Visual Cryptography on Graphs

Steve Lu[*]
University of California, Los Angeles
stevelu@math.ucla.edu

Daniel Manchala[†]
Xerox Corporation
daniel.manchala@xerox.com

Rafail Ostrovsky[‡]
University of California, Los Angeles
rafail@cs.ucla.edu

Appeared in COCOON 2008: 225-234

Abstract

In this paper, we consider a new visual cryptography scheme that allows for sharing of *multiple* secret images on graphs: we are given an arbitrary graph (V, E) where every node and every edge are assigned an arbitrary image. Images on the vertices are "public" and images on the edges are "secret". The problem that we are considering is how to make a construction such that when the encoded images of two adjacent vertices are printed on transparencies and overlapped, the secret image corresponding to the edge is revealed. We define the most stringent security guarantees for this problem (perfect secrecy) and show a general construction for all graphs where the cost (in terms of pixel expansion and contrast of the images) is proportional to the chromatic number of the cube of the underlying graph. For the case of bounded degree graphs, this gives us constant-factor pixel expansion and contrast. This compares favorably to previous works, where pixel expansion and contrast are proportional to the number of images.

Keywords: *Visual Cryptography, Multi-Secret Sharing, Graph Decomposition*

[*] Department of Mathematics. Work done in part while visiting IPAM. Partially supported by Xerox Innovation Group Award, NSF grants 0430254, 0716835, 0716389, and NSF VIGRE grant DMS-0502315

[†] Partially supported by the Xerox Innovation Group Award

[‡] Department of Computer Science and Department of Mathematics. Work done in part while visiting IPAM. Partially supported by Xerox Innovation Group Award, IBM Faculty Award, NSF grants 0430254, 0716835, 0716389 and U.C. MICRO grant.

1 Introduction

Secret sharing, introduced independently by Blakley[Bla79] and Shamir[Sha79], is a scheme for an authority to encode a secret into shares to be distributed to a set of n participants such that only qualified subsets of these participants may reconstruct the secret. It is also required that unqualified subsets learn nothing about the secret. In their works, both Blakley and Shamir describe a k-out-of-n threshold secret sharing scheme, where any subset of at least k participants may reconstruct the secret. In general, there is a set Γ, known as an *access structure*, which denotes the collection of subsets of participants that can recover the secret. Note that Γ must be monotone increasing, i.e. if $A \in \Gamma$ and $A \subset B \subset P$ then $B \in \Gamma$. The study of secret sharing schemes has been generalized to arbitrary access structures[BL90, ISN87]. Multi-secret sharing involves multiple secrets, with possibly different access structures, to be shared across participants. In this scenario, the authority can distribute shares in a way that different qualified participant sets may recover different secrets. These schemes[BSV93, BSC^{+}94, BSSV97, Cre03] perform better than trivially instantiating multiple single-secret sharing schemes.

Visual cryptography schemes (VCS), introduced by Naor and Shamir[NS94], involve a dealer encoding a *secret* (or *target*) *image* into shares to be distributed to n participants. These shares, when printed on transparencies, may be recombined simply by overlapping them. When a qualified subset of the participants overlap their transparencies, a human-recognizable facsimile of the secret image appears. The main benefit of such schemes is that the participants do not need to rely on machines to perform the reconstruction. In a generalization of this scheme, it is sometimes additionally required that each share is a human-recognizable image. In this type of extension, each participant may have their own *source image* (that is known to the authority) and the share generated for each user by the authority must “look” like their source image (see Section 2 for definitions). If the shares are generated in this fashion to match the source images, we call the scheme an Extended Visual Cryptography Scheme (EVCS). Indeed, many researchers have worked on EVCSs, giving constructions and proving bounds for them [ABSS96b, ABSS96a, ABSS01].

1.1 Organization of Our Results

The works [ABSS96b, ABSS96a, ABSS01] focused on the case where there was only one secret image to be reconstructed. In this paper, we consider the natural generalization of this for multiple secret images. The problem our paper addresses is how to extend previous constructions so that each pair of participants may have their own unique secret image that they can reconstruct together. We may treat this as a graph where each vertex represents a participant and each edge represents a secret image. We refer to this model as a Graph-Based Extended Visual Cryptography Scheme (GEVCS). In Section 2, we propose a definition of security and correctness for GEVCSs. We summarize our main results in Section 3 and spend the rest of the paper on the proofs and constructions. We will show first that the definition is satisfiable by a naïve construction in Section 4, then describe a better general construction for any graph in Section 5. In Section 6, we give a sample construction. Finally, in Section 7, we employ our construction on bounded degree graphs to give a GEVCS with constant-factor pixel expansion and contrast. Additionally, in Appendix A, we provide a visual example.

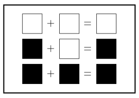

Figure 1: Overlapping Operation

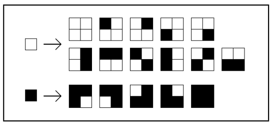

Figure 2: Pixel Expansion

1.2 Comparison to Previous Results

The previous results most relevant to our work are the multi-secret visual cryptography schemes proposed in [Dro96, KI98, IY03, CWL06, WYL$^+$06, YWL$^+$06]. Droste[Dro96] introduced the idea of different resulting images when overlapping different combinations of transparencies. The schemes described in Katoh-Imai [KI98], Iwamoto-Yamamoto [IY03] and Chen-Wu-Laih [CWL06] are special restricted cases of the problem we are addressing. Both Wang et. al.[WYL$^+$06] and Yi et. al.[YWL$^+$06] proposed a scheme for multiple visual secrets and general access structures. Using binary tree graphs for comparison, the [WYL$^+$06, YWL$^+$06] schemes' pixel expansions would grow on the order of the number of nodes, while our main construction has a pixel expansion of no more than 25 for even arbitrarily many nodes. Because of these practical considerations we had in mind (i.e. much better results, and with constant pixel expansion), we chose to use the graph-based model instead of a general access structure for this paper.

Our work differs from other previous results in visual cryptography, such as [NS94, ABSS96b, ABSS96a, ABSS01], by handling multiple secret images. These results use graph-based access structures as an example, however our scheme handles the case of one secret image per edge as opposed to only one secret image per graph structure. On the other hand, there are constructions of (non-visual) secret sharing or multi-secret sharing on a graph-based access structure[Sti94, BSSV95, BSSV97, Cre03, Csi05]. These are special types of access structures in which a graph $G = (V, E)$ is used to represent the sets of qualified participants. Each vertex is treated as a participant, and an edge between two participants indicates the two of them together may recover a secret. The constructions given in this paper involve graph decompositions, but our methods differ from these previous constructions as we must take into account the visual aspects in addition to the multi-secret requirements. We will describe our novel decomposition in the following sections.

1.3 Background

We give a review of extended visual cryptography in the case of 2 participants and produce a 2-out-of-2 scheme (denoted $(2, 2)$-EVCS). We begin by introducing the physical model of the problem.

Physical Model. The physical model of our scheme will use images printed on transparencies (as in [NS94]). Black pixels will be printed onto the transparency making these portions completely opaque, leaving the remaining portion completely transparent (we will refer to these as white pixels). Thus the transparency can be viewed as a Boolean matrix, where a 1 in the (i, j)th entry represents a black pixel at that location and a 0 represents a white pixel. When overlapping two transparencies, the result will have a black pixel where either of the two had a black pixel, and a white pixel only where both have a white pixel (Figure 1). This operation may be viewed as the Boolean OR operation performed entrywise on the two matrices. Because our constructions are all pixel-wise operations, all images are henceforth just a single black or white pixel.

