J. Vis. Commun. Image R. 24 (2013) 318–331

Contents lists available at SciVerse ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Natural language letter based visual cryptography scheme

Hsiao-Ching Lin [a], Ching-Nung Yang [b], Chi-Sung Laih [c,1], Hui-Tang Lin [c,*]

[a] Institute of Computer and Communication Engineering, National Cheng Kung University, Taiwan
[b] Department of Computer Science and Information Engineering, National Dong Hwa University, Taiwan
[c] Department of Electrical Engineering, National Cheng Kung University, Taiwan

ARTICLE INFO

Article history:
Received 19 May 2012
Accepted 17 December 2012
Available online 9 January 2013

Keywords:
Deterministic VCS
Letter-based
Natural language
Probabilistic VCS
Secret sharing
Subpixel
Visual cryptography scheme
Visual secret sharing

ABSTRACT

Naor and Shamir proposed the notion of a (k, n) visual cryptography scheme (VCS), which allows k or more stacked transparent share images to reveal a secret image. It can be used without prerequisite knowledge of cryptography or complex computations. In these schemes, no information about the secret can be obtained from fewer than k shares. Previous VCSs use black and white subpixels to create share images. In this paper, we present a letter-based VCS (LVCS) where pixels are replaced by letters for the share images. Shares can now be constructed using meaningful data as subterfuge all while carrying secret data in plain sight, and an adversary will not recognize them as containing secrets. We prove that the proposed (k, n)-LVCS satisfies contrast and security conditions and secret information may be reconstructed by any k shares but with less than k shares reveal nothing.

1. Introduction

In a (k, n) secret image sharing scheme, the secret is divided into n share images and distributed to n holders in such a way that any k or more shares can reconstruct the secret, but any fewer than $k-1$ shares cannot obtain any information on the secret. There are two major categories of secret image sharing schemes: One is the polynomial-based secret image sharing scheme [1–5] based on Shamir's well-known secret sharing [6], and the other is visual cryptography scheme (VCS). The schemes in [1–5] can reconstruct a secret image without any distortion, but they need Lagrange interpolation polynomial computations. On the other hand, VCS has a novel stack-to-see property where decryption requires neither knowledge of cryptography nor complex computations. Share holders may transfer their shares onto transparencies and superimpose them to visually decode the secret through the human visual system (HVS). In (k, n)-VCS, a secret image is encrypted into n shares by expanding each secret pixel into m (known as the pixel expansion) subpixels. The difference between the secret pixel and the subpixel is that the "secret pixel" denotes the pixel located in the secret image, and the "subpixel" is the pixel of the share. The size of a subpixel is the same as that of the secret pixel and shares are expanded m times. The first VCS was proposed by Naor and Shamir [7] which used whiteness to distinguish black from white. The visual quality of a reconstructed image in VCS is degraded by large pixel expansion, and thus most studies try to enhance visual quality and/or reduce pixel expansion. Size-reduced VCSs schemes were proposed accordingly in [8–14]. Some have no pixel expansion $(m = 1)$ and are known as probabilistic VCS (PVCS) [9–12]. Conventional VCS with fixed m $(m > 1)$ are referred to as deterministic VCS (DVCS).

Interestingly, VCS uses visual authentication by stacking shares to reconstruct the secret as in "seeing-is-believing" [15–17]. One example application is users can enter passwords for an online server while preventing attackers from intercepting the passwords. For example, Naor and Pinkas [15] suggested using VCS in a transparency-on-screen version to authenticate itself to a user sitting in front of the screen. In addition, there were many types of VCS developed to achieve different applications. VCSs with specific features were proposed, e.g., sharing multiple secrets [18–21], cheating prevention [22], and solving share misalignment problems [23,24]. Currently, there is a vast amount of research on VCS and recently a book covering an extensive range of topics related to VCS was published [25].

Shamir's secret sharing scheme [6] is perfectly secure because possessing less than the required number of secret shares provides no information about the secret itself. Naor and Shamir [7] extended this concept into a visual variant resulting in the k out of n secret sharing scheme now known as visual cryptography scheme (VCS), which conforms to the perfect security property [10,16]. Naor and

* Corresponding author. Address: No. 1, University Road, Tainan City 701, Taiwan. Fax: +886 6 2345482.
E-mail addresses: gookey22@gmail.com (H.-C. Lin), cnyang@mail.ndhu.edu.tw (C.-S. Laih), htlin100@gmail.com, htlin@mail.ncku.edu.tw (H.-T. Lin).
[1] Posthumous Attribution.

1047-3203/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jvcir.2013.01.003

Shamir's VCS is considered to be unconditionally secure, which means the chance of disclosure or tampering with secret data is zero even with no limits on the adversary's computational power [26,27]. Based on the concepts of Naor and Shamir's VCS, the stacking secure feature is defined and applied. This feature means stacking less than the number of threshold shares will not disclose the secret and it is implied to have unconditional security [28]. In this paper, we propose a method named LVCS which is derived from VCS and we prove it conforms to the stacking secure feature.

Takizawa and Yamamura [29,30] proposed two secret sharing schemes using natural language letters (Japanese). In the first scheme [29], the secret is revealed as the non-overlapped Japanese characters when stacking shares. All secret characters are placed in non-overlapped locations in each share, and the remaining space is overlapped by successive shares.

For the second scheme [29] (which is presented again in [30]), the secret is revealed as a meaningful sentence in a particular column arising from the alignment of plaintext sentences in a specific sequence. The authors employ a morphological analyzer to carefully design their shares to try to prevent secret information leakage. However, Takizawa et al.'s two schemes cannot ensure security as will be explained later in this work.

The remainder of this work is organized as follows: In Section 2, previous VCSs are described. Motivation and design concept are described in Section 3. In Section 4, we propose a (k,n)-LVCS and an enhancement. Experimental results and applications are given in Section 5. Finally, some concluding remarks are provided in Section 6.

2. Preliminaries

Previous VCSs are black and white pixel-based schemes. There are two types of VCS: one is DVCS and the other is PVCS. For DVCS, we use m subpixels to represent a secret pixel, while PVCS only uses one subpixel to represent a secret pixel. Our LVCS can be implemented using either DVCS or PVCS.

2.1. DVCS

Naor and Shamir's (k,n)-DVCS can be designed using two Boolean $n \times m$ basis matrices, B_1 and B_0, with elements "1" and "0" respectively denoting black and white subpixels. The whiteness (the number of white subpixels in a m-subpixel block) is used to distinguish white from black colors, i.e., "$m-l$"B "l"W subpixels represent the black color and "$m-h$"B "h"W subpixels represent the white color, where $h > l$.

When sharing a black secret pixel, one matrix is randomly chosen from set C_1 which includes all matrices obtained by fully permuting the columns in B_1. For white secret pixels, the set is chosen from C_0 which contains the matrices arising from permuting the columns in B_0. Then, we select a row from the chosen matrix. Let $\mathrm{OR}(C_i|r)$, i=0 and 1, denote the "OR"-ed vector of any r rows of a matrix in C_i, and $H(\cdot)$ be the Hamming weight function. Then, the base matrices of the (k,n)-DVCS satisfy the following conditions:

$$\begin{cases} H(\mathrm{OR}(B_1|r)) \geqslant (m-l) \\ H(\mathrm{OR}(B_0|r)) \leqslant (m-h) \end{cases}, \quad \text{for} \quad r = k \tag{1}$$

where $0 \leqslant l \leqslant h \leqslant m$.

$$H(\mathrm{OR}(C_1|r)) = H(\mathrm{OR}(C_0|r)), \quad \text{for} \quad r \leqslant (k-1) \tag{2}$$

The first condition is often referred to as the contrast condition, and the secret image can be recognized due to the contrast of black and white colors. The second condition is the security condition that assures the (k,n)-DVCS of unconditional security. For DVCS, every m-pixel block in the black color or white color has the same number of black/white subpixels.

Example 1. Construct (2, 2)-DVCS with $m = 2, h = 1 and l = 0$ by using $B_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $B_0 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$. By permuting the columns in B_1 and B_0, we have $C_1 = \left\{ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \right\}$ and $C_0 = \left\{ \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \right\}$. Therefore, $\mathrm{OR}(C_1|2) = (1,1)$, and $\mathrm{OR}(C_0|2) = (1,0)$ or $(0,1)$. So, $H(\mathrm{OR}(C_1|2)) = 2$ and $H(\mathrm{OR}(C_0|2)) = 1$, i.e., the black color is 2B (■■) and the white color is 1B1W (■□) or 1W1B (□■) in the reconstructed image. On the other hand, $\mathrm{OR}(C_1|1) = (1,0)$ or $(0,1)$, and $\mathrm{OR}(C_0|1) = (1,0)$ or $(0,1)$. So, $H(\mathrm{OR}(C_1|1)) = H(\mathrm{OR}(C_0|1)) = 1$, i.e., every share contains 1B1W (■□) or 1W1B (□■) since $H(\mathrm{OR}(B_1|1)) = H(\mathrm{OR}(B_0|1)) = 1$. Finally, one cannot see anything from a single share but can visually decode the secret when stacking both shares.

2.2. PVCS

By choosing every column vector in B_1 and B_0, we construct the black and white collections C_1' and C_0', where there are m n-bit column matrices. When sharing a black pixel, a column matrix is randomly chosen from C_1' and then the element of this column vector corresponding to the relative share is used. Conversely for white pixels, the column is chosen from C_0'. Let $\mathrm{OR}(C_1'|r)$, i = 0 and 1, denote a collection including the results of "OR"-ed r rows in all column matrices, and $P(\cdot)$ be the appearance probability of the "0" (whiteness) in a collection. Then a (k,n)-PVCS should satisfy the following two conditions:

$$\begin{cases} P(\mathrm{OR}(C_1'|r)) \leqslant (p_t - \alpha) \\ P(\mathrm{OR}(C_0'|r)) \geqslant p_t \end{cases}, \quad \text{for } r = k \tag{3}$$

where p_t is a threshold probability and α is a relative difference.

$$P(\mathrm{OR}(C_1'|r)) = P(\mathrm{OR}(C_0'|r)) \quad \text{for } r \leqslant (k-1) \tag{4}$$

Eqs. (3) and (4) are similar to Eqs. (1) and (2), but they are probabilistic. From (2) and (4), the Hamming weight in m subpixels (DVCS) and the probability of whiteness in black and white colors is the same for stacking less than k shares. A secret image can be successfully recognized through their different probabilities of "whiteness" in the reconstructed image. PVCS has no pixel expansion, i.e., the share and secret image are the same size, while the share size of DVCS is m times that of PVCS.

