurs parts des secrets et en faisant l′évolution inverse d′ordre $n+e-2$ de l′AC à partir des $n+1$ dernières configurations calculées en phase 1.

3.2.7 Approche de Z. Eslami et J. Zarepour Ahmadabadi, 2010 [Eslami 2010] :

Z. Eslami et J. Zarepour Ahmadabadi ont évolué le travail précédent fait par [Alvarez 2008] en proposant un schéma à seuil (k,t,n) vérifiable basé sur l′utilisation des ACMLs-1D, le schéma proposé permet de partager k secrets $SC_1, \cdots, SC_k$ entre un groupe de n participants $P_1, \cdots, P_n$, et de ne permettre la reconstruction de ces secrets que si t participants

réunissent leurs parts attribuées du secret $SH_1, \cdots, SH_t$, l'algorithme de cette approche a été divisé en trois phases comme suite :

1. Phase d'initialisation : « Le dealer » définit un AC d'ordre t :
   - Si $t \geq k$,
     - les k configurations initiales de l'AC sont représentées par les k secrets $C^{(0)} = SC_1, C^{(1)} = SC_2, \cdots, C^{(k-1)} = SC_k$ et les configurations qui suivent $\{C^{(k)}, \cdots, C^{(t-1)}\}$ par des chaines binaires aléatoires de même longueur l que les secrets.
   - Sinon,
     - les t configurations initiales de l'AC sont représentées par les t premiers secrets $C^{(0)} = SC_1, C^{(1)} = SC_2, \cdots, C^{(t-1)} = SC_t$ et les $k-t$ secrets restants sont utilisés pour calculer les $\acute{}\beta\check{}$ en phase 2.
2. Phase de partage : « Le dealer » choisit un nombre aléatoire $m \geq max(k,t)$, pour calculer l'évolution d'ordre $m+n-1$ de l'AC de la phase 1 avec la fonction de transition locale $a_j^{(T+1)} = f_{w_1}(N_j^{(T)}) + \cdots + f_{w_{t-1}}(N_j^{T-t+2}) + a_j^{(T-t+1)}(mod\ 2)$, et ceci après avoir sélectionné le rayon de voisinage $1 \leq r \leq \lfloor (l-1)/2 \rfloor$, ainsi que les numéros de règle $1 \leq w_i \leq 2^{2r+1} - 1, (1 \leq i \leq t-1)$, si $t < k$, alors le dealer calcule et publie $\{\beta_1, \cdots, \beta_{k-t}\}$ où $\beta_1 = SC_{t+1} + C^{(t)}(mod\ 2), \cdots, \beta_{k-t} = SC_k + C^{(k-1)}(mod\ 2)$, et les n dernières configurations représenteront les n parts distribuées aux n participants $SH_1 = C^{(m)}, \cdots, SH_n = C^{(m+n-1)}$.
3. Phase de reconstruction : Tous les groupes de t participants $P_{(\alpha+1)}, \cdots, P_{(\alpha+t)}$, ayant des parts de configuration avec des numéros d'ordre consécutifs $SH_{(\alpha+1)}, \cdots, SH_{(\alpha+t)}, (0 \leq \alpha \leq n-t)$, pourront reconstruire les k secrets $SC_1, \cdots, SC_K$, en fessant l'évolution inverse d'ordre $m+\alpha$ à partir des configurations $\tilde{C}^{(0)} = C^{(m+\alpha+t-1)}, \cdots, \tilde{C}^{(t-1)} = C^{(m+\alpha)}$ de l'AC et en se servant de la fonction de transition locale inverse définie par les numéros de règle générés en phase 2 .
   - Si $t \geq k$,
     - les k dernières configurations de l'AC inverse représenteront les k secrets $SC_1 = C^{(0)}, \cdots, SC_k = C^{(k-1)}$.
   - Sinon,
     - les t dernières configurations de l'AC inverse représenteront les t secrets $SC_1 = C^{(0)}, \cdots, SC_t = C^{(t-1)}$ et les $k-t$ secrets restants seront représentés par une addition modulaire des $m-t$ configurations de l'AC inverse et les bêtas calculés en phase 2 $SC_{t+1} = \beta_1 + C^{(t)}, \cdots, SC_k = \beta_{k-t} + C^{(k-1)}(mod\ 2)$.

Le schéma est en plus vérifiable et permet donc aux participants du groupe de vérifier leurs parts attribuées du secret comme suite :

- « Le dealer » choisit un nombre premier p tel que le problème du logarithme discret est insoluble dans $GF(p)$, et un générateur g pour le groupe cyclique de $GF(p)/\{0\}$ puis les publies.
- Avant de distribuer les parts $SH_i, 1 \leq i \leq n$ aux n participants P_i, « le dealer » calcule puis publie $G_i \equiv g^{SH_i}(mod\ p)$.
- Chaque participant du groupe pourra vérifier la validité de sa part après distribution.

❋ L'avantage :

- Schéma parfait et idéal, où on peut partager plusieurs secrets indépendamment au nombre de participant.
- Le schéma est sécurisé et vérifiable.

3.2.8 Approche de W. Xiaotian et al, 2012 [Wu 2012] :

Dans cette approche, les auteurs ont inclus la stéganographie à leur schéma de partage d'image secrète en utilisant les ACMLs-2D, l'algorithme de cette approche est décrit comme suite :

1. Phase d'initialisation : Dans cette première phase les auteurs ont fixé les paramètres utilisés pour construire l'ACML, et déterminer les configurations initiales comme suite :
   - « Le dealer » détermine un nombre k pour convertir tous les pixels $s_{i,j}$ de l'image secrète avec la notation $k\_ary(s_{i,j})$ en $(s_{i,j})_k$.
   - Calcule la longueur maximale $l = \lceil log_k T \rceil$ où T représente le nombre maximal de couleur, puis divise l'image secrète $(SI)_k$ de notation k_ary en l plans $SI_1, \cdots, SI_l, ((SI)_k = SI_1 \times k^{(l-1)} + SI_2 \times k^{(l-2)} + \cdots + SI_l)$, les résidus de l'image de couverture sont obtenus par $RS = CI(mod\ k)$.
   - Génère l fonctions de transition locale $f_{w_1}, \cdots, f_{w_l}$ par l entiers aléatoires $w_1, \cdots, w_l \in [0, 511]$, la fonction de transition locale d'ordre $(l+1)$ de l'ACML est construite par :

$$a_{i,j}^{(t+1)} = f_{w_1}(N_{i,j}^{(t)}) + \cdots + f_{w_l}(N_{i,j}^{(t-l+1)}) + a_{i,j}^{(t-1)}(mod\ 2) \tag{3.5}$$

   - Calcule la valeur du hach de l'image secrète et l'utilise pour initialiser un générateur de nombre aléatoire, une matrice aléatoire RM de taille $m \times n$ est construite par ce générateur.
   - « Le dealer » définit les configurations initiales de l'ACML d'ordre $(l+1)$ par :

$$C^{(0)} = RS \oplus RM, C^{(1)} = SI_1, \cdots, C^{(l)} = SI_L \tag{3.6}$$

2. **Phase de camouflage : Dans cette deuxième phase, « le dealer » évolue l'ACML d'ordre $(l+1)$ pour avoir les $l+1$ données à partager. Le processus est formulé comme suite :**

   - **« Le dealer » génère un nombre aléatoire supérieur ou égal à $l+1$.**
   - **Calcule l'évolution d'ordre u de l'ACML prédéfini d'ordre $(l+1)$ construit dans la phase précédente. Pour chaque étape de temps, une nouvelle configuration $C^{(l+i)}(1 \leq i \leq u)$ est générée puis accordée respectivement à une configuration parmi les $l+1$ configurations précédentes $C^{(i-1)}, \cdots, C^{(l+i-1)}$.**
   - **Sélectionne les $l+1$ dernières configurations $C^{(l+1)}, \cdots, C^{(l+u)}$ pour les utiliser lors de la phase suivante.**

3. **Phase de partage : Cette phase consiste à intégrer les $l+1$ dernières configurations calculées dans l'image de couverture pour créer les $l+1$ images stégos qui vont être partagées aux participants du groupe, la description détaillée est formulée comme suite :**

   - **« Le dealer » génère les $l+1$ images stégos $S_1, \cdots, S_{l+1}$ par :**

   $$S_i = \lfloor CI/k \rfloor \times k + C^{(i+u-1)}, i \in [1, l+1]. \tag{3.7}$$

   - **Le triplet (i, w_i, S_i) est délivré au $i^{\text{ème}}$ participant, et le nombre u est publié.**

4. **Phase de reconstruction : Cette dernière phase consiste à reconstruire l'image secrète ainsi que l'image de couverture, les étapes de reconstruction sont décrites comme suite :**

   - **Les $l+1$ données partagées sont retirées à partir des images stégos par :**

   $$C^{(i+u-1)} = S_i (mod\ k), i \in [1, l+1] \tag{3.8}$$

   - **« Le dealer » défini les $l+1$ configurations initiales de l'ACML inverse d'ordre $l+1$ par :**

   $$\tilde{C}^{(0)} = C^{(l+u)}, \tilde{C}^{(1)} = C^{(l+u-1)}, \cdots, \tilde{C}^{(l)} = C^{(u)}. \tag{3.9}$$

   - **A l'aide de la fonction de transition locale, « le dealer » étire u fois l'ACML inverse à partir des $l+1$ configurations initiales pour obtenir les $l+1$ dernières configurations de l'AC inverse : $\tilde{C}^{(u)}, \cdots, \tilde{C}^{(l+u-1)}, \tilde{C}^{(l+u)}$.**
   - **Les plans de l'image sont atteints par : $SI_l = \tilde{C}^{(u)}, \cdots, SI_1 = \tilde{C}^{(l+u-1)}, R = \tilde{C}^{(l+u)}$, et l'image secrète $(SI)_k$ est reconstruite par : $(SI)_k = SI_1 \times k^{(l-1)} + SI_2 \times k^{(l-2)} + \cdots + SI_l$, après conversions de $(SI)_k$ en décimal.**

- La matrice aléatoire RM est générée par un générateur de nombre aléatoire qui est avec la valeur du hach de l'image secrète. L'image de couverture CI est restaurée par : $CI = \lfloor S_i/k \rfloor \times k + (R \oplus RM)$, où S_i représente une image stégo parmi les $l+1$ autres images.

3.3 La problématique :

Même si toutes les approches basées sur les ACMLs fournissent de meilleures performances par rapport aux schémas de partage standards de Shamir, en offrant une complexité de partage et de reconstruction linéaire en temps, elles ont tous un inconvénient majeur qui les rend inutiles pour des applications réelles, ce qui ne permet pas de définir un mécanisme de partage à seuil (t,n) robuste, puisque la possibilité de reconstruire le secret n'est pas donnée à tous les sous-ensembles de t participants, mais seulement à ceux ayant des parts de configurations consécutives.

Malheureusement, pour des valeurs données de t et n ($t \leq n$), le nombre de sous-ensembles possibles de t parts comportant des éléments consécutifs est égal à $(n - t + 1)$, ce qui représente un nombre très petit par rapport au nombre de sous-ensembles possibles de t parts égal à $(C_n^t = \frac{n!}{t!(n-t)!})$. Par exemple, si $(t = 3)$ et $(n = 10)$, nous aurons $(C_{10}^3 = 120)$ sous-ensembles possibles de t parts, lorsque seul $(10 - 3 + 1 = 8)$ peuvent vérifier la propriété d'éléments consécutifs. Par conséquent, moins de (7%) des sous-ensembles possibles de t participants peuvent récupérer le secret.

Un tel problème rend le système de partage de secret basé sur les ACMLs non robuste, et par conséquent inapproprié pour l'utilisation dans les applications de scénarios réels. Plusieurs améliorations ont été proposées, mais elles ont tous ciblé des améliorations d'autres aspects de partage tels que le support multi-secrets, la vérifiabilité des parts, t-consistance et traçabilité. L'inconvénient principal de robustesse des schémas de partage basé sur les ACML n'a pas encore été abordé.

Nous présentons dans le chapitre qui suit, notre première solution qui répond pour la première fois au problème de la robustesse des schémas de partage de secret basés sur les ACMLs. L'idée de base en est simple : au lieu de définir la part de chaque participant par une seule configuration, un sous-ensemble de configurations est attribué à chaque participant de telle sorte que l'ensemble de l'union de t sous-ensembles de t différents participants contient une séquence unique de t configurations consécutives, tandis que l'union d'un nombre inférieur de sous-ensembles ne permet pas d'obtenir une telle séquence.

3.4 Conclusion :

Dans ce chapitre, une présentation de quelques solutions publiées de partage de secret utilisant les automates cellulaires a été faite, nous avions pu voir d'après les approches proposées, la manière dont laquelle les automates cellulaires sont exploités pour partager un secret donné entre plusieurs participants d'un groupe, ainsi nous avons constaté malgré les différentes recherches faites dans ce domaine que le problème de robustesse lié à tous ces schémas proposés reste ouvert.

Pour cela un nouveau système de partage de secret basé sur l'utilisation des automates cellulaires est présenté dans ce qui suit répondant ainsi au problème posé, les détails et les preuves du système sont donnés dans le chapitre suivant.

Partie III

Applications

Chapitre 4

Utilisation des matrices d′affectation pour la construction d′un modèle robuste de partage de secret basé sur les ACMs

Sommaire

4.1 Introduction :

Dans le chapitre présent, on décrit la solution proposée pour le problème de robustesse des schémas de partage de secret basés sur les ACMLs. A l'opposé des schémas existants, le schéma de partage de secret à seuil (t, n) proposé basé sur les ACMLs permet à chaque sous-ensemble de t participants (soit C_n^t sous-ensembles) de reconstruire le secret, et ceci en utilisant une matrice spécifique d'affectation des configurations de l'AC : qui au lieu de donner une seule configuration à chaque participant comme part du secret, un ensemble spécifique de différentes configurations est affecté à chaque participant de telle manière que l'union de tous t sous-ensembles différents de t participants contiendra toujours t configurations consécutives différentes, alors qu'en regroupant t-1 sous-ensembles ou moins la contrainte ne peut être réalisée. L'affectation des configurations est effectuée selon une matrice spécifique construite en utilisant l'algorithme heuristique proposé, tandis que le partage et la reconstruction sont effectués comme d'habitude en utilisant le mécanisme d'évolution de l'ACML.

Nous verrons dans ce chapitre une description détaillée du schéma de partage de secret proposé, nous analyserons par la suite les résultats obtenus suite à l'expérimentation du schéma, puis nous étudierons à la fin l'idéalité de ce schéma.

4.2 La solution proposée :

Dans cette section, nous présentons le schéma de partage d'image secrète proposé basé sur les ACMLs unidimensionnels. Le partage des images digitales étant un des grands domaines de recherche active lié au partage de secret en raison de leurs caractéristiques spécifiques telle que la redondance, la capacité de données volumineuses et la forte corrélation à travers les blocs de pixels.

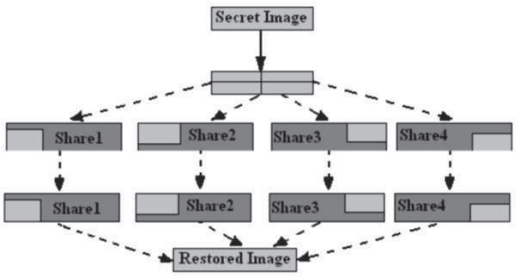


FIGURE 4.1 – Principe de partage d'une image secrète.

Le schéma à seuil (t, n) proposé $(2 < t < n - 1)$ s'avère robuste, puisque

que chaque sous-ensemble d'au moins t participants peut récupérer totalement le secret partagé en réunissant leurs parts. Comme d'habitude, trois phases principales sont nécessaires pour le schéma de partage de secret :

(1) La phase d'initialisation, au cours de laquelle « le dealer » génère les paramètres du schéma et définit l'ACML d'ordre t ;

(2) La phase de partage, où « le dealer » crée les n différentes parts en utilisant l'ACML défini en phase 1, ainsi que les données secrètes ;

(3) Et en fin la phase de reconstruction permettant de récupérer le secret à partir de tout sous-ensemble de t participants possédant des parts différentes du secret.

Notez que nous avons utilisé un ACML de rayon $r = 3$, ce qui fait que les règles de transitions sont dans l'ensemble $\{0, \cdots, 2^{2r+1} - 1\}$.

4.2.1 Construction de la matrice d'affectation :

Comme mentionné auparavant, « le dealer » a besoin d'une matrice d'affectation afin de distribuer les configurations de l'ACML entre les n participants. La matrice est utilisée pour affecter un ensemble de configurations à chaque utilisateur correspondant à une colonne.

La matrice d'affectation notée « A » a n colonnes (correspondantes aux n participants) et C_{n-2}^{t-2} lignes d'éléments de type entier. Elle est construite de telle manière à satisfaire les conditions suivantes :

(1) La combinaison de $t-1$ colonnes ne permet pas de construire une séquence de t nombres consécutifs ;

(2) La combinaison de chaque t colonnes permet de construire une séquence de t nombres consécutifs ;

Théoriquement, la construction d'une telle matrice est un problème combinatoire difficile. Par conséquent, nous avons développé un algorithme heuristique qui permet une telle construction et produit une matrice en respectant les deux conditions mentionnées ci-dessus. En outre, aucune restriction n'est imposée sur la limite supérieure des éléments de la matrice, et les valeurs dupliquées sont autorisées. Chaque colonne i de la matrice correspond à un participant P_i pour $1 \leq i \leq n$. La matrice d'affectation A est construite selon les étapes suivantes :

1. Initialement, la matrice A ayant n colonnes et C_{n-2}^{t-2} lignes est initialisée à zéro ; soit CS l'ensemble de toutes les combinaisons possibles (soit C_{n-2}^{t-2} combinaisons) de (t-2)-uplet dont les indices appartiennent à l'ensemble $\{2, \cdots, n-1\}$ construit avant, supposons que la valeur id de type entier définit la plus petite valeur autorisée pour les éléments de la matrice (peut trivialement être égal à 1). La matrice est construite ligne par ligne en partant par la première.

2. Les premiers éléments de la ligne reçoivent la valeur de id ; puis la première combinaison de CS est sélectionnée
3. Les éléments de la combinaison courante sélectionnée sont utilisés comme indices pour remplir la ligne courante de A : si la combinaison est définie par $< i_1, i_2, \cdots, i_{t-2} >$ alors nous affectons la valeur $id + k$ à chaque élément $A[i_k]$;
4. Chacun des éléments de la matrice dans la ligne courante ayant un indice plus grand que i_{t-2} (la plus grande valeur de l'indice des combinaisons) reçoit la valeur $id + t - 1$;
5. La dernière étape de vérification consiste à regarder les éléments de la ligne courante de 2 à $n-1$ et les tester : si l'élément est resté égal à 0, alors l'élément reçoit la valeur du précédent ;
6. Si toutes les C_{n-2}^{t-2} lignes ont été remplîtes, alors l'algorithme se termine. Dans le cas contraire, on incrémente l'indice de la ligne courante, on met à jour la valeur de id par $id := id + t + 1$ et on retourne à l'étape 2.

Une description du pseudo-algorithme de la procédure de génération de la matrice est donnée dans ce qui suit. Nous considérons $CS[0 \cdot\cdot C_{n-2}^{t-2} - 1]$ un tableau représentant les $t-2$ combinaisons possibles dont les valeurs sont dans l'ensemble $\{2, \cdots, n-1\}$ de sorte que chaque élément $CS[k]$ est une combinaison possible $< i_1^k, i_2^k, \cdots, i_{t-2}^k >$

Entrées : t, n : entier ; CS ; id :entier (valeur du plus petit élément de la matrice) ;

Sortie : la matrice d'affectation A ayant n colonnes et C_{n-2}^{t-2} lignes ;

```
Pour i := 1 à C_{n-2}^{t-2} faire
      Pour j := 1 à n faire {A[i][j] := 0 ;}
Pour i := 1 à C_{n-2}^{t-2} faire
        {A[i][1] := id;
          Pour k := 1 à t − 2 faire {A[i][CS[i][k]] := id + k ;}
          Pour k := CS[i][t − 2] + 1 à n faire {A[i][k] := id + t − 1 ;}
          Pour k := 2 à n − 1 faire
           {Si A[i][k] = 0 alors A[i][k] := A[i][k − 1] ;}
          id := id + t + 1 ;
        } ;
```

L'algorithme proposé est conçu pour assurer les deux conditions mentionnées ci-dessus. L'incrémentation de la valeur id avec un supplémentaire de 1 (le saut) est effectuée ligne par ligne afin d'éviter à deux lignes consécutives de la matrice la séquence de t éléments consécutifs.

Un exemple de construction de la matrice utilisant les valeurs $n = 6$ et $t = 4$ est illustré dans ce qui suit :

D'après les valeurs de paramètre t et n, il est clair que la matrice contient 6 colonnes et $C_4^2 = 6$ lignes. L'ensemble CS des C_4^2 combinaisons d'indices possibles $\{2, 3, 4, 5\}$ est égal à $\{(2,3),(2,4),(2,5),(3,4),(3,5),(4,5)\}$. En appliquant l'algorithme proposé, la matrice suivante est obtenue :

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 & 4 & 4 \\ 6 & 7 & 7 & 8 & 9 & 9 \\ 11 & 12 & 12 & 12 & 13 & 14 \\ 16 & 16 & 17 & 18 & 19 & 19 \\ 21 & 21 & 22 & 22 & 23 & 24 \\ 26 & 26 & 26 & 27 & 28 & 29 \end{bmatrix} \tag{4.1}$$

On peut facilement vérifier que la combinaison de chaque 4 colonnes de la matrice permet d'obtenir une séquence de quatre nombres consécutifs. Par exemple la combinaison des colonnes $1, 2, 3$ et 4 donne la séquence des valeurs consécutives $\{1, 2, 3, 4\}$, tout comme la combinaison des colonnes $2, 3, 5$ et 6 qui donne la séquence des valeurs consécutives $\{21, 22, 23, 24\}$. En revanche, la combinaison de chaque trois colonnes ne permet en aucun cas de donner une séquence de quatre nombres consécutifs.

Un autre exemple peut être illustré pour $n = 7$ et $t = 3$. Ici, l'ensemble CS est simplement l'ensemble des combinaisons possibles dans $\{2, 3, 4, 5, 6\}$ ayant une seule longueur, qui est trivialement égal au même ensemble $\{2, 3, 4, 5, 6\}$. En appliquant l'algorithme on obtient la matrice d'affectation suivante ayant $C_{7-2}^{3-2} = C_5^1 = 5$ lignes et 7 colonnes :

$$A = \begin{bmatrix} 1 & 2 & 3 & 3 & 3 & 3 & 3 \\ 5 & 5 & 6 & 7 & 7 & 7 & 7 \\ 9 & 9 & 9 & 10 & 11 & 11 & 11 \\ 13 & 13 & 13 & 13 & 14 & 15 & 15 \\ 17 & 17 & 17 & 17 & 17 & 18 & 19 \end{bmatrix} \tag{4.2}$$

Les mêmes propriétés existent dans cette matrice l'union des éléments de chaque trois colonnes conduit toujours à une séquence de trois nombres consécutifs.

Dans ce qui suit, nous montrons que pour chaque valeur des paramètres n et t, la matrice d'affectation vérifie toujours les conditions (1) et (2) :

Lemma 4.2.1. *Pour toute valeur de n et t, la matrice d'affectation construite en utilisant l'algorithme proposé possède les propriétés suivantes :*

1. *L'union de l'ensemble des éléments de chaque t colonnes contient toujours une séquence de t nombres consécutifs ;*
2. *L'union de l'ensemble de chaque $t-1$ ou moins de colonnes ne contient aucune séquence de t nombres consécutifs.*

Démonstration. Nous commençons par prouver la première hypothèse. Supposons qu'on a une combinaison $B =< i_1, i_2, \cdots, i_t >$ de t colonnes différentes de la matrice A ($i_k \neq i_j \ \forall 1 \leq k, j \leq t$). Nous supposons également que les éléments de la combinaison sont triés dans l'ordre croissant ($i_1 < i_2 < \cdots < i_t$).

