Traitor-traceable Key Pre-distribution based on Visual Secret Sharing

Sheethal Kumar[1], Jaipal Pai B J[1], Sourav Sen Gupta[2⋆], and Vigesh R Ungrapalli[1]

[1] National Institute of Technology Karnataka, Surathkal, India
sheethalk.49@gmail.com, redevils_jaipal_710@hotmail.com, vigshru23@gmail.com
[2] Indian Statistical Institute, Kolkata, India
sg.sourav@gmail.com

Abstract. In this paper, we study the problem of traitor-traceable key pre-distribution for general access structures. We propose a new scheme for key pre-distribution using visual secret sharing, where the keys are generated based on certain combinatorial block designs. Our scheme naturally extends for general access structures, and provides a flexible many-to-one function using visual secret sharing concepts to efficiently avoid the problem of pixel expansion. In addition, our proposal accommodates a simple traitor-tracing functionality for video broadcast applications; using efficient PBIBD based combinatorial constructs and visual secret sharing based on random grids. In effect, our scheme provides a novel technique for secure video and image broadcast, using general access structures to reduce collusions, trace forgery, and identify traitors in case there is a collusion. We duly analyze and discuss the efficiency of our scheme for varying number of users in the broadcast network.

Keywords: key pre-distribution, general access structure, traitor tracing, combinatorial block design, visual secret sharing, random grids

1 Introduction

Secure transmission of intellectual properties of commercial value to a large number of recipients has beckoned the rise of data encryption in broadcast networks. In the past two decades, several techniques related to encryption of broadcast data have found diverse applications in video distribution, streaming video applications, area-specific media devices (CD, DVD etc.), access-controlled databases, and even in SaaS frameworks of the cloud architecture.

Naive idea to achieve data security in broadcast mode of transmission is to establish a secret-key secure channel between the distributor and each authorized recipient. Establishment of a common secret key between the transmitter and the receiver may be achieved either by using a public key framework for key

⋆ Supported by DRDO sponsored project Centre of Excellence in Cryptology (CoEC), under MOC ERIP/ER/1009002/M/01/1319/788/D(R&D) of ER&IPR, DRDO.

exchange, or by a simpler and alternative method of key pre-distribution. In case of key pre-distribution, it is hard to cater to a large user-base if single unique keys are provided to each user. The most compelling requirement for such an access control structure comes from the domain of image and video broadcast, where the transmitter generally caters to a large number of users.

1.1 Preliminaries of Media Broadcast

Conventional method of video broadcasting involves encrypting the video frames and broadcasting them across the network. It is expected that only authorized users can decrypt the video stream using an authenticated set of keys given to them by the transmitter, and no other individual may have a similar access to the data. The transmitter builds a main key-pool, and pre-distributes authorized subsets of keys to each authorized user. The user may thereafter construct an authorized decoder using his/her user key-set, and access the encrypted video broadcast on the network. The pre-distribution of the user key-sets are expected to satisfy the following conditions.

- When used together, the keys of a particular authorized user key-set must map to a single value – the final decryption key.
- None of the sub-sets of the main key-pool, apart from the authorized user key-sets, should map to the final decryption key.

Many-to-one functions. The key pre-distribution described above suggests access-control using a many-to-one function, which maps each of the authorized user key-sets to the final decryption key; but no other subset of the main key-pool to the final key. By definition, a many-to-one function is a relation $\mathcal{R} \subseteq S \times T$ where every element of the domain of $\mathcal{R}$ relates to exactly one element of its co-domain, that is,

$$(x, y_1) \in \mathcal{R} \wedge (x, y_2) \in \mathcal{R} \;\Rightarrow\; y_1 = y_2 \qquad \text{if and only if} \quad x \in \mathrm{Dom}(\mathcal{R}).$$

The key-sets meant for the authorized users lie within $\mathrm{Dom}(\mathcal{R})$, while the final decryption key is a member of the co-domain of $\mathcal{R}$. Several techniques have been proposed in the literature to construct such functions; many based on suitable combinatorial designs like BIBD and PBIBD [8, 7, 9–11, 13].

A specific technique of realizing these functions may be conceived using Visual Secret Sharing (VSS). The basic concept of Secret Sharing was independently proposed by Shamir [12] and Blakley [2] in 1979. Visual Secret Sharing (VSS) appeared in 1994 as a special sub-domain of secret sharing, where a single image is divided into secret shares and a combination of certain shares (defined by some access structure) results in the original image. The very first VSS scheme was proposed by Naor and Shamir [6], and later this was further extended to general access structure based VSS schemes by Ateniese, Blundo, Santis and Stinson [1]. In 2012, Wu and Sun [15] proposed a general access structure based VSS scheme using random grids, which efficiently overcame the critical problem of pixel expansion.

Traitor-tracing. The many-to-one functions solve the two conditions required for a key pre-distribution scheme. However, this introduces a new problem of potential collusion amongst authorized users to produce a new authorized key-set, which had not been provided by the distributor. Suppose that the main key-pool is $\mathcal{K}$, and N key-sets $K_1, K_2, \ldots, K_N \subset \mathcal{K}$ have been distributed to authorized users. Some of these authorized users may attempt a collusion to construct a new key-set K_{N+1} which maps to the final decryption key as well. The colluding users may choose to sell this pirated key-set K_{N+1} to enable unauthorized users to decrypt the contents of the encrypted media without proper consent or authorization by the original distributor.

In such a scenario, it is required to identify the colluding authorized users, hereafter called *traitors*, who have violated their contract by jointly producing a pirate key-set out of their own authorized keys. Solution to this problem requires a traitor tracing scheme to trace at least one traitor from the coalition of users. Traitor tracing is necessary to provide a legal proof for cheating. Traitor tracing methods were first proposed by Chor, Fiat and Naor [4], and traitor tracing techniques using combinatorial approach were proposed by Stinson and Wei [13], Safavi-Naini and Wang [10], and Ruj and Roy [7]. It is worth noting that traitor-tracing schemes are a special subclass of many-to-one functions, and not all such functions allow for tracing traitors within colluding groups.