Example 2. Construct (2,2)-PVCS using B_1 and B_0 in Example 1. From B_1 and B_0, we have $C_1' = \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}$ and $C_0' = \left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\}$. Therefore, $\mathrm{OR}(C_1'|2) = (1)$, and $\mathrm{OR}(C_0'|2) = (1)$ or (0). So, $P(\mathrm{OR}(C_1'|2)) = 1$ and $P(\mathrm{OR}(C_0'|2)) = 0.5$ satisfy the contrast condition where $p_t = 0.5$ and $\alpha = 0.5$. The black secret pixel in the stacked result is 1B (■) with 100% probability, and the white secret pixel in the stacked result is 1B (■) with 50% probability and 1 W (□) with 50% probability. On the other hand, $\mathrm{OR}(C_1'|1) = \mathrm{OR}(C_0'|1) = (1)$ or (0). Therefore $P(\mathrm{OR}(C_1'|1)) = P(\mathrm{OR}(C_0'|1)) = 0.5$ satisfies the security condition.

Table 1 summarizes the representation of the secret pixel for (2,2)-DVCS and (2,2)-PVCS. Suppose we use the printed-letter image LVCS as the secret image. Figs. 1(a) and (b) respectively reveal two shares and the reconstructed image for (2,2)-DVCS and (2,2)-PVCS. Observe the share size and the secret image size for PVCS is the same, while the share size of (2,2)-DVCS is double that of (2,2)-PVCS. Both schemes recover the secret from stacking two shares. However, the visual quality in Fig. 1(f) is blurred compared to Fig. 1(c).

Table 1
Representation of secret pixels for the (2,2)-DVCS and (2,2)-PVCS

Types	Secret pixel	Subpixels in shares	Stacked result	Secret pixel	Subpixels in shares	Stacked result
DVCS	■	Share 1 ■□ or □■ $C_1 = \left\{ \begin{bmatrix}10\\01\end{bmatrix} \text{ or } \begin{bmatrix}01\\10\end{bmatrix} \right\}$ □■ or ■□ Share 2	■■	□	Share 1 ■□ or □■ $C_0 = \left\{ \begin{bmatrix}10\\10\end{bmatrix} \text{ or } \begin{bmatrix}01\\01\end{bmatrix} \right\}$ ■□ or □■ Share 2	■□ or □■
PVCS	■	Share 1 ■ or □ $C'_1 = \left\{ \begin{bmatrix}1\\0\end{bmatrix} \text{ or } \begin{bmatrix}0\\1\end{bmatrix} \right\}$ □ or ■ Share 2	■	□	Share 1 ■ or □ $C'_0 = \left\{ \begin{bmatrix}1\\1\end{bmatrix} \text{ or } \begin{bmatrix}0\\0\end{bmatrix} \right\}$ ■ or □ Share 2	■ or □

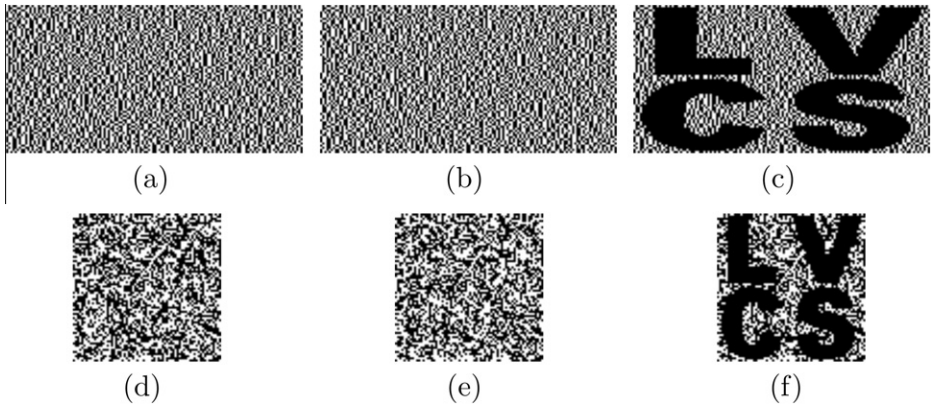

Fig. 1. Shares and stacked results of DVCS: (a) share 1, (b) share 2, (c) stacked results; and PVCS: (d) share 1, (e) share 2, (f) stacked results.

3. Motivation and design concepts

Takizawa et al. proposed two methods of sharing secrets using natural language (Japanese letters). They employed a database, which stored meaningful phrases and a morphological analyzer to design shares in the construction phase. The first method depends on the position of letters. Using their database and morphological analyzer, secret messages complied by specific characters could be placed into specific positions and construct meaningful sentences in combination with other characters. Those secret message characters would not be overlapped when the shares are stacked. For reconstructing the secret messages, participators only stack enough shares and read the unstacked characters. Their second method also depends on the position of letters, but each shares contains a secret letter. For the secret encoding phase, they separate the secret message into different shares with one character of the secret and compose a meaningful sentence using each secret letter with other words. For reconstructing the secret message, participators need to line up all the shares vertically in order, and the secret is revealed as a particular horizontal line. Their two methods both cover the secret in meaningful sentences, but their methods are not secure and limited. We point out the drawbacks of this approach in next section.

3.1. Motivation

In Takizawa et al.'s first scheme [29], the more stacked shares, the higher the possibility of leaking some secret information since the number of non-overlapped characters decreases when more shares are stacked. Fig. 2 (as Fig. 1 shown in [29]) shows the reconstructed secret by the first scheme is "the meeting place is front of Morioka Station", which is not overlapped by other Japanese letters. The secret may be guessed from the non-overlapped letters with less than the necessary shares. This violates the property of thresholds, since it is possible to guess or derive the secret from less than the complete number of shares. In addition, the secret may contain errors when lacking some important (key) shares and the wrong message may be reconstructed. The shares in this method do not possess the characteristic of generality which denotes that any two shares are equivalent for the purposes of decoding since order plays a role in this method. For scheme 2 [29,30] illustrated as Fig. 3 (as Fig. 2 shown in [29]), all shares need to be lined up according to a specific sequence, and secret reconstruction fails when lacking important shares. With more shares lined up, more of the secret is disclosed. It may be possible to guess the secret from the interim state even without all the shares. The second method also does not adhere to property of thresholds and its shares also do not possess generality. Therefore, Takizawa et al.'s two schemes fail to uphold the threshold property of secret sharing.

Fig. 2. Takizawa et al.'s first scheme in [28].

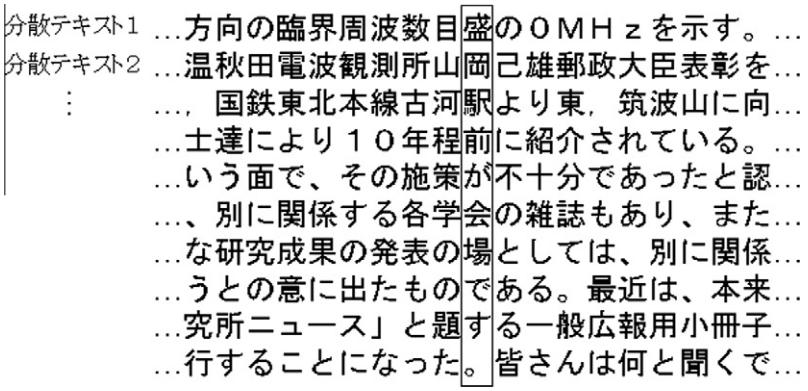

Fig. 3. Takizawa et al.'s second scheme in [28].

Table 2
Average number of pixels $N_i, 1 \leqslant i \leqslant 4$, for i overlapped letters.

Overlapped letters	Combinations	Average pixels
1	26	15.58
2	325	24.69
3	2600	30.53
4	14,950	34.60

Table 3
Six instances for simulation.

Instance	(k,n)-LVCS	Approach	Encoding manner	Enhanced scheme	Results
I	(2,2)	Using basis	DVCS	No	Fig. 4
II	(2,2)	Matrices in	PVCS	No	Fig. 5
III	(2,2)	Example 1	DVCS	Yes	Fig. 6
IV	(2,2)	And 2	PVCS	Yes	Fig. 7
V	(2,3)	Using basis	DVCS	Yes	Fig. 8
VI	(2,3)	Matrices in	PVCS	Yes	Fig. 9
VII	(3,3)	Section	PVCS	No	Fig. 10
VIII	(3,4)	5.1.1	PVCS	No	Fig. 11

In this paper, we attempt to address the issues in Takizawa's schemes. We use natural language letters instead of black and white pixels to design LVCS so shares may have independent meaning, and an adversary does not recognize them as shares. In our LVCS, we retain the nature of conventional VCS using whiteness to distinguish black and white. In addition, we prove that the proposed (k,n)-LVCS achieves stacking security, where one cannot gain any information with less than k shares.

3.2. Design concept

The following example illustrates that superimposing more letters gains more darkness. All 26 capital letters are shown in Appendix A.1 using 10 point Arial typeface within a 9×9-pixel square. The four letters L, V, C, and S have 11, 13, 13, and 15 pixels, respectively. Overlapped letters have 20 pixels (L+V), 16 pixels (L+C), 20 pixels (L+S), 21 pixels (V+C), 24 pixels (V+S), and 17 pixels (C+S). Three overlapped letters have 24 pixels (L+V+C), 28 pixels (L+V+S), 20 pixels (L+C+S), and 25 pixels (V+C+S). All four overlapped letters have 28 pixels.