Prenons $\grave{B} =< i_2, i_3, \cdots, i_{t-1} >$ une sous combinaison de B restreinte par $t-2$ éléments et cela sans le premier et le dernier élément (i_1 et i_t), et soit $Ord(\grave{B})$ l'ordre de $\grave{B}$ dans l'ensemble CS.

Il est clair à partir de l'algorithme que comme $\grave{B} \in CS$, les $t-2$ nombres consécutifs :

$$\begin{aligned}\{id + Ord(\grave{B}) * t + 1, id + Ord(\grave{B}) * t + 2, id + Ord(\grave{B}) * t + 3, \cdots, \\ id + Ord(\grave{B}) * t + t - 2\},\end{aligned} \tag{4.3}$$

seront affectés aux indices de $\grave{B}$ au cours de la première boucle du nombre d'itération $Ord(\grave{B})$ (l'itération en utilisant $\grave{B}$ comme base de combinaison). En outre, la seconde boucle affectera la valeur $id + Ord(\grave{B}) * t + t - 2$ aux indices restants plus grand que i_{t-1}, et comme $i_t > i_{t-1}$, la valeur de la ligne courante dont l'indice est i_t recevra la valeur $id + Ord(\grave{B}) * t + t - 1$. En conséquence, nous obtenons $t-1$ valeurs consécutives $\{id + Ord(\grave{B}) * t + 1, id + Ord(\grave{B}) * t + 2, id + Ord(\grave{B}) * t + 3, \cdots, id + Ord(\grave{B}) * t + t - 2, id + Ord(\grave{B}) * t + t - 1\}$ pour les indices $< i_2, i_2, \cdots, i_t >$.

Maintenant nous considérons le premier indice i_1. Dans la même ligne (correspondant aux nombre $Ord(\grave{B})$ d'itérations), à moins que i_1 soit égal à un ($i_1 = 1$) et dans ce cas la valeur de la ligne en position i_1 est certainement égale à $id + Ord(\grave{B}) * t$ (valeur d'initialisation de chaque ligne). En outre, si $i_1 > 1$, et puisque par hypothèse $i_1 < i_2$, aucune valeur ne peut être affectée à la valeur de la ligne courante en position i_1 durant la seconde boucle (sa valeur reste égale à zéro). Durant la troisième boucle, cette valeur prendra une des positions précédente qui est certainement égale à la valeur de la première colonne dans la ligne courante $id + Ord(\grave{B}) * t$. Par conséquent, dans tous les cas, une séquence $\{id + Ord(\grave{B}) * t, id + Ord(\grave{B}) * t + 1, id + Ord(\grave{B}) * t + 2, \cdots, id + Ord(\grave{B}) * t + t - 1\}$ peut être construite pour la combinaison B qui est une séquence de t nombres consécutifs.

La deuxième hypothèse peut être facilement prouvée : supposant que nous avons une combinaison de $t-1$ colonnes $< i_1, i_2, \cdots, i_{t-1} >$ de A. il est clair que comme un saut est introduit entre chaque deux lignes consécutives (i.e. la valeur du premier élément de la nouvelle ligne est toujours incrémentée de un en fonction de la dernière valeur de la ligne précédente), une séquence de t nombre consécutifs peut être seulement obtenue dans une seule ligne. Maintenant puisque nous avons seulement $t-1$ indices différents pour les $t-1$ colonnes, nous ne pourrons jamais collecter une séquence de t nombres à partir de la même ligne. Par conséquent, aucune séquence de t nombres consécutifs ne peut être collectée. □

Basé sur la construction de la matrice d'affectation en utilisant l'algorithme proposé pour chaque deux valeurs t et n vérifiant $2 < t < n-1$,

nous proposons un schéma de partage de secret basé sur les ACMLs dans la sous-section suivante. Un sous-ensemble de configurations est affecté à chaque participant en utilisant les indices de sa colonne correspondante à la matrice. Notez que l'algorithme retourne une nouvelle valeur du paramètre id qui représente la plus grande valeur des éléments de la matrice, et détermine le nombre de configurations de l'ACML à construire.

4.2.2 La phase d'initialisation :

Durant la phase d'initialisation, « le dealer » responsable de la génération des parts va tout d'abord définir les paramètres du schéma. Les étapes suivantes sont effectuées au cours de cette phase :

1. « Le dealer » génère $t-1$ entiers aléatoires de l'ensemble $\{0, \cdots, 127\}$ afin de définir les $t-1$ règles de transition $w_1, w_2, \cdots, w_{t-1}$, définissant les $t-1$ fonctions de transition locale $f_{w_1}, f_{w_2}, \cdots, f_{w_{t-1}}$.
2. « Le dealer » divise le secret S de $|S|$ octets en t parties $PS_1, PS_2, \cdots, PS_t$, de taille égale à $[|S|/t]$, chaque partie du secret définie une configuration de l'ACML comme suite :

$$C^{(0)} = PS_1, C^{(1)} = PS_2, \cdots, C^{(t-1)} = PS_t, \tag{4.4}$$

   si la valeur de $|S|$ n'est pas divisible sur t, un « padding » est fait pour compléter les cellules dans $C^{(t-1)}$.
3. « Le dealer » génère un nombre entier publique aléatoire $\alpha > n+1$. Ce paramètre est utilisé pour introduire une diffusion suffisante et une confusion entre les configurations de l'ACML et donc il produit d'autres configurations plus aléatoires dans les parts résultantes.
4. « Le dealer » construit une matrice d'affectation A en utilisant les paramètres t,n et $id = \alpha$. La valeur retournée de id (le plus grand élément de la matrice) est affectée à un nouveau paramètre β qui va être utilisé lors de l'étape suivante. Notez que α est le plus petit élément de la matrice A, alors que β est le plus grand.
5. « Le dealer » calcule l'évolution de β – ème ordre de l'ACML, à partir des configurations initiales $\{C^{(0)}, C^{(1)}, \cdots, C^{(t-1)}\}$:

$$\{C^{(0)}, C^{(1)}, \cdots, C^{(t-1)}, C^{(t)}, \cdots, C^{(n)}, \cdots, C^{(\alpha)}, \cdots, C^{(\beta-1)}, C^{(\beta)}\} \tag{4.5}$$

4.2.3 La phase de partage :

Durant la phase de partage, « le dealer » utilise les $\beta - \alpha + 1$ dernières configurations $\{C^{(\alpha)}, C^{(\alpha+1)}, \cdots, C^{(\beta)}\}$ générées pendant la phase précédente (phase d'initialisation) pour construire les parts qui vont être distribuées aux n participants $P_1, P_2, \cdots, P_n$ comme dans ce qui suit :

1. Pour chaque participant P_i, « le dealer » affecte un sous-ensemble de configurations S_i à partir de l'ensemble $\{C^{(\alpha)}, C^{(\alpha+1)}, \cdots, C^{(\beta)}\}$ ayant comme indice les valeurs de la $i^{\text{ème}}$ colonne à partir de la matrice d'affectation A :

$$S_i = \{C^{A[1,i]}, C^{A[2,i]}, \cdots, C^{A[C_{n-2}^{t-2}-1,i]}, C^{A[C_{n-2}^{t-2},i]}\}, \tag{4.6}$$

chaque participant recevra exactement C_{n-2}^{t-2} configurations différentes.

2. Pour chaque participant P_i, les configurations correspondantes de son ensemble S_i sont concaténées pour former la part correspondante. Les parts sont finalement distribuées aux participants à travers un canal sécurisé.

Les deux figures 4.2 et 4.3 montrent un aperçu du fonctionnement des deux phases du schéma proposé :

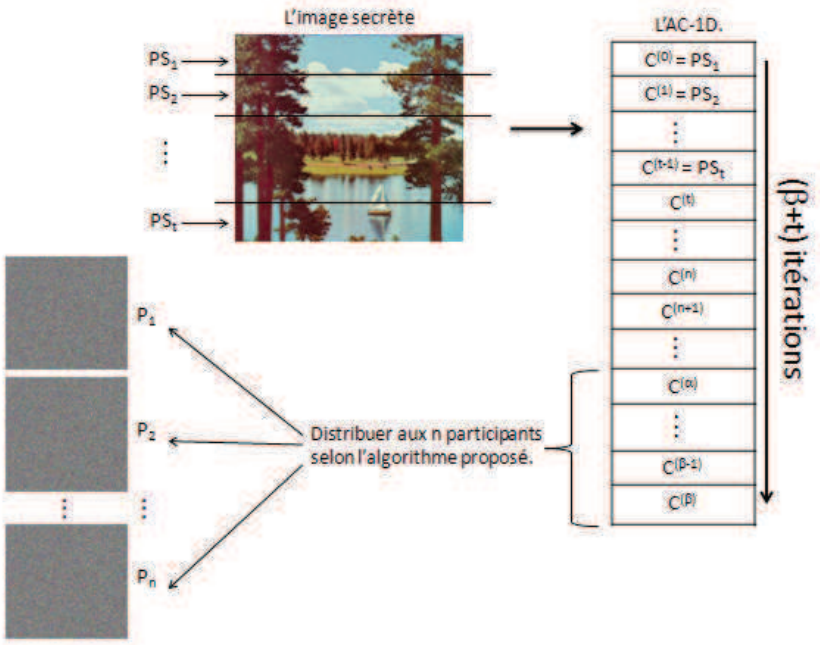


FIGURE 4.2 – Phase de partage proposée de l'approche 1.

Noter que les nombres de toutes les configurations distribuées aux n participants est égal au nombre des éléments de la matrice (égal à $n \times C_{n-2}^{t-2}$) qui est plus petit que le nombre total des configurations générées égal à $\beta - \alpha + 1$. Cette différence est due à deux facteurs : (1) le saut introduit entre les lignes consécutives durant la création de la matrice, et (2) la procédure de répétition introduite pour satisfaire les combinaisons appartenant à la même ligne.

4.2.4 La phase de reconstruction :

Durant la phase de reconstruction, le concessionnaire reconstruit le secret à partir d'au moins t parts appartenant à t participants différents $\{P_{i_1}, P_{i_2}, \cdots, P_{i_t}\}$ selon les étapes suivantes :

1. **Pour chaque participant P_{i_k}, la part correspondante est transformée en un ensemble de configurations selon les valeurs qui correspondent aux éléments de la $(i_k)^{\text{ème}}$ colonne de A. la part du participant P_{i_k} définie l'ensemble suivant (de C_{n-2}^{t-2} configurations) :**

$$S_{i_k} = \{C^{A[1,i_k]}, C^{A[2,i_k]}, \cdots, C^{A[C_{n-2}^{t-2},i_k]}\} \tag{4.7}$$

2. **Le concessionnaire construit l'ensemble S_U contenant l'union de tous les sous-ensembles de configurations des t participants comme suite :**

$$S_U = U_{i=i_1,\cdots,i_t} S_i \tag{4.8}$$

3. **Selon le lemme 4.2.1, il existe toujours une séquence de t nombres consécutifs dans chaque union de t colonnes différentes à partir de la matrice d'affectation A. Le concessionnaire détermine l'ensemble des indices consécutifs à partir de la matrice qui est noté $Seq = \{v_1, v_2, \cdots, v_t\}$. En utilisant Seq, le concessionnaire construit la séquence suivante :**

$$\{C^{(v_1)}, C^{(v_2)}, \cdots, C^{(v_t)}\}, \tag{4.9}$$

   qui représente l'ensemble des configurations consécutives dans l'ensemble S_U.

4. **Comme l'ensemble Seq est ordonné, v_1 est le plus petit entier de la séquence, le concessionnaire calcule alors l'évolution d'ordre (v_1) de l'ACML inverse construit durant la phase d'initialisation (en utilisant les mêmes règles de transitions w_i). l'ACML inverse est exécuté en utilisant les configurations suivantes :**

$$\{\tilde{C}^{(0)} = C^{(v_t)}, \tilde{C}^{(1)} = C^{(v_{t-1})}, \cdots, \tilde{C}^{(t-1)} = C^{(v_1)}\}, \tag{4.10}$$

   et l'évolution inverse de l'ACML est calculée avec v_1 itérations pour obtenir les t configurations représentantes initialement le secret qui a été partagé.

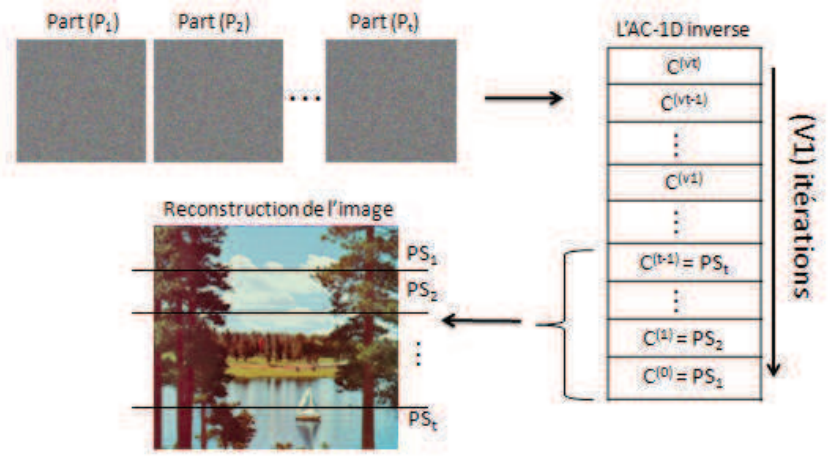


FIGURE 4.3 – Phase de reconstruction proposée de l'approche 1.

Selon la proposition 1 (Chapitre 2), les configurations initiales peuvent toujours être reconstruites. Ce qui fait que le secret peut être reconstruit par la concaténation des configurations découvertes.

4.3 Les résultats d'expérimentation :

Afin d'illustrer les étapes du schéma proposé, nous avons développé ci-dessous un exemple illustratif du partage de secret en utilisant l'approche proposée. Nous avons choisi d'appliquer le schéma pour partager une image en couleur (voir figure 4.4).

Soit un schéma à seuil $(3, 5)$ **de partage de secret à construire (i.e.** $t = 3$ **et** $n = 5$**). Nous définissons tout d'abord les paramètres du schéma en générant les règles de transition. Deux valeurs** w_1 **et** w_2 **sont choisies parmi l'ensemble** $\{0, \cdots, 127\}$ **: supposons** $w_1 = 41$ **et** $w_2 = 35$**. Nous attribuons aussi une valeur aléatoire pour le paramètre** α **(pour assurer une suffisante diffusion et l'aspect aléatoire des parts), nous avons choisi** $\alpha = 25$**. La matrice d'affectation A ayant 5 colonnes et** $C_{5-2}^{3-2} = C_3^1 = 3$ **lignes est générée en utilisant l'algorithme proposé : l'ensemble CS de** $t-2 = 1$ **combinaison dans l'ensemble** $\{2, 3, 4\}$ **est trivialement** $CS = \{2, 3, 4\}$ **(puisque la longueur des combinaisons est égale à un). Par conséquent on obtient la matrice suivante :**

$$A = \begin{bmatrix} P_1 & P_2 & P_3 & P_4 & P_5 \\ 25 & 26 & 27 & 27 & 27 \\ 29 & 29 & 30 & 31 & 31 \\ 33 & 33 & 33 & 34 & 35 \end{bmatrix} \tag{4.11}$$

Où chaque colonne correspond à un participant donné. Comme sortie

de l'algorithme, nous obtenons la valeur du paramètre $\beta = 35$ (la plus grande valeur des éléments de la matrice).

L'image secrète est ensuite décomposée en trois bloques définissants les trois configurations de l'ACML $C^{(0)}$,$C^{(1)}$ et $C^{(2)}$. Comme l'image contient $512 \times 512 = 262\ 144$ pixels chacun sur 3 octets (l'image de couleur est sur 24 bits), la taille de chaque configuration est $262\ 144 * 3/3 = 262\ 144$ octets. On construit à la fin un ACML d'ordre 3 et on l'évolue en utilisant les configurations $C^{(0)}$,$C^{(1)}$ et $C^{(2)}$ pour $\beta = 35$ itérations en utilisant les règles w_1 et w_2. On obtient finalement un ensemble de 35 configurations consécutives $\{C^{(0)}, C^{(1)}, \cdots, C^{(35)}\}$.

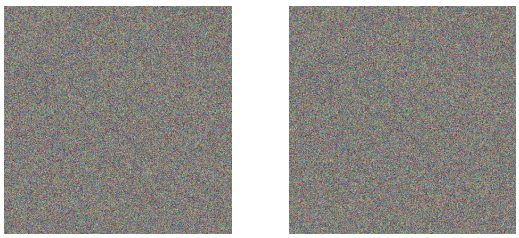
FIGURE 4.4 – L'image secrète (512×512) utilisée pour illustrer le schéma proposé 1.

En utilisant la matrice d'affectation A, les configurations obtenues sont affectées selon l'équation 4.6 afin de construire les différentes parts $S_i (1 \leq i \leq 5)$ comme suite :

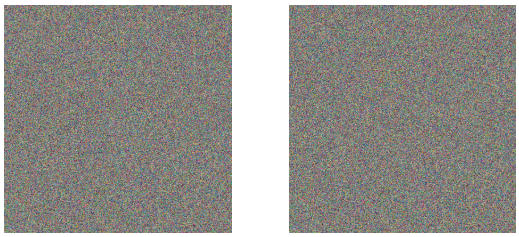
$$\begin{aligned} S_1 &= \{C^{(25)}, C^{(29)}, C^{(33)}\} \\ S_2 &= \{C^{(26)}, C^{(29)}, C^{(33)}\} \\ S_3 &= \{C^{(27)}, C^{(30)}, C^{(33)}\} \\ S_4 &= \{C^{(27)}, C^{(31)}, C^{(34)}\} \\ S_5 &= \{C^{(27)}, C^{(31)}, C^{(35)}\} \end{aligned} \tag{4.12}$$

Pour chaque participant, les configurations sont assemblées et concaténées pour former la part finale. Puisque chaque configuration est sur $262\ 144$ octets, la taille de la part est égale à $262\ 144 \times 3 = 786\ 432$ octets, et peut alors être représentée comme une image en couleur de 512×512 pixels (dans ce cas elle est de même taille que l'image secrète). La figure 4.5 illustre les cinq parts d'image obtenues quand on partage l'image de la figure 4.4 en utilisant l'approche proposée accordée au schéma de partage à seuil $(3, 5)$.

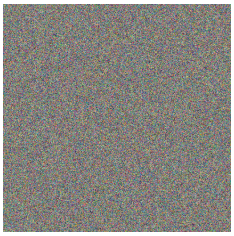
Il est clair à partir de la figure 4.5 que les parts obtenues ont un aspect aléatoire et par conséquent ne révèlent aucune information utile du secret original partagé. Afin de prouver ce fait, plusieurs expérimentations statiques ont été faites sur les parts afin de monter qu'elles sont



(a) P_1 (b) P_2



(c) P_3 (d) P_4



(e) P_5

FIGURE 4.5 – Les cinq parts d'images obtenues de taille (512×512).

indifférentes du bruit aléatoire.

Supposons qu'un sous-ensemble donné de participants se réunissent pour reconstruire l'image secrète. Puisque le schéma est à seuil $(3, 5)$, l'acquisition des parts d'au moins trois participants différents est nécessaire. Supposons que les participants sont P_2, P_4 et P_5, le concessionnaire collecte les parts ((b), (d) et (e)) des trois participants qui sont illustrées dans la figure 4.5. Chaque part est décomposée selon les configurations qui la composent, pour avoir les trois ensemble de configurations S_2, S_4 et S_5 définis dans l'équation 4.12. Le concessionnaire alors utilise la matrice publique d'affectation et construit l'ensemble contenant l'union $S_U = S_2 \cup S_4 \cup S_5$. Après ordonnancement l'ensemble S_U est défini par :

$$S_U = \{C^{(26)}, C^{(27)}, C^{(27)}, C^{(29)}, C^{(31)}, C^{(31)}, C^{(33)}, C^{(34)}, C^{(35)}\} \tag{4.13}$$

L'ensemble S_U contient la séquence des trois configurations consécutives $C^{(33)}$,$C^{(34)}$ et $C^{(35)}$. Selon le schéma de reconstruction, et puisque 33 est le plus petit indice des configurations, le concessionnaire exécute l'inverse de l'ACML (en utilisant les mêmes règles de transition publiques

w_1 et w_2) pour 33 itérations à partir des configurations initiales $C^{(35)}$,$C^{(34)}$ et $C^{(33)}$ (dans l′ordre inverse) pour reconstruire les configurations $C^{(2)}$,$C^{(1)}$ et $C^{(0)}$ qui définissent l′image secrète partagée du départ avec aucune perte d′information.

4.4 L′analyse de sécurité du schéma proposé :

Nous montrons dans ce qui suit que le schéma proposé vérifie la robustesse et la sécurité. La robustesse du schéma signifie que seul tous les ensembles d′au moins t participants peuvent récupérer le secret partagé, alors que ceux avec $t-1$ ou moins de participants ne permettent de révéler aucune information sur le secret. La sécurité du schéma signifie que les parts individuelles sont indifférentes des données générées aléatoirement. La robustesse est montrée ci-dessous en utilisant les hypothèses théoriques, alors que la sécurité du schéma est expérimentalement démontrée en utilisant l′aspect aléatoire des parts produites, nous aborderons ensuite une étude sur l′idéalité du schéma, et à la fin une comparaison est donnée entre quelques solutions existantes et la solution proposée.

4.4.1 La robustesse du schéma :

La robustesse du schéma proposé est liée aux faits suivants :

a) Un mécanisme d′ACML d′ordre t peut reconstruire les t configurations initiales à partir de tout sous-ensemble de t configurations consécutives en utilisant l′ACML inverse correspondant aux règles de transitions linéaires définies ;

b) La reconstruction de t configurations initiales d′un ACML est impossible quand on utilise un nombre de $t-1$ ou moins de configurations ;

c) La matrice d′affectation construite utilisant l′algorithme correspondant proposé assure que l′union des éléments de chaque t colonnes permet de reconstruire une séquence de t valeurs consécutives, alors que l′union de chaque $t-1$ ou moins de colonnes ne le permet pas.

Le premier fait est établi par la preuve de la proposition 1 (Chapitre 2). Il est montré dans [Fredkin 1990] qu′un ACML construit selon les équations 2.6 et 2.7 est toujours réversible, alors que son inversion en utilisant $t-1$ ou moins de configurations est un problème de combinaison difficile. Par conséquent les faits (a) et (b) sont prouvés.

Le fait (c) peut être montré en utilisant le lemme 4.2.1 nous avons établi que l′algorithme proposé pour la construction de la matrice produit toujours des matrices valides. Leur validité est considérée en respectant les deux contraintes du lemme, qui correspondent exactement aux exigences du fait (c). En combinant la vérifiabilité des trois faits (a), (b) et (c), nous concluons que le schéma proposé est robuste.

4.4.2 Les tests statistiques :

L'objectif d'un test statistique est de déterminer l'aspect aléatoire d'une séquence « c » en testant les séquences de bit générées à partir de différents tests statistiques, et ceci en utilisant des batteries de tests dont voici quelques une que nous avons utilisé pour tester les résultats obtenus du schéma proposé :

4.4.2.1 La batterie de test Diehard :

Fournie par un laboratoire de recherche universitaire, Les tests Diehard sont une série de tests empiriques au nombre de 15. Ils ont été originellement écrits en fortran par G. Marsaglia, puis portés en C (DiehardC, la version C de DIEHARD). Une version graphique est également en cours de développement.

Il s'agit de tests basés sur des manipulations de bits ou de matrices obtenues avec l'échantillon. Pour plus de détails, la sortie du programme précise mathématiquement chaque test avant de donner les résultats. Ces tests retournent une valeur de c^2 qui ne doit pas être 0 ou 1, ou trop proche de ces valeurs. Le test est considéré réussi si le c^2 est compris entre 0,025 et 0,975.

Ce test est néanmoins très gourmand, si de nos jours les machines ne mettent que moins de quelques minutes à effectuer les tests, il faut néanmoins un échantillon d'environ 10 Mo pour parvenir à effectuer tous les tests [Nicolas 2000].