1.2 Motivation for this paper

We consider the scenario of video or image broadcast on a network where the media is encrypted for security, and can be decrypted only by the authorized users who possess certain key-sets pre-distributed by the authentic distributor of the media. This warrants for a many-to-one function with general access structure in case of the key pre-distribution. The broadcaster/distributor gives a set of keys to authorized receivers, which maps to a particular final decryption key, and a 'black-box' decoder decrypts the media using this decryption key. We call the domain of the many-to-one function *qualified sets*, which are to be acquired by authorized users.

Dealing with the encryption and decryption of video and image media becomes conceptually easier (and quite efficient) if the keys used in the process are also of a similar format. If images are used as keys, the process of encryption or decryption of digital images and videos (frame-by-frame) may be performed easily by simple arithmetic operations like XOR and OR; especially if we consider visual secret sharing techniques. Moreover, we noticed that visual secret sharing schemes offer a natural many-to-one function, which may be extended to a general access structure to provide a more flexible form of the function.

Although general access structures based on visual secret sharing have been proposed in the literature, we could find none that addresses the problem of traitor tracing in practical scenarios of video broadcast. However, there exist efficient general access structures based on combinatorial block designs to achieve traitor tracing in practice. We attempt at combining the two apparently disjoint ideas to form a general access structure based on visual secret sharing that

would as well provide an efficient mechanism for traitor tracing in practical cases of broadcast of image and videos. A concrete practical instantiation of our motivation for this paper is as follows.

Practical instantiation of our motivation: Consider a video broadcaster who wants only authorized users to access the video. To achieve this, he encrypts the video frame-by-frame using the encryption key – an image similar to a single frame of the video – and broadcasts the resulting data to the network. Decryption by an authorized user will have to use the pre-distributed key-set provided by the broadcaster well in advance. We require the following properties for the scheme.

1. General access structure for the pre-distributed authorized key-sets.
2. Traitor traceable mechanism built within the key-set pre-distribution.
3. Conceptually simple implementation of encryption-decryption routine.

1.3 Contribution of this paper

We propose a visual secret sharing based traitor traceable key pre-distribution scheme which distributes sets of shares to each user, such that, on combining the shares of each authorized user, the final decryption key is recovered in form of an image. The desired properties can be achieved using VSS scheme which distributes the personal sets of keys in form of images for easy media operation during encryption and decryption. Traitor tracing is possible in cases of collusion through the use of a traitor traceable key pre-distribution scheme, as proposed in [7], where the proposed general access structure arising from PBIBD may be implemented through visual secret sharing using random grids, as in [15].

The advantage of our scheme is that if any set formed by collusion is not from any of the authorized user key-sets, then the combination of keys of this set will not yield the secret image. In other words we have a many-to-one function whose domain is strictly the set of authorized user key-sets and range is the secret image. This reduces the chances for collusion, and even if there is any, the broadcaster can trace the traitors quite efficiently.

Practical instantiation of our contribution: In line with our motivating example, as earlier, our proposal may be practically applied in case of video broadcast. Our proposal achieves the following desirable properties.

1. PBIBD based general access structure for traitor-tracing.
2. General access structure based on visual secret sharing.
3. Easy media encryption-decryption using images as shares.

Roadmap. In Section 2, we discuss general access control schemes based on visual secret sharing. Later in Section 3, we propose our scheme for VSS-based traitor tracing and key pre-distribution, describe its details, and analyze the efficiency of our proposal. Finally, Section 4 concludes the paper with a summarized discussion of our proposal, and future scope for research in this direction.

2 VSS based Access Control

In this section, we discuss some aspects regarding VSS based many-to-one functions and access control designs. Based on Naor and Shamir's [6] VSS schemes, several related works have been published over the last two decades. The VSS schemes for general access structure proposed in [13, 15] are noteworthy.

VSS based many-to-one functions. Naor and Shamir [6] proposed a (k, n)-VSS. This scheme is one possible realization of many-to-one functions using VSS, where any subset of size k, taken from the main key-pool, successfully maps to a single image; the 'secret image' of the scheme. The domain of the function is the set of all subsets of size k, and the range is a unique secret image.

Contrary to accepting all subsets of size k in the domain of the many-to-one function, certain access control applications may require to allow only certain subsets (of varying sizes) to map to the secret image. This requires a general access control structure using VSS; a brief overview is as follows.

2.1 General access structure using conventional VSS

Let $P = \{1, 2, \ldots, n\}$ be a set of participants (users) in a VSS scheme and let 2^P denote the power set (set of all subsets) of P. Let $\Gamma_{Qual} \subseteq 2^P$ be the *qualified sets* of the scheme, and $\Gamma_{Forb} \subseteq 2^P$ be the *forbidden sets*. Then the pair $(\Gamma_{Qual}, \Gamma_{Forb})$ is called the access structure of this VSS scheme [13].

In conventional VSS, a secret pixel is encoded into n shared pixels. Each shared pixel contains m black and white sub-pixels, where m is called the pixel expansion of the scheme. The construction can be described by an $n \times m$ boolean matrix $S = [s_{ij}]$, where $s_{ij} = 1$ if and only if the j-th sub-pixel in the i-th share is black. The grey level of each pixel of the final image, obtained by stacking the shares $i_1, i_2, \ldots, i_s$, is interpreted by the human eye as black or white in accordance with the contrast of the pixel. The contrast is proportional to the hamming weight $H(V)$ of the m-dimensional vector V obtained by a logical OR of the row vectors $r_{i_1}, r_{i_2}, \ldots, r_{i_s}$ from the matrix S, associated with the shares given out to the participants $i_1, i_2, \ldots, i_s$, respectively.

