A RAMP CODE FOR FINE-GRAINED ACCESS CONTROL

Kannan Karthik[1]

[1]Department of Electronics and Electrical Engineering,
Indian Institute of Technology Guwahati
Guwahati, Assam 781039
k.karthik@iitg.ernet.in

ABSTRACT

Threshold ramp secret sharing schemes are designed so that (i) certain subsets of shares have no information about the secret, (ii) some subsets have partial information about the secret and (iii) some subsets have complete information to recover the secret. However most of the ramp schemes in present literature do not control the leakage of information in partial access sets, due to which the information acquired by these sets is devoid of structure and not useful for fine-grained access control. Through a non-perfect secret sharing scheme called MIX-SPLIT, an encoding methodology for controlling the leakage in partial access sets is proposed and this is used for fine-grained access to binary strings. The ramp code generated using MIX-SPLIT requires a much smaller share size of $O(n)$, as compared to Shamir's ramp adaptation which incurs a share size of atleast $O(n^2)$ for the same multi-access structure. The proposed ramp code is finally applied towards the protection and fine-grained access of industrial design drawings.

KEYWORDS

MIX-SPLIT, Ramp code, Non-perfect secret sharing, Fine-grained access control, Industrial designs

1. INTRODUCTION

Most initial design blueprints of products such as bikes, cars etc., are a byproduct of collaborative effort between several designers often at dispersed locations. Information flow across teams of designers is critical and must be encouraged since changes in one design facet may require subtle changes in another facet but on a need to know basis only to avoid information *leakage or theft*. A few traitors can always share fragments of the early design blueprints with an external party. Hence, the notion of maintaining absolute transparency in the design and development phase, is a big risk to the company.

This problem can be actively mitigated by providing selective access to groups of designers. Different groups of designers should therefore have restricted access to different portions of the overall design. When the number of users falls below a critical mass/number (e.g. a threshold factor K), then no access should be given to any portion of the blueprint. This is necessary to prevent leakages within small groups. However when multiple groups are involved, a larger fraction of the design can be revealed. Hence any practical selective access scheme must drift

Jan Zizka (Eds) : CCSIT, SIPP, AISC, PDCTA - 2013
pp. 67–79, 2013. © CS & IT-CSCP 2013 DOI : 10.5121/csit.2013.3607

from *no access* to *partial but selective access* and under special circumstances to *complete access*, depending on the access rights, group size and designation of the designers. Hence, the size and composition of a coalition of designers will influence exactly which portion of the protected complete design is available for viewing and modification.

These designs can be protected by encryption, but a complete encryption of the entire mass of data renders it inaccessible when only selective information needs to be examined for further analysis. To permit selective access the design data is granularized by partitioning and each disjoint fragment must be encrypted with a different key. This set of keys is then pooled and split into several shares based on pre-decided access policies, which are then distributed amongst the designers. The access framework must ensure that only the right combination of shares produce a set of viable decryption keys which can be used to decrypt different portions of the design simultaneously.

Most conventional secret sharing schemes such as Shamir's polynomial construction [1] and Blakeley's Geometric approach [2] were designed to protect single secrets based on a threshold number of users and can be termed as *(K,n)* threshold schemes. In these schemes a single secret, $\bar{X}$, is split into *n* shares $\{\bar{S}_1, \bar{S}_2, \ldots, \bar{S}_n\}$ and atleast *K* shares are required for perfect reconstruction. However, *K-1* or fewer shares do not have adequate information to reconstruct the secret either in the information theoretic sense (e.g. Shamir's scheme) or in the computational sense (e.g. Blakeley's scheme). Extensions of these schemes towards more general access structures [3] from plain threshold type structures are possible. Unfortunately single secret sharing schemes are not very useful for selective access applications since the system must handle multiple secrets such as several decryption keys.

In the industrial design application, every decryption key is a link to a certain portion of the design which can be viewed by designated groups of designers and this aspect is captured by the access policy. There may be different access policies for different portions of the design. Applying single secret sharing schemes independently to different secrets based on the type of access policy may not be very efficient in terms of storage as the size of the shares given to different users may become very large. The multi-secret optimization problem was studied by Xiao et. al.[4] by realizing multi-access structures using monotone span programs. The authors showed that if the access structures were of a multi-threshold type, then the lowerbound on the share size is based on the sum of the sizes of the participating coalitions. Only then will the constructions remain information theoretically secure.