The operation of overlapping two transparencies is inherently a destructive operation; one cannot "invert" the opaqueness caused by overlapping with a black pixel. This is apparent by the fact that the OR operation lowers entropy. Thus, in order to retain information, we will introduce some redundancy in the way a black or white pixel may be viewed. We sometimes refer to this process as *encoding* an image, and one should keep in mind the distinction between the original and the encoded image (which contains more information). In particular, we encode 1 pixel as m (usually chosen to be a perfect square) subpixels (known as the *pixel expansion*), each which may be black or white. If the original image was of size $p \times q$, then the encoded image will be of size $p\sqrt{m} \times q\sqrt{m}$. Each of the 2^m colorings of the subpixels of an encoded pixel may be visually interpreted as a single black or white pixel. The natural visual interpretation is to say if there are more than some threshold d black subpixels then view it as black, otherwise view it as white (Figure 2). To accommodate the human eye, we may wish to preclude encodings that appear ambiguous in color. To do this, we can impose a contrast requirement that says an encoding of a white pixel must have less than $d-\alpha$ black subpixels (α is known as the absolute contrast, α/m the relative contrast). If we let 1 indicate a black pixel and 0 indicate a white pixel, then this may be viewed as an error correcting code where any string with Hamming weight greater than d encodes a 1 and any string with Hamming weight less than $d-\alpha$ encodes a 0.

Extended Visual Cryptography. We review the problem of extended visual cryptography for two participants and a dealer. Loosely speaking, the goal of the dealer is to take public images A_1 and A_2 and a secret image B and create secure encoded shares S_1 and S_2 such that S_i "looks like" A_i and the overlap of S_1 and S_2 "looks like" B. Formally, the setup is as follows: each participant has a public image, say "A_1" and "A_2", which are known as the two *source* images. There is a secret image, say "B", known as the *target* image, to be shared between them by a dealer. The dealer must then encode A_1 and A_2 into shares S_1 and S_2 (possibly under different encodings) by selecting the colors of the subpixels in a way so that when S_1 and S_2 are overlapped, the result is an encoding of B (possibly yet another encoding). In addition, like in a secret sharing scheme, we will define a perfect secrecy requirement that should be satisfied.

Contrast Correctness. While many encodings could in theory solve the above problem, we wish to restrict ourselves to only those encodings that satisfy some contrast property. Although this creates a more difficult problem, the effort put into finding a solution is rewarded by the practical property of the scheme that allows the unaided decoding of the images by the human eye. We say a particular encoding is (α-)*contrast correct* if the absolute contrast of the encoding is at least α. Note that in a single visual cryptography scheme, there may be many different encodings, e.g. S_1 encodes A_1 under one encoding, S_2 encodes A_2 using another encoding, and the overlap of S_1 and S_2 encodes B in yet another encoding.

Perfect Secrecy. The shares individually should not reveal any information about the secret image. We view this as a game between a probabilistic poly-time dealer $\mathcal{D}$ and an adversary $\mathcal{A}$ with infinite computational power. The adversary generates the two source images (recall they are treated as single pixels) A_1 and A_2 and an index $i \in \{1,2\}$. The dealer then randomly selects the target image B as either a black or white pixel and creates shares S_1 and S_2 for B and sends S_i back. The adversary must then attempt to guess what the color of B is. We say the dealer's algorithm is *perfectly secret* if the probability that the adversary wins is exactly 1/2. More formally,

$$Pr\left[(A_1, A_2, i) \leftarrow \mathcal{A}, B \leftarrow \{black, white\}, \{S_1, S_2\} \leftarrow \mathcal{D}(A_1, A_2, B), B' \leftarrow \mathcal{A}(S_i); B = B'\right] = \frac{1}{2}$$

An EVCS Construction. We review an EVCS similar to the ones found in [NS94, ABSS01] that solves this problem. As stated before, the scheme will operate on individual pixels, so the input will be *source pixels* A_1 and A_2 and a *target pixel* B. The two shares S_1 and S_2 each consist of m (the pixel expansion) subpixels and together can be represented by a $2 \times m$ Boolean matrix, called a *share matrix*. We can then consider 8 collections $C_0^{00}, C_1^{00}, C_0^{01}, \ldots, C_1^{11}$ of $2 \times m$ matrices to be defined below.

Let $(s_0^1, s_1^1, s_0^2, s_1^2, t_0, t_1) \in \{1 \ldots m\}^6$ define how many black subpixels each source or target pixel should be encoded into, e.g. s_0^1 (resp. s_1^1) is the number of black subpixels a white (resp. black) pixel in the first source image will be encoded into. The collection C_z^{xy} will contain all permutations of the columns of the matrix

$$S_z^{xy} = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 & \ldots & 1 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 & 1 & \ldots & 1 & 1 & \ldots & 1 & 1 & 0 & \ldots & 0 \end{bmatrix}$$

where row 1 contains s_x^1 1's, row 2 contains s_y^2 1's and the OR of the two rows contain t_z 1's. The S_z^{xy}'s are well-defined as long as $s_a^1 + s_b^2 \geq t_c$ (this ensures we have enough black subpixels) and $max(s_a^1, s_b^2) \leq t_c$ (this ensures we do not have too many black subpixels).

Then, to generate the shares for source and target images A_1, A_2, B, the dealer will randomly choose a matrix $M \in C_z^{xy}$ where x is the color of A_1, y is the color of A_2, and z is the color of B, and set S_1 as the first row of the matrix and S_2 as the second row.

2 Our Definitions

In this section, we consider the problem of generating shares for n participants organized in a graph structure. We remind the reader that the graphs are interpreted differently than in Ateniese et. al. [ABSS96b]. We interpret the graph to denote which pairs of participants may overlap their shares to reconstruct the secret image dealt between them. For example, a complete graph would mean any pair of participants may overlap their shares to get a secret image *for that pair*, resulting in a total of $\binom{n}{2}$ possible secret images. In this case, each vertex will have a source image A_i attached to it, and each edge will have a secret target image B_e attached to it. A (probabilistic) polynomial-time computable algorithm that takes these as input and produces image shares S_i (each of length m, the pixel expansion) that satisfy the properties defined below will be referred to as a *Graph-based Extended Visual Cryptography Scheme* or GEVCS. This choice of graph structure is a practical one – indeed an interesting question would be to extend our constructions to general multi-secret access structures.

As a reference, we summarize all the properties of a GEVCS:

- A graph $G = (V, E)$ with n vertices and r edges.

- Source images A_i for each vertex i, each being a black (1) or white (0) pixel.

- Target images B_e for each edge e, each being a black or white pixel.

- Source shares S_i to be generated for each vertex i, each a vector of length m, the pixel expansion. The share matrix M is an $n \times m$ matrix where row i is S_i.

- Target shares T_e, obtained by overlapping S_i and S_j where $e = (i, j)$. Algebraically, this may be written as $T_e = S_i \vee S_j$.

- S_i encodes A_i by having at least s_1^i black pixels if $A_i = 1$ and at most s_0^i black pixels if $A_i = 0$. The contrast is $\alpha_i = s_1^i - s_0^i$.