Let $(\Gamma_{Qual}, \Gamma_{Forb})$ be an access structure defined on a set of n participants. A $(\Gamma_{Qual}, \Gamma_{Forb})$-VSS scheme with pixel expansion m, relative difference $\alpha(m)$, and set of thresholds $\{(X, t_X)\}_{X \in \Gamma_{Qual}}$ is realized using the two collections of $n \times m$ boolean matrices C_0 and C_1 if the following two conditions are satisfied.

1. If $X = \{i_1, i_2, \ldots, i_p\} \in \Gamma_{Qual}$, then the V-vector produced by logical OR of the rows $r_{i_1}, r_{i_2}, \ldots, r_{i_p}$ meets $H(V) \leq t_X - \alpha(m) \cdot m$ for any $M \in C_0$, whereas it results in $H(V) \geq t_X$ for any $M \in C_1$.
2. If $X = \{i_1, i_2, \ldots, i_q\} \in \Gamma_{Forb}$, the two collections of $q \times m$ matrices D_0, D_1 obtained by restricting the $n \times m$ matrices C_0, C_1 to rows $r_{i_1}, r_{i_2}, \ldots, r_{i_q}$, respectively, are indistinguishable from one another.

The first condition is to ensure that any qualified set $Q \in \Gamma_{Qual}$ recovers the 'secret image' whereas the second condition is to ensure that the recovery of the secret image in this scheme is immune against any forbidden set $F \in \Gamma_{Forb}$.

2.2 General access structure using VSS based on random grids

An alternative method of visual secret sharing was proposed by Kafri and Keren [5] in 1987. This was a $(2, 2)$-VSS based on random grids, which eliminates the problems of pixel expansion and shape distortion in practice. The scheme was extended to an (n, n)-VSS scheme in 2008 by Chen and Tsao [3], and was further extended by Wu and Sun [15] in 2012 to a VSS based on general access structure.

Construction of shares: In the VSS scheme based on general access structure, proposed in [15], n binary shares $R_1, \ldots, R_n$ are constructed from an $M \times N$ secret image S for a pre-specified access structure $(\Gamma_{Qual}, \Gamma_{Forb})$. It defines the set of all minimal qualified sets for the VSS as

$$\Gamma_0 = \{Q \in \Gamma_{Qual} \ : \ Q' \notin \Gamma_{Qual} \ \forall \ Q' \subset Q\},$$

and for each pixel $(i, j) \in S$, a minimal qualified set $Q = i_1, \ldots, i_p \in \Gamma_0$ is selected at random $(p \leq n)$. For each of the first $p - 1$ shares in Q, the pixel (i, j) is randomly assigned a value 0 (for white) or 1 (for black).

Thereafter the pixel (i, j) of the p-th share is obtained by performing a logical XOR on the (i, j)-th pixel of each of the $p - 1$ shares and the (i, j)-th pixel of the secret image:

$$r_{i_p}(i, j) = S(i, j) \oplus r_{i_1}(i, j) \oplus r_{i_2}(i, j) \oplus \cdots \oplus r_{i_{p-1}}(i, j) \qquad \forall \text{ pixel } (i, j) \in S.$$

For the remaining $n - p$ binary shares not in the minimal qualified set Q, the (i, j)-th pixel is randomly assigned a value 0 (for white) or 1 (for black). The process is repeated for each pixel in the secret image.

Proof of correctness: As a result of the above construction, participants belonging to any qualified set can reconstruct the image by performing a logical XOR operation on the corresponding pixels of the shares in the qualified sets; however, stacking of the shares in any of the forbidden sets does not provide any information of the secret image. This has theoretically been proved in the original proposal [15].

Contrast of the recovered image: The original scheme [15] also proves a quantitative measure of the contrast of the recovered image. Let $k = |\Gamma_0|$ be the number of minimal qualified sets. $X = \{i_1, i_2, ..., i_p\} \in \Gamma_{Qual}$ be a qualified set whose elements can form d minimal qualified sets $(1 \leq d \leq k)$. Then the contrast of the image by stacking shares $R_{i_1}, ..., R_{i_p}$ from the qualified set X, as proved in [15, Theorem 2], is as follows:

$$\alpha = \frac{2d}{(2^p + 1)k - d} \tag{1}$$

Thus, it is evident that as the number of qualified sets increases, the contrast of the recovered image obtained from each qualified set decreases considerably. In our proposal, we try to rectify this issue and provide perfect contrast.

3 Our proposal – VSS-based traitor-tracing

Traitor tracing schemes were first proposed by Chor, Fiat and Naor [4]. The key distributor distributes a set of keys $K = \{K_1, K_2, ..., K_b\}$ to b users, where the base set or key-pool consists of v keys. Each user U is given a set of k keys which form the personal user key-set, and we denote this set by $P(U)$. If user U is authorized by the broadcaster to access the encrypted content, then $P(U) \in \Gamma_{Qual}$. It is assumed that a set $C = \{K_1, K_2, ..., K_c\}$ of c malicious users (traitors) may collude and construct a pirate decoder F, not originally (or not yet) authorized by the broadcaster, such that

$$F \subset \bigcup_{U \in C} P(U), \quad F \in \Gamma_{Qual}, \quad \text{and} \quad |F| = k.$$

The goal of the broadcaster is to assign the keys to the users in such a way that once the pirate decoder F is found and the keys are examined, at least one traitor within the colluding set C will be identified.

Traitor tracing schemes based on PBIBD. The first PBIBD based traitor tracing scheme was proposed by Ruj and Roy [7]. A partially balanced incomplete block design with m associate classes [14], denoted by PBIBD(m), is a design on a v-set X, with b blocks each of size k and with each element of X being repeated r times, such that if there is an association scheme with m classes defined on X where, two elements x and y are i-th associates ($1 \leq i \leq m$), then they occur together in λ_i blocks. Such a design is denoted by $PB[k, \lambda_1, \lambda_2, ..., \lambda_m; v]$. One may refer to [14] for a comprehensive mathematical treatment of PBIBD and similar combinatorial designs.