For the set of 26 English letters, there are a total 325 combinations (from (A, B) to (Y, Z)) for two overlapping letters, and the average number of pixels for two stacked letters is 24.69. Table 2 shows the average numbers of pixels, N_i where $1 \leqslant i \leqslant 4$, for stacking i letters. The values $N_1 = 15.58$, $N_2 = 24.69$, $N_3 = 30.53$, and $N_4 = 34.60$ confirm the intuitive notion of more darkness when

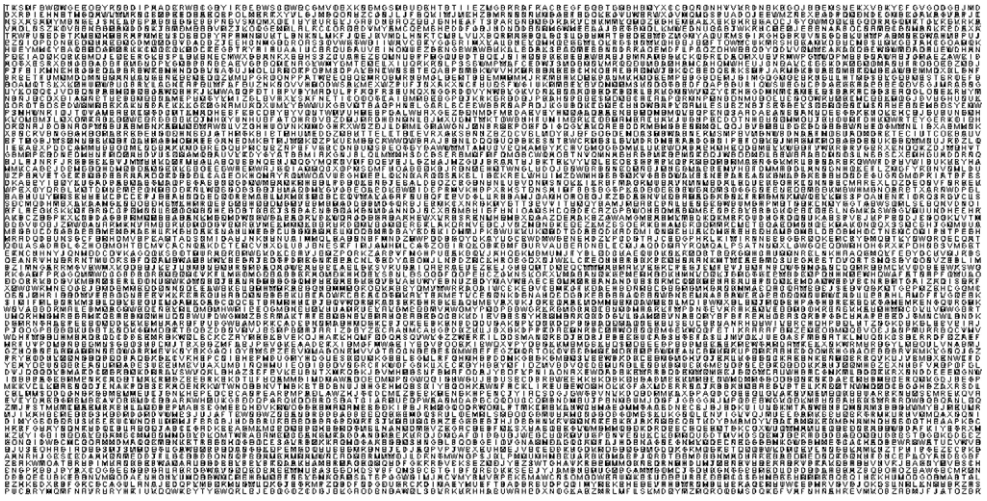

Fig. 4. (2,2)-LVCS by Algorithm 1: Share 1 + Share 2. (CR : 6.8843, BD : 0.4492).

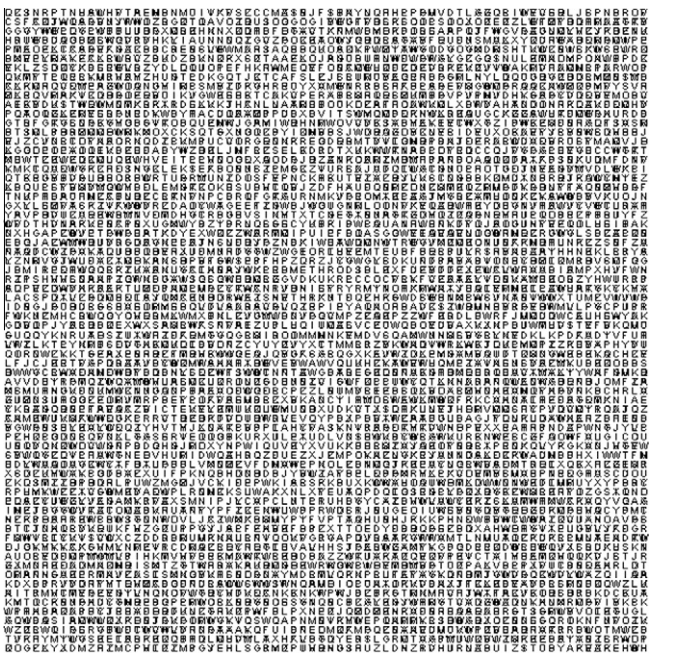

Fig. 5. (2,2)-LVCS by Algorithm 2: Share 1 + Share 2. (CR : 8.4304, BD : 0.4465).

stacking more letters. In this paper, we use overlapped and non-overlapped letters in the reconstructed image to represent black and white colors, respectively, due to the property $N_i > N_1$ for $i \geqslant 2$.

4. The proposed LVCS

The proposed (k,n)-LVCS can be implemented with either (k,n)-DVCS or (k,n)-PVCS. We first give the formal contrast and security conditions of the proposed (k,n)-LVCS. Let OR $(1|r)$ and OR $(0|r)$ be m-tuple letters, and denote the black and white colors when stacking any r shares in our LVCS. Let $N(\cdot)$ be the number of overlapped letters (two or more) in every m letters. Similar to Eqs. (1) and (2), the contrast and security conditions of (k,n)-LVCS when using (k,n)-DVCS are defined as follows:

$$\begin{cases} N(\mathrm{OR}(1|r)) \geqslant (m-l) \\ N(\mathrm{OR}(0|r)) \leqslant (m-h) \end{cases}, \quad \text{for } r = k \tag{5}$$

where $0 \leqslant l \leqslant h \leqslant m$.

$$N(\mathrm{OR}(1|r)) = N(\mathrm{OR}(0|r)) \quad \text{for } r \leqslant (k-1) \tag{6}$$

Let $A(\cdot)$ be the appearance probability of the non-overlapped letters. Similar to Eqs. (3) and (4), the contrast condition and the security condition of (k,n)-LVCS when using (k,n)-PVCS are defined as follows:

$$\begin{cases} A(\mathrm{OR}(1|r)) \leqslant (p_t - \alpha) \\ A(\mathrm{OR}(0|r)) \geqslant p_t \end{cases}, \quad \text{for } r = k \tag{7}$$

where p_t is a threshold probability and α is a relative difference.

$$A(\mathrm{OR}(1|r)) = A(\mathrm{OR}(0|r)) \quad \text{for} \quad r \leqslant (k-1) \tag{8}$$

Both conditions (6) and (8) assure (k,n)-LVCS of security. In (6) and (8), there are overlapped letters every m letters. In (8), we have the same appearance probability of non-overlapped letters. Therefore, no secret information can be revealed from $(k-1)$ or fewer shares. Above the threshold, we have more overlapped letters in the black area than the white. Since $N_i > N_1$ for $i \geqslant 2$, we can reveal the secret by HVS.

We propose two encoding methods of (k,n)-LVCS. The first is a general (k,n)-LVCS for any k and n where $k \leqslant n$, and the second is the enhanced $(2,n)$-LVCS where $2 \leqslant n$.

4.1. The (k,n)-LVCS

We show how to construct (k,n)-LVCS from (k,n)-DVCS and (k,n)-PVCS. We define an operation $T(\cdot)$, which transforms $n \times m$ base matrices B_0 and B_1 in (k,n)-DVCS into two $n \times m$ letter-based matrices L_0 and L_1. Suppose the elements in B_0 and B_1 are $B_0[i,j]$ and $B_1[i,j]$, and the elements in L_0 and L_1 are $L_0[i,j]$ and $L_1[i,j], 1 \leqslant i \leqslant n$ and $1 \leqslant j \leqslant m$. Then, $L_0 = T(B_0)$ and $L_1 = T(B_1)$, where

$$\begin{cases} L_0[i,j] = return\_letter(B_0[i,j]) \\ L_1[i,j] = return\_letter(B_1[i,j]) \end{cases} \tag{9}$$

The $return\_letter(b_i)$ function returns a letter, where b_i is the ith row element of a column in B_0 or B_1. The returned letter is generated according to the following rules:

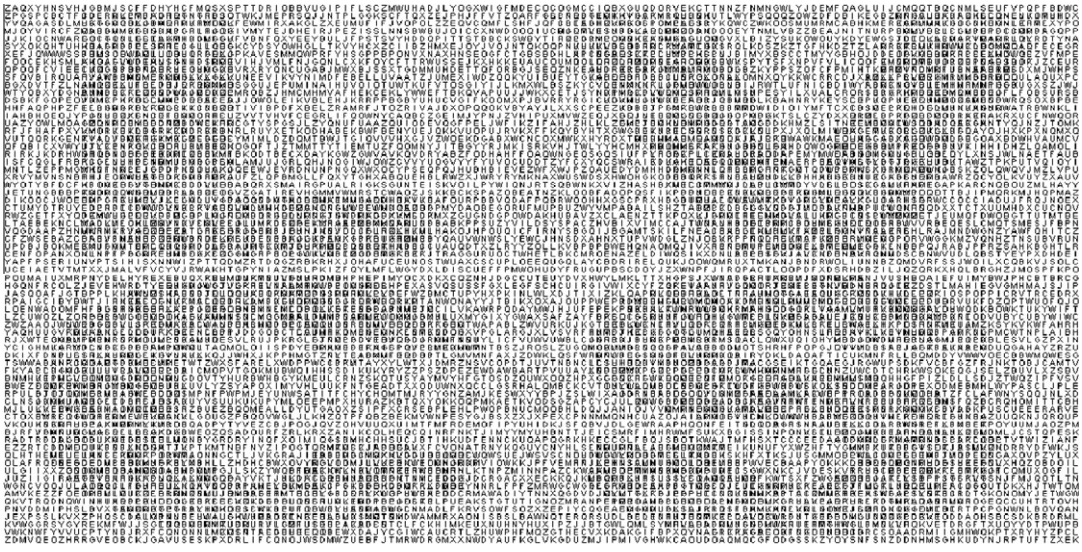

Fig. 6. The enhanced (2,2)-LVCS by Algorithm 1: Share 1 + Share 2. (CR : 8.8986, BD : 0.4237).

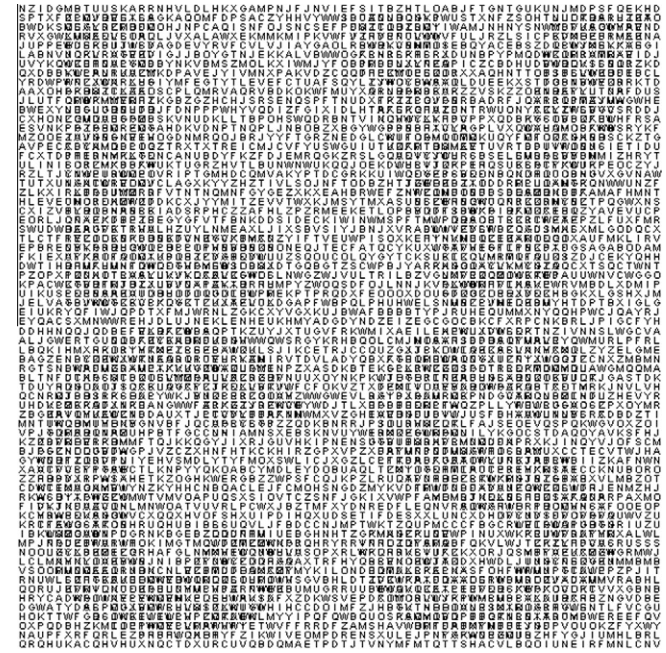

Fig. 7. The enhanced (2,2)-LVCS by Algorithm 2: Share 1 + Share 2. (CR : 9.8970, BD : 0.4203).