- T_e encodes B_e by having at least t_1^e black pixels if $B_e = 1$ and at most t_0^e black pixels if $B_e = 0$. The contrast is $\alpha_e = t_1^e - t_0^e$.
- A security property loosely defined as: Fix an edge $e^\star = (i^\star, j^\star)$. A computationally unbounded adversary cannot distinguish between whether $B_{e^\star}$ is black or white even when given every A_i, every B_e for $e \neq e^\star$, and every S_i for $i \neq i^\star$.

2.1 Contrast Correctness for GEVCS

When generating shares S_i (and overlapped shares T_e) given a graph $G = (V, E)$ with n vertices and r edges, we define the following contrast properties. Each S_i should have at least s_1^i 1's when encoding a 1 and at most s_0^i 1's when encoding a 0. We can similarly parameterize these thresholds for the T_e and obtain t_1^e and t_0^e. Define the *(absolute) contrast* to be $\alpha_i = s_1^i - s_0^i$, $\alpha_e = t_1^e - t_0^e$. Define the *relative contrast* to be $\frac{\alpha}{m}$, the ratio between the absolute contrast and the pixel expansion. In essence, it is the relative contrast that affects how clear the final images will appear to be to the human eye.

Definition 2.1. We say a GEVCS satisfies the *contrast correctness* property with parameters $(s_0^i, s_1^i, t_0^e, t_1^e)$ if for every possible set of source images A_i and target images B_e, each share S_i that is generated is a valid encoding (under these parameters) of A_i, and overlapping two of them along an edge e results in a valid encoding of B_e.

2.2 Security for GEVCS

Visual cryptography schemes traditionally come with a guarantee of security by means of defining perfect secrecy. Usually, a set of forbidden players is not allowed to learn any information about the (one) secret image even under the possibility of collusion. In our scheme, participants share different secrets with different people, thus we need to take this into account when defining security.

Take the example of a GEVCS scheme on a military chain-of-command, represented by a graph. A general may have different secrets when overlapping with his different lieutenants. These secrets may be highly sensitive, and one of the benefits of having a scheme with source images is that the shares may be rather inconspicuous, e.g. printed as a picture of a common object, or may be used to authenticate the carrier of the image, e.g. printed as a photograph of the soldier. While these natural images may be used to mislead potential adversaries, we still demand a secrecy guarantee for such schemes. We would like to guarantee that even if all of the lieutenants were captured and their shares and source images were collected (along with all possible overlaps of their shares), the general's source image (but not his share), and all but one lieutenant revealed (under interrogation) the secret target images they shared with the general, then still no information should be revealed regarding the one honest remaining lieutenant's secret image with the general.

To further illuminate this point, consider a graph $G = (V, E)$ and the set of source images A_i, the set of (secret) target images B_e, the generated source images S_i, and the overlapped source images T_e. Select an edge, $e^\star$, and a vertex on that edge, $i^\star$, and suppose all of the source images on A_i were revealed, along with all of target images B_e, on the edges $E \setminus e^\star$. Furthermore, reveal all of the shares S_i in $V' \setminus i^\star$. Perfect secrecy guarantees that the adversary should learn nothing about the original target image $B_{e^\star}$. We may once again view this as a game between the dealer and an adversary with infinite computational power. As we operate on the image pixel by pixel, security will be defined on a single pixel. The adversary starts with a graph $G = (V, E)$ and selects a vertex $i^\star$ and an edge $e^\star$ and generates source images A_i for each $i \in V$ and target images B_e for each $e \in E \setminus \{e^\star\}$ and sends this to the dealer. $B_{e^\star}$ is randomly chosen to be black or white. After applying the GEVCS to generate shares S_i, the adversary obtains every share except $S_{i^\star}$. The adversary must then guess whether $B_{e^\star}$ is 0 or 1. Formally, we have the definition:

Definition 2.2. We say a GEVCS (a probabilistic polynomial-time algorithm named $\mathcal{D}$) is *secure* or *perfectly secret* for G if for any adversary $\mathcal{A}$ we have:

$$Pr\Big[(\{A_i\}, \{B_e\}_{e \neq e^\star}, i^\star, e^\star) \leftarrow \mathcal{A}(G), B_{e^\star} \overset{\text{R}}{\leftarrow} \{0,1\},$$

$$\{S_i\} \leftarrow \mathcal{D}(G, \{A_i\}, \{B_e\}), B \leftarrow \mathcal{A}(\{S_i\}_{i \neq i^\star}); B = B_{e^\star}\Big] = \frac{1}{2}$$

In some of the constructions, the GEVCS will deal the shares by sampling from a collection of matrices. On a graph $G = (V, E)$ there will be collections $C^{\{a_i\}}_{\{b_e\}}$, one for each possible assignment of 0's and 1's to $\{a_i\}_{i \in V}, \{b_e\}_{e \in E}$. We will also make use of a so-called *basis matrix* – this $n \times m$ (n being the number of participants and m being the pixel expansion) matrix contains the m subpixels to be assigned to player i in row i. The collections will arise as all matrices obtained by permuting the columns of the basis matrices, thus we will have one basis matrix for each possible assignment of the vertex and edge source images. We will parameterize the basis matrices for a graph G and source images $\{a_i\}_{i \in V}$ and target images $\{b_e\}_{e \in E}$ by $S^{\{a_i\}}_{\{b_e\}}$. Our constructions will give an explicit algebraic formula to compute the basis matrix from given values of $\{a_i\}$ and $\{b_e\}$.

2.3 Graph Theoretic Terminology

A *star* is a connected graph that has at most one vertex, known as the *center*, with degree greater than 1. A *star forest* is a graph where each connected component is a star. If we let $G = (V, E)$ be a graph, given a set of subgraphs $H_1, \ldots, H_k$ we say that they are a *graph (resp. star, star forest) cover* of G if every edge in E is contained in at least one H_i and each subgraph is a graph (resp. star, star forest). We let $N(v)$ denote the neighbors of a vertex v, or in other contexts, the neighborhood of v, i.e. the star centered at v with all its neighbors as points.

A subset $I \subset V$ of vertices is called an *independent set* if every edge has at most one endpoint in I. A *maximal independent set (MIS)* $I \subset V$ is one such that adding any vertex $v \in V \setminus I$ will result in a non-independent set. Note this is different from the notion of a *maximum independent set*, which is an independent set such that no other independent set has more elements than it. Finding a maximal independent set is quite easy while finding a maximum independent set is NP-hard.

Let $H_1, \ldots, H_k$ be subgraphs of $G = (V, E)$ such that each $H_i = (V_i, E_i)$ is a subgraph, each $v \in V$ belongs to at most one H_i, and there are no edges in G between any two vertices that are not in the same subgraph, i.e. $\nexists i \neq j, v_i \in V_i, v_j \in V_j (v_i, v_j) \in E$. There can still be edges in G between vertices in the *same* H_i. In this case, we say that $H_1, \ldots, H_k$ form an *independent subgraph set* and $H = \bigcup_{i=1}^{k} H_i$ is an *independent subgraph of* G. In addition, if each H_k is a star, we say H is an *independent star forest subgraph of* G. In Figure 3 we decompose a graph (top) into a union of star subgraphs. Notice that each edge is contained in at least one star.

