Compte-rendu hebdomadaire

Compte-rendu 2

Valentin Noyé

6 octobre 2025 - 10 octobre 2025

GitHub : https://github.com/owilo/SecretSharing

1 Résumé des tâches réalisées cette semaine

1. Lecture d'une étude sur l'ensemble des techniques de partage d'images secrètes [1].
2. Section 2.1.1 – Recherche sur le théorème des restes chinois [1–5].
3. Section 2.1.2 – Recherche sur le fonctionnement des codes de Reed-Solomon [5, 6].
4. Section 2.2 – Étirement d'histogramme afin de forcer les pixels à être contenus sur des nuances entre 0 et 250.
5. Section 2.3 – Reconstruction de parts bruitées avec JPEG.

2 Travail réalisé

2.1 Recherches en arithmétique des corps finis

2.1.1 Théorème des restes Chinois

Parmi les articles proposés, certains effectuent le partage d'images secrètes sur la base du théorème des restes chinois [1–5]. Supposons que nous souhaitons résoudre pour x un système de congruences de la forme :

$$\begin{aligned} x &\equiv a_1 \bmod m_1 \\ x &\equiv a_2 \bmod m_2 \\ &\dots \\ x &\equiv a_n \bmod m_n \end{aligned}$$

La méthode ne fonctionne que lorsque tous les m_i sont premiers entre eux. On définit M :

$$M = \prod_{i=1}^{n} m_i$$

Et pour tout $i = \{1, \dots, n\}$,

$$M_i = \frac{M}{m_i}$$

Pour tout M_i, on détermine l'inverse modulaire y_i modulo m_i :

$$M_i y_i \equiv 1 \bmod m_i$$

Cela peut se faire avec la méthode d'Euclide étendue ou selon le petit théorème de Fermat. La solution unique a cette équation est donc la suivante :

$$x \equiv \sum_{i=1}^{n} a_i M_i y_i \bmod M$$

2.1.2 Codes de Reed-Solomon

Dans les recherches menées, il existe certaines méthodes qui emploient les codes correcteurs d'erreur de Reed-Solomon afin de garantir l'intégrité des parts [5, 6].

Notons un message $m = \{m_1, m_2, \ldots, m_k\}$ composé d'octets où $m_i \in GF(2^8)$. Nous voulons encoder ce message de taille k dans un message de taille n où $2t = n-k$ données supplémentaires représentent les octets de parité $r = \{r_1, r_2, \ldots, r_{2t}\}$. La méthode de Reed-Solomon permet de détecter jusqu'à $2t$ erreurs, et d'en corriger t dans le message reçu.

Construction du message par la méthode de Reed-Solomon systématique Dans la méthode originale employée par Reed et Solomon, le message m est encodé dans les points du polynôme de degré $k-1$ passant par l'ensemble de ces points, puis les octets de parités r sont évalués à différents autres points du polynôme. Si une valeur du message venait à être érronée, elle n'appartiendrait pas au polynôme et serait corrigée selon la parité.

Le problème, c'est que le nouveau message formé doit alors encoder les coefficients du polynôme ainsi que les octets de parité, le message original m n'est plus présent dans le code de Reed-Solomon.

Dans la version dite « systématique » de Reed-Solomon, les octets du message m sont insérés en tant que coefficients d'un polynôme $M(x)$ de degré $k-1$:

$$M(x) = m_1 + m_2 x + \ldots + m_k x^{k-1}$$

Dans $GF(2^8)$, nous définissons $f(x) = x^8+x^4+x^3+x^2+1$ le polynôme irréductible afin d'obtenir la relation isomorphoque suivante :

$$GF(2^8) \cong \frac{GF(2)[x]}{f(x)}$$

Il en découle de cette configuration une valeur $\alpha = x$ représentant l'élément primitif de $GF(2^8)$ tel que $\alpha^n \equiv 1 \bmod f(x)$.