1. *return_letter*(0) returns a letter same as generated at element 0 in this column
2. *return_letter*(1) returns a letter different from all previous letters in this column
3. for initialization, the function *return_letter*(·) returns any letter

We outline two encoding methods based on (k,n)-DVCS and (k,n)-PVCS in Algorithms 1 and 2, respectively.

Algorithm 1. The proposed (k,n)-LVCS using (k,n)-DVCS

Input: A $(W \times H)$ binary secret image I; two $n \times m$ matrices B_0 and B_1 in (k,n)-DVCS
Output: n random letter-based shares $S_i, i \in [1,n]$, with size $(m \times W \times H)$.
1: $L_0 = T(B_0)$ and $L_1 = T(B_1)$;
2: /* convert the Boolean matrix to letter-based matrix */
3: Construct the sets T_0 and T_1 from L_0 and L_1;
4: /* the set T_0 (respectively T_1) includes all matrices obtained by permuting the columns in L_0 (respectively L_1) */
5: **for** all $W \times H$ elements in I **do**
6: **if** the element in I = 0
7: **then** Randomly select one matrix from T_0;
8: put the ith row (m letters) in the selected matrix to $S_i, i \in [1,n]$;
9: **else**
10: Randomly select one matrix from T_1;
11: put the ith row (m letters) in the selected matrix to $S_i, i \in [1,n]$;
12: **end for**
13: Output $(S_1, S_2, \ldots, S_n)$.

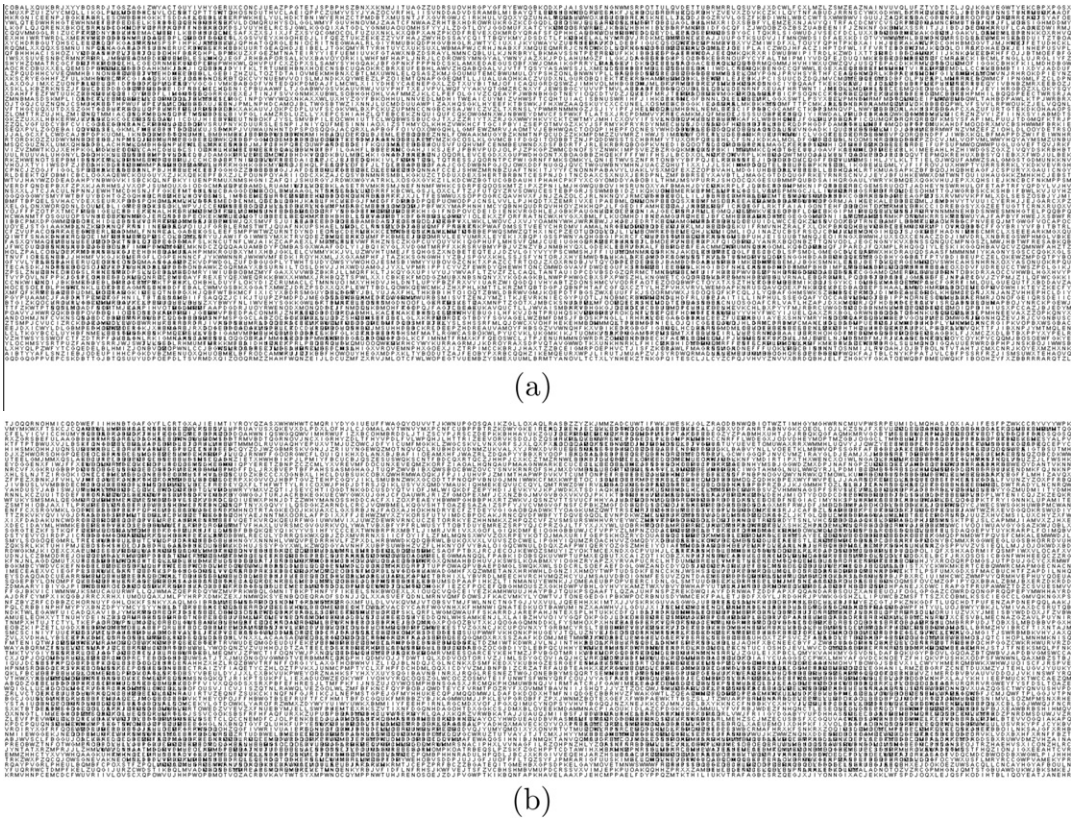

Fig. 8. The enhanced (2,3)-LVCS by Algorithm 1: (a) stack any two shares (CR : 10.0280, BD : 0.4566), (b) stack all three shares (CR : 10.5609, BD : 0.4528).

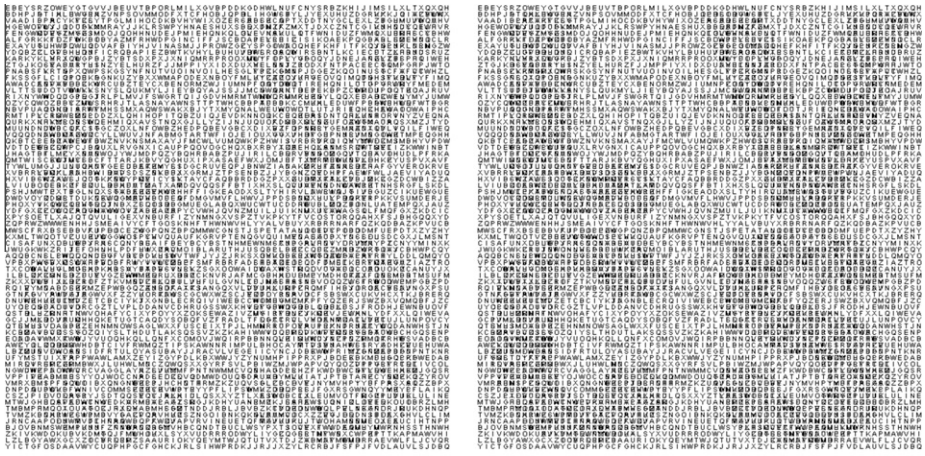

Fig. 9. The enhanced (2,3)-LVCS by Algorithm 2: (a) stack any two shares ($CR: 8.0259, BD: 0.4351$), (b) stack all three shares ($CR: 9.5421, BD: 0.4183$).

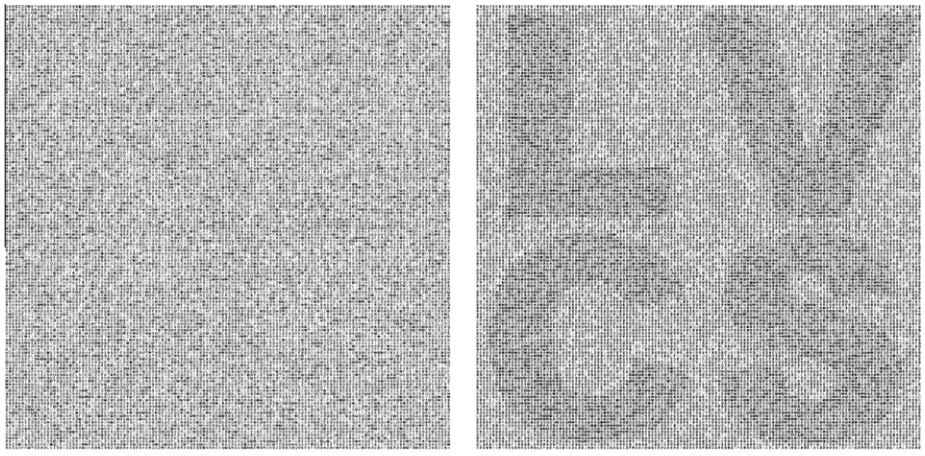

Fig. 10. The (3,3)-LVCS by Algorithm 2: (a) stack any two shares, (b) stack all three shares ($CR: 9.0809, BD: 0.4383$).

Algorithm 2. The proposed (k,n)-LVCS using (k,n)-PVCS

Input: A $(W \times H)$ binary secret image I; two $n \times m$ matrices B_0 and B_1 in (k,n)-PVCS

Output: n random letter-based shares $S_i, i \in [1,n]$, with size $(m \times W \times H)$.

1: $L_0 = T(B_0)$ and $L_1 = T(B_1)$;
2: /* convert the Boolean matrix to letter-based matrix */
3: Construct the sets T_0 and T_1 from L_0 and L_1;
4: /* the set T_0 (respectively T_1) includes all matrices obtained by permuting the columns in L_0 (respectively L_1)*/
5: **for** all $W \times H$ elements in I**do**
6: **if** the element in $I = 0$
7: **then** Randomly select one column $(L_0[1,t], L_0[2,t], \ldots, L_0[n,t]$;
8: $1 \leqslant t \leqslant m$) from $T_0[i,j]$; put $L_0[i,t]$ (one letter) to $S_i, i \in [1,n]$;
9: **else**
10: Randomly select one column $(L_1[1,t], L_1[2,t], \ldots, L_1[n,t]$;
11: $1 \leqslant t \leqslant m$) from $T_1[i,j]$; put $L_1[i,t]$ (one letter) to $S_i, i \in [1,n]$;
12: **end for**
13: Output $(S_1, S_2, \ldots, S_n)$.