The proposal of [7] achieves a PBIBD-based key distribution scheme for N users, with the size of each personal set $|P(U)| \sim O(N^{1/2})$, and a mechanism to trace a collusion of at most $|C| \sim O(N^{1/4})$ traitors. In this paper, we try to combine this idea with VSS-based general access structures, while keeping the bounds on personal key-set and collusion detection the same as [7].

3.1 Combination of VSS and PBIBD

In this section, we describe our proposal of mapping of PBIBD scheme to a general access structure based on VSS. The PBIBD-based scheme distributes a set of k keys based on the number of users b. In our proposal, each key in the PBIBD scheme is mapped to a particular share obtained from a secret image in the VSS scheme. Thus, the concept of many-to-one function realized by the PBIBD approach is naturally extended to the VSS scheme.

The set of shares in the VSS scheme corresponding to the qualified user key-set, when combined together, results in the original secret image. However, when the set of shares corresponding to the set of keys other than the qualified user key-set are combined together, the secret image is not recovered. Thus, the set of shares corresponding to the user key-set in the PBIBD scheme forms the access

structure of the VSS scheme, where the set of shares given to authorized users form the qualified sets, and the other sets automatically become forbidden. To achieve this map from PBIBD based key distribution scheme to VSS, we need a general access structure based VSS, with a mechanism to map the set of user key-sets in the PBIBD scheme to the qualified sets for generating shares.

Choosing the appropriate VSS construction. In the general access structure based VSS scheme provided in [13], the construction of shares is based on the minimal forbidden sets, and not on the minimal qualified sets. The complexity of determining all the forbidden sets, given a small number of qualified sets, increases exponentially with the increase in the number of shares. Thus the complexity of the determining all the shares using the construction in [13] is impractical if only the qualified sets are specified. However, that is the situation in our case, as PBIBD based traitor-tracing generally prescribes only the qualified sets in the access structure.

In order to reduce the complexity of the construction of shares, we consider the random grid based VSS scheme, proposed in [15], as this construction considers the minimal qualified sets for designing the access structure. In practical applications, the number of users U and hence the number of personal user key-sets $P(U)$ is usually high. But it is evident from Eqn. (1) that as the number of qualified sets increases, the contrast of the image obtained by combining all the shares in a qualified set decreases exponentially, which results in poor 'visual' recovery of the secret image. To overcome this problem we introduce a new scheme based on random grids in the next section.

3.2 Design of the proposed scheme

As discussed earlier, we propose a key-distribution scheme based on VSS and general access structure, along the following line of construction.

Step 1 – PBIBD based general access structure for traitor-tracing.
Step 2 – General access structure based on visual secret sharing.
Step 3 – Visual secret sharing using random grid based approach.

Designing the access control structure. Assuming b users in the scheme, each user is assigned k keys as his/her user key-set, where $k \sim O(b^{1/2})$. The PBIBD based construction with $PB[k, \lambda_1, \lambda_2, ..., \lambda_m; v]$ gives us the set of all personal user key-sets $Z = \{P(B_1), P(B_2), ...P(B_b)\}$ where $P(B_i)$ is the user key-set of the i-th user U_i. This set Z forms the access structure for the VSS based key pre-distribution scheme.

Let $(\Gamma_{Qual}, \Gamma_{Forb})$ be the access structure, where $\Gamma_{Qual} = Z$, and let Γ_0 be the set of all minimal qualified sets. As no subset of a set $P(B_i) \in Z$ belongs to Z, the minimal qualified set is Z itself, that is, we may write $Z = \Gamma_{Qual} = \Gamma_0$.

Designing the VSS scheme for the access control structure. Now we design a random grid based VSS scheme which provides access control as well as good contrast of the recovered image irrespective of the size of the minimal qualified set $\Gamma_0 = Z$.

In [15], for a given pixel in the secret image, a qualified set is selected at random and then the corresponding pixels of the shares are calculated. Hence, a pixel in the recovered image will have the same value as that in the secret image only when the shares of that particular random qualified set are stacked together. However, as discussed earlier, we know that this results in poor contrast.

Design for high contrast image recovery: We consider all shares in the qualified sets of Z simultaneously for a particular pixel in the secret image. For a pixel $(i, j) \in S$ of the secret image, we calculate the corresponding (i, j)-th pixel of all the shares in the first qualified set $P(B_1) \in Z$; followed by that in all shares of the second qualified set $P(B_2) \in Z$, and the process is repeated for all the qualified sets $P(B_i) \in Z$.

Dummy share: Now it may be so that for a qualified set $P(B_j) \in Z$, all the shares within have already been computed as a part of the previously considered qualified sets $P(B_1), P(B_2), \ldots, P(B_{j-1}) \in Z$. In such a scenario, we add a dummy share to $P(B_j)$ such that, when all the shares of the qualified set $P(B_j)$ along with the dummy share are stacked together, the secret image is recovered with a very high contrast. Thus with a little overhead (at most one per user) of dummy shares, we achieve near-perfect contrast in the recovered image, which is a significant improvement over [15].

Designing the traitor-tracing mechanism. To achieve the traitor-tracing feature, we have to make sure that the traitor tracing technique based on PBIBD, as proposed in [7], is still applicable to our new modified sets of secret user keys.

Security analysis: There are two possible scenarios.

- *Case 1.* The collusion is such that a colluded set consists one of the dummy keys that we have introduced in the user key-sets. If this happens, the traitor(s) contributing the dummy key will be trivially exposed, as all dummy keys are randomly and uniquely chosen for each user.
- *Case 2.* The other possibility is that there are no dummy keys in the pirate user key-set resulting from the collusion. This reduces the problem to traitor tracing without considering the dummy keys at all; precisely the condition where the traitor tracing techniques of [7] are applicable.

Thus, we conclude that our modified scheme satisfies the requirement for traitor tracing, in a way similar to that proposed in the PBIBD construction of [7].

Suppose the pirate user key-set formed through collusion is present in the set of qualified sets Γ_{Qual}, but has not been distributed to an authorized user yet. This is possible as some extra qualified sets are expected to be kept in reserve for the scope of expansion in user base. In such a case, the traitors can be traced as in [7], within a bound of $O(N^{1/4})$ traitors, where N is the number of users.