Les parts obtenues par l'exemple pris dans la section 4.3 ont été analysés en utilisant la batterie de tests Diehard, les résultats sont reportés dans le tableau suivant :

Test Name	Averaged P-value	Interpretation
BIRTHDAY SPACINGS	0.910134	Pass
OVERLAPPING PERMUTATION	0.972213	Pass
RANK TEST 31x31	0.692522	Pass
RANK TEST 32x32	0.562338	Pass
MONKEY DNA	0.781003	Pass
COUNT-THE-1's TEST	0.578523	Pass
PARKING LOT	0.359782	Pass
MINIMUM DISTANCE	0.294752	Pass
RUNDOM SPHERE	0.568922	Pass
The SQEEZE test	0.847215	Pass
OVERLAPPING SUMS	0.684211	Pass
The RUNS -up test, down test-	0.920325 0.531017	Pass

CRAPS -no of wins, thrwos/game-	**0.875423 0.72087**	**Pass**
RANK TEST	**0.835214**	**Pass**
MONKEY 20 BITS PER WORD	**0.870254**	**Pass**
MONKEY OPSO, OQSO	**0.915472**	**Pass**

Tableau 4.1 – Les résultats de la batterie de tests Diehard de l′approche proposée 1.

4.4.2.2 La batterie de test ENT :

Fournie par un organisme de standard technologique, le programme de test ENT applique cinq différents tests pour des séquences d′octets stockées dans des fichiers et communique le résultat de ces tests. Le programme est utile pour évaluer les générateurs de nombre aléatoire pour le chiffrement et les applications statistiques d′échantillonnage, et d'autres applications où la densité de l'information d'un fichier est d'intérêt.

Les parts obtenues par l′exemple pris dans la section 4.3 ont été également analysés en utilisant la batterie de tests ENT, les résultats sont reportés dans le tableau qui suit :

Test	Value	Norm
Entropy	7.999925	$Max = 8.0$
Arithmetic mean	127.209	$127.5 = random$
Monte Carlo 0.00059	3.1485967(error=0.24)	π value
Serial correlation coefficient	0.00059	0.0
Optimum compression	0.00001	0.0

Tableau 4.2 – Les résultats de la batterie de tests ENT de l′approche proposée 1.

- **A partir des résultats obtenus par les deux tableaux, on peut facilement voir que les parts contiennent de très bonnes propriétés statiques puisqu′elles passent la majorité des tests appliqués. Ces résultats impliquent que les parts sont indifférentes des images aléatoires, et par conséquent, le schéma de partage est sûr.**

4.4.2.3 L′histogramme :

Une image histogramme montre comment les pixels dans une image graphique sont distribués en traçant le nombre de pixels correspondant à chaque intensité de couleur. En ce qui concerne notre travail, nous avions choisi de traiter des images secrètes en couleurs, les tracés et l′analyse des histogrammes obtenus par les résultats de l′exemple précédent (section 4.3) sont donnés ci-dessous :

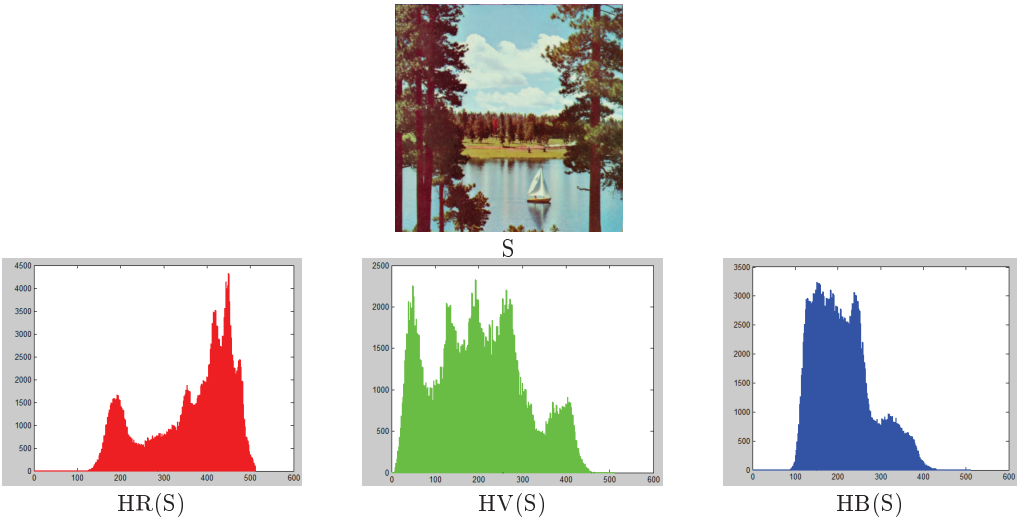
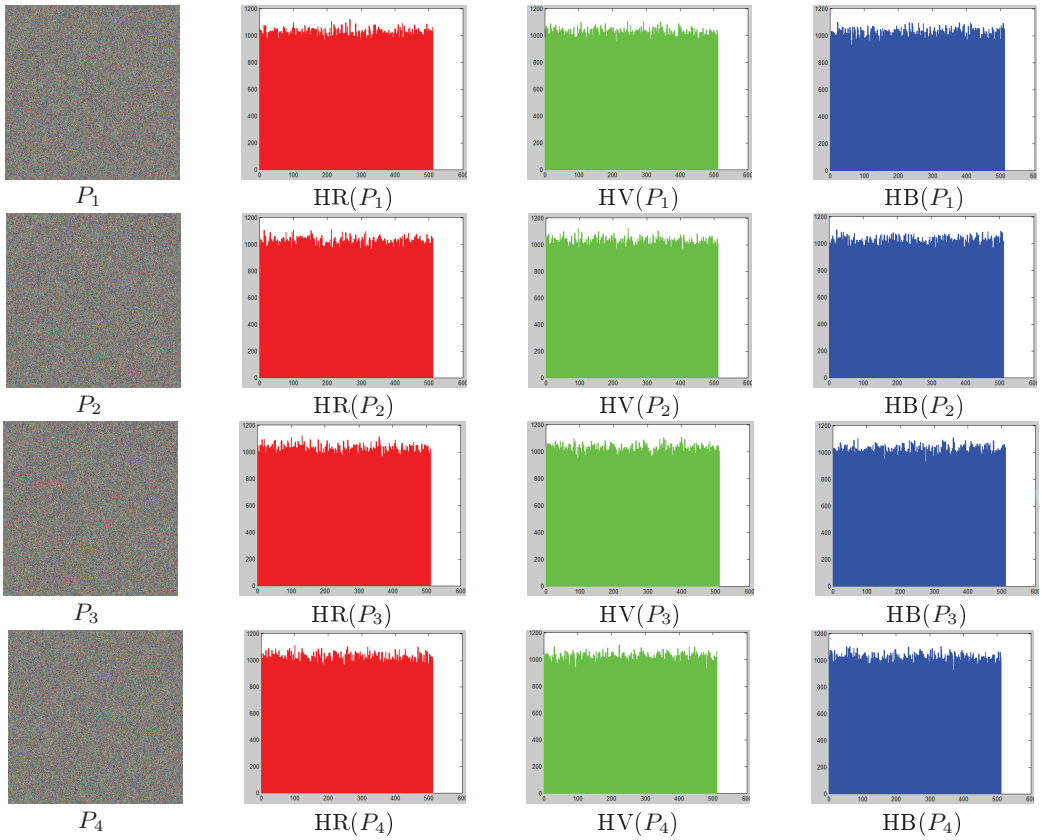
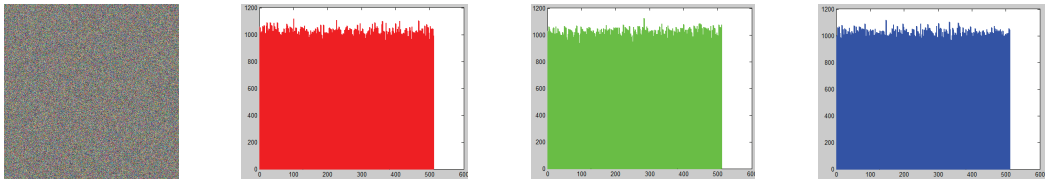


FIGURE 4.6 – L'image secrète (S) et ces histogrammes respectifs HR(S), HV(S) et HB(S).

Ci-dessus, les histogrammes (RVB) correspondants à l'image secrète « S ». Les histogrammes concernant les cinq parts d'images sont donnés dans la figure qui suit :





P_5 HR(P_5) HV(P_5) HB(P_5)

FIGURE 4.7 – Les parts d′images attribuées aux 5 participants et leurs histogrammes respectifs.

Nous remarquons que les histogrammes des parts d′images produites attribuées aux 5 participants sont uniformément distribués par rapport aux histogrammes de l′image secrète ; ce qui fait que le schéma de partage proposé rend la dépendance des propriétés statistiques des parts d′images et de l'image secrète quasi aléatoires.

4.4.2.4 Corrélation des pixels adjacents :

Dans des images claires, les pixels voisins sont fortement corrélés, ces derniers peuvent être horizontalement, verticalement ou diagonalement adjacents. On étudie dans ce qui suit la corrélation entre les pixels adjacents de l′image secrète ainsi que des parts d′images attribuées aux cinq participants (pour le schéma $(3, 5)$ vu précédemment). La corrélation entre les pixels adjacents est calculée par les équations suivantes :

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i; \tag{4.14}$$

$$Cov(x, y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)); \tag{4.15}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2; \tag{4.16}$$

$$r_{xy} = \frac{Cov(x, y)}{\sqrt{D(x)}.\sqrt{D(y)}} \tag{4.17}$$

Où x et y sont des valeurs de deux pixels adjacents dans l′image ; $E(x)$ est l′estimation de l′espérance mathématique de x ; $Cov(x, y)$ est l′estimation de la covariance entre x et y ; $D(x)$ est l′estimation de la variance de x ; et r_{xy} une mesure statistique permettant de donner une idée sur la capacité de résister aux attaques qui réduisent l′espace d′une recherche exhaustive.

Nous avons choisi au hasard 26 214 *paires* de deux pixels adjacents (10% du contenu de l′image) dans chaque image, ensuite nous avons calculé les coefficients de corrélation entre les pixels adjacents r_{xy} de chaque paire en

utilisant les formules précédentes. Le résultat est donné dans le tableau suivant :

	L'image secrète (S)	Les parts d'images				
		P_1	P_2	P_3	P_4	P_5
Horizontale	0.95468644	0.00421311	0.00168088	0.00372221	0.00266158	0.00334804
	0.96602840	0.00136965	0.00102929	0.00449957	0.00652558	0.01050025
	0.96853450	0.00300805	0.00620248	0.00345756	0.00314551	0.01031825
Verticale	0.95608853	0.00200226	0.00056388	0.00634062	0.00734023	0.00229356
	0.97144801	0.00327909	0.00935689	0.00123518	0.00226171	0.00811948
	0.97174259	0.00337598	0.00784140	0.00922335	0.00505332	0.00668062
Diagonale	0.94117830	0.00530222	0.01301617	0.00973062	0.00062397	0.00787370
	0.95252594	0.00285020	0.00125183	0.00950160	0.00771283	0.00290177
	0.95345991	0.00521881	0.00221661	0.00565583	0.00750014	0.00188833

Tableau 4.3 – Coefficients de corrélation entre deux pixels adjacents dans chaque image.

- **A partir du tableau 4.3, nous pouvons remarquer que les coefficients de corrélation pour l'image secrète sont voisin de 1 ce qui montre que les pixels sont fortement corrélés. Alors que pour les parts d'images attribuées aux 5 participants, les coefficients de corrélation sont voisins de 0 ce qui prouve qu'il n'y a pas de corrélation entre l'image secrète et les parts d'images, et donc pas de similitude entre eux.**

Les tracés de corrélation des pixels adjacents horizontaux, verticaux et diagonaux voisins de l'image secrète et des parts d'images attribuées aux 5 participants sont donnés dans ce qui suit :

a) **Corrélation horizontale**
La figure 4.8, montre les distributions de deux pixels adjacents horizontaux de l'image secrète S :

S

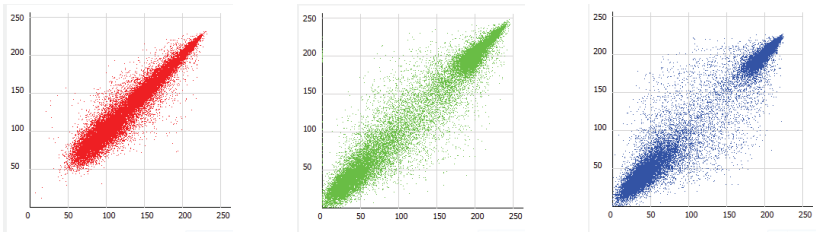


FIGURE 4.8 – Analyse de corrélation de deux pixels adjacents horizontaux de l′image secrète.

La figure 4.9 montre les distributions de deux pixels adjacents horizontaux des parts d′images obtenues.

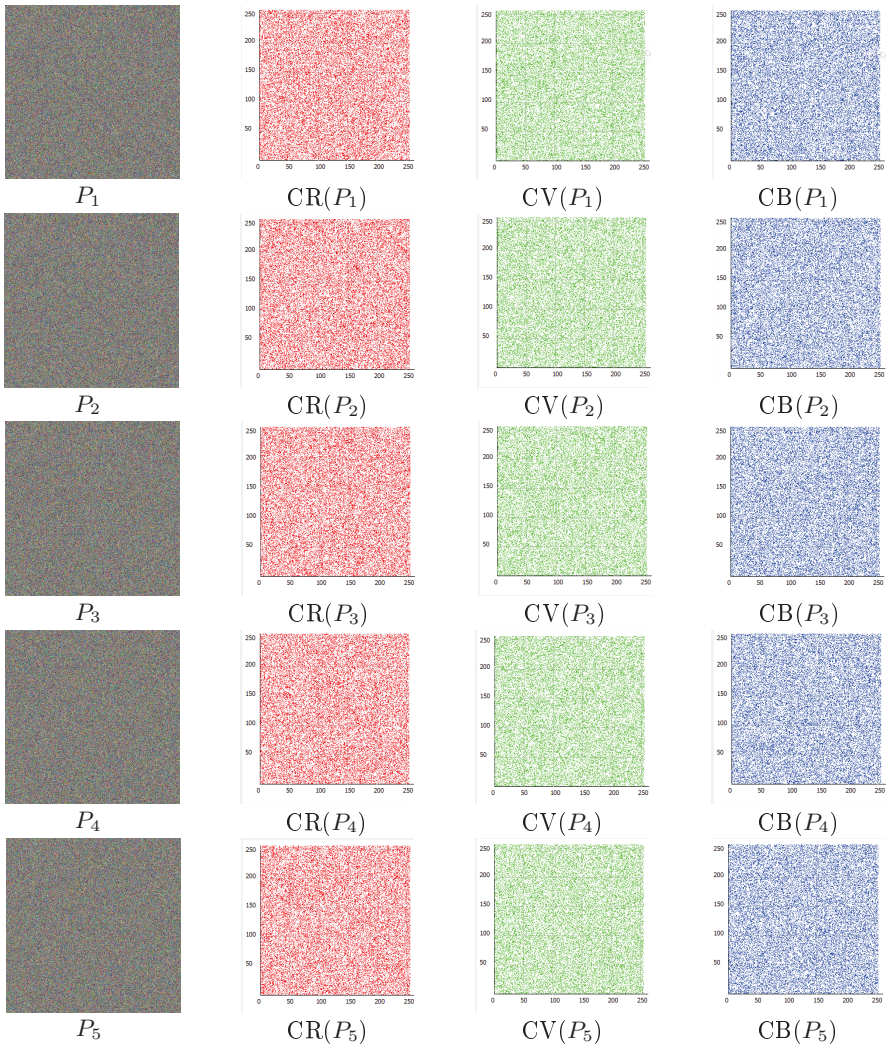


FIGURE 4.9 – Analyse de corrélation de deux pixels adjacents horizontaux des 5 parts d′images produites.

b) Corrélation verticale

La figure 4.10 montre les distributions de deux pixels adjacents verticaux de l'image secrète S :

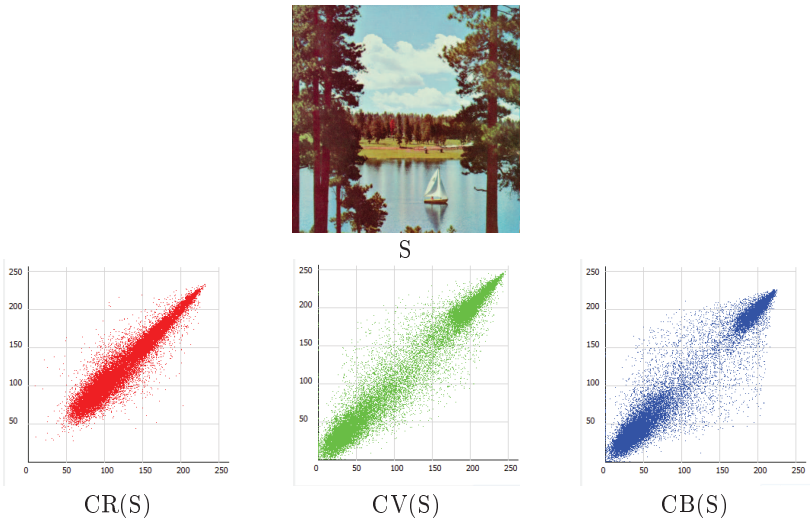
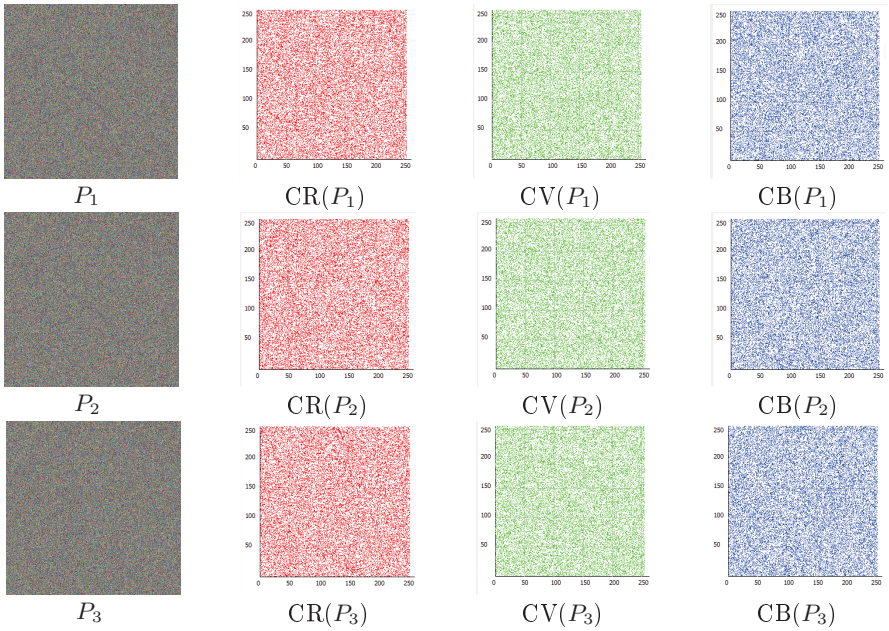


FIGURE 4.10 – Analyse de corrélation de deux pixels adjacents verticaux de l'image secrète.

La figure 4.11 montre les distributions de deux pixels adjacents verticaux des parts d'images obtenues.



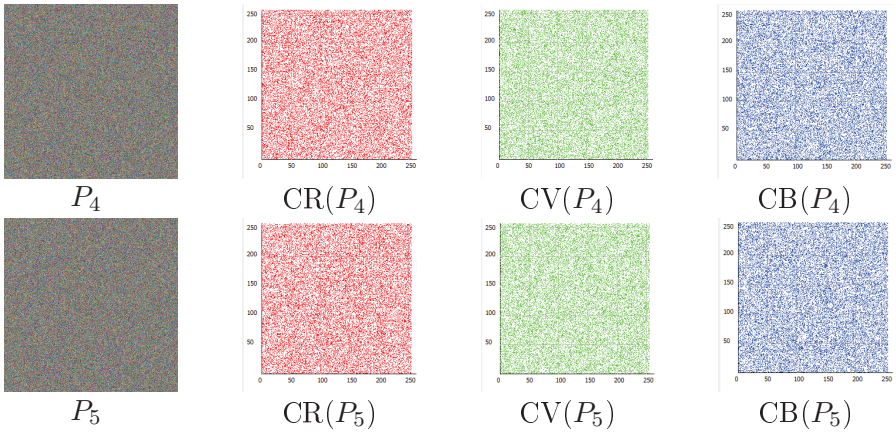


FIGURE 4.11 – Analyse de corrélation de deux pixels adjacents verticaux des 5 parts d′images produites.

c) Corrélation diagonale

La figure 4.12 montre les distributions de deux pixels adjacents diagonaux de l′image secrète S :

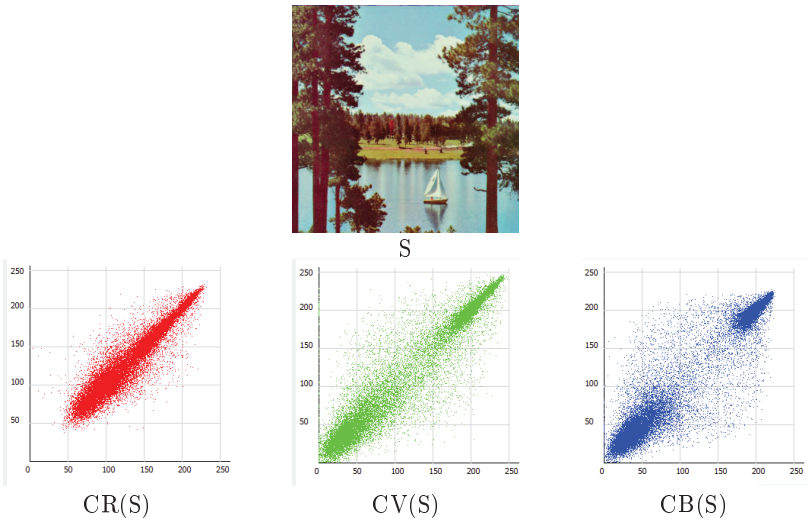


FIGURE 4.12 – Analyse de corrélation de deux pixels adjacents diagonaux de l′image secrète.

La figure 4.13 montre les distributions de deux pixels adjacents diagonaux des parts d′images obtenues.

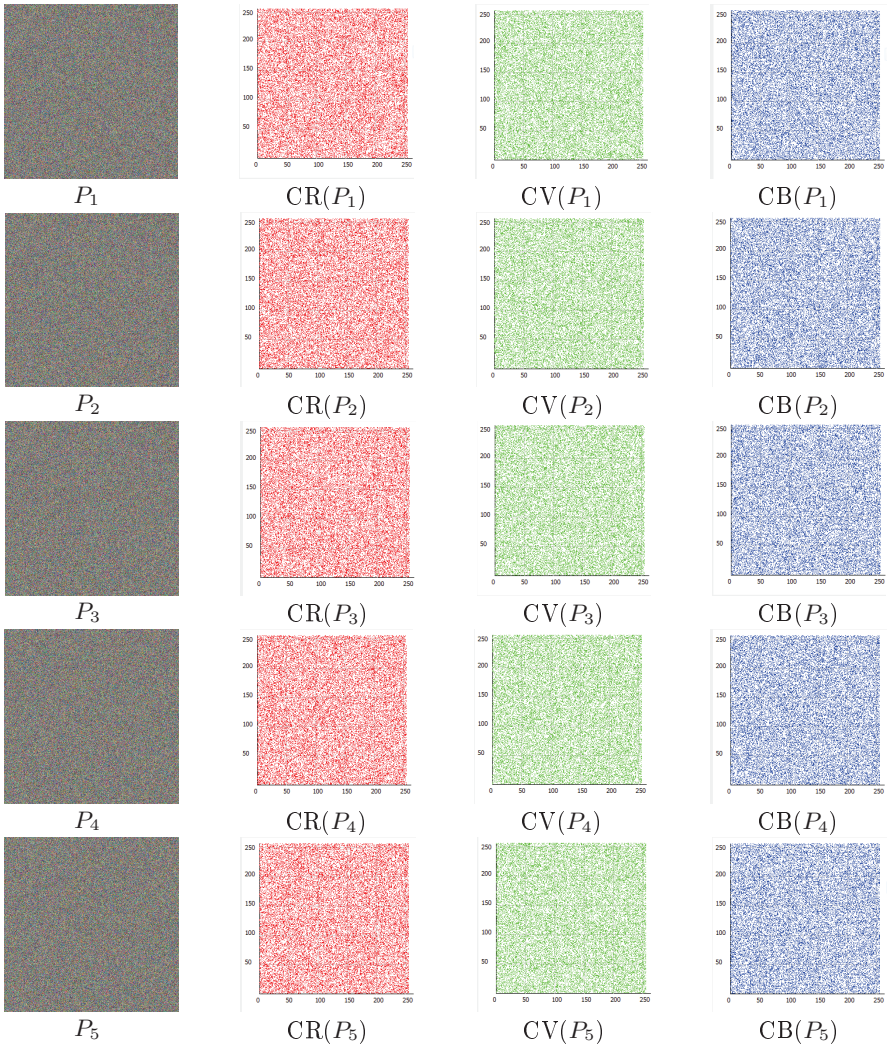


FIGURE 4.13 – Analyse de corrélation de deux pixels adjacents diagonaux des 5 parts d′images produites.