On définit donc le polynôme générateur $g(x)$ de degré $2t$:

$$g(x) = (x-\alpha^1)(x-\alpha^2)\ldots(x-\alpha^{2t})$$

Pour encoder le message $M(x)$, celui-ci est multiplié par x^{2t}, ce qui permet d'encoder la parité dans les plus faibles degrés. La parité est alors calculée selon une division polynomiale $M(x)x^{2t}/g(x)$ pour produire un reste $R(x)$ tel que le nouveau message envoyé est encodé sous le polynôme

$$C(x) = M(x)2^{2t} + R(x)$$

Génération des syndromes À la réception, on obtient un message $C'(x) = c'_0 + c'_1 x + \ldots + c'_n x^{n-1}$. Celui-ci peut contenir des erreurs ou non. Pour le vérifier, nous calculons un ensemble de $2t$ syndromes S_i :

$$S_i = \sum_{j=0}^{n-1} c'_j (\alpha^i)^j$$

Rappelons que tout arithmétique est effectuée dans $GF(2^8)$. Si tous les $S_i = 0$, alors le message n'a pas d'erreur et les k premiers octets (m) peuvent être utilisés. Sinon, il existe au moins une erreur.

Construction du polynôme correcteur d'erreur par algorithme de Berlekamp-Massey et recherche de Chien Si $\exists S_i$, nous avons une erreur. À partir des syndromes S_i, nous construisons le polynôme correcteur d'erreur $\Lambda(x)$ tel que :

$$\Lambda(x) = \prod_{i=1}^{\nu}(1 - x\alpha^{p_i})$$

avec ν le nombre d'erreurs et p_i la position de l'erreur d'indice i. Par simplicité, dénotons $X_i = \alpha^{p_i}$:

$$\Lambda(x) = \prod_{i=1}^{\nu}(1 - xX_i)$$

Or nous ne connaissons pas ν et p_i, mais nous avons l'ensemble des syndromes S_i. Le procédé de construction de ce polynôme peut être mené de deux manières différentes : par algorithme d'Euclide ou par algorithme de Berlekamp-Massey. En particulier, l'algorithme de Berlekamp-Massey a pour but est de trouver à partir de ces syndromes le registre à décalage à rétroaction linéaire (LFSR) le plus court capable de les générer, et c'est cet algorithme qui est préféré. Par simplicité, il n'est pas détaillé ici.

Ici, $\Lambda(X_i^{-1}) = 0$ correspondrait à une erreur en position p_i. On emploie alors très simplement une recherche de Chien où pour les déterminer, on vérifie $\Lambda(X_i^{-1}) = 0$ pour tout $i = 0, 1, \ldots, n-1$.

Calcul des magnitudes d'erreur par la méthode de Forney et correction du message On définit un polynôme à partir des syndromes :

$$S(x) = S_1 + S_2x + \ldots + S_{2t}x^{2t-1}$$

puis le polynôme d'évaluation de l'erreur :

$$\Omega(x) = \Lambda(x)S(x) \bmod x^{2t}$$

Ce polynôme de degré inférieur à t encode à la fois la position et la magnitude des erreurs. La formule de Forney définit pour chaque erreur p_i :

$$E_{p_i} = -\frac{\Omega(X_i^{-1})}{\Lambda'(X_i^{-1})}$$

où $\Lambda'(x)$ représente la dérivée formelle de $\Lambda(x)$ dans $GF(2^8)$, c'est à dire que nous omettons les degrés d'ordre pair.

Enfin, nous reconstruisons le message d'origine en corrigeant les erreurs :

$$C'_{p_i} \leftarrow C'_{p_i} + E_{p_i}$$

2.2 Partage d'image secrète sur GF(251) avec étirement d'histogramme

Nous comparons à présent ce qu'il se passe lorsque nous appliquons un étirement d'histogramme sur l'image d'entrée plutôt que de limiter les nuances à 250 afin de les contenir dans GF(251). La figure 1 montre tout d'abord nos images de test.

(a) Grenouille

(b) Stop

(c) Poivrons

FIGURE 1 – Comparaison des images après étirement ou limitation des nuances à 250.

Si nous nous intéressons à l'histogramme pour l'image « Stop » en figure 2, nous constatons que les nuances blanches sont réduites à 250, formant ainsi un pic lorsque les valeurs sont limitées entre 0 et 250. Si nous appliquons en revanche un étirement d'histogramme, nous notons l'apparence de certains pics à des intervalles régulières.