Advantage over [7]: If two or more users collude and form a pirate set of keys which is not present in the set of qualified sets Γ_{Qual}, then the access control structure achieved through our random grid based VSS assures that stacking the shares present in this pirate set does not reveal the secret image. In other words, if the pirate user key-set is not in Γ_{Qual}, our scheme assures that this key-set is not present in the set of minimal qualified sets $\Gamma_0 = Z$ of our design as well. This was not guaranteed in the scheme proposed by Ruj and Roy [7].

Algorithm for the proposed scheme. The discussion above confirms that we have created a VSS based on random grid methods to address the problem of general access structure based key pre-distribution with traitor tracing. Algorithm 1 illustrates our design.

Input: Number of users b, size of user key-set k and an $M \times N$ binary image Im.
Output: b user key-sets $Z = \{PS_1, PS_2, \ldots, PS_b\}$, v binary shares $R_1, \ldots, R_n$ corresponding to v keys of the key-pool, and d dummy shares.

Generate a PBIBD framework PB$[k, \lambda_1, \lambda_2, ..., \lambda_m; v]$;
Generate the user key-sets $Z = \{P(B_1), P(B_2), \ldots, P(B_b)\}$ using PB;
Construct the access structure $(\Gamma_{Qual}, \Gamma_{Forb})$, where $\Gamma_{Qual} = \Gamma_0 = Z$;

for *each minimal qualified set* $P(B_i) \in Z = \Gamma_0$ **do**
  $PS_i = P(B_i)$;
  if *last share in* PS_i *is already visited* **then**
    if *there exists a share* $X \in PS_i$, *which is not already visited* **then**
      Swap last share with X;
    end
    else
      Create a dummy share d of size $M \times N$;
      Expand the key-set $PS_i = P(B_i) \cup d$;
    end
  end
  for *each share* $S \in PS_i$ **do**
    for *each pixel* $(i, j) \in S$ **do**
      if S *is not visited and it is not the last share in* PS_i **then**
        $S(i, j) = \text{RandomBit}(0, 1)$;
      end
      if S *is the last share in* PS_i **then**
        $S(i, j) = \bigoplus_{R \in PS_i, R \neq S} R(i, j) \ \oplus \ Im(i, j)$;
      end
    end
  end
end

Algorithm 1: The proposed VSS scheme based on a traitor-traceable access structure generated by PBIBD.

3.3 Implementation of the proposed scheme

In this section we demonstrate an example implementation of our proposed key pre-distribution scheme. We consider the PBIBD construction $PB[10, 1, 0; 82]$ from [7], with total number of users $N = 41$. Each user is given a set of $k = 10$ keys. For completeness, we replicate the construction from [7], as follows.