- ✤ **Les figures 4.8, 4.10 et 4.12 représentent les corrélations de pixels adjacents de l′image secrète s′alignant sur la première bissectrice montrant ainsi une forte corrélation entre les pixels adjacents.**
- ✤ **Les figures 4.9, 4.11 et 4.13 montrent une désamination presque de manière aléatoire des pixels adjacents appartenant aux parts d′images attribuées aux 5 participants, ce qui renvoi à un algorithme robuste à toute attaque statistique.**

4.4.2.5 La taille des parts et l′étude de l′idéalité du schéma :

Comme le schéma proposé affecte plusieurs configurations à la même personne, la taille de la part est parfois plus grande que la taille du secret. Selon le schéma de partage, si on considère que la taille du secret est don-

née par |S|, le seuil et le nombre de participant sont t et n respectivement, alors la taille estimée de chaque part est donnée par :

$$|Part| = \frac{C_{n-2}^{t-2} * |S|}{t} \tag{4.18}$$

Dépendant du seuil et du nombre de participant, la taille de la part diffère et, est expérimentalement grande pour quelques combinaisons de t et n. Nous avons étudié l′évolution de la taille des parts par rapport aux paramètres du schéma afin de définir les conditions efficaces. La figure 4.14 montre l′évolution de la taille des parts par rapport à t et n. Nous remarquons d′après la figure que le schéma s′avère efficace dans une certaine région suffisamment large d′espace au contraire de l′autre. Selon ces résultats, on conclut que l′utilisation du schéma proposé est conditionnée par la relation entre les valeurs de t et n, et même si le schéma assure une robustesse complète. Il est largement inidéal quand t est dans le voisinage de n/2.

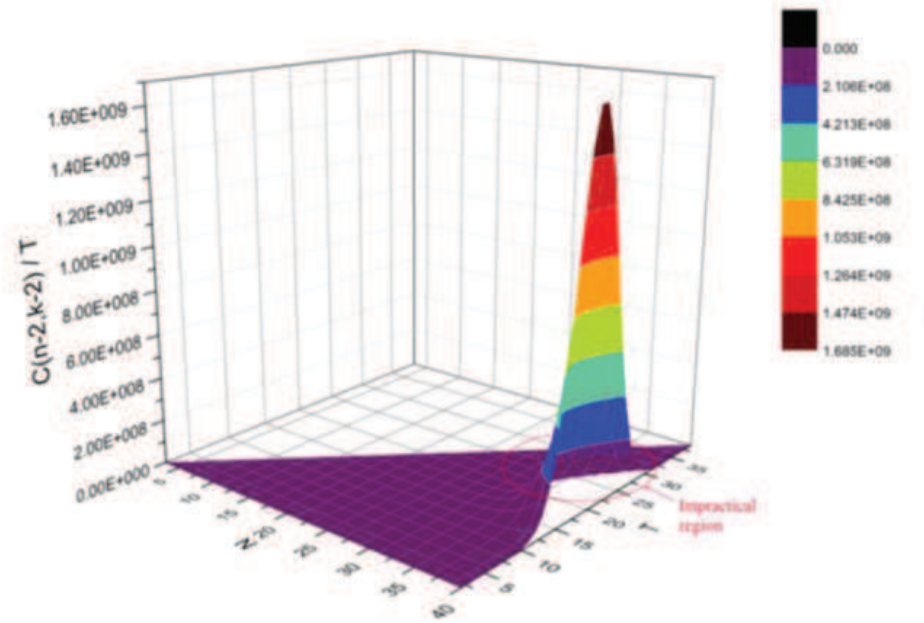


FIGURE 4.14 – Estimation entre la taille du secret et celle des parts par rapport aux valeurs possibles (t, n).

Pour d′autres illustrations des propriétés et capacités du schéma proposé, une étude comparative a été donnée dans le tableau 4.4, où le schéma est comparé à quelques autres schémas de partage récents existants par rapport à plusieurs paramètres de performance. Avec respect aux schémas basés sur les ACs, le schéma proposé est le seul qui assure

la robustesse, alors qu'à l'idéalité, elle est assurée seulement si t et n n'appartiennent pas à la région inadaptée. La complexité de calcule linéaire est un autre avantage du schéma qui n'est pas assurée par tous les schémas n'utilisant pas les ACMLs, et qui ont au moins une complexité polynomiale.

(t,n) Sharing Scheme	Sharing complexity	Reconstruction complexity	Lossless	Scalability for large data	Robustness
Thien et al.[Thien 2002]	$O(\|S\| * nlog^2 n)$	$O(\|S\| * tlog^2 t)$	Yes	No	Yes
Yang et al.[Yang 2011]	$O(\|S\| * nlog^2 n)$	$O(\|S\| * tlog^2 t)$	No	No	No
Lin et al.[Lin 2010]	$O(\|S\| * nlog^2 n)$	$O(\|S\| * tlog^2 t)$	Yes	No	Yes
Hadian et al.[Dehkordi 2008]	$O(\|S\| * nlog^2 n)$	$O(\|S\| * tlog^2 t)$	Yes	No	Yes
Eslami et al.[Eslami 2010]	$O(\|S\| * n)$	$O(\|S\| * t)$	Yes	Yes	No
Wu et al.(2012)[Wu 2012]	$O(\|S\| * n)$	$O(\|S\| * t)$	No	Yes	No
Wu et al.(2013)[Wu 2013]	$O(\|S\| * n)$	$O(\|S\| * t)$	No	Yes	No
Proposed	$O(\|S\| * n)$	$O(\|S\| * t)$	Yes	Yes	Yes

Tableau 4.4 – Comparaison entre les schémas de partage existants avec le schéma proposé 1.

4.5 Conclusion :

Cette première solution proposée à été réalisée dans le but de répondre au problème de robustesse commun à tous les schémas de partage de secret basés sur les ACMLs. Dont seulement les participants détenant des parts consécutives sont autorisés à reconstruire le secret, alors que le reste des participants dont un nombre très grand ne le sont pas.

Pour résoudre le problème nous avons proposés d'affecter plusieurs configurations au lieu d'une seule à chaque utilisateur ceci afin de permettre à chaque sous-ensemble d'au moins t participants de reconstruire complètement le secret qui a été partagé. Pour cela une matrice d'affectation spécifique a été heuristiquement générée en utilisant un algorithme que nous avons proposé, et utilisé pour définir un nouveau mécanisme de partage/reconstruction. En plus de la robustesse, le schéma proposé c'est avéré sûr puisque aucune information des parts attribuées

aux n participants n′est révélée, et idéal dans certain cas (selon les valeurs prises des paramètres t et n, comme vu précédemment).

Chapitre 5

Nouvelle approche de partage de secret à seuil par le biais des systèmes surdéterminés sur des corps de Galois.

Sommaire

5.1 Introduction :

Dans ce chapitre, on présente une deuxième solution pour le problème de partage multi-secret cette fois-ci, ce dernier a déjà été traité bien plus qu'une fois en utilisant différentes techniques basées sur les schémas standards de Shamir, Blakely et CRT, avec bien plus d'autres, mais tout de même la complexité de calcule de ces schémas et leur utilisation dans un corps fini de Rijndael ($GF(2^8)$) entrainent quelques inconvénients, pour cela nous avons proposé un nouveau schéma de partage multi-secret basé sur l'utilisation des systèmes d'équations linéaires surdéterminés définis sur un corps fini de Rijndael ($GF(2^8)$). Cette nouvelle solution permet de faire le partage et la reconstruction des secrets en un temps linéaire par rapport à leurs tailles, au nombre de participant et au seuil de partage. En plus, l'utilisation d'un corps de Rijndael permet d'avoir un codage optimal des pixels par rapport au corps premiers.

Nous décrirons dans ce chapitre quelques éléments mathématiques de base, tel que les systèmes d'équations linéaires et les principes des corps finis qu'on va exploiter dans notre schéma, suivit d'une description détaillée du schéma de partage multi-secret proposé, nous expérimenterons le schéma par des exemples, puis nous finirons avec une analyse sur la sécurité du schéma.

5.2 Les corps finis de Galois :

La structure de corps fini intervient dans divers domaines mathématiques, en particulier dans la théorie de Galois sur la résolution des équations algébriques où ils se sont introduits pour la première fois. Pour cette raison, en hommage au mathématicien français Evariste Galois (1811-1832), ces corps sont appelés les corps de Galois.

Définition 8. *(Anneau commutatif). Un anneau est un ensemble A muni de deux lois internes :*

- ☞ *La loi additive structure A en un groupe abélien (commutatif),*
- ☞ *La loi multiplicative est associative et distributive par rapport à l'addition.*

- ➦ *Un anneau commutatif est un anneau dont la multiplication est elle aussi commutative.*

Un corps fini est un anneau commutatif sur lequel tous les éléments à l'exception de l'élément neutre de la loi additive, possède un inverse par rapport à la loi multiplicative [McEliece 2012].

Un corps fini de Galois est un corps dont le cardinal qu'on appel son ordre est fini, et qui est représenté par un nombre primaire p^r (puissance

entière d'un nombre premier), les opérations de multiplication commutative, addition, soustraction et de division sont explicitement définies.

Un corps fini à p^r éléments noté $GF(p^r)$, caractérisé par le nombre premier p et de dimension r (entier positif), est isomorphe au corps des polynômes de degrés strictement inférieur à r (à coefficients dans $GF(p)$) modulo un polynôme (irréductible dans $GF(p)$) de degré r. $GF(p)$ est le corps de base (ou sous corps premier, ground field) pour $GF(p^r)$.

5.2.1 Les corps de Rijndael (GF(2^8)) :

Les corps finis de Rijndael sont un cas particulier des corps de Galois ayant une dimension r égal à 8, et sont généralement dénotés par $GF(2^8)$. Les éléments du coprs de Rijndael sont définis sur 8 bits (1 octet), et toute opération arithmétique liée est comme un résultat défini en utilisant les mêmes formes. L'addition et la soustraction sur $GF(2^8)$ sont perfomées en utilisant l'opération du ou exclusive (le xor) ; alors que la multiplication et la division sont performées en utilisant des algorithmes spécifiquement développés sur la base de multiplication polynomiale modulo un polynôme irréductible de degré 8 [Silverman 1999]. Pour des applications pratiques, on utilise des tables de consultation pour implémenter la multiplication, la division, l'exponentiation et l'inverse multiplicatif qui fournissent des calculs extrêmement rapides et conduisent à des performances optimales lors de l'application des régimes à base de Rijndael [Nagaraj 2010]. L'Advanced Encryption Standard (AES) est l'une des applications les plus connues dans le domaine Rijndael assurant des fonctions de chiffrement/déchiffrement optimisées en utilisant l'implémentation des tables de consultation.

Puisque le champ Rijndael englobe 256 valeurs possibles d'un octet, son utilisation dans la représentation de pixels d'image est optimale et efficace en terme d'espace. Les deux éléments, secrets et parts peuvent être traités au sein de la même structure (pixel), et aucune réduction/agrandissement est nécessaire contrairement aux champs premiers nécessitant de telles opérations. Par conséquent, l'utilisation de ce champ dans le schéma proposé fournit un partage d'images sans perte et une représentation de parts optimales .

5.3 Les systèmes d'équations linéaires :

En mathématiques et particulièrement en algèbre linéaire, un système d'équations linéaires est un ensemble d'équations linéaires qui portent sur les mêmes inconnues.

Définition 9. *(Système d'équations linéaires). Un ensemble fini d'équations linéaires dont les variables $x_1, \cdots, x_n$ doivent êtres toutes vérifiées par les éven-*

tuelles solutions du système (une solution étant un $n-$uplet de valeurs particulières $b_1, \cdots, b_n$).

En général, un système linéaire de n équations à t inconnues se présente sous la forme suivante :

$$\begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,t}x_t = b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,t}x_t = b_2 \\ \vdots \\ a_{n,1}x_1 + a_{n,2}x_2 + \cdots + a_{n,t}x_t = b_n \end{cases} \tag{5.1}$$

Où $x_1, \cdots, x_t$ sont les inconnues, et les nombres $a_{i,j}$ les coefficients du système, la solution est un n-uplet appartenant au corps fini F_q et qui satisfait à la fois les n équations. Un système d'équations linéaires peut aussi s'écrire sous la forme matricielle : $A.X = B$ avec :

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,t} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,t} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,t} \end{pmatrix}; X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_t \end{pmatrix} etB = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \tag{5.2}$$

On peut distinguer trois sortes de système d'équations linéaires dont voici ci-dessous :

5.3.1 Les systèmes d'équations linéaires indéterminés :

Dans le cas où le nombre d'équation (n) est inférieur au nombre d'inconnues (t) $(n < t)$, le système est appelé indéterminé, et il admet soit un nombre infini de solutions ou pas de solution.

5.3.2 Les systèmes d'équations linéaires déterminés :

Dans le cas où le nombre d'équation (n) est égal au nombre d'inconnus (t) $(n = t)$, le système est appelé déterminé, et il admet généralement une solution unique si la matrice A est régulière/inversible (de rang maximal) et admet un inverse A^{-1} sur le corps F_q. La solution du système est donnée par $X = A^{-1}.B$;

Proposition 3. *Les éléments $C_1, \cdots, C_n$ de R^n sont linéairement indépendants si, et seulement si on a $det(C_1, \cdots, C_n) \neq 0$.*

Définition 10. *Soit $A \in R_p^n$. Le rang de A est le nombre maximal de colonnes de A linéairement indépendantes (dans R^n). On le note rg(A) ou p(A).*

Proposition 4. *Le rang d'une matrice $A \in R_p^n$ est l'ordre du plus grand mineur non nul de A. c'est aussi le nombre maximal de lignes linéairement indépendantes de A.*

Theorem 5.3.1. *(Matrice inversible et rang). Une matrice carrée de taille n est inversible si et seulement si elle est de rang n.*

Démonstration. Soit A une matrice carrée d'ordre n. Soit f l'endomorphisme de K^n dont la matrice dans la base canonique est A. On a les équivalences suivantes :

$$\begin{aligned} A \text{ de rang } n &\leftrightarrow f \text{ de rang } n \\ &\leftrightarrow f \text{ surjective} \\ &\leftrightarrow f \text{ bijective} \\ &\leftrightarrow A \text{ inversible} \end{aligned}$$

Nous avons utilisé le fait qu'un endomorphisme d'un espace vectoriel de dimension finie est bijectif si et seulement s'il est surjectif et le théorème sur la caractérisation de la matrice d'un isomorphisme.

□

5.3.3 Les systèmes d'équations linéaires surdéterminés :

Si le nombre d'équations (n) est supérieur au nombre de variables (t) $(n > t)$, le système est appelé surdéterminé, et il admet soit une solution, un nombre infini de solutions ou pas de solution, selon les caractéristiques de la matrice A et la relation d'indépendance qui existe entre les vecteurs des lignes de la matrice.

5.3.3.1 Système d'équations linéaires surdéterminé ayant une seule solution :

Pour construire un système surdéterminé d'équations linéaires qui admet une solution unique $S = (s_1, \cdots, s_t)$, il suffit de construire une matrice $A_{n\times t}$ de rang plein, tels que les vecteurs définis par toute combinaison de t lignes de A soient linéairement indépendants. La résolution du système peut être effectuée en inversant toute sous-matrice d'ordre $(t \times t)$ à partir de A, et en multipliant le résultat par les coefficients correspondants au sous vecteur B.

Pour des applications pratiques, deux principales matrices fournissent la propriété du rang plein sur un corps fini donné F_q, est qui sont comme suite :

La construction d'une matrice non carrée de rang plein peut être effectuée en utilisant plusieurs matrices spécifiquement définies tels que la matrice de Vandermonde ou la matrice de Cauchy. Ces deux types de matrices fournissent la propriété du rang plein et peuvent être construites sur un domaine fini quelconque pour des dimensions données n et t en utilisant des formulations spécifiques.

a) **La matrice de Vandermonde :**
La matrice de Vandermonde ayant les termes d'une progression géométrique dans chaque ligne, peut être construite en utilisant un vecteur de n éléments distincts $(\alpha_1, \alpha_2, \cdots, \alpha_n)$ de F_q comme suite :

$$V_{ij} = (\alpha_i^{j-1})_{\substack{i=1,\cdots,t \\ j=1,\cdots,n}}, V = \begin{pmatrix} 1 & \alpha_1 & \alpha_1^2 & \cdots & \alpha_1^{t-1} \\ 1 & \alpha_2 & \alpha_2^2 & \cdots & \alpha_2^{t-1} \\ \vdots & \vdots & \ddots & \vdots & \\ 1 & \alpha_n & \alpha_n^2 & \cdots & \alpha_n^{t-1} \end{pmatrix} \tag{5.3}$$

Soit M une matrice de Vandermonde, le déterminant d'une telle matrice est $det(M) = \prod_{1 \leq j < k < t}(x_{i_k} - x_{i_j})$ mod p. Comme tous les x_{i_j} sont distincts dans $Z_p, x_{i_k} - x_{i_j} \neq 0$ mod p et comme Z_p est un corps, $det(M) \neq 0$.

b) **La matrice de Cauchy :**
La matrice de Cauchy est une autre matrice de rang plein qui nécessite moins de temps de calcul. En outre, il a été montré que chaque sous-matrice d'une matrice de Cauchy est également une matrice de Cauchy. Une matrice de Cauchy d'ordre $(n \times t)$ est construite en utilisant deux séquences d'éléments $(\alpha_1, \alpha_2, \cdots, \alpha_n)$ et $(\beta_1, \beta_2, \cdots, \beta_t)$ de F_q, en utilisant la définition suivante :

$$C_{ij} = \left(\frac{1}{\alpha_i + \beta_j}\right)_{\substack{j=1,\cdots,t \\ i=1,\cdots,n}}, C = \begin{pmatrix} \frac{1}{\alpha_1+\beta_1} & \frac{1}{\alpha_1+\beta_2} & \cdots & \frac{1}{\alpha_1+\beta_t} \\ \frac{1}{\alpha_2+\beta_1} & \frac{1}{\alpha_2+\beta_2} & \cdots & \frac{1}{\alpha_2+\beta_t} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\alpha_n+\beta_1} & \frac{1}{\alpha_n+\beta_2} & \cdots & \frac{1}{\alpha_n+\beta_t} \end{pmatrix}, \alpha_i + \beta_j \neq 0 \ \forall i, j, \tag{5.4}$$

$(\frac{1}{\alpha_i+\beta_j})$ est le multiplicatif inverse de $(\alpha_i + \beta_j)$ sur F_q, et $\alpha_i + \beta_j \neq 0$. Les deux séquences (α_i) et (β_j) doivent être des séquences injectives (qui ne contiennent pas des éléments répétitifs)

Lemma 5.3.2. *(Déterminant de Cauchy).*

Soit la matrice de Cauchy de la forme $C = \left(\frac{1}{u_i+v_j}\right)$, alors :

$$det(M) = \frac{\prod_{i<j}(u_j - u_i)\prod_{i<j}(v_j - v_i)}{\prod_{i,j}(u_i - v_j)} \tag{5.5}$$

Theorem 5.3.3. *$\forall n \in N$, la matrice A_n est inversible. Son déterminant est l'inverse d'un entier :*

$$det(A_n) = \frac{(1!2!3!\cdots(n-1)!)^4}{1!2!3!\cdots(2n-1)!} = \left[\prod_{k=1}^{n-1}(2k+1)(\tfrac{2k}{2})^2\right]^{-1} \tag{5.6}$$

5.4 La solution proposée :

Dans cette section, on décrit le schéma à seuil (m, t, n) de partage multi-secret proposé permettant de partager m images secrètes entre un ensemble de n participants, de telle sorte que la reconstruction des images se fait seulement si t participants au moins réunissent leurs parts attribuées des secrets, le schéma proposé repose sur l'utilisation des systèmes d'équations linéaires surdéterminés définis sur un champ de Rijndael, ce qui permet d'apporter de meilleures performances par rapport aux méthodes classiques de partage de secret en terme de : représentation des parts (parts optimales) ce qui rend le schéma idéal avec une complexité de partage/reconstruction linéaire, une forte t-consistance des parts, et un partage de secrets dynamique.

Deux phases constituent le schéma de partage multi-secret proposé :

(1) La phase de partage, dont laquelle « le dealer » partage les m images secrètes $I_1, I_2, \cdots, I_m$ entre les n participants ;

(2) la phase de reconstruction qui consiste à reconstruire les images secrètes à partir de tout sous-ensemble d'au moins t participants ;

Dans chacune des deux phases, deux cas sont étudiés : le premier cas où le nombre de secret est inférieur ou égal au seuil $(m \leq t)$; et le deuxième cas où le nombre de secret est supérieur au seuil $(m > t)$. Les images secrètes à partager doivent être de même taille (dans le cas contraire un « padding » est nécessaire pour compléter les plus petites), chaque image est définie par $l \times w$ pixels, et chaque pixel de l'image est représenté par un octet appartenant à un champ de Rijndael ($[0, 255]$).

5.4.1 La phase de partage :

Dans cette phase, «le dealer» génère les n parts et les distribue aux n participants du groupe selon les étapes suivante correspondantes aux deux cas $(m \leq t)$ et $(m > t)$:

➥ Si $m \leq t$,

1. Le concessionnaire (D) crée la matrice de Cauchy C d'ordre $n \times t$ en utilisant deux séquences d'éléments $(\alpha_1, \alpha_2, \cdots, \alpha_n)$ et $(\beta_1, \beta_2, \cdots, \beta_t)$ sur $GF(2^8)$ comme suite :

$$C = \begin{pmatrix} \frac{1}{\alpha_1+\beta_1} & \frac{1}{\alpha_1+\beta_2} & \cdots & \frac{1}{\alpha_1+\beta_t} \\ \frac{1}{\alpha_2+\beta_1} & \frac{1}{\alpha_2+\beta_2} & \cdots & \frac{1}{\alpha_2+\beta_t} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\alpha_n+\beta_1} & \frac{1}{\alpha_n+\beta_2} & \cdots & \frac{1}{\alpha_n+\beta_t} \end{pmatrix} \tag{5.7}$$

2. Pour chaque ensemble de m pixels extraits des images secrètes $I_1, I_2, \cdots, I_m$:

2.1. D définit le vecteur $PI = (PI_1, PI_2, \cdots, PI_m)$ contenant les pixels des images secrètes :

- ✦ **Si $m < t$,**
  - ✧ **Sélectionner des octets aléatoires dans un corps de Rijndael $R_1, R_2, \cdots, R_{t-m}$ pour les $t - m$ pixels restants, $\forall i > m : PI_i = R_{i-m}$.**
  - ✧ **Mettre $PI = (PI_1, PI_2, \cdots, PI_m, R_1, R_2, \cdots, R_{t-m}) = (PI_1, PI_2, \cdots, PI_m, \cdots, PI_t)$.**
- ✦ **Si $m = t$ alors,**
  - ✧ **Le vecteur PI reste inchangeable.**

2.2. D calcule le vecteur $PH = (PH_1, PH_2, \cdots, PH_n)$ comme suite :

$$PH = \begin{pmatrix} PH_1 \\ PH_2 \\ \vdots \\ PH_n \end{pmatrix} = \begin{pmatrix} \frac{1}{\alpha_1+\beta_1} & \frac{1}{\alpha_1+\beta_2} & \cdots & \frac{1}{\alpha_1+\beta_t} \\ \frac{1}{\alpha_2+\beta_1} & \frac{1}{\alpha_2+\beta_2} & \cdots & \frac{1}{\alpha_2+\beta_t} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\alpha_n+\beta_1} & \frac{1}{\alpha_n+\beta_2} & \cdots & \frac{1}{\alpha_n+\beta_t} \end{pmatrix} \begin{pmatrix} PI_1 \\ PI_2 \\ \vdots \\ PI_t \end{pmatrix} \quad (5.8)$$

Chaque pixel $PH_i, 1 \leq i \leq n$, est stocké dans une part d'image SH_i.