The *cube* of G is a new graph $G^3 = (V, E')$ where $(v, w) \in E'$ if v and w are connected by a path of at most length 3. A *coloring* of a graph is an assignment of a color to each vertex so that no edge has its two endpoints the same color. The *chromatic number* of a graph is the minimum number of colors required to color the graph. The *degree* of a graph is the maximum of the degrees of all its vertices and *(d-)bounded degree graphs* are those which have degree at most d.

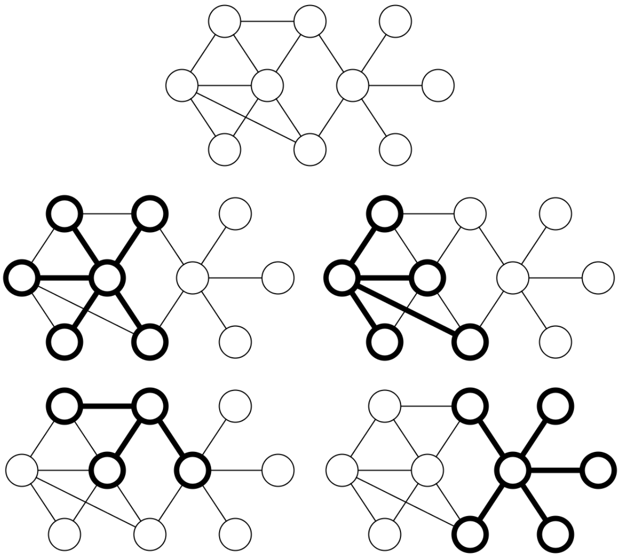

Figure 3: Star Forest Decomposition

3 Main Result

Our main result is stated as follows:

Theorem 3.1. *Let $G = (V, E)$ be a graph where no vertex has degree greater than d and let χ be the chromatic number of G^3. Then there exists a GEVCS on G with pixel expansion at most $m = \chi(5d + 1)$, and absolute contrast 2 for each source image on a vertex and 4 for each target image on an edge. Furthermore, we give an explicit construction for a GEVCS with pixel expansion at most $m = (d^3 + 1)(5d + 1)$.*

We will build up to this result in the remainder of the paper.

4 Warming Up: A Naïve Construction

For practical applications of GEVCSs, we wish to maximize the contrast and minimize the pixel expansion for the encoded images. This involves selecting better contrast parameters so that the contrast is increased. The question is whether or not we can construct a GEVCS to satisfy the chosen parameters. We will instead construct a GEVCS for a given graph G, then evaluate the contrast and pixel expansion necessitated by the construction. We begin by exploring a naïve construction of a GEVCS for a complete graph that involves a pixel expansion of $m = 2n^2 - n$ with relative source contrast $\frac{1}{m}$ and relative target contrast $\frac{2}{m}$. Compare this to the optimal lower bounds in the recent work of Blundo et. al. [BCS06]. They show a tight lower bound of $m \approx n^2/4$ with relative contrast $\frac{1}{m}$ for a $(2, n)$-VCS that has no source images (the shares are not required to look like anything) and only a single secret image to recover.

4.1 Satisfying Both Security and Contrast for General Graphs

We present a construction of a GEVCS on any graph satisfying certain contrast parameters. This construction will turn out to be perfectly secret as well. For any complete graph $G = (V, E)$ of n vertices we give a construction with a pixel expansion of $m = 2n^2 - n$ and will determine the parameters $s_0^i, s_1^i, t_0^e, t_1^e$ after the construction. For each possible assignment of $\{a_i\}, \{b_e\}$ we will construct the basis matrix $S_{\{b_e\}}^{\{a_i\}}$ (we will write S for ease of reading). Each basis matrix will contain a so-called "source-contrast" block, U, meant to allow the source subpixels to pass the threshold for a black pixel, followed by n "target-contrast" blocks $T_1, \ldots, T_n$ meant to control the number of black subpixels of the target image. First define the $n \times n$ matrix U as:

$$U = \begin{bmatrix} a_1 & 1 & 1 & \ldots & 1 \\ 1 & a_2 & 1 & \ldots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \ldots & a_n \end{bmatrix}$$

If a_i is black then row i will have one extra black subpixel. This will be used to differentiate a black pixel from a white pixel of the source image. The remaining matrices, T_i, will be used to control the darkness of the target image. We define each T_i to be the $n \times 2(n-1)$ matrices:

$$T_i = \begin{bmatrix} b_{(i,1)} & 0 & 0 & \ldots & 1 - b_{(i,1)} & 0 & 0 & \ldots \\ 0 & b_{(i,2)} & 0 & \ldots & 0 & 1 - b_{(i,2)} & 0 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 1 & 1 & 1 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & b_{(i,n)} & 0 & 0 & \ldots & 1 - b_{(i,n)} \end{bmatrix}$$

The form of these matrices is to start with an $(n-1) \times (n-1)$ matrix with diagonal entries $b_{(i,j)}$ (there is no $b_{(i,i)}$), then concatenate a matrix with diagonal entries $1 - b_{(i,j)}$, then insert a new row i which consists of $n - 1$ zeroes followed by $n - 1$ ones. Notice when any row j is overlapped with row i, the result will contain all 1's in the right half, and all zeroes except $b_{(i,j)}$ in the left half. On the other hand, when any row j is overlapped with row $k \neq i$, the result will contain exactly two 1's.

Finally we let $S = U||T_1|| \cdots ||T_n$ (where $||$ denotes horizontal matrix augmentation), a matrix with $2n^2 - n$ columns and n rows. We now count how many 1's there are in each row i corresponding to a white (resp. black) source pixel. There will be $n - 1$ (resp. n) 1's in the U block, there will be a single 1 in each T_j block with $j \neq i$, and there will be $n - 1$ 1's in the T_i block. Thus we can set $s_0^i = 3(n-1)$ and $s_1^i = 3(n-1) + 1$. We may similarly count how many 1's there are in the overlap of two rows i and j with a white (resp. black) corresponding target pixel: n in the U block, n (resp. $n + 2$) in the T_i and T_j blocks, and 2 in each T_k block for $k \neq i, j$. Thus we can set $t_0^e = 4n - 4$ and $t_1^e = 4n - 2$. Then our construction satisfies contrast for these parameters on a complete graph.

To ensure security, we randomly permute the columns of the matrix S before setting share S_i as the ith row. We now prove the construction satisfies the security definition in the previous section. *Proof.*

Let $\mathcal{A}$ be an adversary with infinite computational power which will play against an honest dealer $\mathcal{D}$ as defined in Definition 2.2. Let $\{a_i\}, \{b_e\}_{e \neq e^\star}$ be the images generated by $\mathcal{A}$ and without loss of generality take $i^\star = 1, e^\star = (1,2)$. Using the construction above, let S (resp. S') be the basis matrix associated with $b_{(1,2)} = 0$ (resp. 1). Let C (resp. C') be the collection of matrices obtained by taking all permutations of S (resp. S'). A $j^\star$ is randomly chosen and the construction calls for the dealer to randomly sample a matrix from C if $j^\star = 0$ and from C' otherwise. Because the adversary does not receive S_1 (the first row), it appears as if the dealer were sampling from C or C' restricted to the $(n-1) \times m$ submatrix obtained by removing the first row. To complete the proof, we exhibit an identification between the matrices in C restricted to $(n-1) \times m$ submatrices and the restricted matrices in C' thereby showing the adversary has no information as to what $j^\star$ is. As only $b_{(1,2)}$ differs between the two, the only difference between S and S' is between the first two rows of T_1, T_2 and T_1', T_2'. As an example, we write down the first two rows for comparison of each of the matrices when $n = 4$:

$$T_1 = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} T_2 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

$$T_1' = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} T_2' = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Then, by the permutation τ which swaps columns 1 and n of T_1, we see that $\tau(S)$ is indistinguishable from S' when restricted. Any matrix in C' can be written as $\sigma(S')$ for some column permutation σ, and $\sigma(S')$ is identical to $\sigma\tau(S) \in C$ when restricted. This shows the collections C and C' are identical when restricted, and therefore this scheme preserves perfect secrecy. □