Theorem 1. *The (k,n)-LVCSs using Algorithms 1 and 2 satisfy the contrast and security conditions.*

Proof. We first prove (k,n)-LVCS with Algorithm 1 satisfies contrast condition (Eq. 5) and security condition (Eq. 6). If the stacked result of m-tuples OR $(C_i|r)$ is 1, it implies that there is at least one "1" in this column. Due to the function return_letter (b_i), there are at least two different characters overlapped in the column of L_i. On the other hand, if the stacked result of OR $(C_i|r)$ is 0, it implies that this is the all-0 column, so the column of L_i will have the same character. From the above description, the stacked results "1" and "0" in OR $(C_i|r)$ denote the overlapped characters and non-overlapped characters in OR $(i|r)$, respectively. So, we have $H(\mathrm{OR}(B_i|r)) = N(\mathrm{OR}(i|r)), i = 0, 1$. Therefore, from Eqs. (1) and (2), we can derive Eqs. (5) and (6). For Algorithm 2, by a similar argument, we have $P(\mathrm{OR}(C_i'|1)) = A(\mathrm{OR}(i|r)),\ i = 0, 1$. Therefore, we can derive Eqs. (7) and (8) from Eqs. (3) and (4). □

Example 3. Construct the (2,2)-LVCS by Algorithms 1 and 2 using B_1 and B_0 in Example 1. The letter set A, B, ..., Z is chosen from A.2. Possible L_1 and L_0 by Algorithm 1 are shown as: $L_1 = \begin{bmatrix} AB \\ EC \end{bmatrix}$ and $L_0 = \begin{bmatrix} DL \\ YL \end{bmatrix}$. When stacking two shares, in the black secret pixel,

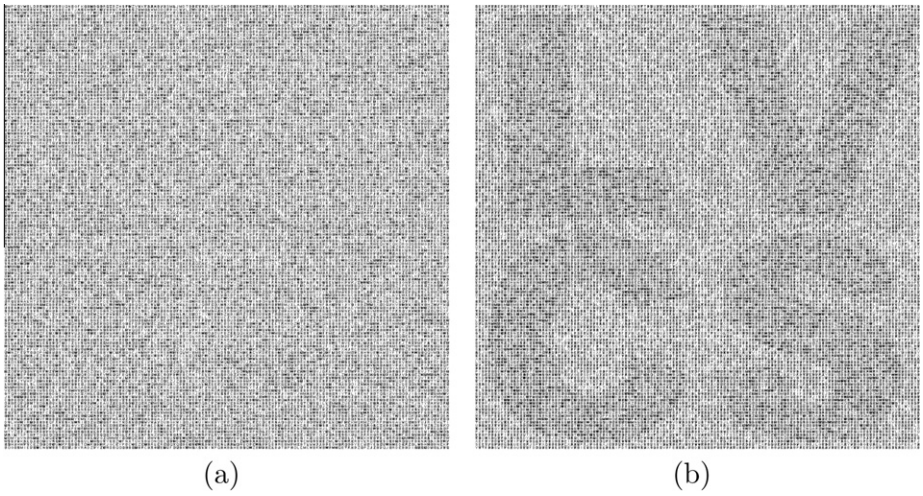

Fig. 11. The (3,4)-LVCS by Algorithm 2: (a) stack any two shares, (b) stack any three shares. (CR : 9.4126, BD : 0.4424).

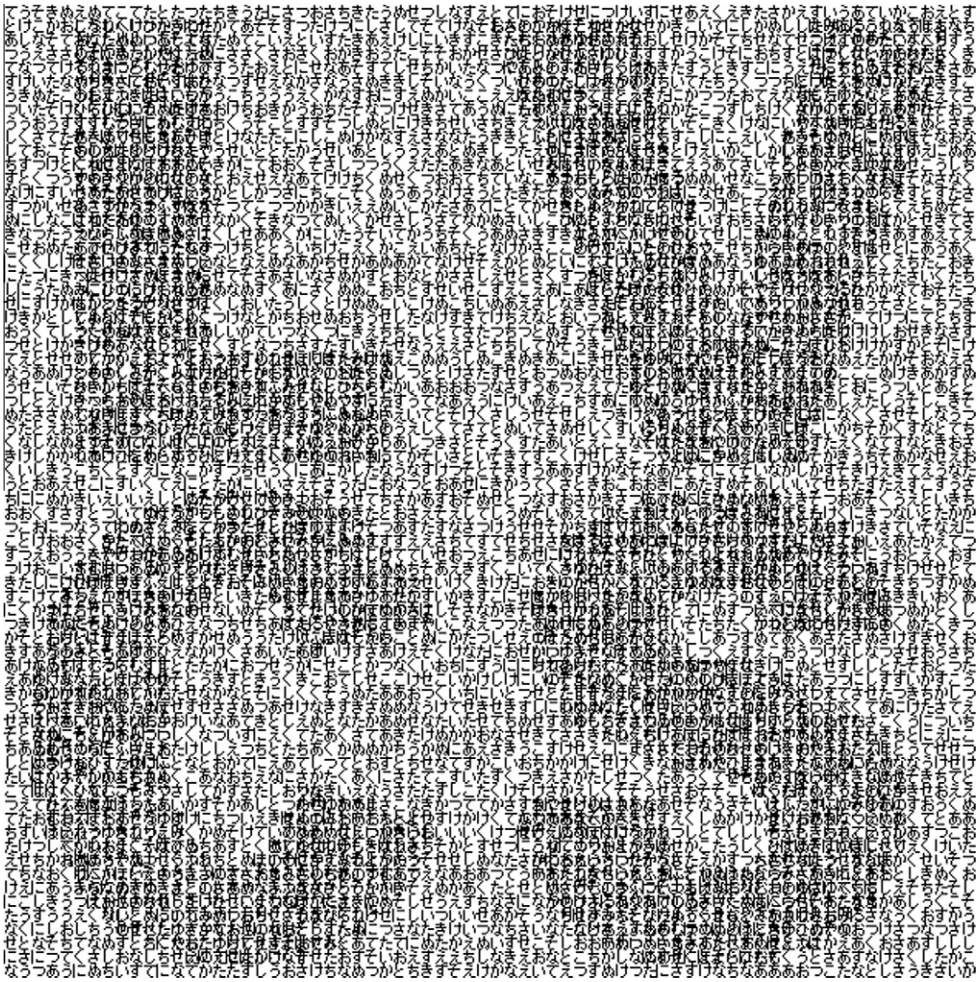

Fig. 12. The enhanced (2,2)-LVCS using Algorithm 2 for Hiragana (CR : 15.9676, BD : 0.3870)

we have two overlapped letters $\boxed{\text{A+E}}$ and $\boxed{\text{B+C}}$. There is one non-overlapped letter $\boxed{\text{L+L}}$ and one overlapped letter $\boxed{\text{D+Y}}$ in the white secret pixel. Since the overlapped letter is darker than the single non-overlapped letter, we can regard the overlapped letter and the non-overlapped letter as black and white subpixels. We then have 2 overlapped letters and 0 non-overlapped letters (denoted as [2\0]) to represent a black secret pixel. On the other hand, we have [1\1] to represent a white secret pixel. Our LVCS

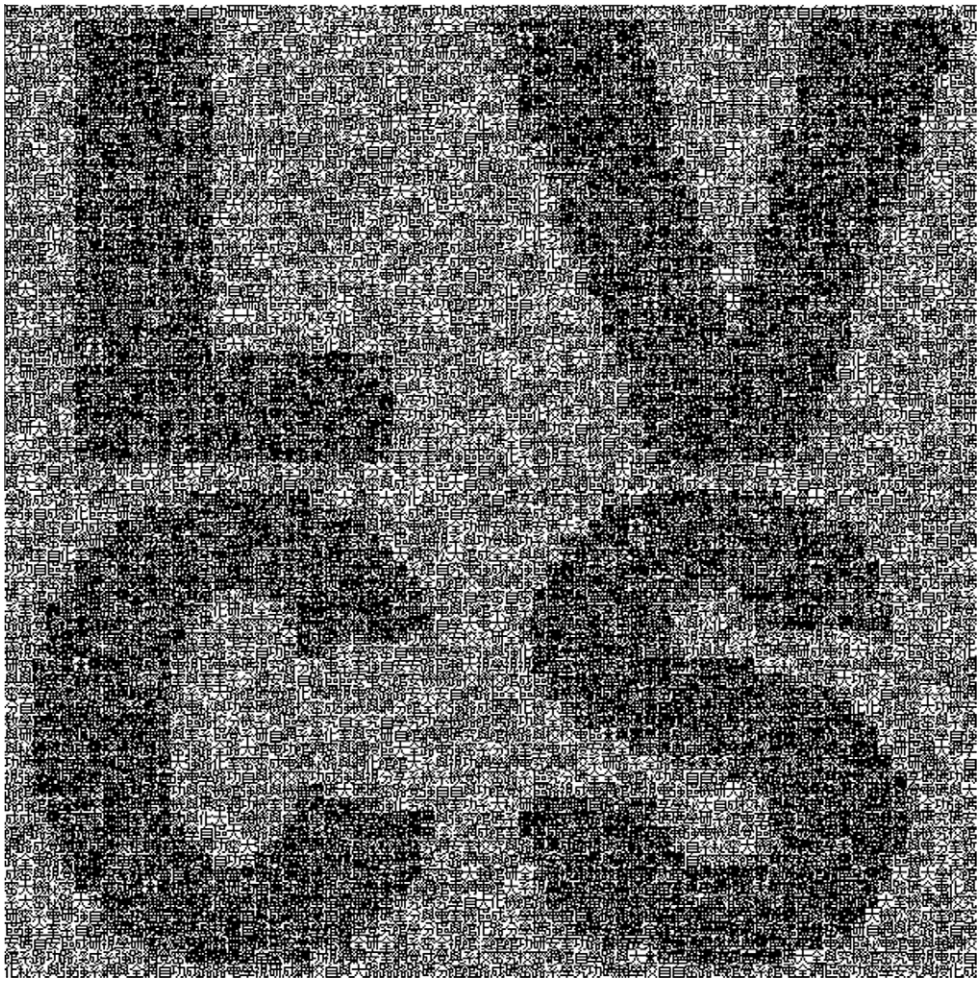

Fig. 13. The enhanced (2,2)-LVCS using Algorithm 2 for Chinese (CR : 19.8819, BD : 0.3928)

by Algorithm 2 has overlapped letters with 100% probability in black color. In this example ($L_1 = \begin{bmatrix} AB \\ EC \end{bmatrix}$), so we have one overlapped letter $\boxed{\text{A+E}}$ or $\boxed{\text{B+C}}$. However, there is 50% probability showing the non-overlapped in white color (one overlapped letter $\boxed{\text{D+Y}}$ or one non-overlapped letter $\boxed{\text{L+L}}$).