3. D publie la matrice de Cauchy et distribue pour chaque participant $P_i, 1 \leq i \leq n$, à travers un canal sécurisé la part (i, SH_i), où i représente le numéro d'ordre du participant et SH_i la part finale (après avoir assemblé et concaténé tous les pixels PH_i correspondants) du participant i.

➥ **Si $m > t$,**

1. D crée la matrice de Cauchy C d'ordre $(n + m - t) \times m$ en utilisant les deux séquences d'éléments $(\alpha_1, \alpha_2, \cdots, \alpha_{n+m-t})$ et $(\beta_1, \beta_2, \cdots, \beta_m)$ sur $GF(2^8)$ comme suite :

$$C = \begin{pmatrix} \frac{1}{\alpha_1+\beta_1} & \frac{1}{\alpha_1+\beta_2} & \cdots & \frac{1}{\alpha_1+\beta_m} \\ \frac{1}{\alpha_2+\beta_1} & \frac{1}{\alpha_2+\beta_2} & \cdots & \frac{1}{\alpha_2+\beta_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\alpha_{n+m-t}+\beta_1} & \frac{1}{\alpha_{n+m-t}+\beta_2} & \cdots & \frac{1}{\alpha_{n+m-t}+\beta_m} \end{pmatrix} \quad (5.9)$$

2. Pour chaque ensemble de m pixels extraits des images secrètes $I_1, I_2, \cdots, I_m$:

2.1. D définit le vecteur $PI = (PI_1, PI_2, \cdots, PI_m)$ représentant les pixels des images secrètes.

2.2. D calcule le vecteur $PH = (PH_1, PH_2, \cdots, PH_{n+m-t})$ comme suite :

$$PH = \begin{pmatrix} PH_1 \\ PH_2 \\ \vdots \\ PH_{n+m-t} \end{pmatrix} = \begin{pmatrix} \frac{1}{\alpha_1+\beta_1} & \frac{1}{\alpha_1+\beta_2} & \cdots & \frac{1}{\alpha_1+\beta_m} \\ \frac{1}{\alpha_2+\beta_1} & \frac{1}{\alpha_2+\beta_2} & \cdots & \frac{1}{\alpha_2+\beta_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\alpha_{n+m-t}+\beta_1} & \frac{1}{\alpha_{n+m-t}+\beta_2} & \cdots & \frac{1}{\alpha_{n+m-t}+\beta_m} \end{pmatrix} \begin{pmatrix} PI_1 \\ PI_2 \\ \vdots \\ PI_m \end{pmatrix} \tag{5.10}$$

Chaque pixel $PH_i, 1 \leq i \leq n+m-t$, est stocké dans une part d'image SH_i.

3. D publie la matrice de Cauchy et les $m-t$ dernières parts $SH_i, n \leq i \leq n+m-t$, et distribue pour chaque participant $P_i, 1 \leq i \leq n$, à travers un canal sécurisé la part (i, SH_i).

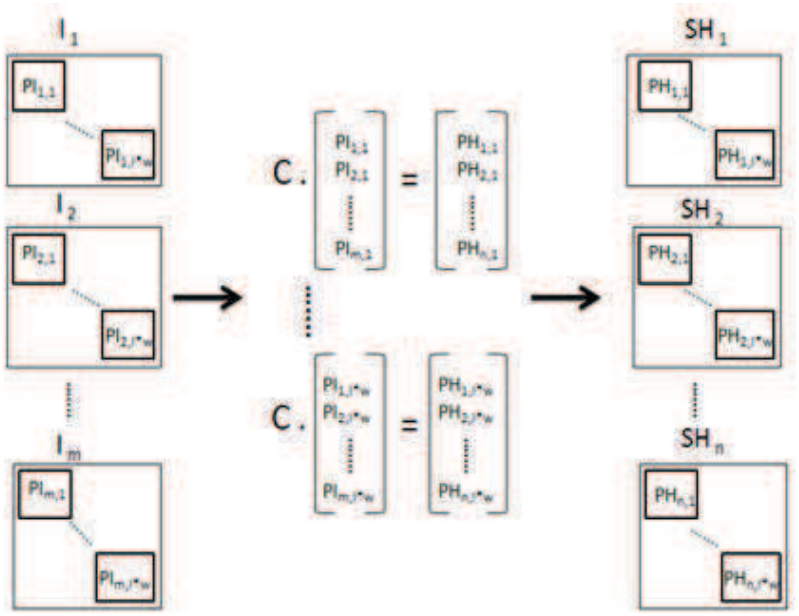


FIGURE 5.1 – La phase de partage proposée de l'approche 2.

5.4.2 La phase de reconstruction :

Dans cette phase, les m images secrètes sont reconstruites à partir de tout groupe de t participants selon les étapes suivante correspondantes

au deux cas $(m \leq t)$ et $(m > t)$:

➥ **Si $m \leq t$,**

1. **Chaque participant partage sa part (i, SH_i) avec le reste des $t-1$ participants à travers un canal sécurisé.**
2. **La partie de confiance construit une sous-matrice $\acute{C}$ d'ordre $t \times t$ à partir de la matrice C selon les indices i des t participants, puis calcule l'inverse de cette sous-matrice $\acute{C}^{-1}$ sur $GF(2^8)$.**

$$\acute{C} = \begin{pmatrix} \frac{1}{\alpha_{i_1}+\beta_1} & \frac{1}{\alpha_{i_1}+\beta_2} & \cdots & \frac{1}{\alpha_{i_1}+\beta_t} \\ \frac{1}{\alpha_{i_2}+\beta_1} & \frac{1}{\alpha_{i_2}+\beta_2} & \cdots & \frac{1}{\alpha_{i_2}+\beta_t} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\alpha_{i_t}+\beta_1} & \frac{1}{\alpha_{i_t}+\beta_2} & \cdots & \frac{1}{\alpha_{i_t}+\beta_t} \end{pmatrix} \tag{5.11}$$

3. **Pour chaque ensemble de t pixels extrait des parts d'images $SH_{i_1}, SH_{i_2}, \cdots, SH_{i_t}$ des t participants :**

   3.1. **D définit le vecteur $PH = (PH_{i_1}, PH_{i_2}, \cdots, PH_{i_t})$, afin de trouver les pixels des images secrètes reconstruites comme suite :**

$$SC = \begin{pmatrix} SC_1 \\ SC_2 \\ \vdots \\ SC_t \end{pmatrix} = \acute{C}^{-1}. \begin{pmatrix} PH_{i_1} \\ PH_{i_2} \\ \vdots \\ PH_{i_t} \end{pmatrix} \tag{5.12}$$

   ✦ **Si $m < t$ alors,**

   ✧ **On considère seulement les m valeurs initiales $SC_i, 1 \leq i \leq m$ qui vont être stockées dans les m images reconstruites, le reste des pixels sont stockés dans $t - m$ images aléatoires respectivement.**

   ✦ **Si $m = t$,**

   ✧ **Chaque pixel $SC_i, 1 \leq i \leq t$, est stocké dans une image correspondante.**

   Selon le cas, les pixels $SC_i, 1 \leq i \leq t$ sont assemblés et concaténés pour former les images reconstruites.

➥ **Si $m > t$,**

1. **Chaque participant partage sa part (i, SH_i) avec le reste des $t-1$ participants via un canal sécurisé.**
2. **La partie de confiance construit une sous-matrice $\acute{C}$ d'ordre $m \times m$ à partir de la matrice C selon les indices i des t participants ainsi que des $m - t$ indices des dernières parts, puis**

calcule l'inverse de cette sous-matrice $\acute{C}^{-1}$ sur $GF(2^8)$.

$$\acute{C} = \begin{pmatrix} \frac{1}{\alpha_{i_1}+\beta_1} & \frac{1}{\alpha_{i_1}+\beta_2} & \cdots & \frac{1}{\alpha_{i_1}+\beta_m} \\ \frac{1}{\alpha_{i_2}+\beta_1} & \frac{1}{\alpha_{i_2}+\beta_2} & \cdots & \frac{1}{\alpha_{i_2}+\beta_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\alpha_{i_t}+\beta_1} & \frac{1}{\alpha_{i_t}+\beta_2} & \cdots & \frac{1}{\alpha_{i_t}+\beta_m} \\ \frac{1}{\alpha_{n+1}+\beta_1} & \frac{1}{\alpha_{n+1}+\beta_2} & \cdots & \frac{1}{\alpha_{n+1}+\beta_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\alpha_{n+m-t}+\beta_1} & \frac{1}{\alpha_{n+m-t}+\beta_2} & \cdots & \frac{1}{\alpha_{n+m-t}+\beta_m} \end{pmatrix} \tag{5.13}$$

3. Pour chaque ensemble de t pixels extraits des parts d'image $SH_{i_1}, SH_{i_2}, \cdots, SH_{i_t}$ attribuées aux t participants, complété des $m-t$ pixels appartenant aux $m-t$ parts publiques :

   3.1. D définit le vecteur $PH = (PH_{i_1}, PH_{i_2}, \cdots, PH_{i_t}, PH_{i_{n+1}}, \cdots, PH_{i_{n+m-t}})$, afin de trouver les m pixels SC_i, $1 \leq i \leq m$, comme suite :

$$SC = \begin{pmatrix} SC_1 \\ SC_2 \\ \vdots \\ SC_m \end{pmatrix} = \acute{C}^{-1}. \begin{pmatrix} PH_{i_1} \\ PH_{i_2} \\ \vdots \\ PH_{i_t} \\ PH_{n+1} \\ \vdots \\ PH_{n+m-t} \end{pmatrix} \tag{5.14}$$

   Les pixels $SC_i, 1 \leq i \leq m$ calculés à chaque fois vont être assemblés et concaténés pour reconstruire les images secrètes respectives.

Les figures 5.1 et 5.2 montrent le fonctionnement général du schéma de partage multi-secret proposé ; noter que en plus des n parts d'image attribuées aux n participants, $m-t$ images publiques sont générées aléatoirement dans le cas $(m > t)$ lors de la création des parts dans la phase de partage ; alors que lorsque le nombre de secrets est inférieur au seuil $(m < t), t-m$ images aléatoires sont générées en plus des m images secrètes reconstruites durant la deuxième phase.

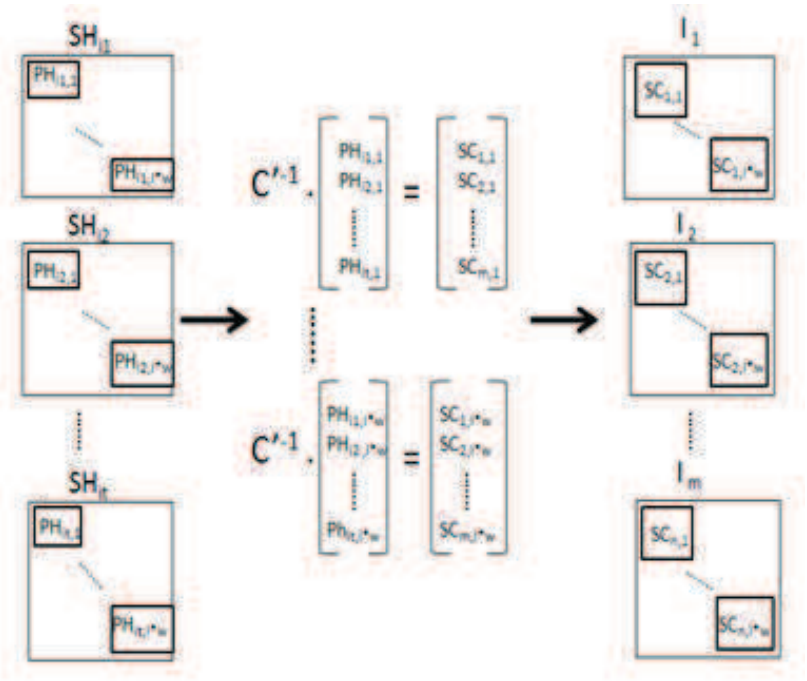


FIGURE 5.2 – La phase de reconstruction proposée de l'approche 2.

❋ **L'avantage de la solution proposée :**

- **Le schéma proposé permet d'obtenir de meilleures performances suite à la complexité de calcul linéaire qu'offre le schéma par rapport au nombre de participants (n), le seuil de partage (t) et de la taille des secrets.**
- **Le schéma est idéal : contrairement aux schémas de partage existants opérant sur un corps fini premier ce qui permet de réduire les pixels des images (définis dans $GF(2^8)$) modulo un nombre premier en conduisant soit à une perte d'information (si le nombre premier est 251 par exemple), ou à une augmentation des taille des parts construites (si par exemple le nombre premier est 257) ; le schéma proposé n'utilise aucun modulo premier, ce qui fait que la taille des parts d'images ainsi que des images reconstruites reste la même que celles des images secrètes avec aucun pixel de plus ou de moins.**

5.5 Les résultats d'expérimentation :

Dans cette section, on expérimente les résultats du schéma de partage multi-secret proposé, supposons par un exemple le schéma à seuil $(4, 3, 6)$ où le concessionnaire désire partager 4 images secrètes en couleur I_1, I_2, I_3

et I_4 de même taille 512×512 pixels (montrées en figure 5.3) entre 6 participants, de telle sorte que la reconstruction des images se fait que si 3 participants parmi les 6 réunissent leurs parts attribuées des secrets.

(a) I_1 (b) I_2

(c) I_3 (d) I_4

FIGURE 5.3 – Les 4 images secrètes à partager.

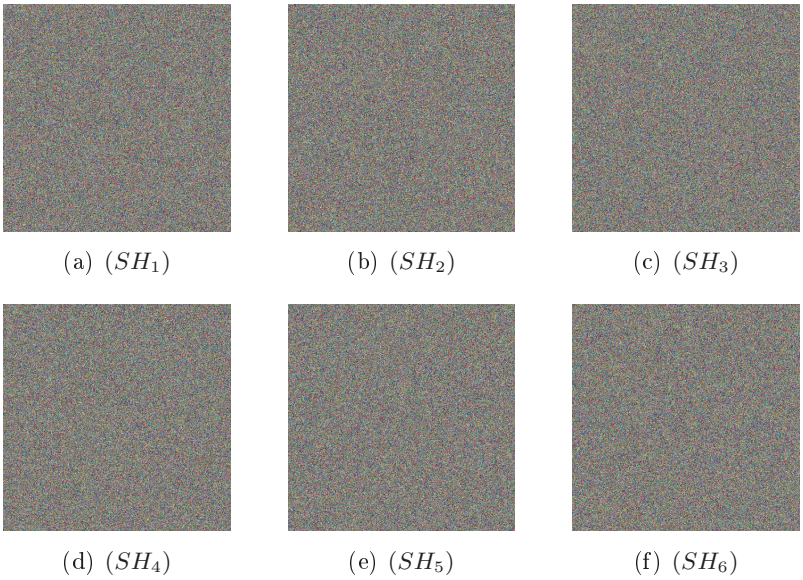
Comme on est dans le cas $m > t$, le concessionnaire D définit la matrice de Cauchy d'ordre (7×4) à l'aide des deux séquences d'éléments générées aléatoirement $\alpha = (234, 189, 243, 102, 166, 96, 50)$ et $\beta = (52, 47, 203, 229)$ sur $GF(2^8)$ comme suite :

$$C = \begin{pmatrix} 25 & 184 & 237 & 150 \\ 49 & 68 & 218 & 19 \\ 158 & 153 & 80 & 76 \\ 104 & 136 & 8 & 29 \\ 68 & 49 & 236 & 213 \\ 129 & 147 & 60 & 204 \\ 122 & 131 & 212 & 214 \end{pmatrix} \tag{5.15}$$

Puis pour tout ensemble de 4 pixels (PI_1, PI_2, PI_3, PI_4) appartenant chacun à une image secrète I_1, I_2, I_3 et I_4 respectivement, D calcule le vecteur PH qui contient sept pixels $(PH_1, PH_2, PH_3, PH_4, PH_5, PH_6, PH_7)$:

$$PH = \begin{pmatrix} PH_1 \\ PH_2 \\ PH_3 \\ PH_4 \\ PH_5 \\ PH_6 \\ PH_7 \end{pmatrix} = \begin{pmatrix} 25 & 184 & 237 & 150 \\ 49 & 68 & 218 & 19 \\ 158 & 153 & 80 & 76 \\ 104 & 136 & 8 & 29 \\ 68 & 49 & 236 & 213 \\ 129 & 147 & 60 & 204 \\ 122 & 131 & 212 & 214 \end{pmatrix} \begin{pmatrix} PI_1 \\ PI_2 \\ PI_3 \\ PI_4 \end{pmatrix} \tag{5.16}$$

Le pixel PH_i donné à chaque opération est stocké dans la part d'image SH_i appropriée ; à la fin, la concaténation des pixels dans chaque part forme la part finale qui est affectée à chaque participant $P_i, 1 \leq i \leq 6$. La figure 5.4 montre les 6 parts d'images obtenues. D publie la matrice de Cauchy ainsi que la dernière part générée aléatoirement SH_7 (voir la figure 5.5), et distribue via un canal sécurisé respectivement les parts $(i, SH_i), 1 \leq i \leq 6$ aux 6 participants.



(a) (SH_1) (b) (SH_2) (c) (SH_3)

(d) (SH_4) (e) (SH_5) (f) (SH_6)

FIGURE 5.4 – Les six parts (512×512) attribuées aux participants P_1, P_2, P_3, P_4, P_5 et P_6 respectivement.

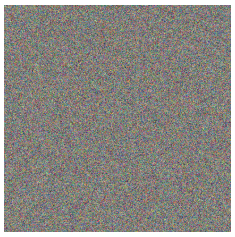


FIGURE 5.5 – La part SH_7 (512×512) publiée.

Pour reconstruire les 4 images secrètes la collaboration de 3 participants est nécessaire, supposons par exemple les participants P_2, P_4 et P_5, en utilisant le numéro d'ordre i de ces 3 participants (lignes 2, 4 et 5 de la matrice C) et la dernière ligne $(m - t)$ de C, la sous-matrice $\acute{C}$ est

construite de cette manière :

$$\acute{C} = \begin{matrix} P_2 \\ P_4 \\ P_5 \\ (n+m-t)^{\text{ème}} ligne \end{matrix} \begin{pmatrix} 49 & 68 & 218 & 19 \\ 104 & 136 & 8 & 29 \\ 68 & 49 & 236 & 213 \\ 122 & 131 & 212 & 214 \end{pmatrix} \tag{5.17}$$

Puis pour chaque ensemble de 4 pixels (PH_1, PH_2, PH_3, PH_4) appartenant chacun à une part d'image PH_1, PH_2, PH_3 et PH_4 respectivement, D calcule le vecteur SC qui contient quatre pixels (SC_1, SC_2, SC_3, SC_4) en utilisant l'inverse de la sous-matrice $\acute{C}$ comme suite :

$$SC = \begin{pmatrix} SC_1 \\ SC_2 \\ SC_3 \\ SC_4 \end{pmatrix} = \begin{pmatrix} 33 & 237 & 185 & 16 \\ 73 & 83 & 247 & 124 \\ 201 & 28 & 205 & 8 \\ 37 & 187 & 150 & 150 \end{pmatrix} \begin{pmatrix} PH_2 \\ PH_4 \\ PH_5 \\ PH_7 \end{pmatrix} \tag{5.18}$$

Le pixel SC_i donné à chaque opération est stocké dans l'image appropriée, à la fin la concaténation des pixels dans chaque image forme les 4 images secrètes reconstruites.

Nous prenons un autre exemple de schéma à seuil $(3, 3, 5)$, où on désire partager 3 images secrètes (voir figure 5.6) entre cinq participants, et ne permettant qu'à seulement trois participants parmi les cinq de les reconstruire, cette fois ci le nombre de secret m est égal au seuil t $(m = t)$, « le dealer » définit la matrice de Cauchy d'ordre (5×3) à l'aide des deux séquences d'éléments générées aléatoirement $\alpha = (135, 37, 133, 79, 186)$ et $\beta = (62, 181, 14)$ sur $GF(2^8)$ comme suite :

$$\acute{C} = \begin{pmatrix} 100 & 111 & 49 \\ 128 & 24 & 74 \\ 123 & 72 & 45 \\ 209 & 232 & 95 \\ 124 & 150 & 235 \end{pmatrix} \tag{5.19}$$

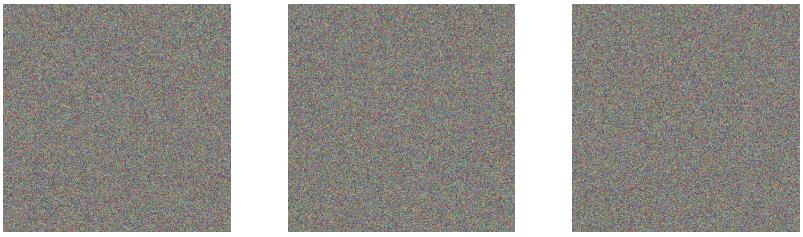
Pour tout ensemble de 3 pixels (PI_1, PI_2, PI_3) appartenant chacun à une image secrète, « le dealer » calcule le vecteur PH qui contient cinq pixels $(PH_1, PH_2, PH_3, PH_4, PH_5)$:

$$PH = \begin{pmatrix} PH_1 \\ PH_2 \\ PH_3 \\ PH_4 \\ PH_5 \end{pmatrix} = \begin{pmatrix} 100 & 111 & 49 \\ 128 & 24 & 74 \\ 123 & 72 & 45 \\ 209 & 232 & 95 \\ 124 & 150 & 235 \end{pmatrix} \begin{pmatrix} PI_1 \\ PI_2 \\ PI_3 \end{pmatrix} \tag{5.20}$$

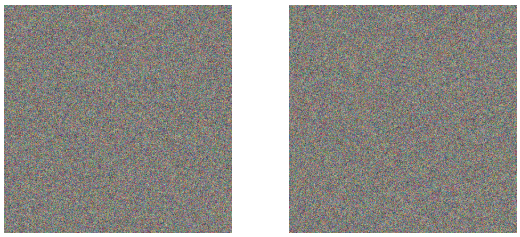
(a) I_1 (b) I_2 (c) I_3

FIGURE 5.6 – Les 3 images secrètes à partager.

Les pixels PH_i donnés à chaque opération sont stockés et concaténés avec les précédents dans les parts d'image SH_i appropriées affectée à chaque participant $P_i, 1 \leq i \leq 5$. La figure 5.7 montre les 5 parts d'images obtenues.



(a) (SH_1) (b) (SH_2) (c) (SH_3)



(d) (SH_4) (e) (SH_5)

FIGURE 5.7 – Les parts d'images distribuées aux cinq participants.