However if information theoretic secrecy is sacrificed, information storage can be made more efficient. A non-perfect secret sharing scheme in contrast consists of three different types of sets of shares: (i) Access sets which constitute subsets of users who have full information about the secret, (ii) Partial access sets which include sets of users who possess partial information about the secret and (iii) Non-access sets in which users do not have any information about the secret. Examples are *(d,k,n)* ramp schemes of Blakley and Meadows [5]which are an extension of *(K,n)* threshold schemes. If we denote $Size(\bar{S}_i)$ as the size of the share $\bar{S}_i$ in bits and $Size(\bar{X})$ as the secret size, for any perfect secret sharing scheme it holds that $Size(\bar{S}_i) \geq Size(\bar{X})$[6]. On the other hand there exist non-perfect schemes with $Size(\bar{S}_i) < Size(\bar{X})$[5]. Thus one of the advantages of non-perfectness is the smaller share size at the expense of information-theoretic secrecy. Unfortunately most ramp schemes do not control the information leakage occurring in the partial access sets and hence non-perfect schemes have largely remained a subject of purely academic interest. Variations in the form of Shamir's ramp scheme have been proposed for applications such as efficient information dispersal [7] and image splitting [8].

If the information flow and leakage in non-perfect secret sharing schemes can be controlled, several interesting applications can be conceived [9]. The MIX-SPLIT cipher which first originated as a non-perfect multi-secret sharing scheme [10] was found to have a variety of applications when its information leakage characteristics were regulated [9][11]. In this paper we explore the application of this non-perfect scheme towards fine-grained access control through a ramp construction.

The MIX-SPLIT algorithm specific to access control is given in Section 2. The rules for partition visibility for multi-secret access are given in Section 3, while the actual code construction is given in Section 4. The proposed ramp code is examined for its granularity and efficiency (in terms of share size) in Section 5. Finally in Section 6, an industrial design protection application is provided.

2. MIX-SPLIT APPLIED TO FINE-GRAINED ACCESS CONTROL

Consider a sequence of independent and identically distributed (I. I. D.) binary random variables, $\bar{X} = [x_1, x_2, \dots, x_L]$ based on a series of unbiased coin-flip experiments. It is impossible to decompose this block into smaller sequences based on statistical disparities. In an access control application this interesting sequence represents a block of encryption (or decryption) key-strings which are mixed to form $\bar{X}$. If these key strings are created from independent and unbiased coin-flip experiments, then their mixture is guaranteed to take the form of the statistically homogeneous block $\bar{X}$. The overall access control process comprises of the following steps:

Step 1: Partitioning and forming the homogenous block

Let $[\bar{K}_1, \bar{K}_2, \dots, \bar{K}_v]$ be v different L_p-bit key strings. These strings are first concatenated and then shuffled to form a homogeneous block $\bar{X}$ of length $L = L_p \times v$ bits. The shuffling disperses the key-strings over the entire block $\bar{X}$. Let $P = \{1,2,3,\dots,L\}$ be the set of all possible bit-positions within the block $\bar{X}$. When the key-strings disperse, they occupy a distinct group of positions within the homogeneous block. This group forms what we define as a hidden *Partition*. There will be exactly v disjoint and equal length partitions $P_1, P_2, \dots, P_v$ such that $P = P_1 \cup P_2 \cup \dots \cup P_v$. Thus after mixing we are left with the one-to-one map,

$$\bar{K}_j = \bar{X}(P_j) \qquad \dots\dots\dots\dots(1)$$

for $j = 1,2,3,\dots,v$, where $\bar{X}(P_j)$ represents the information corresponding to the bit positions specified in partition P_j. This type of mixing is called *micro-mixing*, where the shuffling is performed at the bit-level. The hidden secrets are all the v, $\bar{X}$-fragments $\bar{X}(P_j)$, $j=1,2,\dots,v$, where, $\bar{X}(P_j) = [x_{j1}, x_{j2}, \dots, x_{jLp}]$, with $j_k \in \{1,2, \dots, L\}$ and the length of each subsequence is $L_p = L/v$. From this sequence $\bar{X}$ one can derive another sequence $\bar{Y}$ as:

$$\bar{Y} = BIT\_CMP[\bar{X}] \qquad \dots\dots\dots\dots(2)$$

where, the function *BIT_CMP[.]* is the bit-complement of any binary string vector. For example if $\bar{X} = [1,1,0,1,0,1]$, then $\bar{Y} = [0,0,1,0,1,0]$. Observe that $\bar{Y}$ inherits the statistical characteristics of $\bar{X}$, i.e. $y_i \perp y_j$ (columnwise independence). However, given $\bar{X}$, $\bar{Y}$ is completely deterministic. The need for a complementary sequence $\bar{Y}$ is for allowing conditional visibility of portions of $\bar{X}$ when a coalition of shares are brought together (which otherwise cannot be separated).