This construction extends to any graph by deleting every column j and $n - 1 + j$ (the columns containing entries $b_{(i,j)}$) from T_i if (i,j) is not an edge. This results in a pixel expansion of $n + \sum_{i=1}^n 2d_i = n + 4e$ where d_i is the degree of vertex i and e is the number of edges. The contrast parameters will be $s_0^i = (n-1) + 2d_i$ (there are $n-1$ black pixels in U, one black pixel in each T_j where (i,j) is an edge, and d_i black pixels in T_i), $s_1^i = n + 2d_i$, and if $e = (i,j)$ we have $t_0^e = n + (d_i - 1) + (d_j - 1) + d_i + d_j$ (there are n black pixels in U, one black pixel in T_k for each $(i,k) \in E$ and each $(j,k) \in E$, d_i in T_i, and d_j in T_j), and $t_1^e = n + (d_i - 1) + (d_j - 1) + (d_i + 1) + (d_j + 1)$.

We mention an additional property, known as *smoothness*, that will be important in a later section: overlapping two shares that do not have an edge between them will result in the same number of 1's regardless of the source and target images. To see why the smoothness property holds, let i and j be vertices such that (i,j) is not an edge in G. The overlap of S_i and S_j will have n black pixels in U block. In a T_k block where $k \neq i, j$ there will be black pixels depending on whether or not (i,k) and (j,k) are edges. In T_i and T_j, since the columns for $b_{(i,j)}$ are deleted their overlap will be some constant number of 1's equal to the number of columns in T_i or T_j that are not deleted: this is exactly the degree of vertices i and j, respectively. Thus, regardless of the images, there will always be $n + 2d_i + 2d_j$ black pixels in their overlap.

5 GEVCS for a General Graph

With examples of secure schemes shown to exist in the previous section, we now move to give constructions with better bounds on the pixel expansion and contrast. The idea is that we can view the act of overlapping a transparency with your neighbors as a local process so that we may seek to decompose our graphs into sufficiently independent local pieces, build a GEVCS for each piece, then patch them together in a meaningful way.

5.1 Building Blocks

Our construction idea is to construct GEVCSs for building blocks, then somehow combine them to form any graph. We apply the naïve construction for a small subgraph, then describe how these subgraphs may be patched together. Indeed, the idea of "patching together" several schemes has also been investigated by Droste[Dro96]. We now present two different ways the GEVCSs described above may be patched together to form a GEVCS on a graph G. We take a graph cover of G and first show how to generate shares for an independent subgraph set in parallel. Then we will show how to take these sets of subgraphs and combine their shares sequentially. An example of how these are used is given in Section 6.

Construction 5.1 (Parallel Sharing on Independent Subgraphs)**.** Let H be an independent subgraph of G. We can write $H = \bigcup_{i=1}^{k} H_i$, where the H_i are the independent pieces (recall this means that each H_i has no edges connecting to an H_j). Obtain a GEVCS for each subgraph using the naïve construction above. Let m be the maximum pixel expansion over all the subgraphs. Construct a new distribution of share matrices for $H = \bigcup_{i=1}^{k} H_i$ by first sampling a share matrix from each GEVCS on H_j. The new matrix will have one row for each vertex in H, and because each vertex is uniquely contained in some H_j we may assign to it the corresponding row from the GEVCS on H_j (also, pad them with 0's at the end so that each row is of length m).

By observing that this is simply sampling multiple GEVCSs on independent subgraphs, we obtain the following lemma:

Lemma 5.2. *By sampling the share matrix according to the distribution in Construction 5.1, we obtain a secure GEVCS on H. The pixel expansion is equal to the maximum pixel expansion of the GEVCSs on the individual subgraphs and maintains the same contrast parameters for each vertex and edge. Also, this scheme satisfies the* smoothness *property.*

The smoothness property mentioned above is used to maintain perfect secrecy. In addition to this construction, we have a second construction to patch together all of the independent subgraphs of G.

Construction 5.3 (Sequential Sharing on Dependent Subgraphs)**.** Let $K_1, \ldots, K_\ell$ be a graph cover of $G = (V, E)$ where each K_k is an independent subgraph. Use Construction 5.1 on the K_i to obtain GEVCS schemes on each of these. For each K_i, first pad the shares in its GEVCS with rows $i \in V \setminus V_i$ by filling the all these rows with 1's. Each of these matrices will have n rows, and we can then concatenate them horizontally. This completes the construction of a new distribution of shares on G.

Lemma 5.4. *By sampling the share matrix according to the distribution in Construction 5.3, we obtain a secure GEVCS on G. The pixel expansion is equal to the sum of the pixel expansions of the GEVCS on each of the subgraphs. The contrast parameters are dependent on how many times a vertex or edge appears in the decomposition; in terms of absolute contrast, a source image has absolute contrast equal to the number of times its vertex appears in the covering, and a target image has absolute contrast equal to twice the number of times its edge appears in the covering.*

Proof Sketch:

(Contrast Correctness) We consider the contrast on an edge e_{ij} in G. The overlap of share i and share j will contain a number of 1's equal to the sum of the overlap of share i and share j in each $K_1, \dots, K_\ell$. Thus we may consider three cases: if $e_{ij} \in K_k$, if $e_{ij} \notin K_k$ but both nodes i and j are in K_k, and if at least one node is not in K_k. In the first case, the contrast property of the GEVCS on H_k will contribute to the overlap being darker if the secret pixel on e_{ij} is black. In the second case, by the special property of the GEVCS, we have that there will always be a constant number of 1's, thus not affecting the darkness either way. In the last case, the share of the node not in H_k will be all 1's, hence the overlap will always be completely black. Thus each block satisfies the contrast property, and after summing over all the $1, \dots, \ell$ blocks, we still satisfy the contrast property.

(Perfect Secrecy) To show this construction is secure, let $e^\star$ be the edge the adversary wishes to attack. Without loss of generality, assume $e^\star \in K_1, \dots, K_\ell$. We construct a series of hybrid matrices where the ith matrix is sampled from a distribution where $e^\star$ is white in $K_1, \dots, K_i$ but black in $K_{i+1}, \dots, K_\ell$. By the perfect secrecy property on each K_i, the view of the adversary remains the same between any two consecutive steps in the hybrid. After completion of the entire hybrid, we have changed the color of $e^\star$ of our construction from white to black and shown the view of the adversary does not change. This shows perfect secrecy for Construction 5.3. □

5.2 Construction of a GEVCS for a General Graph

Given a graph G with an independent subgraph cover $K_1, \dots, K_\ell$ we can construct a GEVCS for G by applying the two previous constructions 5.1 and 5.3. First construct a GEVCS for each component of K_i using the naïve GEVCS construction described above. Then combine the shares in parallel by construction 5.1 to obtain GEVCSs for each independent subgraph K_i. This will be followed by combining the shares sequentially by construction 5.3 to finally obtain a GEVCS on G.