Pour reconstruire les 3 images secrètes la collaboration de 3 participants est nécessaire, supposons par exemple les participants P_1, P_3 et P_4, en utilisant le numéro d'ordre i de ces 3 participants (lignes 1, 3 et 4 de la matrice C), la sous-matrice $\acute{C}$ est construite :

$$\acute{C} = \begin{matrix} P_1 \\ P_3 \\ P_4 \end{matrix} \begin{pmatrix} 100 & 111 & 49 \\ 123 & 72 & 45 \\ 209 & 232 & 95 \end{pmatrix} \tag{5.21}$$

Puis pour chaque ensemble de 3 pixels (PH_1, PH_2, PH_3) appartenant chacun à une part d'image PH_1, PH_2 et PH_3 respectivement, D calcule le

vecteur SC qui contient trois pixels (SC_1, SC_2, SC_3) en utilisant l'inverse de la sous-matrice $\acute{C}$ comme suite :

$$SC = \begin{pmatrix} SC_1 \\ SC_2 \\ SC_3 \end{pmatrix} = \begin{pmatrix} 181 & 99 & 158 \\ 107 & 33 & 99 \\ 130 & 246 & 221 \end{pmatrix} \begin{pmatrix} PH_1 \\ PH_3 \\ PH_4 \end{pmatrix} \tag{5.22}$$

Le pixel SC_i donné à chaque opération est stocké dans l'image appropriée, à la fin la concaténation des pixels dans chaque image forme les 3 images secrètes reconstruites.

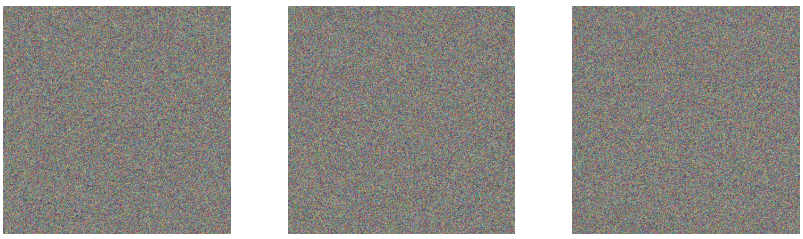
On prend un troisième exemple de schéma $(3, 5, 8)$ cette fois avec un seuil supérieur au nombre de secret, dans lequel on désir partager les trois images secrètes de la figure 5.6 entre 8 participants en permettant la reconstruction de ces dernières que si une collaboration de cinq participants parmi huit est faite. Le concessionnaire commence par définir la matrice de Cauchy d'ordre (8×5) à l'aide des deux séquences d'éléments générées aléatoirement $\alpha = (89, 88, 111, 227, 48, 125, 55, 70)$ et $\beta = (192, 14, 151, 100, 196)$ sur $GF(2^8)$ comme suite :

$$C = \begin{pmatrix} 220 & 97 & 168 & 12 & 9 \\ 11 & 37 & 58 & 171 & 172 \\ 183 & 87 & 66 & 152 & 60 \\ 81 & 33 & 233 & 176 & 138 \\ 227 & 21 & 5 & 129 & 3 \\ 154 & 217 & 243 & 222 & 100 \\ 201 & 34 & 28 & 140 & 234 \\ 228 & 48 & 113 & 57 & 161 \end{pmatrix} \tag{5.23}$$

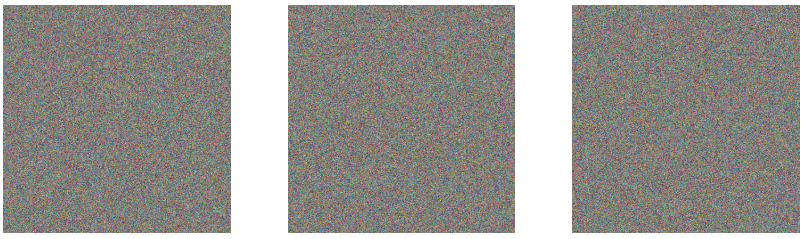
Calcule le vecteur PH à partir de la matrice C et du vecteur PI complété de pixels des deux images générées aléatoirement :

$$PH = \begin{pmatrix} PH_1 \\ PH_2 \\ PH_3 \\ PH_4 \\ PH_5 \\ PH_6 \\ PH_7 \\ PH_8 \end{pmatrix} = \begin{pmatrix} 220 & 97 & 168 & 12 & 9 \\ 11 & 37 & 58 & 171 & 172 \\ 183 & 87 & 66 & 152 & 60 \\ 81 & 33 & 233 & 176 & 138 \\ 227 & 21 & 5 & 129 & 3 \\ 154 & 217 & 243 & 222 & 100 \\ 201 & 34 & 28 & 140 & 234 \\ 228 & 48 & 113 & 57 & 161 \end{pmatrix} \begin{pmatrix} PI_1 \\ PI_2 \\ PI_3 \\ R_1 \\ R_2 \end{pmatrix} \tag{5.24}$$

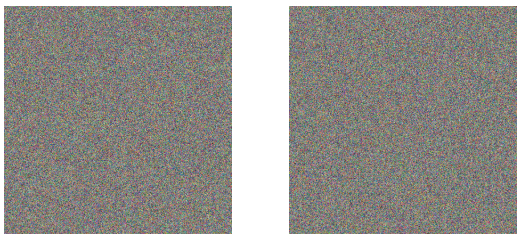
Chaque pixel PH_i est stocké et concaténé avec les autres pour former les parts d'image (comme le montre la figure 5.8) qui vont être distribuées aux huit participants.



(a) (SH_1) (b) (SH_2) (c) (SH_3)

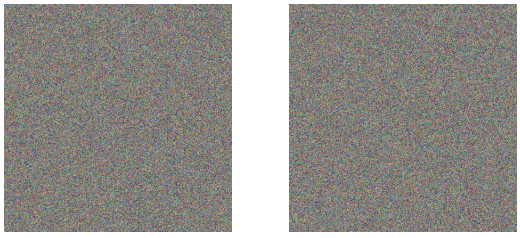


(d) (SH_4) (e) (SH_5) (f) (SH_6)



(g) (SH_7) (h) (SH_8)

FIGURE 5.8 – Les parts attribuées à chaque participant.



(a) (IG_1) (b) (IG_2)

FIGURE 5.9 – Les deux images générées aléatoirement.

Supposons par exemple les participants P_1, P_2, P_3, P_4 et P_5 qui veulent

reconstruire les images secrètes :

$$\acute{C} = \begin{matrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \end{matrix} \begin{pmatrix} 220 & 97 & 168 & 12 & 9 \\ 11 & 37 & 58 & 171 & 172 \\ 183 & 87 & 66 & 152 & 60 \\ 81 & 33 & 233 & 176 & 138 \\ 227 & 21 & 5 & 129 & 3 \end{pmatrix} \tag{5.25}$$

« Le dealer » calcule :

$$SC = \begin{pmatrix} SC_1 \\ SC_2 \\ SC_3 \\ R_1 \\ R_2 \end{pmatrix} = \begin{pmatrix} 40 & 210 & 99 & 19 & 11 \\ 243 & 183 & 44 & 233 & 54 \\ 77 & 172 & 143 & 110 & 131 \\ 8 & 189 & 184 & 183 & 48 \\ 164 & 34 & 42 & 254 & 41 \end{pmatrix} \begin{pmatrix} PH_1 \\ PH_2 \\ PH_3 \\ PH_4 \\ PH_5 \end{pmatrix} \tag{5.26}$$

À la fin de la concaténation de tous les pixels, les trois images secrètes sont reconstruites, les images générées aléatoirement peuvent également être reconstruites.

5.6 L′analyse de sécurité :

Dans ce qui suit, une analyse concernant la sécurité du schéma proposé est donnée ; on montre dans ce qui suit que le schéma est parfait (1), idéal (2), ainsi nous appliquerons quelques tests sur les résultats obtenus du schéma proposé.

5.6.1 Schéma parfait et idéal :

- **(1) Le schéma de partage multi-secret proposé est parfait puisque la coopération de t participants permet de révéler les m images secrètes alors que toute combinaison de moins de t participants ne le permet pas :**
  - **Si $m \leq t$, et qu′un sous-ensemble de $t-1$ participants souhaite reconstruire les secrets, l′ensemble contenant les pixels $\{PH_{i_1}, PH_{i_2}, \cdots, PH_{i_{t-1}}\}$ appartenant aux parts d′image $\{SH_{i_1}, SH_{i_2}, \cdots, SH_{i_{t-1}}\}$ respectivement, entrainera à la résolution du système indéterminé suivant de $t-1$ équations et t variables :**

$$\begin{pmatrix} \frac{1}{\alpha_{i_1}+\beta_1} & \frac{1}{\alpha_{i_1}+\beta_2} & \cdots & \frac{1}{\alpha_{i_1}+\beta_t} \\ \frac{1}{\alpha_{i_2}+\beta_1} & \frac{1}{\alpha_{i_2}+\beta_2} & \cdots & \frac{1}{\alpha_{i_2}+\beta_t} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\alpha_{i_{t-1}}+\beta_1} & \frac{1}{\alpha_{i_{t-1}}+\beta_2} & \cdots & \frac{1}{\alpha_{i_{t-1}}+\beta_t} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_t \end{pmatrix} = \begin{pmatrix} PH_{i_1} \\ PH_{i_2} \\ \vdots \\ PH_{i_{t-1}} \end{pmatrix} \tag{5.27}$$

Par conséquent, il admet un ensemble infini de solutions, et donc la solution SC (vecteur contenant les m pixels de chaque image secrète) ne peut être calculée.

- Si $m > t$, et qu′un sous-ensemble de $t-1$ participants souhaite reconstruire les secrets, l′ensemble de $t-1+m-t = m-1$ pixels $PH = (PH_{i_1}, PH_{i_2}, \cdots, PH_{i_{t-1}}, PH_{n+1}, PH_{n+2}, \cdots, PH_{n+m-t})$, contenant l′ensemble des $t-1$ pixels appartenant aux parts d′image $\{SH_{i_1}, SH_{i_2}, \cdots, SH_{i_{t-1}}\}$ combiné aux pixels des images publiques qui ont été générées $\{SH_{n+1}, SH_{n+2}, \cdots, SH_{n+m-t}\}$, conduit à la résolution du système indéterminé suivant de $m-1$ équations et m variables :

$$\begin{pmatrix} \frac{1}{\alpha_{i_1}+\beta_1} & \frac{1}{\alpha_{i_1}+\beta_2} & \cdots & \frac{1}{\alpha_{i_1}+\beta_m} \\ \frac{1}{\alpha_{i_2}+\beta_1} & \frac{1}{\alpha_{i_2}+\beta_2} & \cdots & \frac{1}{\alpha_{i_2}+\beta_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\alpha_{i_{t-1}}+\beta_1} & \frac{1}{\alpha_{i_{t-1}}+\beta_2} & \cdots & \frac{1}{\alpha_{i_{t-1}}+\beta_m} \\ \frac{1}{\alpha_{n+1}+\beta_1} & \frac{1}{\alpha_{n+1}+\beta_2} & \cdots & \frac{1}{\alpha_{n+1}+\beta_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\alpha_{n+m-t}+\beta_1} & \frac{1}{\alpha_{n+m-t}+\beta_2} & \cdots & \frac{1}{\alpha_{n+m-t}+\beta_m} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} PH_{i_1} \\ PH_{i_2} \\ \vdots \\ PH_{i_{t-1}} \\ PH_{n+1} \\ \vdots \\ PH_{n+m-t} \end{pmatrix} \tag{5.28}$$

Ce qui entraine à un ensemble infini de solutions, par conséquent la solution SC (vecteur contenant les m pixels de chaque image secrète) ne peut être calculée.

➥ (2) Le schéma de partage multi-secret proposé utilise les systèmes d′équations linéaires définis sur $GF(2^8)$ pour partager et reconstruire les images secrètes de $l \times w$ pixels, qui consistent (respectivement selon le cas $m \leq t$ / $m > t$) à donner n / $n+m-t$ solutions dans $GF(2^8)$ pour n / $n+m-t$ équations de t / m variables représentant au moins m pixels des m images secrètes avec chaque pixel représenté par un octet dans $GF(2^8)$, et puisque aucun modulo premier n′est utilisé comme dans le cas des solution classiques, les solutions calculées représentantes les pixels des parts d′image seront stockées exactement de la même manière dont elle sont définies c-à-d. par un octet rendant ainsi la taille des parts d′images invariante par rapport aux images secrètes avec aucun bit de plus ou de moins, ce qui rend le schéma idéal.

5.6.2 Tests statistiques :

Afin de vérifier qu′aucune information n′est révélée à partir des parts générées du schéma de partage multi-secret proposé attribuées aux participants, nous avons analysé l′aspect aléatoire des parts d′image en utilisant deux batteries de tests : Diehard et ENT (voir le chapitre précédent

pour plus d'information sur les deux batteries de tests). Les résultats obtenus des tests sont stockés dans les tableaux 5.1 et 5.2 respectivement comme suite :

Test Name	Averaged P-value	Interpretation
BIRTHDAY SPACINGS	0.716048	Pass
OVERLAPPING PERMUTATION	0.087598	Pass
RANK TEST 31x31	0.836817	Pass
RANK TEST 32x32	0.743973	Pass
BITSTREAM TEST	0.5013985	Pass
OPSO/OQSO/DNA	0.4053087 0.5194893 0.4743677	Pass
COUNT-THE-1's TEST	0.665823	Pass
PARKING LOT TEST	0.259022	Pass
3DSPHERES TEST	0.971851	Pass
SQEEZE test	0.033041	Pass
OVERLAPPING SUMS	0.753636	Pass
RUNS up/down test	0.3885035 0.4830975	Pass
3DSPHERES test	0.971851	Pass

Tableau 5.1 – Les résultats des tests Diehard de l'approche proposée 2.

Test	Value	Norm
Entropy	7.999988	$Max = 8.0$
Arithmetic mean	127.5023	$127.5 = random$
Monte Carlo	3.141384335 (error=0,01%)	π value
Serial correlation coefficient	0.000113	0.0

Tableau 5.2 – Les résultats des tests ENT de l'approche proposée 2.

Il est clair qu'à partir des résultats obtenus accordés aux tableaux 5.1 et 5.2 que tous les tests appliqués ont été passés avec succès ce qui montre que les parts d'images générées possèdent de très bonnes propriétés statistiques ce qui implique que le schéma de partage assure le sécurité des images.

5.6.2.1 L'histogramme :

Ci-dessous les tracés et l'analyse des histogrammes de chaque image secrète ainsi des résultats obtenus par le dernier exemple vu dans la section 5.5, qui concerne le schéma $(3, 5, 8)$, la figure suivante représente les histogrammes correspondants à chaque image secrète respectivement :

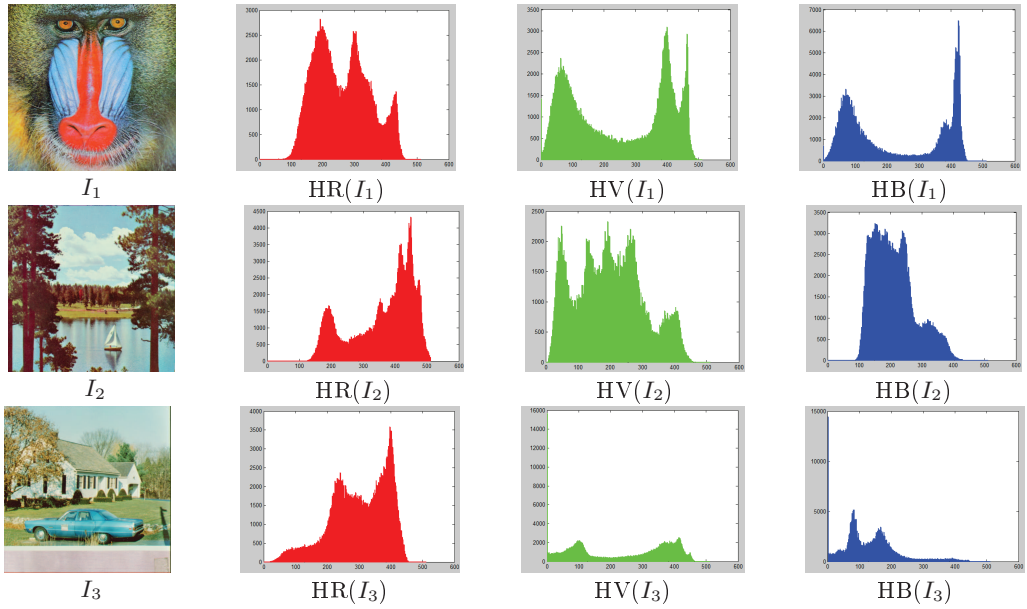
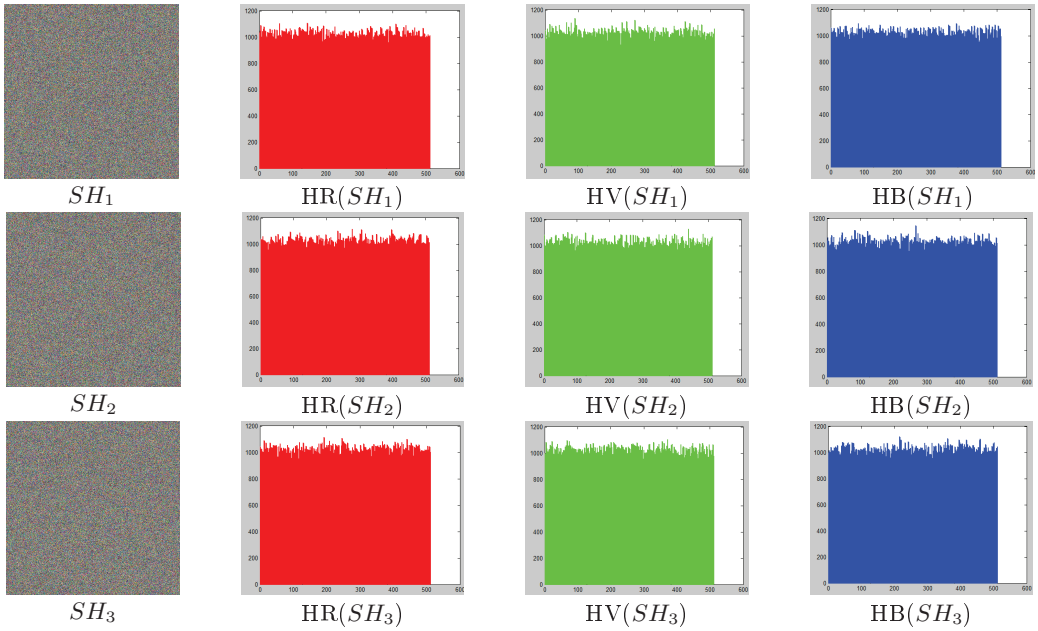


FIGURE 5.10 – Les trois images secrètes I_1,I_2,I_3 et leurs histogrammes respectifs HR, HV, HB.

Les histogrammes concernant les huit images chiffrées sont donnés dans la figure qui suit :



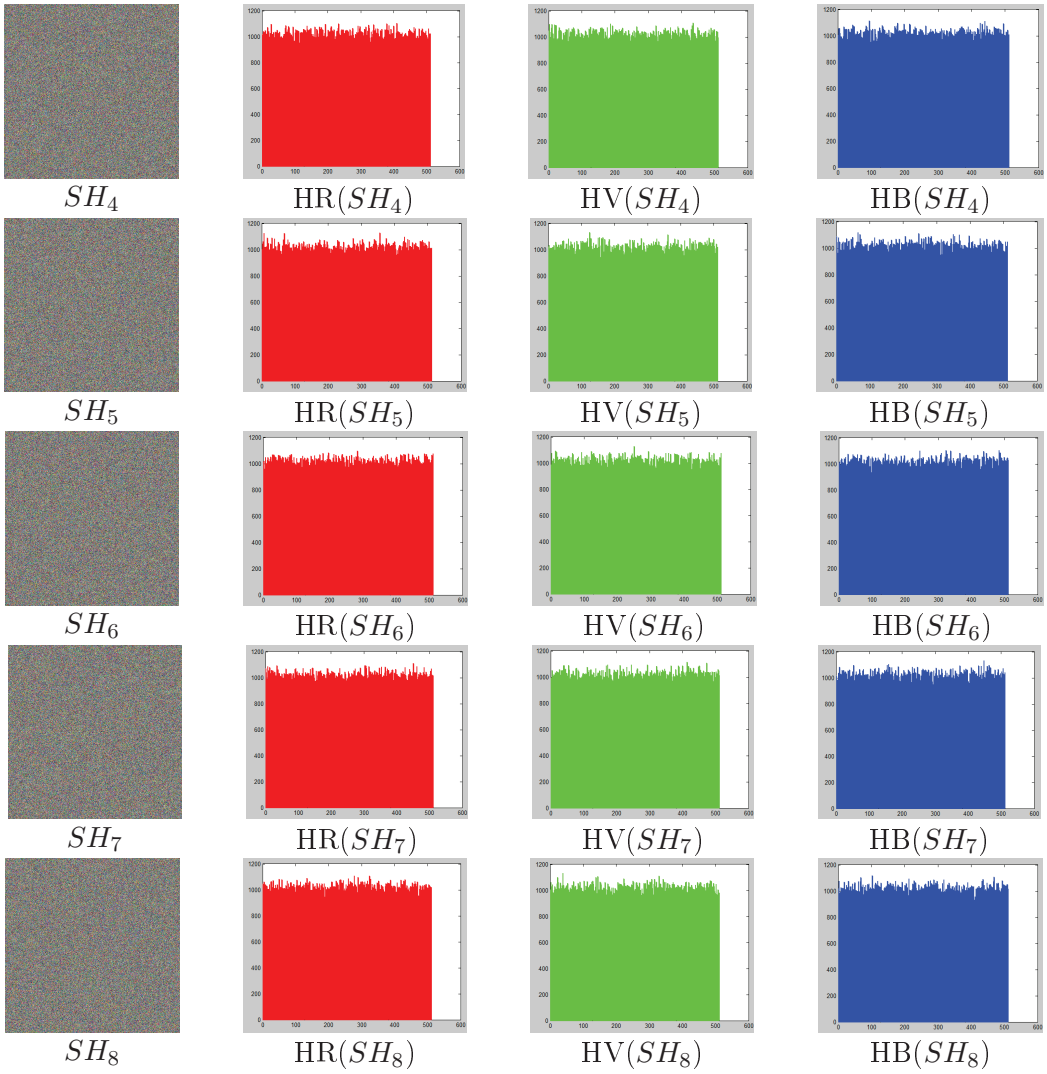
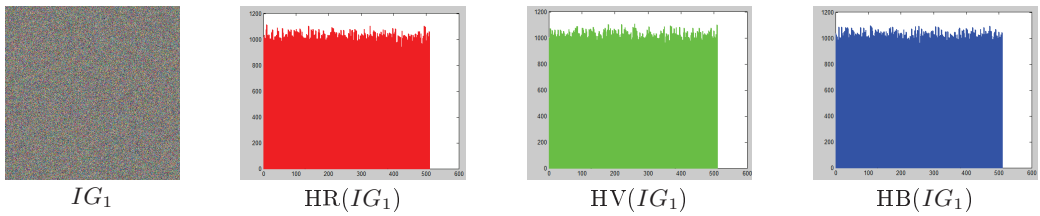


FIGURE 5.11 – Les parts d'images attribuées aux huit participants et leurs histogrammes respectifs.

Ainsi, ci-dessous la figure 5.12 qui représente les histogrammes des deux images générées aléatoirement lors de la phase de reconstruction des secrets :



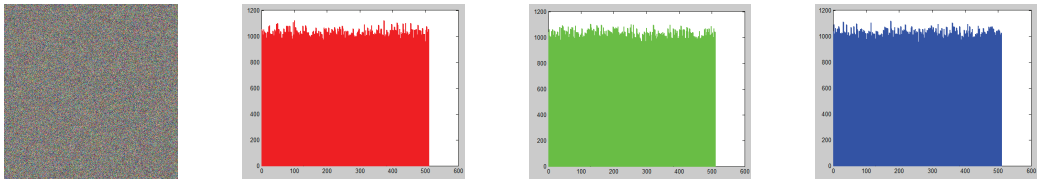


FIGURE 5.12 – Les deux parts d′images générées aléatoirement ainsi que leurs histogrammes respectifs.