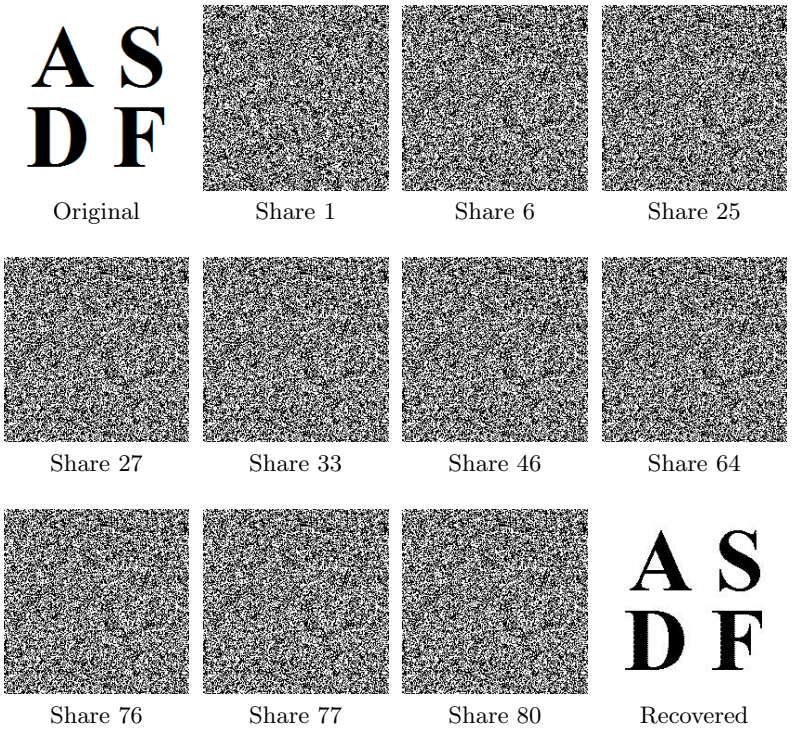
$P(B_1) = \{1, 6, 25, 27, 33, 46, 64, 76, 77, 80\}$
$P(B_2) = \{2, 7, 26, 28, 34, 47, 65, 77, 78, 81\}$
$P(B_3) = \{3, 8, 27, 29, 35, 48, 66, 78, 79, 82\}$
$P(B_4) = \{4, 9, 28, 30, 36, 49, 67, 79, 80, 42\}$
$P(B_5) = \{5, 10, 29, 31, 37, 50, 68, 80, 81, 43\}$
$P(B_6) = \{6, 11, 30, 32, 38, 51, 69, 81, 82, 44\}$
$P(B_7) = \{7, 12, 31, 33, 39, 52, 70, 82, 42, 45\}$
$P(B_8) = \{8, 13, 32, 34, 40, 53, 71, 42, 43, 46\}$
$P(B_9) = \{9, 14, 33, 35, 41, 54, 72, 43, 44, 47\}$
$P(B_{10}) = \{10, 15, 34, 36, 1, 55, 73, 44, 45, 48\}$
$P(B_{11}) = \{11, 16, 35, 37, 2, 56, 74, 45, 46, 49\}$
$P(B_{12}) = \{12, 17, 36, 38, 3, 57, 75, 46, 47, 50\}$
$P(B_{13}) = \{13, 18, 37, 39, 4, 58, 76, 47, 48, 51\}$
$P(B_{14}) = \{14, 19, 38, 40, 5, 59, 77, 48, 49, 52\}$
$P(B_{15}) = \{15, 20, 39, 41, 6, 60, 78, 49, 50, 53\}$
$P(B_{16}) = \{16, 21, 40, 1, 7, 61, 79, 50, 51, 54\}$
$P(B_{17}) = \{17, 22, 41, 2, 8, 62, 80, 51, 52, 55\}$
$P(B_{18}) = \{18, 23, 1, 3, 9, 63, 81, 52, 53, 56\}$
$P(B_{19}) = \{19, 24, 2, 4, 10, 64, 82, 53, 54, 57\}$
$P(B_{20}) = \{20, 25, 3, 5, 11, 65, 42, 54, 55, 58\}$
$P(B_{21}) = \{21, 26, 4, 6, 12, 66, 43, 55, 56, 59\}$
$P(B_{22}) = \{22, 27, 5, 7, 13, 67, 44, 56, 57, 60\}$
$P(B_{23}) = \{23, 28, 6, 8, 14, 68, 45, 57, 58, 61\}$
$P(B_{24}) = \{24, 29, 7, 9, 15, 69, 46, 58, 59, 62\}$
$P(B_{25}) = \{25, 30, 8, 10, 16, 70, 47, 59, 60, 63\}$
$P(B_{26}) = \{26, 31, 9, 11, 17, 71, 48, 60, 61, 64\}$
$P(B_{27}) = \{27, 32, 10, 12, 18, 72, 49, 61, 62, 65\}$
$P(B_{28}) = \{28, 33, 11, 13, 19, 73, 50, 62, 63, 66\}$
$P(B_{29}) = \{29, 34, 12, 14, 20, 74, 51, 63, 64, 67\}$
$P(B_{30}) = \{30, 35, 13, 15, 21, 75, 52, 64, 65, 68\}$
$P(B_{31}) = \{31, 36, 14, 16, 22, 76, 53, 65, 66, 69\}$
$P(B_{32}) = \{32, 37, 15, 17, 23, 77, 54, 66, 67, 70\}$
$P(B_{33}) = \{33, 38, 16, 18, 24, 78, 55, 67, 68, 71\}$
$P(B_{34}) = \{34, 39, 17, 19, 25, 79, 56, 68, 69, 72\}$
$P(B_{35}) = \{35, 40, 18, 20, 26, 80, 57, 69, 70, 73\}$
$P(B_{36}) = \{36, 41, 19, 21, 27, 81, 58, 70, 71, 74\}$
$P(B_{37}) = \{37, 1, 20, 22, 28, 82, 59, 71, 72, 75\}$
$P(B_{38}) = \{38, 2, 21, 23, 29, 42, 60, 72, 73, 76\}$
$P(B_{39}) = \{39, 3, 22, 24, 30, 43, 61, 73, 74, 77\}$
$P(B_{40}) = \{40, 4, 23, 25, 31, 44, 62, 74, 75, 78\}$
$P(B_{41}) = \{41, 5, 24, 26, 32, 45, 63, 75, 76, 79\}$

User set-keys in the proposed scheme: In the above PBIBD construction, there are 82 unique keys. An additional 22 keys are appended as dummy shares (to certain sets) after applying Algorithm 1. The key-sets after applying Algorithm 1 are as follows, where the user key-sets from PS_{20} to PS_{41} have been appended with one dummy share each; numbered 83 to 104 in the presentation.

$PS_1 = \{1, 6, 25, 27, 33, 46, 64, 76, 77, 80\}$
$PS_2 = \{2, 7, 26, 28, 34, 47, 65, 77, 78, 81\}$
$PS_3 = \{3, 8, 27, 29, 35, 48, 66, 78, 79, 82\}$
$PS_4 = \{4, 9, 28, 30, 36, 49, 67, 79, 80, 42\}$
$PS_5 = \{5, 10, 29, 31, 37, 50, 68, 80, 81, 43\}$
$PS_6 = \{6, 11, 30, 32, 38, 51, 69, 81, 82, 44\}$
$PS_7 = \{7, 12, 31, 33, 39, 52, 70, 82, 42, 45\}$
$PS_8 = \{8, 46, 32, 34, 40, 53, 71, 42, 43, 13\}$
$PS_9 = \{9, 47, 33, 35, 41, 54, 72, 43, 44, 14\}$
$PS_{10} = \{10, 48, 34, 36, 1, 55, 73, 44, 45, 15\}$
$PS_{11} = \{11, 49, 35, 37, 2, 56, 74, 45, 46, 16\}$
$PS_{12} = \{12, 50, 36, 38, 3, 57, 75, 46, 47, 17\}$
$PS_{13} = \{13, 51, 37, 39, 4, 58, 76, 47, 48, 18\}$
$PS_{14} = \{14, 52, 38, 40, 5, 59, 77, 48, 49, 19\}$
$PS_{15} = \{15, 53, 39, 41, 6, 60, 78, 49, 50, 20\}$
$PS_{16} = \{16, 54, 40, 1, 7, 61, 79, 50, 51, 21\}$
$PS_{17} = \{17, 55, 41, 2, 8, 62, 80, 51, 52, 22\}$
$PS_{18} = \{18, 56, 1, 3, 9, 63, 81, 52, 53, 23\}$
$PS_{19} = \{19, 57, 2, 4, 10, 64, 82, 53, 54, 24\}$
$PS_{20} = \{20, 25, 3, 5, 11, 65, 42, 54, 55, 58, 83\}$
$PS_{21} = \{21, 26, 4, 6, 12, 66, 43, 55, 56, 59, 84\}$
$PS_{22} = \{22, 27, 5, 7, 13, 67, 44, 56, 57, 60, 85\}$
$PS_{23} = \{23, 28, 6, 8, 14, 68, 45, 57, 58, 61, 86\}$
$PS_{24} = \{24, 29, 7, 9, 15, 69, 46, 58, 59, 62, 87\}$
$PS_{25} = \{25, 30, 8, 10, 16, 70, 47, 59, 60, 63, 88\}$
$PS_{26} = \{26, 31, 9, 11, 17, 71, 48, 60, 61, 64, 89\}$
$PS_{27} = \{27, 32, 10, 12, 18, 72, 49, 61, 62, 65, 90\}$
$PS_{28} = \{28, 33, 11, 13, 19, 73, 50, 62, 63, 66, 91\}$
$PS_{29} = \{29, 34, 12, 14, 20, 74, 51, 63, 64, 67, 92\}$
$PS_{30} = \{30, 35, 13, 15, 21, 75, 52, 64, 65, 68, 93\}$
$PS_{31} = \{31, 36, 14, 16, 22, 76, 53, 65, 66, 69, 94\}$
$PS_{32} = \{32, 37, 15, 17, 23, 77, 54, 66, 67, 70, 95\}$
$PS_{33} = \{33, 38, 16, 18, 24, 78, 55, 67, 68, 71, 96\}$
$PS_{34} = \{34, 39, 17, 19, 25, 79, 56, 68, 69, 72, 97\}$
$PS_{35} = \{35, 40, 18, 20, 26, 80, 57, 69, 70, 73, 98\}$
$PS_{36} = \{36, 41, 19, 21, 27, 81, 58, 70, 71, 74, 99\}$
$PS_{37} = \{37, 1, 20, 22, 28, 82, 59, 71, 72, 75, 100\}$
$PS_{38} = \{38, 2, 21, 23, 29, 42, 60, 72, 73, 76, 101\}$
$PS_{39} = \{39, 3, 22, 24, 30, 43, 61, 73, 74, 77, 102\}$
$PS_{40} = \{40, 4, 23, 25, 31, 44, 62, 74, 75, 78, 103\}$
$PS_{41} = \{41, 5, 24, 26, 32, 45, 63, 75, 76, 79, 104\}$