The final pixel expansion and contrast can be counted as follows. If each K_i is written as a union of its independent pieces $K_i = \bigcup_{j=1}^{k_i} H_{ij}$ then the pixel expansion of the parallel sharing will be the maximum of the pixel expansions of the naïve construction on all of the H_{ij}. We write n_{ij} and e_{ij} for the number of vertices and edges in H_{ij}, respectively, and obtain the pixel expansion for the GEVCS on K_i to be $m_i = \max_j\{n_{ij} + 4e_{ij}\}$. The sequential sharing will then give us the final pixel expansion $m = \sum_i m_i$. Similarly, we know the absolute contrast of the naïve construction is 1 for each source image on a vertex and 2 for each target image on an edge. Thus overall, the absolute contrast of a vertex is the number of subgraphs H_{ij} which contain it, and for an edge it is twice the number of subgraphs which contain it.

We make the observation that if one takes a coloring of the cube of a graph G, one can make an independent star forest cover of G by taking K_i to be the union of all stars around centers of color i. If it uses χ colors, then there will be χ of the K_i's. This is explained in further detail in Section 7. By combining the constructions above with the independent star forest cover in the following section, we obtain the main theorem: Theorem 3.1.

6 Example Construction of a GEVCS on a Graph

We construct a GEVCS for the graph seen in Figure 4. Label the vertices 1..6 top to bottom, left to right. As an example, we will use the source and target images as seen on the left of the figure. We will decompose the graph as $K_1 \cup K_2$ where K_1 (center of the figure) is the union of two independent pieces H_{11} (top bold portion) and H_{12} (bottom bold portion) and K_2 (right of the figure) is just H_{21} (bold). The basic construction for a share matrix for each of the H's are as follows (the vertical line separates the U and T_i blocks as in the naïve construction):

$$\text{Share Matrix for } H_{11} = \left[\begin{array}{cc|cc|cc} 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 \end{array}\right]$$

$$\text{Share Matrix for } H_{12} = \left[\begin{array}{ccc|cc|cc|cccc} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \end{array}\right]$$

$$\text{Share Matrix for } H_{21} = \left[\begin{array}{cccc|cc|cccccc|cc|cc} 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right]$$

When treated as subgraphs in G the rows correspond to the vertex number as follows:

$$\text{Share Matrix for } H_{11} = \left[\begin{array}{cc|cc|cc} 1 & 1 & 0 & 1 & 0 & 1 \\ - & - & - & - & - & - \\ 1 & 1 & 0 & 1 & 0 & 1 \\ - & - & - & - & - & - \\ - & - & - & - & - & - \\ - & - & - & - & - & - \end{array}\right]$$

$$\text{Share Matrix for } H_{12} = \left[\begin{array}{ccc|cc|cc|cccc} - & - & - & - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - & - & - & - \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \end{array}\right]$$

$$\text{Share Matrix for } H_{21} = \left[\begin{array}{cccc|cc|cccccc|cc|cc} 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - \end{array}\right]$$

In the actual construction, we would sample a random permutation of the columns of these matrices. We then apply parallel sharing on the matrices for H_{11} and H_{12} to obtain a share matrix for K_1. We pad H_{11} with 1's at the end to make it align with H_{12}. The share matrix for K_2 is just that of H_{21}.

$$\text{Share Matrix for } K_1 = \left[\begin{array}{ccccccccccc} 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ - & - & - & - & - & - & - & - & - & - & - \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \end{array}\right]$$

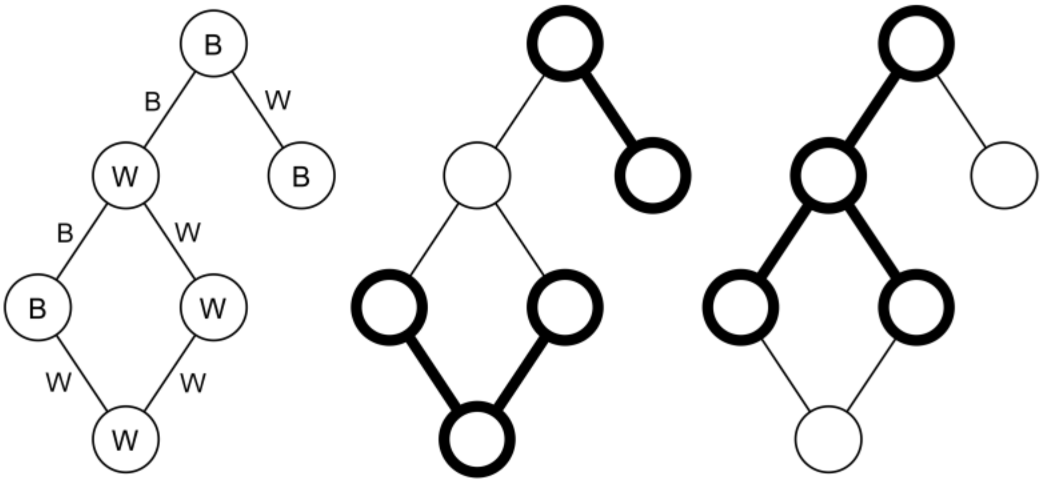

Figure 4: GEVCS Construction

Finally, we apply sequential sharing between K_1 and K_2 to obtain a share matrix for G, completing the construction. We accomplish this by concatenating the two matrices horizontally and fill the remaining blanks with 1's (the vertical line separates K_1 and K_2):

$$\text{Share Matrix for } G = \left[\begin{array}{ccccccccccc|cccccccccccccccc}
1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\
1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{array}\right]$$

7 Independent Star Forest Covers

In this section we describe how to find independent star forest covers for graphs to supply as input to our algorithm in the previous section. We describe the general construction of independent star forests for any graph G, and mention this construction leads to parameters depending only on the maximum degree of vertices of the graph.

First we give an example of an independent star forest decomposition on a tree.

Example Using Trees. Given a graph G that is a tree, we can decompose G into four independent star forests. Define V_j for $j = 1, 2, 3, 4$ to be the set of vertices whose distance from the root is $j \bmod 4$, then define $K_i = \bigcup_{v \in V_i} N(v)$. Then K_1, K_2, K_3, K_4 is an independent star forest cover of G. Indeed, the edge $e = (v, w)$ (where v is the parent of w) is covered by H_i where i is the distance of v from the root $\bmod 4$.

7.1 Algorithm for Finding Independent Star Forest Cover

We begin by making the observation that given a k-coloring of G^3, we can decompose G as follows: Let $K_i = \bigcup_{v \text{ has color } i} N(v)$. Note this is an independent star forest cover as an edge between $N(v)$ and $N(w)$ implies there is a path of at most length 3 between v and w which translates to an edge (v, w) in G^3, hence they cannot be of the same color. Each edge is covered exactly twice.