Les histogrammes des parts d′images attribuées aux n participants, ainsi que des deux parts d′images générées aléatoirement, sont uniformément distribués par rapport aux histogrammes des images secrètes ; ce qui fait que le schéma de partage multi-secret proposé rend la dépendance des propriétés statistiques des parts d′images quasi aléatoires, ce qui permet de rendre la cryptanalyse de plus en plus difficile.

5.6.2.2 Corrélation des pixels adjacents :

Une analyse de corrélation des pixels adjacents horizontaux, verticaux et diagonaux voisins est donnée dans le cadre des images secrètes et des parts d′images attribuées aux n participants ($n = 5$), pour un exemple de schéma $(3, 3, 5)$.

a) **Corrélation horizontale**
La figure 5.13, montre les distributions de deux pixels adjacents horizontaux des 3 images secrètes comme suite :

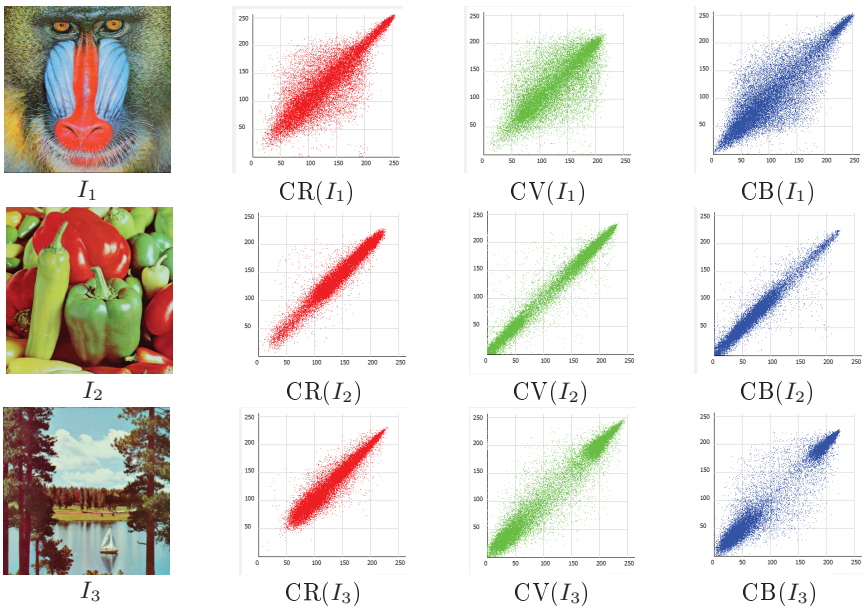


FIGURE 5.13 – Analyse de corrélation de deux pixels adjacents horizontaux des 3 images secrètes.

La figure 5.14 montre les distributions de deux pixels adjacents horizontaux des parts d'images obtenues.

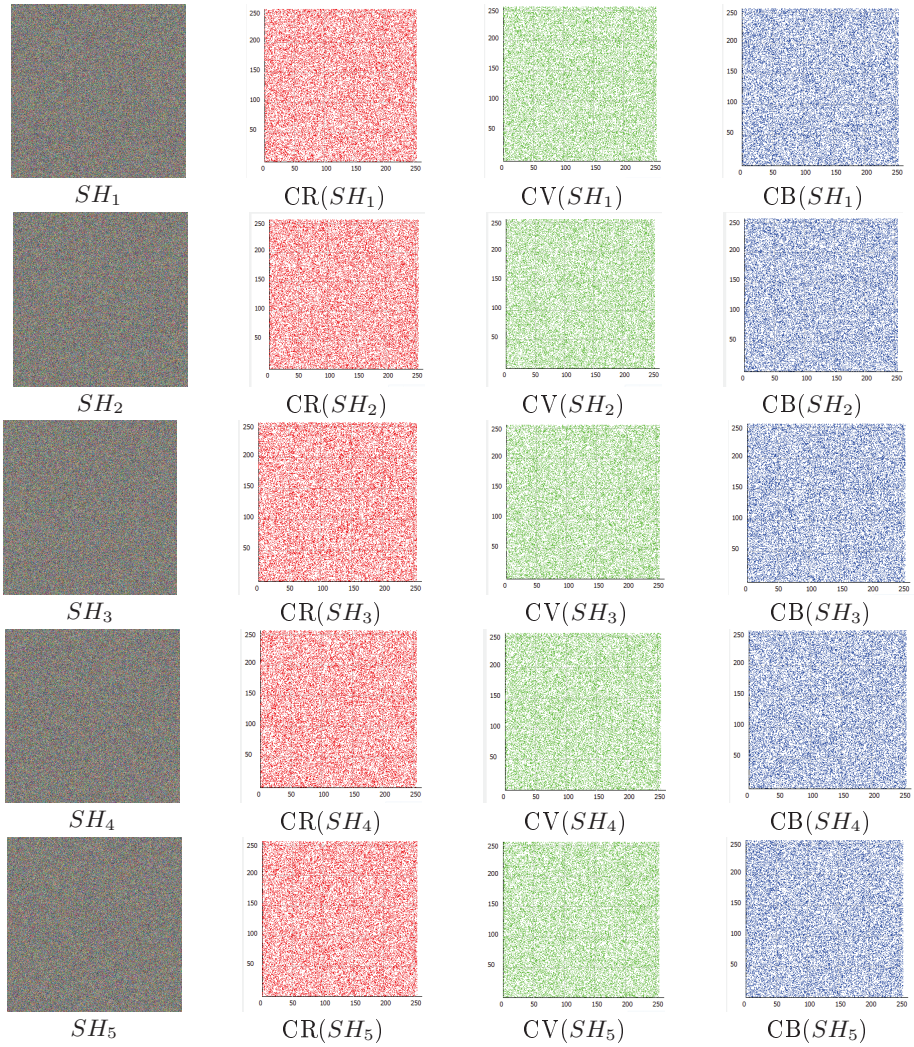
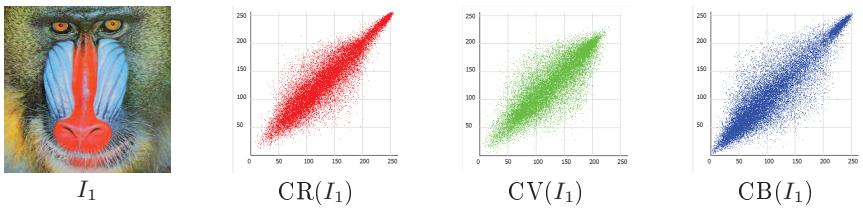


FIGURE 5.14 – Analyse de corrélation de deux pixels adjacents horizontaux des 5 parts d'images produites.

b) Corrélation verticale

La figure 5.15 montre les distributions de deux pixels adjacents verticaux des 3 images secrètes comme suite :



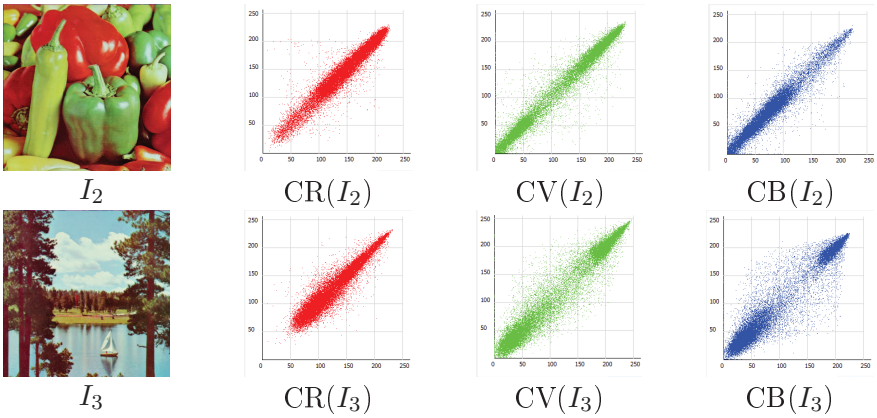
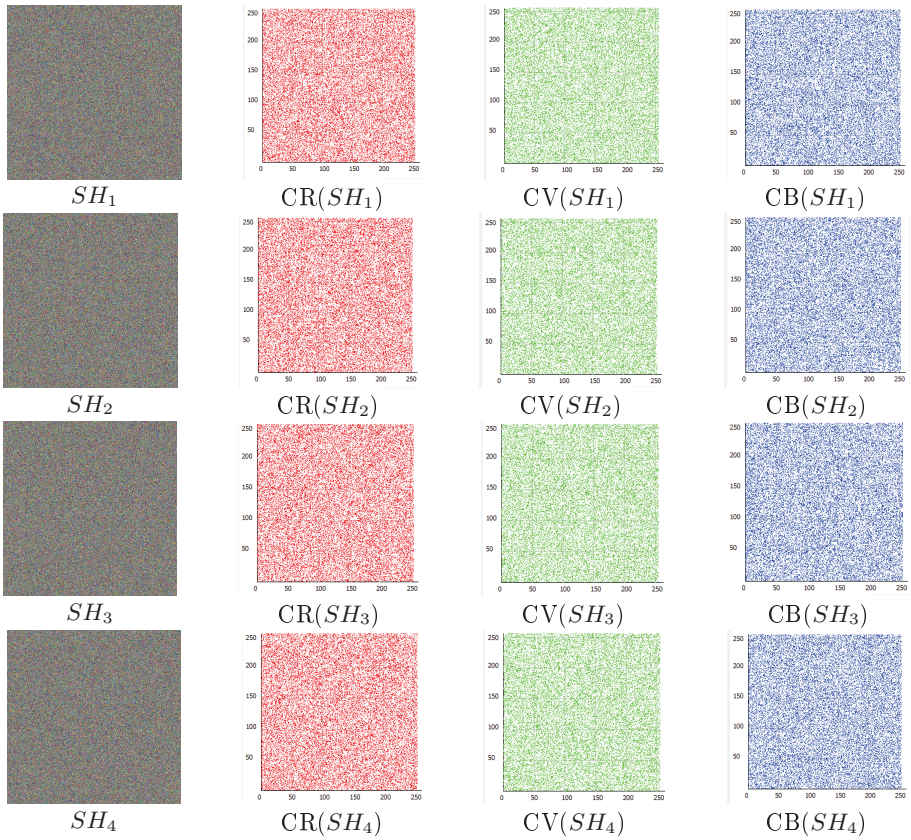


FIGURE 5.15 – Analyse de corrélation de deux pixels adjacents verticaux des images secrètes.

La figure 5.16 montre les distributions de deux pixels adjacents verticaux des parts d′images obtenues.



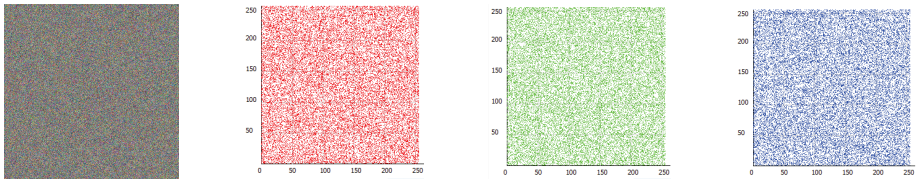


FIGURE 5.16 – Analyse de corrélation de deux pixels adjacents verticaux des 5 parts d'images produites.

c) Corrélation diagonale

La figure 5.17 montre les distributions de deux pixels adjacents diagonaux des 3 images secrètes comme suite :

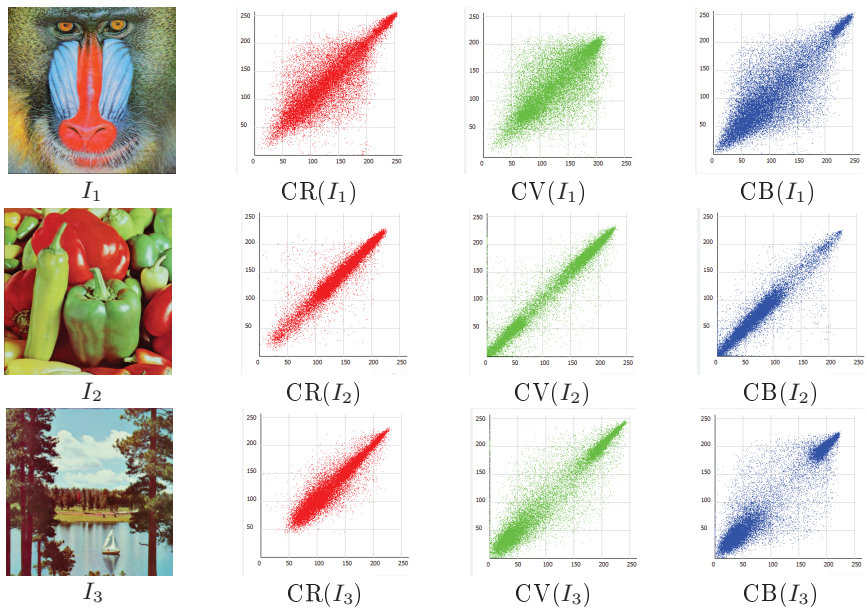
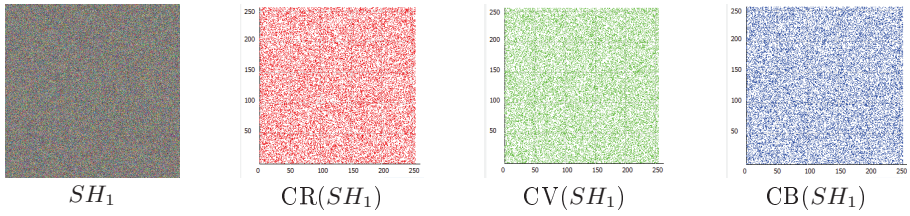


FIGURE 5.17 – Analyse de corrélation de deux pixels adjacents diagonaux des images secrètes.

La figure 5.18 montre les distributions de deux pixels adjacents diagonaux des parts d'images obtenues.



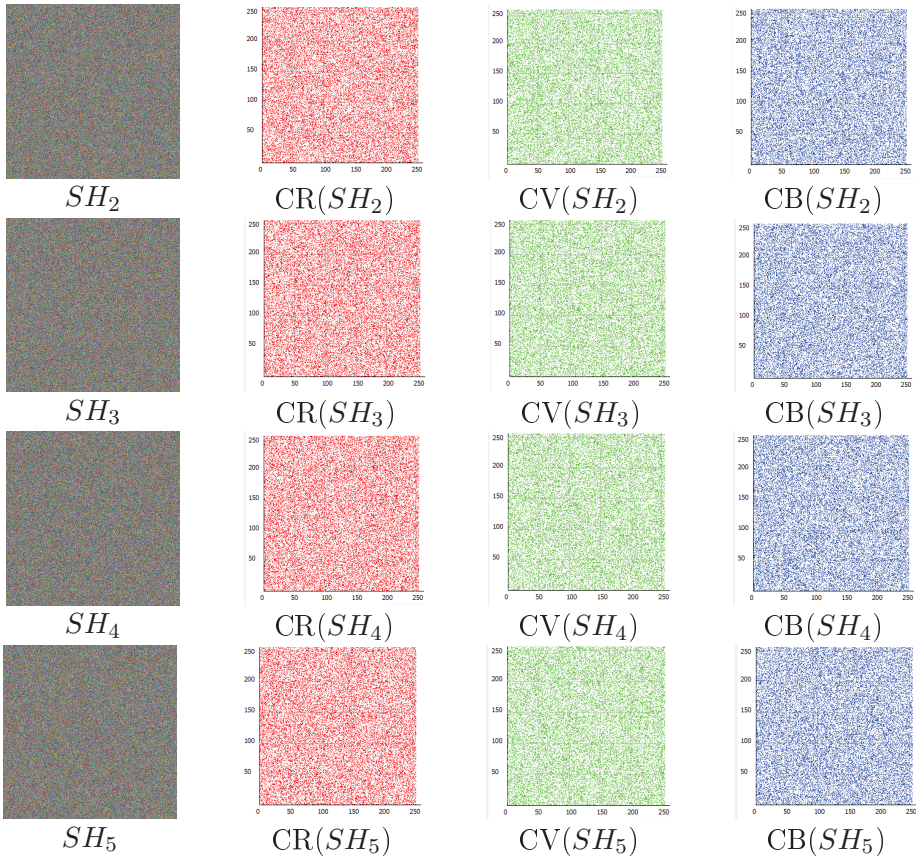


FIGURE 5.18 – Analyse de corrélation de deux pixels adjacents diagonaux des 5 parts d′images produites.

✤ **Il ressort des figures 5.13, 5.14, 5.15, 5.16, 5.17 et 5.18 que dans le cas des images secrètes, les pixels adjacents ont des corrélations fortes et s′alignent sur la première bissectrice. Alors que dans le cas des parts d′images attribuées aux 5 participants, les pixels adjacents sont disséminés presque de manière aléatoires ce qui renvoi à un algorithme robuste résistant à toute attaque statistique.**

Un tableau comparatif entre l′approche proposée et quelques solutions de partage de secret existantes est donné ci-dessous (dans le tableau 5.3) afin d′illustrer les propriétés et capacités du schéma proposé par rapport à plusieurs paramètres de performance. Tout en respectant les schémas basés sur Shamir et CRT, le schéma proposé est le seul qui assure l′idéalité. La complexité de calcule linéaire est un autre avantage du schéma qui n′est pas assurée par ces schémas qui ont au moins une complexité polynomiale.

Sharing Scheme	Techniques based	Multi-secret	Scalability for image	Sharing cost	Reconstruction cost	Strong t-consistency	Ease of management
Harn et Lin[Harn 2010]	Polynomial	No	No	$O(n^2)$	$O(m^2)$	Yes	No
Liu et al.[Liu 2012]	Polynomial	Yes	No	$O(n^2)$	$O(m^2)$	Yes	No
Cheraghi [Cheraghi 2014]	Polynomial/CFB	Yes	No	$O(n^2)$	$O(m^2)$	No	Yes
Chen et Fu[Chen 2008]	Geometry	No	Yes	$O(n)$	$O(m)$	No	Yes
Subba et Chakravarthy [Bhagvati 2014]	CRT	Yes	No	$O(n.log^2(n))$	$O(m.log^2(m))$	No	Yes
Proposed scheme	Overdetermined Sys. of eq.	Yes	Yes	$O(n)$	$O(m)$	Yes	Yes

Tableau 5.3 – Comparaison entre les schémas de partage existants avec le schéma proposé 2.

5.7 Conclusion :

Dans notre travail de thèse, une deuxième solution a été proposée pour partager et reconstruire selon un schéma à seuil (m, t, n) plusieurs images secrètes simultanément en seulement une session avec aucune contrainte de limitation sur le nombre de secret, et comme le schéma de partage proposé repose sur l'utilisation des systèmes d'équations linéaires surdéterminés dans $GF(2^8)$, cela entraine plusieurs avantages dont : (1) la complexité de calcule linéaire du partage et de la reconstruction des secrets, (2) puisque les pixels des images secrètes sont codés sur un octet dans $GF(2^8)$ et que aucun modulo premier n'est utilisé ceci mène à une représentation optimale des parts d'images attribuées aux participants ce qui implique qu'aucune augmentation ou réduction de la taille des images secrètes n'est faite et par conséquent le schéma est idéal, (3) une reconstruction parfaite des images secrètes en respectant la taille de ces dernières, (4) en plus le schéma est robuste puisqu'il se caractérise par la propriété de t-consistance qui offre la possibilité à tout groupe d'au moins t participants de reconstruire les images secrètes contrairement aux groupes de moins de t participants, (5) ainsi le partage dynamique qui est facilement implémenté en ajoutant ou supprimant des participants à l'aide de seulement de la matrice C sans altérer les parts distribuées auparavant.

Conclusion générale

Dans cette thèse nous nous sommes focalisés sur le problème de partage de secret en cryptographie. Pour cela nous avons proposé de concevoir et développer deux nouveaux schémas permettant de résoudre le problème de manière plus efficace et en utilisant de nouvelles techniques plus performantes.

Le premier schéma basé sur les automates cellulaires a été conçu dans le but de résoudre le problème de robustesse dans ces schémas. Car même si tous les schémas existants basés sur les automates cellulaires fournissent une complexité de calcul linéaire pour le partage et la reconstruction de secret(s), ils restent cependant d'un inconvénient majeur qui ne permet pas d'assurer un mécanisme de partage robuste puisque seules les parts définies par des numéros de configurations consécutifs peuvent être utilisées pour récupérer le secret. Par conséquent, un nombre très grand de sous-ensembles de participants autorisés sont incapables de reconstituer les données initialement partagées. Pour résoudre le problème, nous avons proposé d'affecter plusieurs configurations à chaque utilisateur afin de permettre à chaque sous-ensemble d'au moins t participants d'obtenir un accès à la reconstruction complète du secret. pour cela nous avons généré une matrice d'affectation spécifique en utilisant l'algorithme proposé, ainsi nous l'avons utilisé pour définir un nouveau mécanisme de partage/reconstruction de secret(s). Le schéma proposé a été montré comme robuste, et a été soumis à plusieurs analyses comparatives expérimentales afin d'illustrer la confidentialité des parts produites. La sécurité du schéma proposé a été établie, et les conditions de son idéalité ont été illustrées par rapport aux valeurs des paramètres t et n.

Contrairement aux schémas existants de partage de secret et non-utilisant les ACs, celui proposé assure une complexité de partage/reconstruction linéaire ce qui conduit à des performances plus rapide et évolutives, en plus de la propriété de robustesse complète fournie par rapport aux autres schémas existants basés sur les ACs. Toutefois, on peut avoir un coût élevé avec une taille des parts assez grande par rapport à la taille du secret pour certaines valeurs de t et n, induisant une non-idéalité du schéma. Après avoir effectué notre étude sur les ACs, nous concluons cette première approche par le fait qu'on ne peut fournir une solution idéale au problème de partage sans perdre la robustesse.

Le deuxième schéma qui a été proposé concerne le partage multi-secret, pour cette deuxième approche nous nous sommes servis des systèmes d'équations linéaires surdéterminés définis sur un corps fini de Rijndael ; aussi le schéma proposé a montré son efficacité en terme de partage d'images où la représentation des pixels est effectuée sur un champ

fini de Rijndael, et ceci contrairement aux autres schémas existants fonctionnant à la Shamir ou utilisant CRT opérant sur des modulo premier, où on aura soit une perte/expansion d'information, le deuxième schéma proposé s'est avéré optimal dans le sens où on atteint une représentation optimale des parts, et avec une complexité de calcul linéaire par rapport au nombre de participants, le seuil du partage et de la taille des secrets. Par conséquent, les meilleures performances sont obtenues lors de partage d'images numériques de taille très grandes.

Perspectives :

Pour une continuation de notre travail, plusieurs perspectives peuvent être envisagées :

- Optimisation de la matrice d'affectation proposée dans le premier schéma (Chapitre 4) pour trouver une solution avec moins de ligne dans la matrice ce qui permet d'optimiser encore plus la taille des parts produites.
- Proposition d'un schéma de partage de secret vérifiable dans lequel on pourra vérifier la validité des parts dans les deux cotés (client/serveur), et où la détection de tricheurs se ferra.
- Amélioration du coût de communication en offrant la possibilité à toutes personnes ayant rapport avec le secret de générer sa propre part.
- ajout/suppression dynamique de secrets avec un minimum coût de calcul et de communication.
- Réalisation d'un cryptosystème à clé secrète en se servant du principe de partage de secret, où pour chiffrer une image on a besoin de la décomposer en m blocs, chaque bloc étant considéré comme un secret, l'appel d'un schéma (t,m) de partage de secret permettra de générer m parts qui remplaceront chaque bloc de l'image, chaque part est décomposée en son tours en m blocs, et chaque bloc est partagé en m parts à l'aide du schéma de partage de secret (t,m), et ainsi de suite, l'opération est refaite pour tous les blocs de l'image jusqu'à atteindre le plus petit bloc de 128 bits de même taille que la clé dont va prendre AES par exemple, (ou n'importe quel autre système de chiffrement) comme entrée et appliquer la méthode de chiffrement à la clé donnée.

Bibliographie

[Alvarez 2005] G Alvarez, A Hernández Encinas, L Hernández Encinas et A Martín del Rey. *A secure scheme to share secret color images.* Computer physics communications, vol. 173, no. 1, pages 9–16, 2005. (Cité en pages vii, 52 et 55.)