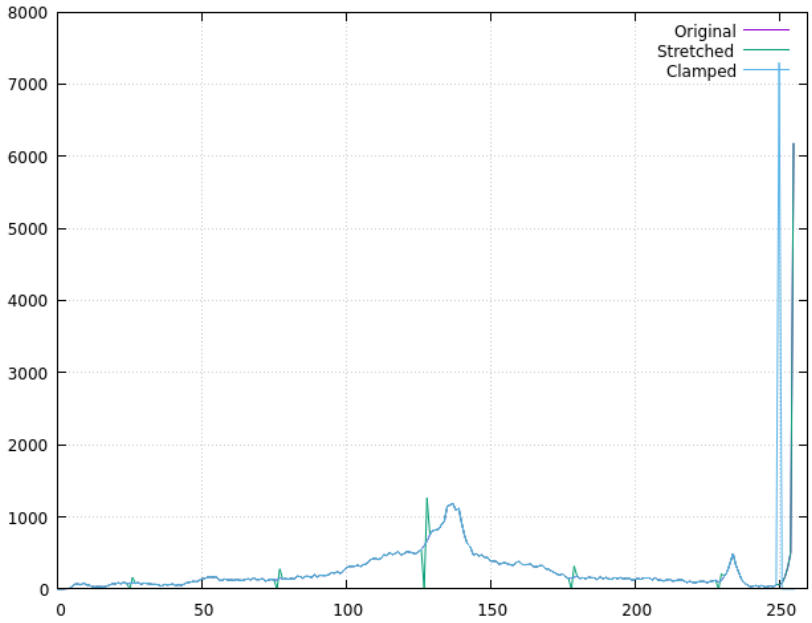


FIGURE 2 – Comparaison des histogrammes après étirement ou limitation des nuances à 250.

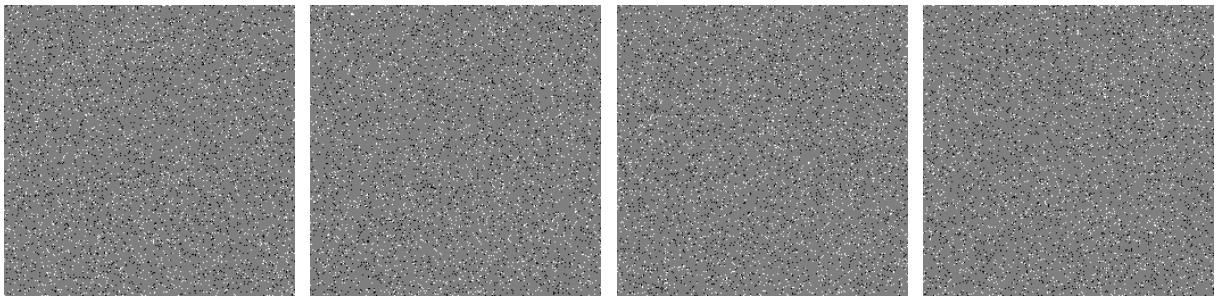
En comparant les images sur différentes métriques dans le tableau 1, nous constatons que l'efficacité des méthodes dépend de la distribution des nuances hautes dans l'image. En effet, un étirement d'histogramme fonctionne mieux sur des images très lumineuses tandis que limiter les valeurs à 250 fonctionne parfaitement sur des images sans valeur de blanc supérieure à 250.

Méthode de réduction à GF(251)	**Grenouille**	**Stop**	**Poivrons**
Limite des valeurs à 250	81,8240 dB	44,0964 dB	∞ dB
Étirement d'histogramme	65,3092 dB	66,0466 dB	65,9094 dB

TABLE 1 – Comparaison des métriques selon la méthode de réduction des pixels à GF(251).