Our construction in the previous section can therefore theoretically be made with pixel expansion and contrast parameters dependent only on the chromatic number of G^3 and the degree of G. However, it is NP-hard to find the chromatic number, so instead we apply a less optimal solution to color the graph. We remodel the algorithm found in Luby [Lub86] into the algorithm in presented in Figure 5.

```
$i \leftarrow 0$
Construct $G^3 = (V, E)$
while $(V, E)$ is not empty do
  $i \leftarrow i + 1$
  Find a Maximal Independent Set $S$
  Color all the vertices in $S$ by color $i$
  $V \leftarrow V \backslash S$
end while
```

Figure 5: Coloring G^3

This algorithm will use at most $d^3 + 1$ colors (cf. [Lub86] Section 7) if G is of degree d. This is because at each stage if the node itself is not colored then at least one neighbor is colored (by the property of a maximal independent set). Thus at the next stage, its degree will drop by at least 1, and since each vertex in G^3 has at most degree d^3, we arrive at the conclusion of at most $d^3 + 1$ colors.

Combining this algorithm with the construction from the previous section gives rise to a construction of a GEVCS on any graph, and for d-bounded degree graphs a constant-factor (on the order of d^4) pixel expansion and contrast as stated in our main theorem. Unlike the naïve construction, this construction is independent of the number of participants.

8 Conclusion and Open Problems

In this paper we presented a Graph-based Extended Visual Cryptography Scheme. We provided a new security definition for such schemes and proved such schemes can always be constructed with sufficient parameters. We then considered a construction of a GEVCS on a star graph, then showed how to combine these into a GEVCS for any arbitrary graph. Finally, we described a complete construction (via an explicit independent star cover) of a GEVCS on any d-bounded degree graph with parameters depending only on d, thus giving an upper bound on the parameters for the scheme.

Because GEVCS is an extension of EVCS, certain theoretical bounds on pixel expansion contrast are carried over from previous works. One question to ask is whether or not these bounds can be tightened in this new setting. Further investigation into different types of graph decompositions and coverings may lead to better parameters.

References

[ABSS96a] Giuseppe Ateniese, Carlo Blundo, Alfredo De Santis, and Douglas R. Stinson. Constructions and bounds for visual cryptography. In Friedhelm Meyer auf der Heide and Burkhard Monien, editors, *ICALP*, volume 1099 of *Lecture Notes in Computer Science*, pages 416–428. Springer, 1996.

[ABSS96b] Giuseppe Ateniese, Carlo Blundo, Alfredo De Santis, and Douglas R. Stinson. Visual cryptography for general access structures. *Electronic Colloquium on Computational Complexity (ECCC)*, 3(12), 1996.

[ABSS01] Giuseppe Ateniese, Carlo Blundo, Alfredo De Santis, and Douglas R. Stinson. Extended capabilities for visual cryptography. *Theor. Comput. Sci.*, 250(1-2):143–161, 2001.

[BCS06] Carlo Blundo, Stelvio Cimato, and Alfredo De Santis. Visual cryptography schemes with optimal pixel expansion. *Theor. Comput. Sci.*, 369(1–3):169–182, 2006.

[BL90] Josh Cohen Benaloh and Jerry Leichter. Generalized secret sharing and monotone functions. In *CRYPTO '88: Proceedings of the 8th Annual International Cryptology Conference on Advances in Cryptology*, volume 403 of *Lecture Notes in Computer Science*, pages 27–35, London, UK, 1990. Springer-Verlag.

[Bla79] George Blakley. Safeguarding cryptographic keys. In *Proc. Am. Federation of Information Processing Soc.*, pages 313–317, 1979.

[BSC$^+$94] Carlo Blundo, Alfredo De Santis, Giovanni Di Crescenzo, Antonio Giorgio Gaggia, and Ugo Vaccaro. Multi-secret sharing schemes. In *CRYPTO '94: Proceedings of the 14th Annual International Cryptology Conference on Advances in Cryptology*, pages 150–163, London, UK, 1994. Springer-Verlag.

[BSSV95] Carlo Blundo, Alfredo De Santis, Douglas R. Stinson, and Ugo Vaccaro. Graph decompositions and secret sharing schemes. *J. Cryptology*, 8(1):39–64, 1995.

[BSSV97] Carlo Blundo, Alfredo De Santis, Roberto De Simone, and Ugo Vaccaro. Tight bounds on the information rate of secret sharing schemes. *Des. Codes Cryptography*, 11(2):107–110, 1997.

[BSV93] Carlo Blundo, Alfredo De Santis, and Ugo Vaccaro. Efficient sharing of many secrets. In *STACS '93: Proceedings of the 10th Annual Symposium on Theoretical Aspects of Computer Science*, pages 692–703, London, UK, 1993. Springer-Verlag.

[Cre03] Giovanni Di Crescenzo. Sharing one secret vs. sharing many secrets. *Theor. Comput. Sci.*, 295(1-3):123–140, 2003.

[Csi05] Laszlo Csirmaz. Secret sharing schemes on graphs. Cryptology ePrint Archive, Report 2005/059, 2005. `http://eprint.iacr.org/`.

[CWL06] K.Y. Chen, W.P. Wu, and C.S. Laih. On the (2,2) visual multi-secret sharing schemes, 2006.

[Dro96] Stefan Droste. New results on visual cryptography. In *Advances in Cryptology – CRYPTO 96*, volume 1109 of *Lecture Notes in Computer Science*, pages 401–415. Springer, 1996.

[ISN87] M. Itoh, A. Saito, and T. Nishizeki. Secret sharing scheme realizing general access structure. In *IEEE Globecom*, pages 99–102, 1987.

[IY03] Mitsugu Iwamoto and Hirosuke Yamamoto. A construction method of visual secret sharing schemes for plural secret images. *IEICE Trans. on Fundamentals*, E86.A(10):2577–2588, 2003.

[KI98] Taku Katoh and Hideki Imai. An extended construction method for visual secret sharing schemes. *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, 81(7):55–63, 1998.

[Lub86] Michael Luby. A simple parallel algorithm for the maximal independent set problem. *SIAM J. Comput.*, 15(4):1036–1055, 1986.

[NS94] Moni Naor and Adi Shamir. Visual cryptography. In *Advances in Cryptology – EUROCRYPT'94*, volume 950 of *Lecture Notes in Computer Science*, pages 1–12, 1994.

[Sha79] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[Sti94] Douglas Stinson. Decomposition constructions for secret sharing schemes. *IEEE Transactions on Information Theory*, 40(1):118–125, 1994.

[WYL+06] DaoShun Wang, Feng Yi, Xiaobo Li, Ping Luo, and Yiqi Dai. On the analysis and generalization of extended visual cryptography schemes, 2006.

[YWL+06] Feng Yi, Daoshung Wang, Ping Luo, Liansheng Huang, and Yiqi Dai. Multi secret image color visual cryptography schemes for general access structures. *Progress in Natural Science*, 16(4):431–436, 2006.

A Visual Example

Figures 6 and 7 show an example in the case of 3 users with secrets between each of the three possible pairs.