Step 2: Mixing $\bar{X}$ and $\bar{Y}$ and splitting into *n* shares

Once the two complementary but homogeneous blocks $\bar{X}$ and $\bar{Y}$ are created, a *macro-mixing* of the fragments of $(\bar{X}, \bar{Y})$ which contain the encryption key-strings,
i.e. $\bar{X}(P_j)$ and $\bar{Y}(\mathrm{d}\ P_j)$, $j = 1,2,3...,v$ is done in a controlled fashion to produce the shares. Each of the *n* shares can be written as,

$$\bar{S}_i = (\bar{S}_{i1}||\bar{S}_{i2}||...||\bar{S}_{iv}) \qquad \text{.....(3)}$$

where, the sequence $\bar{S}_{ij}$ is chosen according to a pre-designed codebook. The share inheritance is represented by the relation,

$$\begin{aligned} \bar{S}_{ij} &= \bar{X}(P_j) \text{ if } c_{ij} = 1 \\ \bar{S}_{ij} &= \bar{Y}(P_j) \text{ if } c_{ij} = 0 \end{aligned} \qquad \text{.....(4)}$$

The binary value $c_{ij} \in \{0,1\}$ is a part of the codebook,

$$\mathbf{C} = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1v} \\ \cdots & \cdots & \cdots & \cdots \\ c_{n1} & c_{n2} & \cdots & c_{nv} \end{pmatrix} \qquad \text{.....(5)}$$

where, *n* represents the number of users and *v* the number of partitions.

Step3: Retrieval of fragments of $\bar{X}$

In the problem of fine grained access we are not just interested in the entire block of data $\bar{X}$, but instead in extracting some of its fragments for which we require selective partition visibility depending on which combination of shares are stacked. This is precisely the reason why there is a need for choosing $\bar{Y}$ as a complementary sequence. When we *chop* $(\bar{X}, \bar{Y})$ and *stitch* the shares using a codebook which controls the *spatial redundancy and piecewise distribution* across multiple shares, we can ensure that a particular piece $\bar{X}(P_j)$ can be made visible provided the corresponding spatial inter-relationship across the stack remains unique to that segment alone. The conditional visibility (or invisibility) of the partitions depend on some basic rules discussed in the next section.

3. RULES FOR PARTITION RELEASE IN MIX-SPLIT

Each share has a unique signature defined by a unique *v*-bit codeword. A coalition of *t* shares, B_t, is defined by a $t \times v$ binary signature matrix $\mathbf{A}$ which is obtained by row sampling of the codebook $\mathbf{C}$. Hence from the point of view of a MIX-SPLIT specific construction, it is more appropriate to refer to a particular coalition in terms of its group signature $\mathbf{A}$ (as $B_t(\mathbf{A})$).

We may represent each block matrix $\mathbf{A}$ in terms of its individual columns, $\mathbf{A} = [\bar{w}_1, \bar{w}_2, \ldots, \bar{w}_v]$, where $\bar{w}_j$ is a $t \times 1$ binary column vector. Now we define a set $\mathbf{VP}[B_t(\mathbf{A})]$ as the set of visible partitions in the coalition of shares B_t defined by the group signature $\mathbf{A}$. We now formulate five rules which determine the visibility of the partitions.