Example verification for correctness. We stack the shares of the qualified set $P(B_1) = \{1, 6, 25, 27, 33, 46, 64, 76, 77, 80\}$, as shown in Table 3.3. It is clear that contrast of the recovered image is same as that of the original secret image.

Table 1. Original Image, Recovered Image and the shares of Set PS_1.



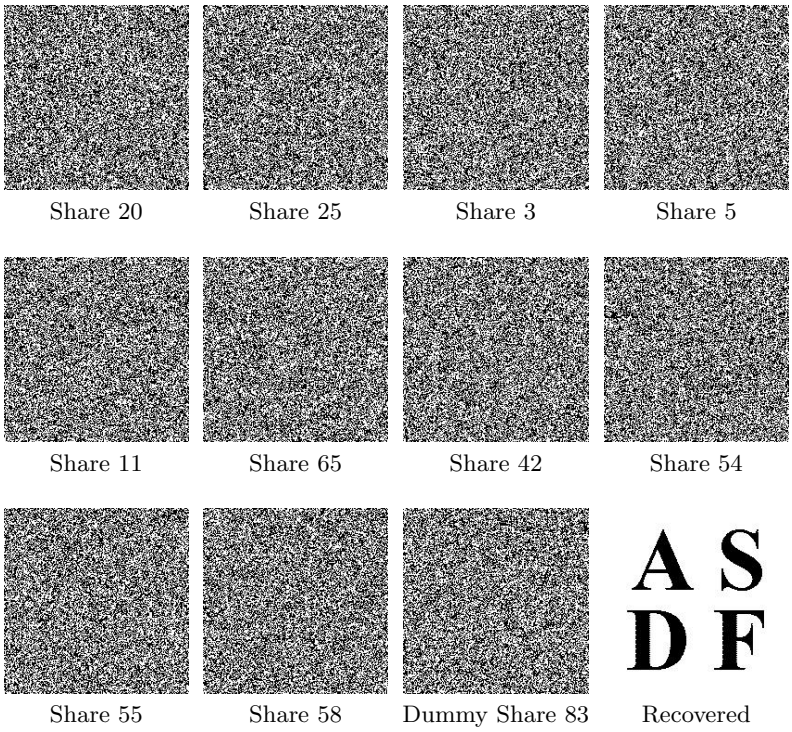
For the user key-set $P(B_{20}) = \{20, 25, 3, 5, 11, 65, 42, 54, 55, 58\}$ from the PBIBD construction of [7], notice that as all the shares have been calculated in the previous sets $P(B_1), \ldots, P(B_{19})$. Thus, Algorithm 1 introduces a dummy share 'Share 83' in $P(B_{20})$; turning the new user key-set into

$$PS_{20} = \{20, 25, 3, 5, 11, 65, 42, 54, 55, 58, 83\}.$$

Stacking of the shares in this set, along with the dummy share 83, also reveals the original image with full contrast, as expected from our scheme. The results are shown in Table 3.3.

Collusion resistance: Let users U_1 and U_2 collude to form a pirate key-set $F = \{1, 2, 6, 7, 9, 28, 62, 69, 76, 77\}$, as in the example of [7]. Note that F is not present in the set of minimal qualified sets $\Gamma_0 = Z$ of our design. Hence, stacking the shares of F would not reveal the secret image in our access control structure.

Table 2. Shares of set PS_{20} and Recovered Image



Efficiency of the proposed scheme. The proposed scheme was implemented on a 2.3GHz Intel processor running GNU/Linux. Table 3 illustrates the efficiency of our proposal through experiments with different number of users.

Table 3. Experimental results to check the efficiency of the proposed scheme.