4.2. The enhanced $(2, n)$-LVCS

We can further enhance the contrast of the reconstructed image for $k = 2$ in (k, n)-LVCS. We add a new rule for the *return_letter*(b_i) function. When applying $L_0 = T(B_0)$, *return_letter*(b_i) returns the same letter for columns with all ones. By using this new function $T(\cdot)$ in Algorithms 1 and 2, we can construct the enhanced $(2, n)$-LVCSs.

Theorem 2. *The enhanced $(2, n)$-LVCSs by Algorithms 1 and 2 satisfy the contrast and security conditions.*

Proof. We first prove that the $(2, n)$-LVCSs by Algorithm 1 satisfies conditions (Eq. 5) and (Eq. 6). In the enhanced $(2, n)$-LVCS, a column of B_0 with all 1's returns the same character in a column of L_0. Due to this modification, we have $N(\text{OR}(0|r)) \leqslant H(\text{OR}(C_0|r))$ for $r = 2$, and $N(\text{OR}(0|r)) = H(\text{OR}(C_0|r))$ for $r = 1$. Note, we still have $N(\text{OR}(1|r)) = H(\text{OR}(C_1|r))$. Since $H(\text{OR}(B_0|2)) \leqslant (m - h)$, it implies $N(\text{OR}(0|2)) \leqslant (m - h)$. Also, by $N(\text{OR}(0|1)) = H(\text{OR}(C_0|1))$, $H(\text{OR}(C_1|1)) = H(\text{OR}(C_0|1))$, and $N(\text{OR}(1|1)) = H(\text{OR}(C_1|1))$, we have $N(\text{OR}(0|1)) = N(\text{OR}(1|1))$. So, the enhanced $(2, n)$-LVCS satisfies Eq. (5) and (6). By a similar argument, we can prove the (k, n)-LVCS using Algorithm 2 satisfies Eq. (7) and (8). □

Example 4. Construct the enhanced (2,2)-LVCS by Algorithms 1 and 2 using B_1 and B_0 in Example 1. Using the same letter for all-1 columns, two possible L_1 and L_0 by Algorithm 1 are shown below: $L_1 = \begin{bmatrix} AB \\ EC \end{bmatrix}$ and $L_0 = \begin{bmatrix} DL \\ DL \end{bmatrix}$. (note: L_0 is different from Example 3). We now have [2\0] and [0\2] for the black and white secret pixels in the reconstructed image. When randomly choosing one column in L_1 and L_0 to implement the enhanced (2,2)-LVCS by Algorithm 2, we have non-overlap with 100% probability in white color. Therefore, the contrast is enhanced.

5. Experiment and application

Our LVCS example uses the English alphabet. Because we use overlapped and non-overlapped characters to represent the black and white pixels, our scheme can also be arbitrarily implemented in different languages.

5.1. Evaluating indexes

In our experiments, we apply two indexes, Contrast ratio (CR) and baseline difference (BD), to evaluate the quality of reconstructed images. CR represents the average of the number of different pixels between two differently colored neighboring pixels, which is also visually represented as the edge of the secret message pattern. We use CR to compute the difference between letters representing each pixel when the color of pixel changes (i.e. black to white or vice versa). The calculated directions are left to right and top to down. Letter pixels and the number of change points

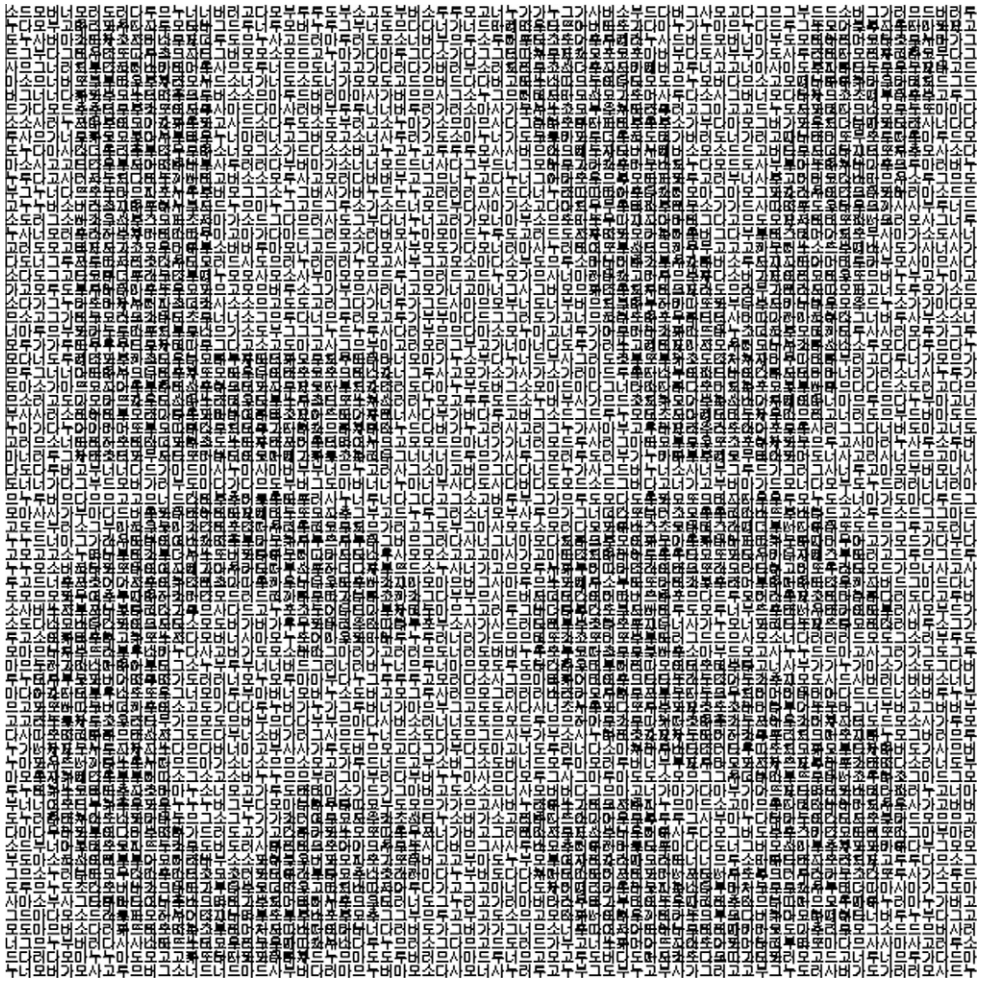

Fig. 14. The enhanced (2,2)-LVCS using Algorithm 2 for Hangul (CR : 12.9142, BD : 0.4100)

are denoted as PL and cpt. The pixel expansion in deterministic VCS is denoted as pn. The function of CR is as follows:

$$CR = \left[\sum_{i=1,j=1}^{m-1,n-1} |PL_{(i,j)} - PL_{(i,j+1)}| + \sum_{i=1,j=1}^{m-1,n-1} |PL_{(i,j)} - PL_{(i+1,j)}|\right] \Big/ (cpt \cdot pn) \tag{10}$$

Where i and j are the position in secret message image, and m and n are the dimensions of the images.

In LVCS, pixels are represented by letters. The baseline pixel is either all black or white depending on the secret image. Another evaluation index is baseline difference (BD), which means the number of different pixels between the letter and an all black or all white block divided by the dimension of the reconstructed image. The function of BD is as follows:

$$BD = \sum_{i=1,j=1}^{m,n} (Letter_{(i,j)} \oplus Block_{(i,j)})| \Big/ (m \times n) \tag{11}$$

CR represents the average contrast of image, and therefore higher values are better. On the contrary, BD means the difference between the letter and the baseline block so the smaller the value of ER the better.

5.1.1. LVCS using english characters

Eight instances are used to evaluate the performance of our approach: I to VI are $(2,n)$-LVCS for $n = 2$ and 3, respectively; VII is a (k,k)-LVCS for $k = 3$; and VIII is an example for $(k,n) = (3,4)$. The enhanced scheme only can be used for $k = 2$, therefore it is used in instances III to VI. All of the instances are shown in Table 3. Instances I to IV are using B_1 and B_0 mentioned in Example 1, and other bases of instances V to VIII are described as follows:

Instances V and VI: Construct the enhanced (2,3)-LVCSs with $B_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ and $B_0 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$.

Instance VII: Construct the (3,3)-LVCS with $B_1 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$ and $B_0 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$.

Instance VIII: Construct the (3,4)-LVCS with $B_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$ and $B_0 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$.

A printed-letter LVCS is used as a secret image. Also, English letters are chosen from A.1 (Arial typeface and font size 10). All instances show that our (k,n)-LVCSs can reveal the secret.

The reconstructed images from stacking two shares for Algorithms 1 and 2 (Instances I and II) are shown in Figs. 4 and 5, respectively. Although the share size of Instance I is doubled that of Instance II, it has a clearer secret. Obviously, Instances III and IV enhance the contrast of Instances I and II, respectively. As shown in Figs. 6 and 7, the enhanced schemes have clearer definition. For easier comprehension, Fig. 7 is enlarged as an example and illustrated as A.1 in Appendix.