Rule 1: Complementary and repetitive columns lead to inseparable partitions

$\mathbf{VP}[B_t(\mathbf{A})] = \phi$ IF for every $\bar{w}_j \in \mathbf{A}$, BIT_CMP$[\bar{w}_j] \in \mathbf{A}$ even though $\bar{w}_j$ may be distinct.
Example:

$$\mathbf{A} = \begin{bmatrix} 0\,1\,0\,1 \\ 0\,1\,1\,0 \\ 0\,1\,1\,0 \end{bmatrix} \qquad \ldots\ldots\ldots\ldots(6)$$

Given above is a codebook for a stack of three shares $\{\bar{S}_i, \bar{S}_j, \bar{S}_k\}$ from top to bottom. Note in this example every column has a corresponding column which is its bit complement. Consequently the stack equations remain non-unique. The bit positions which are leaked out based on columnwise stack element comparisons are,

$$P_A = \bigcup\{r_A\},\ \text{s.t. } \bar{S}_i(r_A) = \bar{S}_j(r_A) = \bar{S}_k(r_A) \qquad \ldots\ldots\ldots\ldots(7)$$

$$P_B = \bigcup\{r_B\},\ \text{s.t. } \bar{S}_i(r_B) \neq [\bar{S}_j(r_B) = \bar{S}_k(r_B)] \qquad \ldots\ldots\ldots\ldots(8)$$

Where, $r_A, r_B \in P$, with $P = P_1 \cup P_2 \cup P_3 \cup P_4$ with $P_A = P_1 \cup P_2$ and $P_B = P_3 \cup P_4$. The pairs [P_1 , P_2] and [P_3 , P_4] remain mixed and cannot be separated. However, since each stack comparison operation tends to narrow down the search for the partitions P_i , *i=1,2,3,4* this is a non-perfect scheme. For a sufficiently large set P_A it is very difficult to split this into two constituent partitions P_1 , P_2 without prior information. Thus no partitions are visible (or can be separated) from complementary patterns.

A gentle extension of this rule is possible by observing that repetitive columns also form an inseparable pair as the stack equations will remain the same.

$$\mathbf{A_1} = \begin{bmatrix} 0\,1\,1\,1 \\ 0\,1\,0\,0 \\ 0\,1\,0\,0 \end{bmatrix} \qquad \ldots\ldots\ldots\ldots(9)$$

Rule 2: Rowsampling of a complementary pattern is complementary

IF $\mathbf{VP}[B_t(\mathbf{A})] = \phi$, THEN $\mathbf{VP}[B_{t-1}(\mathbf{B})] = \phi$

where, **B** is a sub-block obtained by rowsampling of **A** (in this case by dropping any one of the rows in **A**). Consequently, by induction, dropping multiple rows will not reveal any of the partitions.

Rowsampling matrix **A** given in the example for Rule 1, we get,

$$\mathbf{B} = \begin{bmatrix} 0\,1\,0\,1 \\ 0\,1\,1\,0 \end{bmatrix} \qquad \ldots\ldots\ldots\ldots(10)$$

Since each column in **B** has a complementary counterpart, from Rule 1, it follows that $\mathbf{VP}[B_{t=2}(\mathbf{B})] = \phi$.

Rule 3: Single share is always mixed (no partitions visible)

$\mathbf{VP}[B_{t=1}(\mathbf{A})] = \phi$ irrespective of the choice of **A**.

As a special case we observe that since two statistically similar sequences are mixed, no information regarding the partitions is leaked out from a single share.

Rule 4: Atleast one co-ordinate becomes visible if a column is distinct

$\mathbf{VP}[B_t(\mathbf{A})] \neq \phi$ IF atleast one of the columns $\bar{w}_j$ is distinct (i.e. only one of its kind) AND its complementary counterpart BIT_CMP[$\bar{w}_j$] is not in **A**.

Example:

$$\mathbf{A}_2 = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \quad \text{.....(11)}$$

The first column is unique. Partition revealed is P_1 as $\mathbf{VP}[B_{t=2}(\mathbf{A_2})] = \{P_1\}$**.** The other two partitions P_2 and P_3 are mixed and inseparable as their columns form a complementary pair.

Rule 5: Adding to visible stack

IF $\mathbf{VP}[B_t(\mathbf{A})] \neq \phi$ THEN $\mathbf{VP}[B_{t+1}(\mathbf{D})] \neq \phi$

where, **D** is obtained by stacking another share (or codeword) on top of matrix **A**. To generalize this, further note that by adding to the stack, one cannot convert a pair of columns which are non-complementary into a complementary pair. This means that the number of visible partitions can only increase with the addition of new rows. Furthermore,

$$\mathbf{VP}[B_t(\mathbf{A})] \subseteq \mathbf{VP}[B_{t+1}(\mathbf{D})] \quad \text{.....(12)}$$

only if **A** can be derived from **D** directly through row-sampling.