[Alvarez 2008] G Alvarez, L Hernández Encinas et A Martín del Rey. *A multisecret sharing scheme for color images based on cellular automata.* Information Sciences, vol. 178, no. 22, pages 4382–4395, 2008. (Cité en pages vii, 52, 57 et 58.)

[Asmuth 1983] Charles Asmuth et John Bloom. *A modular approach to key safeguarding.* IEEE transactions on information theory, vol. 30, no. 2, pages 208–210, 1983. (Cité en pages 13, 19, 21, 23 et 25.)

[Benaloh 1986] Josh Cohen Benaloh. *Secret sharing homomorphisms : Keeping shares of a secret secret.* In Conference on the Theory and Application of Cryptographic Techniques, pages 251–260. Springer, 1986. (Cité en page 11.)

[BENTOUILA 2013] Sara BENTOUILA et Mohamed Kamel FARAOUN. *Cryptographie symétrique des images numériques par les automates cellulaires.* PhD thesis, 2013. (Cité en page 49.)

[Bhagvati 2014] Chakravarthy Bhagvati. *CRT based threshold multi secret sharing scheme.* International Journal of Network Security, vol. 16, no. 4, pages 249–255, 2014. (Cité en pages vi, 6, 23 et 118.)

[Blakley 1979] George Robert Blakley. *Safeguarding cryptographic keys.* Proc. of the National Computer Conference1979, vol. 48, pages 313–317, 1979. (Cité en pages 1, 13, 17, 21, 25 et 53.)

[Brickell 1990] Ernest F Brickell et Douglas R Stinson. *The detection of cheaters in threshold schemes.* In Proceedings on Advances in cryptology, pages 564–577. Springer-Verlag New York, Inc., 1990. (Cité en page 21.)

[Chen 2008] Chien-Chang Chen, Wen-Yin Fu et Chaur-Chin Chen. *A Geometry-Based Secret Image Sharing Approach.* J. Inf. Sci. Eng., vol. 24, no. 5, pages 1567–1577, 2008. (Cité en pages vi, 6, 21 et 118.)

[Cheraghi 2014] Abbas Cheraghi. *Sharing several secrets based on Lagrangeś interpolation formula and Cipher feedback mode.* International Journal of Nonlinear Analysis and Applications, vol. 5, no. 2, pages 60–66, 2014. (Cité en pages vi, 6, 23 et 118.)

[Chor 1985] Benny Chor, Shafi Goldwasser, Silvio Micali et Baruch Awerbuch. *Verifiable secret sharing and achieving simultaneity in the presence of faults.* In Foundations of Computer Science, 1985., 26th Annual Symposium on, pages 383–395. IEEE, 1985. (Cité en page 21.)

[Dehkordi 2008] Massoud Hadian Dehkordi et Samaneh Mashhadi. *New efficient and practical verifiable multi-secret sharing schemes.* Information Sciences, vol. 178, no. 9, pages 2262–2274, 2008. (Cité en page 88.)

[del Rey 2005] A Martín del Rey, J Pereira Mateus et G Rodríguez Sánchez. *A secret sharing scheme based on cellular automata.* Applied mathematics and computation, vol. 170, no. 2, pages 1356–1364, 2005. (Cité en pages vii, 52 et 55.)

[Eslami 2010] Z Eslami et J Zarepour Ahmadabadi. *A verifiable multi-secret sharing scheme based on cellular automata.* Information Sciences, vol. 180, no. 15, pages 2889–2894, 2010. (Cité en pages vii, 52, 58 et 88.)

[Fouque 2001] Pierre-Alain Fouque. *Le partage de clés cryptographiques : Théorie et Pratique.* PhD thesis, 2001. (Cité en page 12.)

[Fredkin 1990] Edward Fredkin. *An informational process based on reversible universal cellular automata.* Physica D : Nonlinear Phenomena, vol. 45, no. 1-3, pages 254–270, 1990. (Cité en pages 45, 49 et 77.)

[Gardner 1970] Martin Gardner. *Mathematical games : The fantastic combinations of John Conwayś new solitaire game life.* Scientific American, vol. 223, no. 4, pages 120–123, 1970. (Cité en pages 40 et 42.)

[Guan 1987] Puhua Guan. *Cellular automaton public key cryptosystems.* Complex Systems, vol. 1, no. 1, pages 51–56, 1987. (Cité en page 50.)

[Harn 2010] Lein Harn et Changlu Lin. *Strong (n, t, n) verifiable secret sharing scheme.* Information Sciences, vol. 180, no. 16, pages 3059–3064, 2010. (Cité en pages vi, 6, 11, 22 et 118.)

[He 1994] Jingmin He et Edward Dawson. *Multistage secret sharing based on one-way function.* Electronics Letters, vol. 30, no. 19, pages 1591–1592, 1994. (Cité en page 10.)

[Hernández Encinas 2009] Luis Hernández Encinas, Fernando Hernández Álvarez et Raúl Durán Díaz. *Graphic multi-secret sharing schemes with one-dimensional cellular automata.* 2009. (Cité en pages vii, 52 et 58.)

[Herzberg 1995] Amir Herzberg, Stanisław Jarecki, Hugo Krawczyk et Moti Yung. *Proactive secret sharing or : How to cope with perpetual leakage.* In Annual International Cryptology Conference, pages 339–352. Springer, 1995. (Cité en page 11.)

[Kaced 2012] Tarik Kaced. *Partage de secret et théorie algorithmique de línformation.* PhD thesis, Citeseer, 2012. (Cité en pages 9 et 11.)

[Kari 1990] Jarkko Kari. *Reversibility of 2D cellular automata is undecidable.* Physica D : Nonlinear Phenomena, vol. 45, no. 1-3, pages 379–385, 1990. (Cité en page 50.)

[Kari 1992] Jarkko Kari. *Cryptosystems based on reversible cellular automata.* Manuscript, August, 1992. (Cité en page 50.)

[Lin 2010] Pei-Yu Lin et Chi-Shiang Chan. *Invertible secret image sharing with steganography.* Pattern Recognition Letters, vol. 31, no. 13, pages 1887–1893, 2010. (Cité en page 88.)

[Liu 1968] Chung Laung Liu. *Introduction to combinatorial mathematics.* 1968. (Cité en page 7.)

[Liu 2012] Yan-Xiao Liu, Lein Harn, Ching-Nung Yang et Yu-Qing Zhang. *Efficient (n, t, n) secret sharing schemes.* Journal of Systems and Software, vol. 85, no. 6, pages 1325–1332, 2012. (Cité en pages vi, 6, 22 et 118.)

[Marañón 2003] GonzaloÁlvarez Marañón, Luis Hernández Encinas, Ascensión Hernández Encinas, Ángel Martín del Rey et Gerardo Rodríguez Sánchez. *Graphic cryptography with pseudorandom bit generators and cellular automata.* In International Conference on Knowledge-Based and Intelligent Information and Engineering Systems, pages 1207–1214. Springer, 2003. (Cité en pages vii, 52, 53, 56 et 57.)

[Marañón 2005] Gonzalo Álvarez Marañón, Luis Hernández Encinas et Ángel Martín del Rey. *A new secret sharing scheme for images based on additive 2-dimensional cellular automata.* In Iberian Conference on Pattern Recognition and Image Analysis, pages 411–418. Springer, 2005. (Cité en pages vii, 52 et 56.)

[McEliece 2012] Robert J McEliece. Finite fields for computer scientists and engineers, volume 23. Springer Science & Business Media, 2012. (Cité en page 91.)

[Meier 1991] Willi Meier et Othmar Staffelbach. *Analysis of pseudo random sequences generated by cellular automata.* In Workshop on the Theory and Application of of Cryptographic Techniques, pages 186–199. Springer, 1991. (Cité en page 50.)

[Mignotte 1982] Maurice Mignotte. *How to share a secret.* In Workshop on Cryptography, pages 371–375. Springer, 1982. (Cité en page 19.)

[Mohamed 2014] Faraoun Kamel Mohamed. *A parallel block-based encryption schema for digital images using reversible cellular automata.* Engineering Science and Technology, an International Journal, vol. 17, no. 2, pages 85–94, 2014. (Cité en page 50.)

[Nagaraj 2010] Srinivisan Nagaraj, Kishore Bhamidipati et G Apparao. *An Approach to Security Using Rijndael Algorithm.* International Journal of Computer Applications Volume 8, no. 5, 2010. (Cité en page 92.)

[Nicolas 2000] Jombart Nicolas. *How to genrate random numbers using a computer.* 2000. (Cité en page 78.)

[Ostrovsky 1991] Rafail Ostrovsky et Moti Yung. *How to withstand mobile virus attacks.* In Proceedings of the tenth annual ACM symposium on Principles of distributed computing, pages 51–59. ACM, 1991. (Cité en page 11.)

[Pedersen 1991] Torben Pryds Pedersen. *A threshold cryptosystem without a trusted party.* In Workshop on the Theory and Application of of Cryptographic Techniques, pages 522–526. Springer, 1991. (Cité en page 22.)

[Renner 2014] Soline Renner. *Protection des algorithmes cryptographiques embarqués.* PhD thesis, Université de Bordeaux, 2014. (Cité en pages 8, 10 et 11.)

[Sandhya Sarma 2013] K.N. Sandhya Sarma, Hemraj S. Lamkuche et S. Umamaheswari. *A Review of Secret Sharing Schemes.* Information Technology, vol. 5, pages 67–72, 2013. (Cité en page 11.)

[Seredyński 2003] Franciszek Seredyński, Pascal Bouvry et Albert Y Zomaya. *Cellular programming and symmetric key cryptography systems.* In Genetic and Evolutionary Computation Conference, pages 1369–1381. Springer, 2003. (Cité en page 50.)

[Shamir 1979] Adi Shamir. *How to share a secret.* Communications of the ACM, vol. 22, no. 11, pages 612–613, 1979. (Cité en pages 1, 13, 14, 17, 21, 22, 25 et 53.)

[Shannon 1949] Claude E Shannon. *Communication theory of secrecy systems.* Bell system technical journal, vol. 28, no. 4, pages 656–715, 1949. (Cité en page 10.)

[Silverman 1999] Joseph H Silverman. *Fast multiplication in finite fields GF (2N).* In International Workshop on Cryptographic Hardware and Embedded Systems, pages 122–134. Springer, 1999. (Cité en page 92.)

[Szaban 2006] Miroslaw Szaban, Franciszek Seredynski et Pascal Bouvry. *Collective behavior of rules for cellular automata-based stream ciphers.* In 2006 IEEE International Conference on Evolutionary Computation, pages 179–183. IEEE, 2006. (Cité en page 50.)

[Testa 2008] Joseph S Testa. Investigations of cellular automata-based stream ciphers. ProQuest, 2008. (Cité en page 50.)

[Thien 2002] Chih-Ching Thien et Ja-Chen Lin. *Secret image sharing.* Computers & Graphics, vol. 26, no. 5, pages 765–770, 2002. (Cité en page 88.)

[TINDO 2006] Gilbert TINDO. *CCPBAC : un cryptosystème à clés publiques basés sur des automates cellulaires.* In 8ème Colloque Africain sur la Recherche en Informatique, 2006. (Cité en page 51.)

[Toffoli 1990] Tommaso Toffoli et Norman H Margolus. *Invertible cellular automata : A review.* Physica D : Nonlinear Phenomena, vol. 45, no. 1, pages 229–253, 1990. (Cité en pages 46 et 50.)

[Ulam 1952] Stanislaw Ulam. *Random processes and transformations.* In Proceedings of the International Congress on Mathematics, volume 2, pages 264–275. Citeseer, 1952. (Cité en page 29.)

[Von Neumann 1966] John Von Neumann, Arthur W Burks*et al. Theory of self-reproducing automata.* IEEE Transactions on Neural Networks, vol. 5, no. 1, pages 3–14, 1966. (Cité en page 29.)

[Wolfram 1985] Stephen Wolfram. *Cryptography with cellular automata.* In Conference on the Theory and Application of Cryptographic Techniques, pages 429–432. Springer, 1985. (Cité en page 1.)

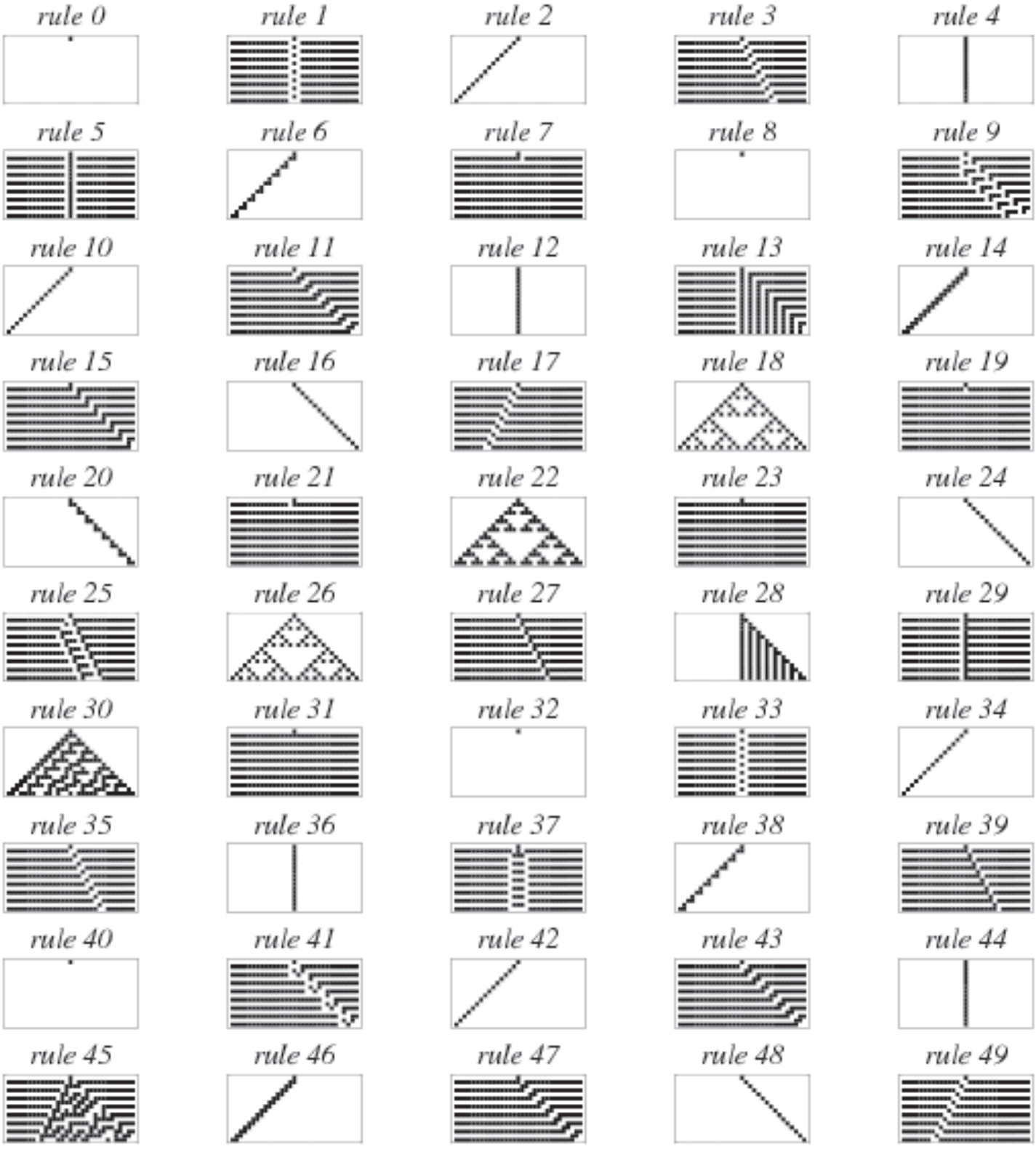
[Wu 2012] Xiaotian Wu, Duanhao Ou, Qiming Liang et Wei Sun. *A user-friendly secret image sharing scheme with reversible steganography based on cellular automata.* Journal of Systems and Software, vol. 85, no. 8, pages 1852–1863, 2012. (Cité en pages viii, 52, 60 et 88.)

[Wu 2013] Xiaotian Wu et Wei Sun. *Secret image sharing scheme with authentication and remedy abilities based on cellular automata and discrete wavelet transform.* Journal of Systems and Software, vol. 86, no. 4, pages 1068–1088, 2013. (Cité en page 88.)

[Yang 2011] Ching-Nung Yang et Yu-Ying Chu. *A general (k, n) scalable secret image sharing scheme with the smooth scalability.* Journal of Systems and Software, vol. 84, no. 10, pages 1726–1733, 2011. (Cité en page 88.)

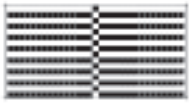
Annexe

Diagrammes espace-temps des 256 automates cellulaires élémentaires


rule 0
rule 1
rule 2
rule 3
rule 4
rule 5
rule 6
rule 7
rule 8
rule 9
rule 10
rule 11
rule 12
rule 13
rule 14
rule 15
rule 16
rule 17
rule 18
rule 19
rule 20
rule 21
rule 22
rule 23
rule 24
rule 25
rule 26
rule 27
rule 28
rule 29
rule 30
rule 31
rule 32
rule 33
rule 34
rule 35
rule 36
rule 37
rule 38
rule 39
rule 40
rule 41
rule 42
rule 43
rule 44
rule 45
rule 46
rule 47
rule 48
rule 49

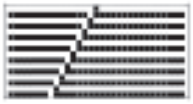
rule 50

rule 51



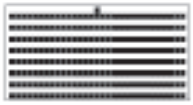
rule 52

rule 53



rule 54

rule 55

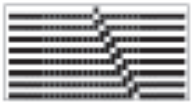


rule 56

rule 57

rule 58

rule 59

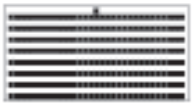


rule 60

rule 61

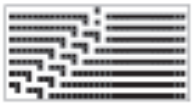
rule 62

rule 63



rule 64

rule 65



rule 66

rule 67

rule 68

rule 69

rule 70

rule 71

rule 72

rule 73

rule 74

rule 75

rule 76

rule 77

rule 78

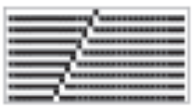
rule 79

rule 80

rule 81

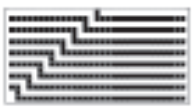
rule 82

rule 83



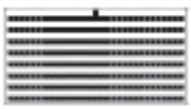
rule 84

rule 85



rule 86

rule 87



rule 88

rule 89

rule 90

rule 91

rule 92

rule 93

rule 94

rule 95

rule 96

rule 97

rule 98

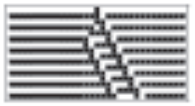
rule 99

rule 100

rule 101

rule 102

rule 103

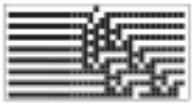


rule 104

rule 105

rule 106

rule 107

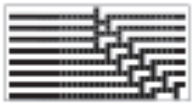


rule 108

rule 109

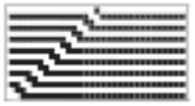
rule 110

rule 111



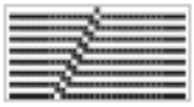
rule 112

rule 113



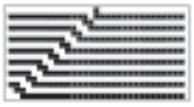
rule 114

rule 115



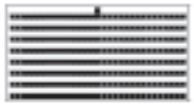
rule 116

rule 117



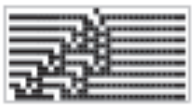
rule 118

rule 119



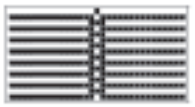
rule 120

rule 121



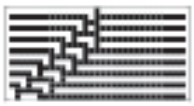
rule 122

rule 123



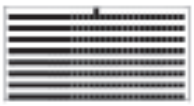
rule 124

rule 125



rule 126

rule 127



rule 128

rule 129

rule 130

rule 131

rule 132

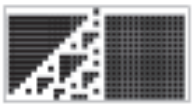
rule 133

rule 134

rule 135

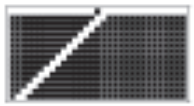
rule 136

rule 137



rule 138

rule 139



rule 140

rule 141

rule 142

rule 143

rule 144

rule 145

rule 146

rule 147

rule 148

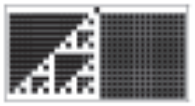
rule 149

rule 150

rule 151

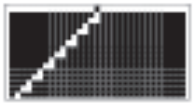
rule 152

rule 153



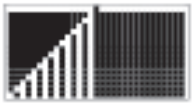
rule 154

rule 155



rule 156

rule 157



rule 158

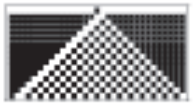
rule 159

rule 160

rule 161

rule 162

rule 163



rule 164

rule 165

rule 166

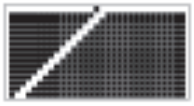
rule 167

rule 168

rule 169

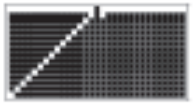
rule 170

rule 171



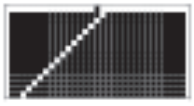
rule 172

rule 173



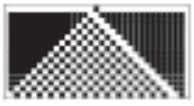
rule 174

rule 175



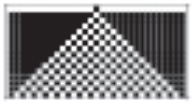
rule 176

rule 177



rule 178

rule 179

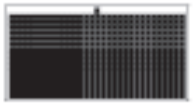


rule 180

rule 181

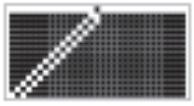
rule 182

rule 183



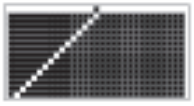
rule 184

rule 185



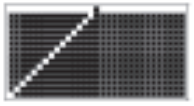
rule 186

rule 187



rule 188

rule 189



rule 190

rule 191

rule 192

rule 193

rule 194

rule 195

rule 196

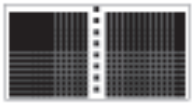
rule 197

rule 198

rule 199

rule 200

rule 201

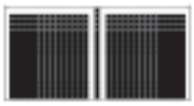


rule 202

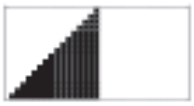
rule 203

rule 204

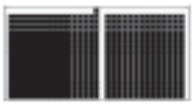
rule 205



rule 206



rule 207

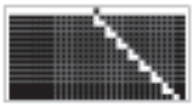


rule 208

rule 209

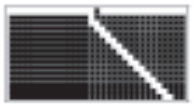
rule 210

rule 211



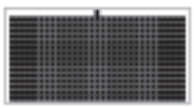
rule 212

rule 213



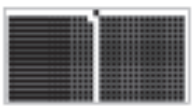
rule 214

rule 215



rule 216

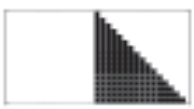
rule 217



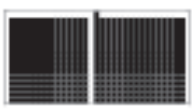
rule 218

rule 219

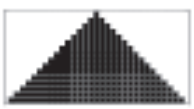
rule 220



rule 221



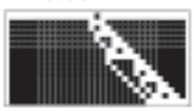
rule 222



rule 223

rule 224

rule 225

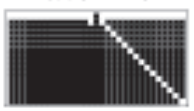


rule 226

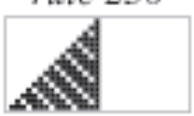
rule 227

rule 228

rule 229

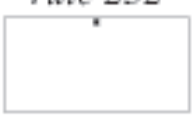


rule 230



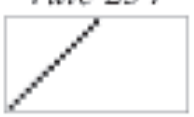
rule 231

rule 232

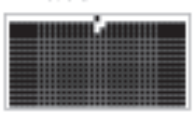


rule 233

rule 234

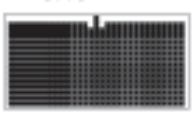


rule 235

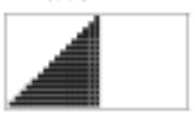


rule 236

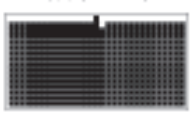
rule 237



rule 238



rule 239



rule 240

rule 241

rule 242

rule 243

rule 244

rule 245

rule 246

rule 247

rule 248

rule 249

rule 250

rule 251

rule 252

rule 253

rule 254

rule 255