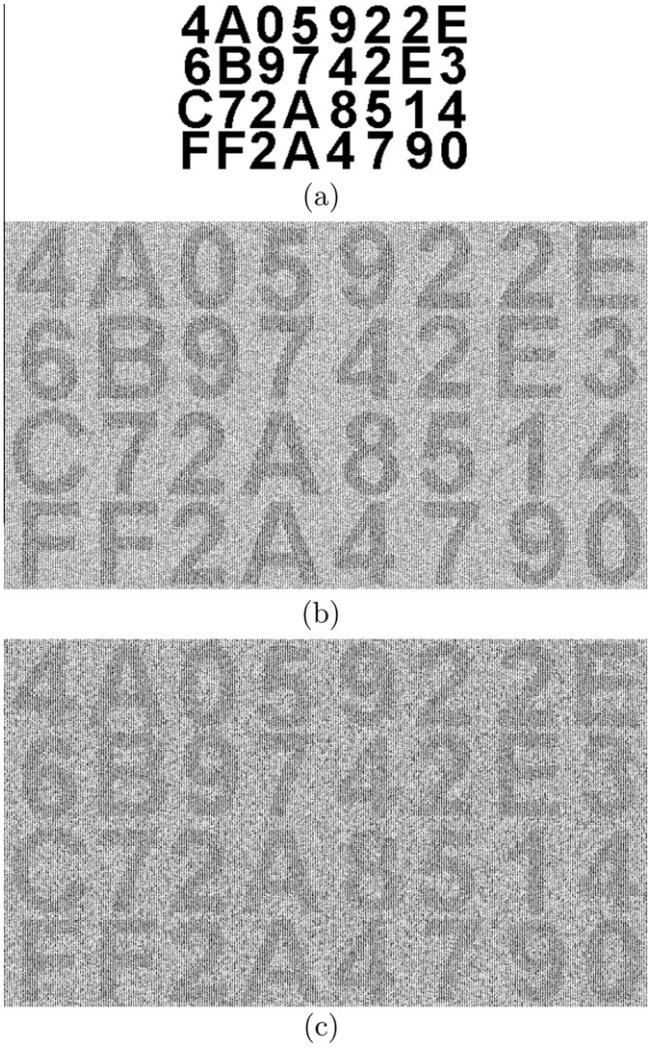

Fig. 15. Error tolerance: (a) the secret, (b) reconstructed in no error (CR : 9.3263, BD : 0.4271), (c) reconstructed in 5% error (CR : 8.4595, BD : 0.4343).

For instance V, Figs. 8(a) and (b) show the reconstructed images from two and three shares. At this time, we have [3\0] and [0\3] for black and white secret pixels when stacking three shares. However, we have [2\1] and [0\3] for black and white secret pixels when stacking two shares. Therefore, Fig. 8(b) is clearer than Fig. 8(a). For Instance VI, Fig. 9 shows the probabilistic characteristic of (2,3)-LVCS, which has small share size and a blurred secret. Instance VII is a (k,k)-LVCS example with $k = 3$. Stacking all three shares reconstructs the secret (Fig. 10(b)) but less than three shares reveals nothing (Fig. 10(a)). Instance VIII is an example for $(k,n) = (3,4)$, where the secret can be reconstructed while stacking any three shares (Fig. 11). From the simulation results of Figs. 4 to 11 and comparing the data of CR and BD, we can make some inferences: CR should be more than 8.4 and BD should be less than 4.4, then the reconstructed image can be recognized by human visualization system.

VCS has poor visual quality of the reconstructed image, which comes from its intrinsic property using the OR-operation for decoding by HVS. As is known, the monotone property of the OR-operation causes such visual effects that a black subpixel in one of the transparencies cannot be undone by the color of another subpixel in other transparencies laid over it. Poor visual quality is more serious in our LVCS due to the replacement of a pixel by a letter. From experiments, our enhanced $(2,n)$-LVCS has acceptable visual quality. Obtaining better visual quality deserves further research.

5.1.2. LVCS using other languages

The LVCS approach is character-set agnostic so any set of characters can be used. In this section, LVCS was implemented in other languages, e.g., Japanese (Hiragana), traditional Chinese, and Korean (Hangul). The Hiragana character set is chosen from Unicode (range: 3040–309F), and there are 46 Hiragana characters selected in our experiments for convenience. For Chinese and Korean, we choose 34 Traditional Chinese characters and 35 Hangul characters. To demonstrate the letters (Japanese, Chinese, and Hangul) in a reconstructed image, we enlarge the secret image and use the enhanced (2,2)-LVCS using Algorithm 2. Figs. 12–14 show the reconstructed images for Hiragana, Chinese, and Hangul, respectively.

5.2. Error tolerance

When applying LVCS in a distributed network to share the secret image, n shares are delivered and stored in various distributed storage nodes. In general, the communication channel is often subject to noise disturbances. A bit error will bring about a wrong pixel in the shares of conventional VCS. However, our LVCS uses overlapped letters as black pixels no matter which letters we use. Even when there is an error, our LVCS has high probability to show the correct color.

In this experiment, we use Radix-64 characters in our (2,2)-LVCS to show the error tolerance capability. A bit error will bring about a wrong pixel in the shares for conventional VCS. In our LVCS, even when there is an error in a character, it still has high probability to show a correct black color. Fig. 15 shows the stacked results from the noise corrupted shares. Fig. 15(a) is the secret image, and Fig. 15(b) is the reconstructed image from two stacked shares with no errors. Fig. 15(c) is the reconstructed images with shares having BER = 5%. It is observed that we still successfully reveal the secret at BER = 5% for each share. Our LVCS allows the shares to tolerate errors and thus it can be used in VCS-based authentication [15,16], where shares are transmitted through network.

5.3. Application: using a code book as a meaningful share

LVCS can be applied for communication or transmitting message privately. For example, (2,2)-LVCS can be used by both sides who want to exchange messages with each other. However, transmitting both shares on-line is insecure and inefficient. For security and convenience, they can secretly tell the other side to choose a specific book as a code book for the shares. Then, both ends can design the second share according to the code book (first share) and secret message. Afterwards, the second share can be sent to the other party for deriving the secret message. To design meaningful shares, we need a database with a lot of meaningful phrases and a morphological analyzer to compose meaningful sentences. The steps of design a meaningful share with (2,2)-LVCS are as follows:

1. Choose a secret image to be encoded.
2. Choose a paragraph and rearrange or typeset it as Share 1 according to the size of secret image.
3. Comparing the secret image and Share 1, copy the same letters from Share 1 to Share 2 according to the position of white pixels in the secret message; find out and blank the specific positions in Share 2 according to the black pixels in secret message.
4. Use a morphological analyzer to analyze the letters in Share 2, and pick up the appropriate phrases or letters from the database to fill the blanks and derive Share 2.

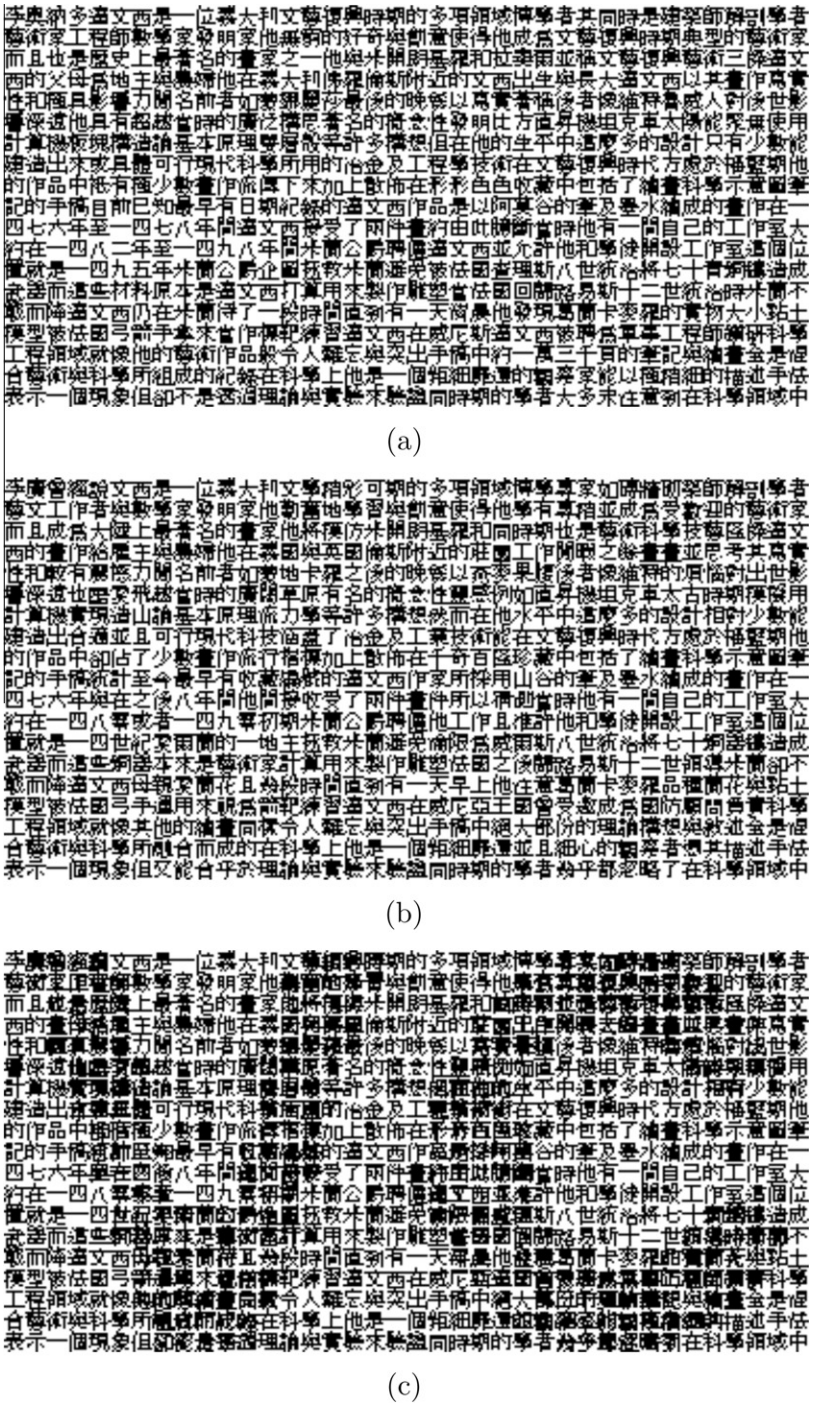

Fig. 16. Error tolerance: (a) Share 1, (b) Share 2, (c) reconstructed image (CR : 23.6392, BD : 0.4052).