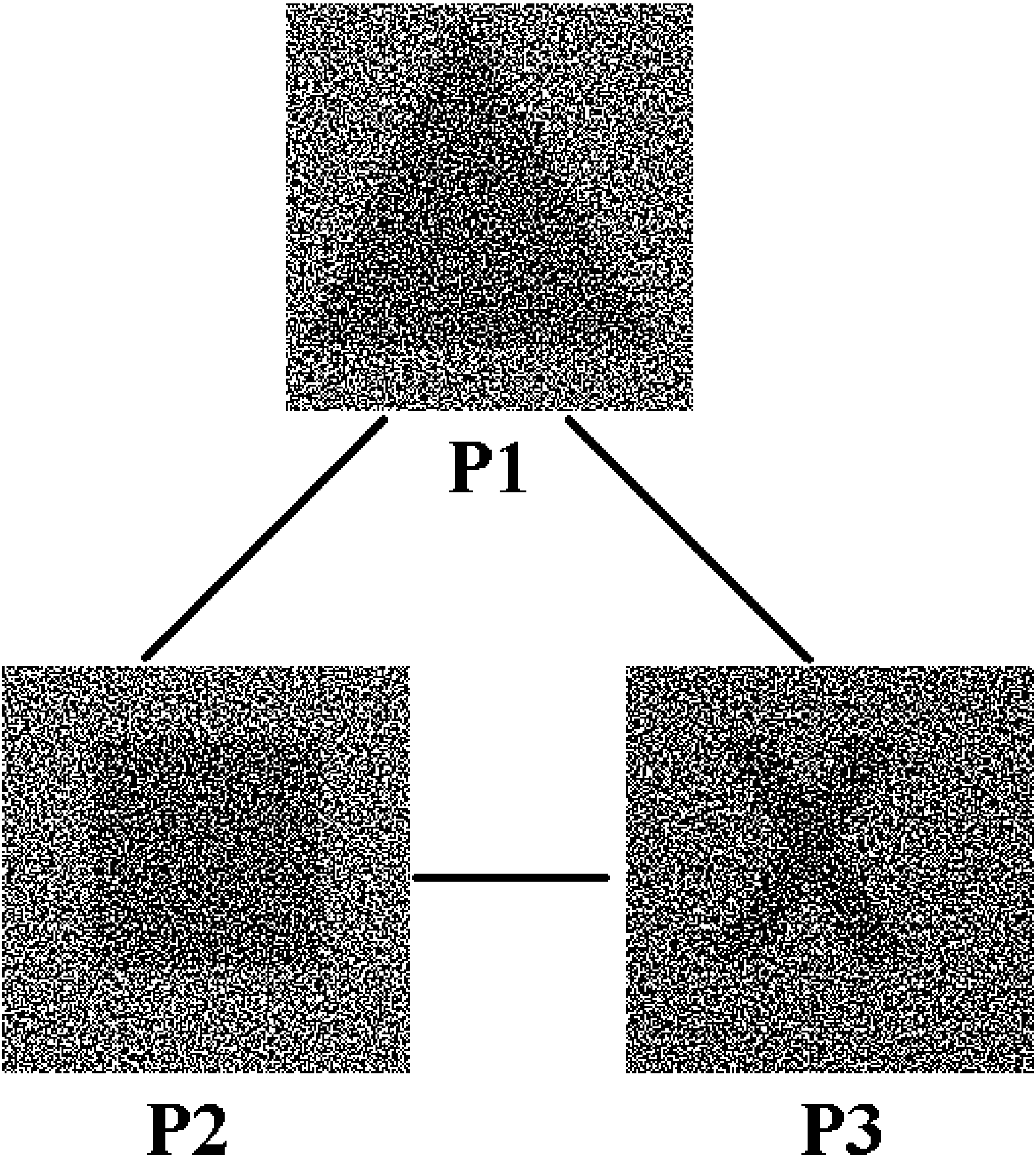

Figure 6: Player 1 holds a triangle, Player 2 holds a square, Player 3 holds a cross

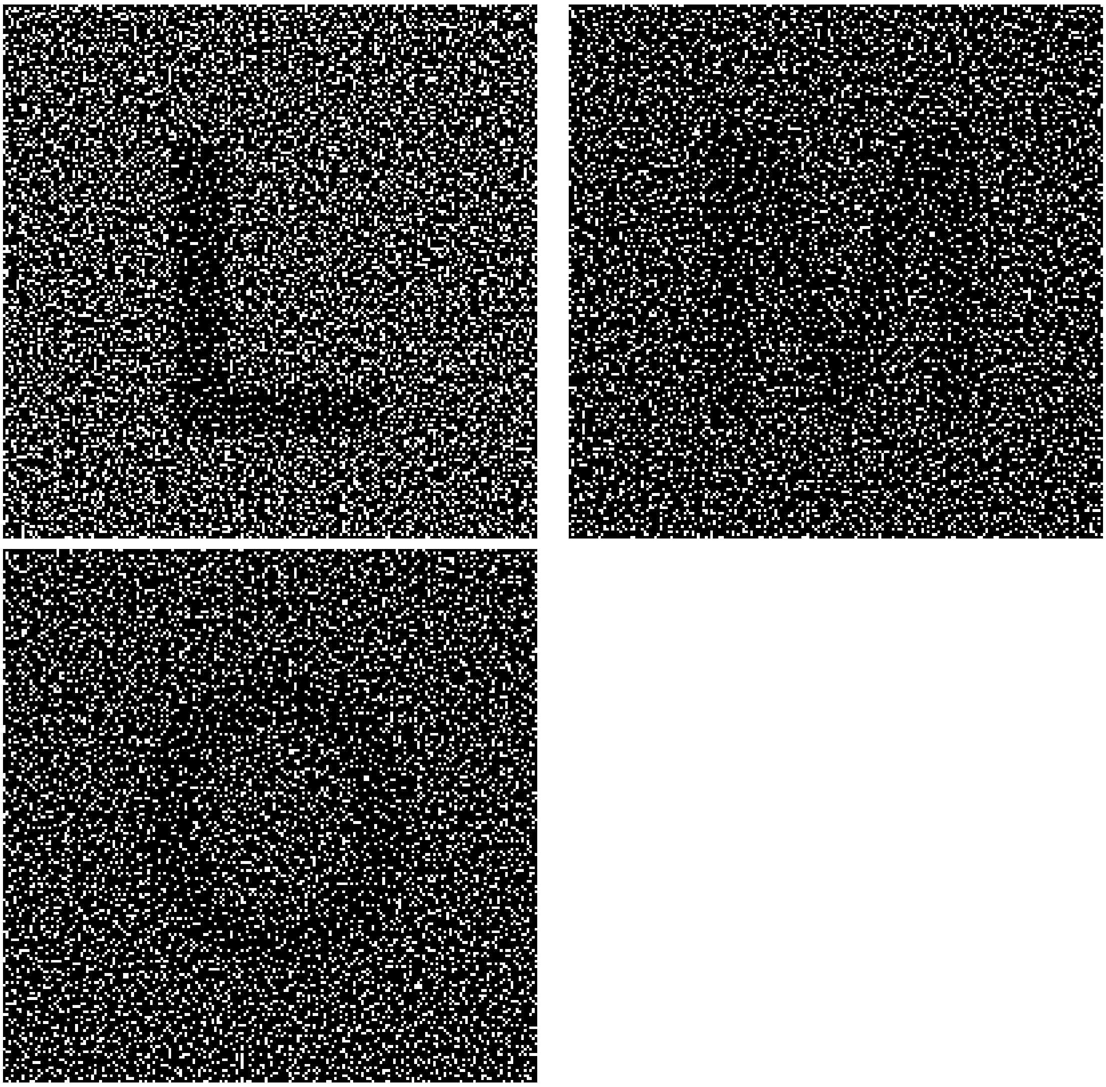

Figure 7: Player 1 and 3 overlap to recover L, Player 1 and 3 overlap to recover M, Player 2 and 3 overlap to recover O