2.3 Bruitage des parts d'une image avec JPEG

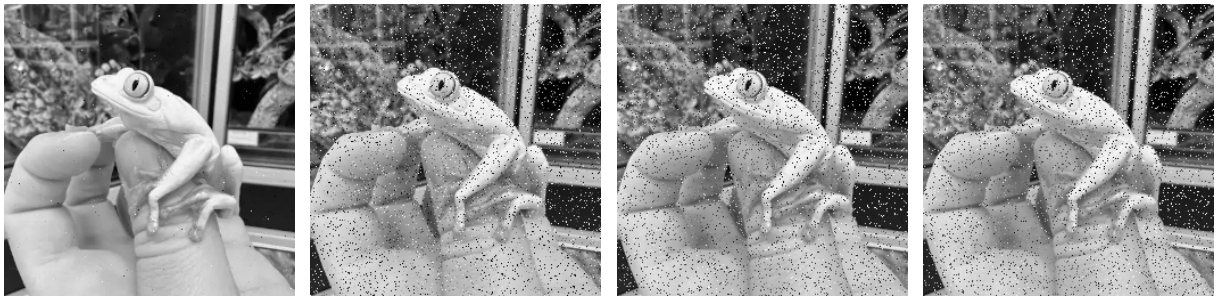
Dans cette section, nous nous intéressons au bruitage des parts à l'aide de JPEG. Pour ce faire, l'image est décomposée en $n = 6$ parts. Nous décidons d'un seuil $k = 3$ pour la reconstruction. Les deux premières parts sont maintenues intactes tandis que nous bruitons les 4 dernières avec un facteur de qualité de 100. Dans la figure 3, l'erreur obtenue est mise en évidence par la présence de pixels blancs (erreur de 1) et de pixels noirs (erreur de -1). Les pixels gris indiquent une valeur inchangée.



(a) Part 3 (b) Part 4 (c) Part 5 (d) Part 6

FIGURE 3 – Carte des erreurs introduites par compression JPEG des parts 4 à 6.

En reconstruisant l'image avec les parts inchangées ainsi qu'une part bruitée, nous obtenons des résultats tels que dans la figure 4 où il est possible de voir que l'erreur s'accentue au fur et à mesure que la troisième part s'éloigne des deux premières $x_1 = 1$ et $x_2 = 2$.



(a) $x_i = \{1, 2, 3\}$ (b) $x_i = \{1, 2, 4\}$ (c) $x_i = \{1, 2, 5\}$ (d) $x_i = \{1, 2, 6\}$

FIGURE 4 – Reconstructions de l'image selon les parts sélectionnées.

En particulier, le tableau 2 met en évidence ce phénomène. Nous constatons ici que le nombre de pixels changés par JPEG n'influe pas grandement sur la qualité visuelle de l'image reconstruite.

Parts	**PSNR (dB)** $\uparrow$	**NPCR (%)** $\downarrow$	**UACI (%)** $\downarrow$
$x_i = \{1, 2, 3\}$	37,3137	8,5556	0,13391
$x_i = \{1, 2, 4\}$	18,1062	8,4396	2,98437
$x_i = \{1, 2, 5\}$	17,9690	8,4534	3,14649
$x_i = \{1, 2, 6\}$	17,6887	8,4839	3,11521

TABLE 2 – Comparaison des métriques de reconstruction selon les parts sélectionnées.

Pour la suite, il serait intéressant d'étudier comment la disposition des parts influe sur la qualité de la reconstruction.

3 Travail à effectuer

1. Approfondir la recherche et le développement de méthodes de partage d'images secrètes de l'état de l'art.
2. Continuer la recherche ainsi que l'analyse de la reconstruction d'image secrètes lorsque des parts sont bruitées.

4 Activités

— **Réunions :**

1. Réunion SecMul – Présentations de Norman et de Khélian – Mercredi 08/10/2025

Références

[1] Sanchita Saha, Arup Kumar Chattopadhyay, Anup Kumar Barman, Amitava Nag, and Sukumar Nandi. Secret image sharing schemes : A comprehensive survey. *IEEE Access*, 11 :98333–98361, 2023.

[2] P.K. Meher and J.C. Patra. A new approach to secure distributed storage, sharing and dissemination of digital image. In *2006 IEEE International Symposium on Circuits and Systems*, pages 4 pp.–376, 2006.

[3] Chin-Chen Chang, Ngoc-Tu Huynh, and Hai-Duong Le. Lossless and unlimited multi-image sharing based on chinese remainder theorem and lagrange interpolation. *Signal Processing*, 99 :159–170, 2014.

[4] Shyong Jian Shyu and Ying-Ru Chen. Threshold secret image sharing by chinese remainder theorem. In *2008 IEEE Asia-Pacific Services Computing Conference*, pages 1332–1337, 2008.

[5] Chaoying Wang, Yong Peng, Zhibiao Liang, Yu Wang, Gang Ke, and Zhiping Jin. Reversible extended secret image sharing with ability to correct errors based on chinese remainder theorem. *Heliyon*, 9(4) :e14918, 2023.

[6] Sébastien Beugnon, Pauline Puteaux, and William Puech. Privacy protection for social media based on a hierarchical secret image sharing scheme. pages 679–683, 09 2019.