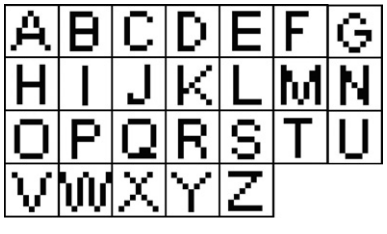

Fig. A.1. 26 English letters.

For illustrating our method, we apply (2,2)-LVCS with Chinese letters and use "VC" as the secret image. The partial biography of Leonardo daVinci was extracted from the Chinese version of the wikipedia website as Share 1. According to the position of the white pixels in secret image, relative letters are copied to Share 2. After analyzing the blank space, meaningful phrases or letters are chosen from the database to represent the black pixels in Share 2. Finally, Share 2 is constructed as another meaningful paragraph. Our example is illustrated as Fig. 16.

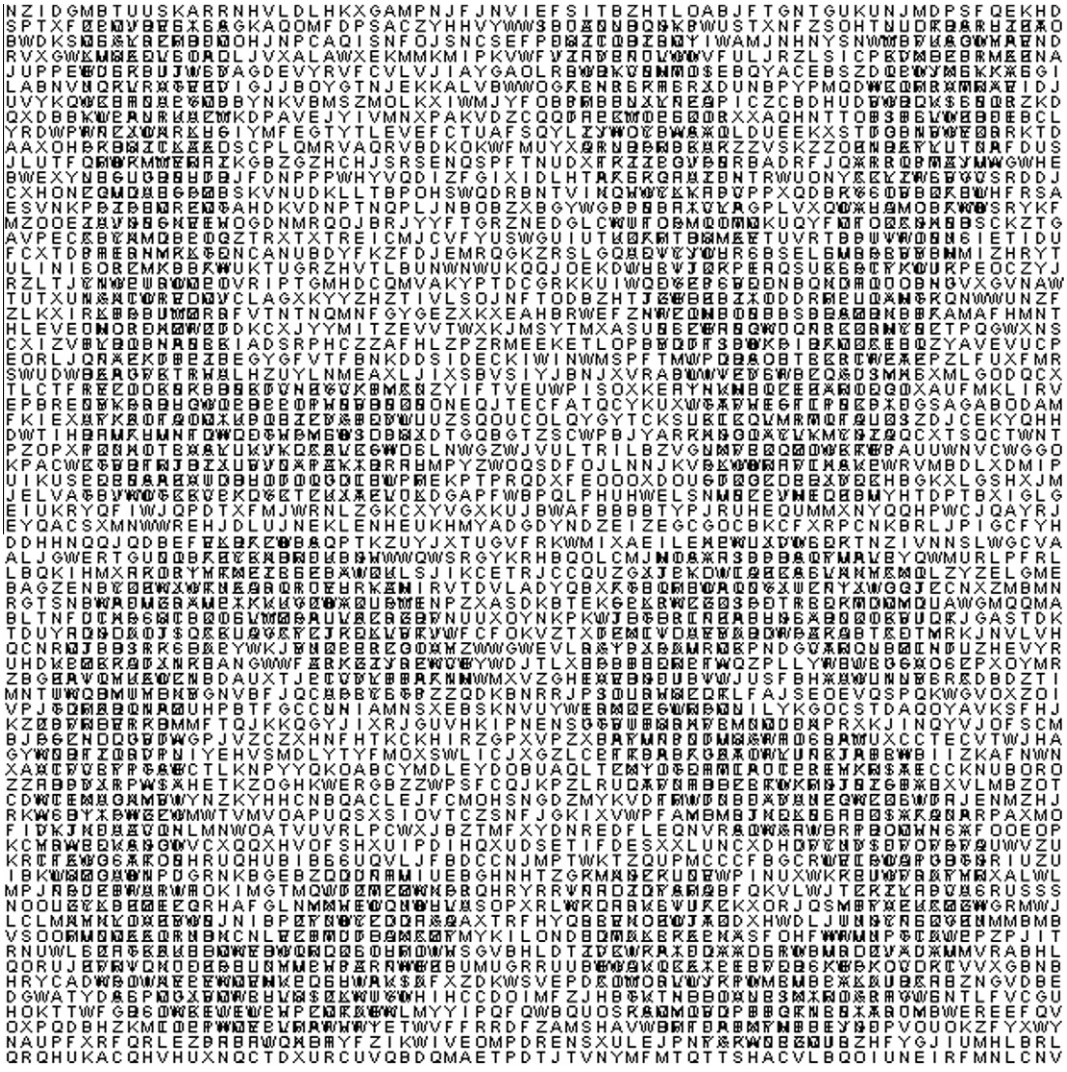

Fig. A.2. Enlargement of Fig. 7.

6. Conclusion

In this paper, we propose (k, n)-LVCS using overlapped and non-overlapped characters instead of black and white pixels. Conventional VCS can be transformed into LVCS by a $T(\cdot)$ operation. In addition, we prove that our proposed LVCS satisfies contrast and security conditions. Our experiments demonstrate that our LVCS scheme is feasible and some possible applications of LVCS are provided. In addition, our LVCS can be extended for other language character sets.

Appendix A

See Figs. A.1 and A.2.

References

[1] C. Thien, J. Lin, Secret image sharing, Computers and Graphics 26 (2002) 765–770.

[2] R. Wang, C. Su, Secret image sharing with smaller shadow images, Pattern Recognition Letters 27 (2006) 551–555.

[3] C. Yang, T. Chen, K. Yu, C. Wang, Improvements of image sharing with steganography and authentication, Journal of Systems and Software 80 (2007) 1070–1076.

[4] C. Chang, Y. Hsieh, C. Lin, Sharing secrets in stego images with authentication, Pattern Recognition 41 (2008) 3130–3137.

[5] C.N. Yang, S.M. Huang, Constructions and properties of k out of n scalable secret image sharing, Optics Communications 283 (2010) 1750–1762.

[6] A. Shamir, How to share a secret, Communications of the ACM 22 (1979) 612–613.

[7] M. Naor, A. Shamir, Visual cryptography, in: A. DeSantis (Ed.), Advances in Cryptology – EUROCRYPT'94, Lecture Notes in Computer Science, Perugia, Italy, vol. 950, 1994, pp. 1–12.

[8] H. Kuwakado, H. Tanaka, Size-reduced visual secret sharing scheme, IEICE – Transactions on Fundamentals of Electronics Communications and Computer Sciences E87-A (2004) 1193–1197.

[9] R. Ito, H. Kuwakado, H. Tanaka, Image size invariant visual cryptography, IEICE – Transactions on Fundamentals of Electronics Communications and Computer Sciences E82-A (1999) 2172–2177.

[10] C. Yang, New visual secret sharing schemes using probabilistic method, Pattern Recognition Letters 25 (2004) 481–494.

[11] S. Cimato, R.D. Prisco, A.D. Santis, Probabilistic visual cryptography schemes, The Computer Journal 49 (2006) 97–107.

[12] D. Wang, F. Yi, X. Li, Probabilistic visual secret sharing schemes for grey-scale images and color images, Information Sciences 181 (2011) 2189–2208.

[13] S.-J. Lin, S.-K. Chen, J.-C. Lin, Flip visual cryptography (fvc) with perfect security, conditionally-optimal contrast, and no expansion, Journal of Visual Communication and Image Representation 21 (2010) 900–916.

[14] F. Liu, T. Guo, C. Wu, L. Qian, Improving the visual quality of size invariant visual cryptography scheme, Journal of Visual Communication and Image Representation 23 (2012) 331–342.

[15] M. Naor, B. Pinkas, Visual authentication and identification, Advances in Cryptology-Crypt'97 1294 (1997) 322–336.

[16] C. Yang, T. Chen, Security analysis on authentication of images using recursive visual cryptography, Cryptologia 32 (2008) 131–136.

[17] J.M. McCune, A. Perrig, M.K. Reiter, Seeing-is-believing: using camera phone for human-verifiable authentication, in: IEEE Symposium on Security and Privacy, 2005, pp. 110–124.
[18] S. Shyu, S. Huang, Y. Lee, R. Wang, K. Chen, Sharing multiple secrets in visual cryptography, Pattern Recognition 40 (2007) 3633–3651.
[19] J. Feng, H. Wu, C. Tsai, Y. Chang, Y. Chu, Visual secret sharing for multiple secrets, Pattern Recognition 41 (2008) 3572–3581.
[20] C.N. Yang, T.H. Chung, A general multi-secret visual cryptography scheme, Optical Communication 283 (2010) 4949–4962.
[21] K.H. Lee, P.L. Chiu, A high contrast and capacity efficient visual cryptography scheme for the encryption of multiple secret images, Optical Communication 284 (2011) 2730–2741.
[22] D.S. Tsai, T.H. Chen, G. Horng, A cheating prevention scheme for binary visual cryptography with homogeneous secret images, Pattern Recognition 40 (2007) 2356–2366.
[23] C.N. Yang, A. Peng, T. Chen, Mtvss: (m)isalignment (t)olerant (v)isual (s)ecret (s)haring on resolving alignment difficulty, Signal Processing 89 (2009) 1602–1624.
[24] F. Liu, C.K. Wu, X.J. Lin, The alignment problem of visual cryptography schemes, Designs, Codes and Cryptography 50 (2009) 215–227.
[25] S. Cimato, C. Yang, Visual cryptography and secret image sharing, Digital Imaging and Computer Vision, CRC Press, Taylor & Francis, 2011.
[26] S. Alharthi, P.K. Atrey, An improved scheme for secret image sharing, IEE, International Conference on Multimedia and Expo (ICME), 2010, pp. 1661–1666.
[27] C. Cachin, Entropy measures and unconditional security in cryptography, Ph.D. thesis, Swiss Federal Institute of Technology Zurich, 1997.
[28] T. Guo, F. Liu, C.K. Wu, On the equivalance of two difinitions of visual cryptography scheme, Information Security Practice and Experience 7232 (2012) 217–227.
[29] O. Takizawa, A. Yamamura, A proposal of secret sharing using natural language text, in: IPSJ Computer Security Symposium, 2001, pp. 343–348.
[30] O. Takizawa, A. Yamamura, K. Makino, Secret sharing scheme using natural language text, Journal of the National Institute of Information and Communications Technology 52 (2005) 173–183.