Example:

To construct **D** a new row $\bar{S}_{new} \equiv [1,0,0]$ is added to the stack of $\mathbf{A_2}$.

$$\mathbf{D} = \left[\begin{array}{ccc} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \hline 1 & 0 & 1 \\ 1 & 1 & 0 \end{array}\right] \quad \text{.....(13)}$$

Three different stack comparisons between $\{\bar{S}_i, \bar{S}_j, \bar{S}_k\}$ will reveal all three partitions. The three disjoint sets of bit positions P_A, P_B, P_C which are revealed as a result of the stack comparison are,

$$\begin{aligned} P_A &= \bigcup \{r_A\}, \text{ s.t. } \bar{S}_i(r_A) = \bar{S}_j(r_A) = \bar{S}_k(r_A) \\ P_B &= \bigcup \{r_B\}, \text{ s.t. } (\bar{S}_i(r_B) = \bar{S}_j(r_B)) \neq \bar{S}_k(r_B) \\ P_C &= \bigcup \{r_C\}, \text{ s.t. } (\bar{S}_i(r_C) = \bar{S}_k(r_C)) \neq \bar{S}_j(r_C) \end{aligned} \quad \text{.....(14)}$$

These sets of positions will directly correspond to the original partitions $P_A = P_1$, $P_B = P_2$ and $P_C = P_3$ respectively. As a result $\mathbf{VP}[B_{t=3}(\mathbf{D})] = \{P_1, P_2, P_3\}$**.**

4. RAMP CODE

To construct the ramp code we first generate a binary column vector $\bar{w}_0$ with Hamming weight $r = n\text{-}1$ and length $n > 4$ and append all possible permutations of this vector to the code. There will be a total of n permutations of the column vector including the original vector itself. The resultant

codebook is of size $n \times n$ and each share has a size of $n \cdot L_p$ bits. We first claim that this code is a *3-out-of-n* selective access code which has a *ramp, multi-access* characteristic. The structure of this code with the number of partitions *v=n*, threshold *K=3* and number of users *n*, denoted as *(v,K,n)* selective access code (SAC), is shown below,

$$\mathbf{C}_{(n,3,n)} = \begin{bmatrix} 0\,1\,1\,1\cdots 1\,1\,1 \\ 1\,0\,1\,1\cdots 1\,1\,1 \\ 1\,1\,0\,1\cdots 1\,1\,1 \\ \vdots \quad\quad \vdots \\ 1\,1\,1\,1\cdots 1\,0\,1 \\ 1\,1\,1\,1\cdots 1\,1\,0 \end{bmatrix} \qquad \ldots\ldots\ldots\ldots(15)$$

When any two share codes are stacked to form the row-sampled matrix $\mathbf{D}_{t=2}$, say for example the first two shares $\bar{S}_1, \bar{S}_2$,

$$\mathbf{D}_{t=2} = \begin{bmatrix} 0\,1\,1\,1\cdots 1\,1\,1 \\ 1\,0\,1\,1\cdots 1\,1\,1 \end{bmatrix} \qquad \ldots\ldots\ldots\ldots(16)$$

there will be three distinct column identities, $[0,1]^T$, $[1,0]^T$ and $[1,1]^T$. The first two form a complementary pair while the third type will be repeated *n-2* times. Hence none of the partitions will be visible for *t=2* or fewer shares (Rule 1).

When any three shares are stacked, say $\bar{S}_i, \bar{S}_j, \bar{S}_k$, there will be three distinct columns which will result in three different stack equations. Exactly three distinct partitions P_i, P_j and P_k, where, *i,j,k* $\in$ *{1,2,..,n}* will be revealed through each of the stack equations. The remain *n-3* partitions will remain invisible since they all share the same column stack equation, $[1,1,1]^T$ (Rule 1).

In general, when $t \geq 3$ shares are fused, *t* partitions will be revealed as there will be *t* distinct stack equations. As the positions of the zeroes is different for each share, the set of partitions revealed will also be different. For such a scheme the total number of variations in access control possibilities by choosing different combinations of shares, is very large. This code can be called as *ramp code* since each share added to a legitimate coalition increases the number of partitions by exactly one and keeps the combination of partitions released, unique to that coalition. Only when the size of the coalition becomes *n-1* or larger, all the *v* partitions are released.