Number of Users	Size of Key-pool	Number of Dummy keys	Time in seconds for	
			Share generation	Share stacking
3	4	1	1.210 sec	0.306 sec
12	40	3	10.830 sec	0.427 sec
16	17	6	1.908 sec	0.110 sec
25	30	13	12.600 sec	0.314 sec
41	82	22	33.470 sec	0.400 sec

The only extra overhead in our case, compared to the proposal of [7], originates due to the dummy shares. From Table 3 it is clear that the number of dummy shares is reasonably less than the number of users in each case. For N users, each user gets k keys, where $k \sim O(N^{1/2})$, and a collusion of up to $O(N^{1/4})$ traitors can be traced. Compared to [7], we do not lose on these bounds.

3.4 Advantages over existing schemes

To the best of our knowledge, there exists no scheme for using VSS-based general access structures in traitor tracing. However, there exist several VSS-based general access structures and PBIBD-based traitor tracing schemes, independently.

Comparison with VSS-based general access structures: In general access structures based on VSS, the contrast of the final recovered image is quite poor. We propose the use of *dummy shares* in VSS based on random grids to solve this problem. Our proposal achieves full contrast in image recovery, and is better in this sense from the schemes proposed in [3, 13, 15]. In addition, the use of random grids eliminate the problem of pixel expansion and shape distortion.

Comparison with PBIBD-based traitor tracing schemes: We borrow the basic ideas of traitor-tracing from the PBIBD-based scheme of [7], and combine it with the aforesaid VSS-based general access structure. We show that even with the inclusion of dummy shares in the key-sets, our proposal matches the traitor-tracing bounds of [7], where each user gets a personal key-set of size $O(N^{1/2})$, and a collusion of size up to $O(N^{1/4})$ can be traced.

We improve upon the idea of [7] in one major aspect. In case of [7], the colluding traitors could generate a new key-set which did not belong to the preset user-sets, but still qualifies as a valid key. However in our proposal, if the colluded set of shares did not belong to the preset qualified user-sets, then it will never be accepted as a valid key. Thus, compared to [7], we reduce the possibility of collusions to a considerable extent.

4 Conclusion

We have constructed a key distribution scheme which is based on VSS. We have used general access structure based VSS for key distribution, where the access structure is defined by PBIBD blocks to reduce the size of the main key pool. We have used the concept of random grids in our VSS scheme which eliminates the problem of pixel expansion and shape distortion of the recovered image.

In addition, we have introduced the concept of dummy shares in the PBIBD construction to completely recover the secret image in VSS with a contrast same as that of the original. Using VSS as the many-to-one function, our key pre-distribution scheme reduces the number of possible collusions, and provides a mechanism alike [7] to trace collusions up to $O(N^{1/4})$ where N is the number of total users in the design.

Our proposed scheme is particularly suitable for applications like video and image broadcast, where the recovered VSS image can directly be applied as a key towards data encryption by simple logical operations between the similar data structures. In such cases, our proposal also guarantees low collusion and traitor tracing for the distributor based on the PBIBD access control structure; once again, particularly suitable for commercial broadcast applications.

Future scope: One may try to design new schemes in this line to eliminate the necessity of dummy shares. A practical implementation of the proposed scheme for secure video broadcast would also be an appealing direction.

Acknowledgments. The authors would like to thank the anonymous reviewers for their valuable comments that helped improve the quality of the paper. Sheethal Kumar, Jaipal Pai B J and Vigesh R Ungrapalli would also like to thank Prof. Bimal Roy, Director, Indian Statistical Institute, for supporting them during the tenure of the project at ISI Kolkata under a Summer Internship grant from Microsoft Research India, and for sparking their motivation towards this project through an instructive talk on combinatorial designs in cryptology.

References

1. G. Ateniese, C. Blundo, A. D. Santis, and D. R. Stinson. Visual cryptography for general access structures. *Inf. Comput.*, 129(2):86–106, 1996.
2. G. Blakley. Safeguarding cryptographic keys. In *Proceedings of the 1979 AFIPS National Computer Conference*, pages 313–317. AFIPS Press, 1979.
3. T.-H. Chen and K.-H. Tsao. Visual secret sharing by random grids revisited. *Pattern Recognition*, 42(9):2203–2217, 2009.
4. B. Chor, A. Fiat, and M. Naor. Tracing traitors. In Y. Desmedt, editor, *CRYPTO*, volume 839 of *Lecture Notes in Computer Science*, pages 257–270. Springer, 1994.
5. O. Kafri and E. Keren. Encryption of pictures and shapes by random grids. *Opt. Lett.*, 12(6):377–379, 1987.
6. M. Naor and A. Shamir. Visual cryptography. In A. D. Santis, editor, *EUROCRYPT*, volume 950 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 1994.
7. S. Ruj and B. K. Roy. Key distribution schemes using combinatorial designs to identify all traitors. *Congressus Numerantium*, 193:195–214, 2008.
8. S. Ruj and B. K. Roy. Key predistribution using combinatorial designs for grid-group deployment scheme in wireless sensor networks. *TOSN*, 6(1), 2009.
9. S. Ruj and B. K. Roy. Key pre-distribution using partially balanced designs in wireless sensor networks. *IJHPCN*, 7(1):19–28, 2011.
10. R. Safavi-Naini and Y. Wang. A combinatorial approach to asymmetric traitor tracing. In D.-Z. Du, P. Eades, V. Estivill-Castro, X. Lin, and A. Sharma, editors, *COCOON*, volume 1858 of *Lecture Notes in Computer Science*, pages 416–425. Springer, 2000.
11. R. Safavi-Naini and Y. Wang. Sequential traitor tracing. In M. Bellare, editor, *CRYPTO*, volume 1880 of *Lecture Notes in Computer Science*, pages 316–332. Springer, 2000.
12. A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.
13. D. R. Stinson and R. Wei. Combinatorial properties and constructions of traceability schemes and frameproof codes. *SIAM J. Discrete Math.*, 11(1):41–53, 1998.
14. A. P. Street and D. J. Street. *Combinatorics of experimental design.* Oxford : Clarendon Press ; New York : Oxford University Press, 1987.
15. X. Wu and W. Sun. Visual secret sharing for general access structures by random grids. *IET Information Security*, 6(4):299–309, 2012.