4.1 Code Characteristics When n ≤ 4

Note that the choice of *n* should be greater than *4*, otherwise this scheme will not acquire a ramp characteristic with threshold *K=3*. When *n=2*, this code is transformed into a *2-out-of-2* conditional unlocking code,

$$\mathbf{C}_{(2,2,2)+\mathrm{UNLK}} = \begin{bmatrix} 1\,0 \\ 0\,1 \end{bmatrix} \qquad \ldots\ldots\ldots\ldots(17)$$

where, a different sequence *[0,0]* or *[1,1]* needs to be stacked above the *2* shares to unlock the mixed partitions. When *n=3*, this code is transformed into a *2-out-of-3* selective access code (shown below) which has been discussed in detail in the Geometric interpretation of MIX-SPLIT [11].

$$\mathbf{C}_{(3,2,3)} = \begin{bmatrix} 1\,1\,0 \\ 1\,0\,1 \\ 0\,1\,1 \end{bmatrix} \qquad \ldots\ldots\ldots\ldots(18)$$

Here, the homogeneous block $\bar{X}$ is split into three disjoint partitions and each partition can be mapped to a co-ordinate in 3-dimensional space restricted to a UNIT cube. The three partitions put together therefore give $\bar{X}$ a certain position/location within this UNIT cube. The region of uncertainty for localizing $\bar{X}$ is a plane when any two shares are fused and becomes a point when all shares are stacked. When *n=4*, the scheme becomes a *3-out-of-4* joint access but without a ramp or selective access characteristic. This happens because when any three shares are stacked, there are four distinct stack equations and so all four partitions are released simultaneously.

$$\mathbf{C}_{(4,3,4)} = \begin{bmatrix} 0\,1\,1\,1 \\ 1\,0\,1\,1 \\ 1\,1\,0\,1 \\ 1\,1\,1\,0 \end{bmatrix} \qquad \ldots\ldots\ldots\ldots(19)$$

5. CHARACTERISTICS OF THE RAMP CODE

5.1 Granularity

In the ramp code, every legitimate coalition of shares $B_t, t \geq 3$ exposes a selective combination of fragments from the homogeneous block $\bar{X} = [\bar{X}(P_1)\,||\bar{X}(P_2)\,||\,\ldots||\bar{X}(P_v)]$. Let N_c be the total number of legitimate access combinations with,

$$N_c = \binom{n}{3} + \binom{n}{4} + .. + \binom{n}{n} \qquad \ldots\ldots\ldots\ldots(20)$$

and the coalitions of shares $A_1, A_2, \ldots, A_{Nc}$ represent only the valid access combinations for which atleast one of the partitions are revealed. Let the corresponding sets of partitions released be $\mathbf{VP}(A_1), \mathbf{VP}(A_2), \ldots, \mathbf{VP}(A_{Nc})$ respectively. A particular set of partitions $\mathbf{VP}(A_i)$ is said to be *distinct* if the combination of partitions is unique to that specific coalition. This holds true only if,

$$|\mathbf{VP}(A_i) \,\triangle\, \mathbf{VP}(A_j)| > 0 \;\forall j \neq i \qquad \ldots\ldots\ldots\ldots(21)$$

with $j \in \{1,2,3,..,N_c\}$. The operator 'Δ' indicates a symmetric set difference and $|\mathbf{VP}(A_i)|$ represents the size of the set (or the number of elements in the set) $\mathbf{VP}(A_i)$. If N_{dist} indicates the total number of distinct sets of partitions, it is obvious that,

$$N_{dist} \leq 2^v - 1 \qquad \ldots\ldots\ldots\ldots(22)$$

where, v is the total number of partitions. We may therefore define granularity as the fraction of legitimate access combinations which produce distinctive sets of partitions. This measure which quantifies the diversity of the multi-access space can be expressed as,

$$GRAN_{v,K,n} = \frac{N_{dist}}{2^v - 1} \qquad \ldots\ldots\ldots\ldots(23)$$

It may be observed that $GRAN_{v,K,n} \in [0,1]$. In the case of the ramp code *(v=n, K=3, n)* (Section 4), when shares $\bar{S}_i, \bar{S}_j, \bar{S}_k$ are stacked, three unique partitions are released P_i, P_j and P_k. Every new share added to the coalition helps release a new partition. All partitions are released when the coalition size becomes $t = n\text{-}1$. Thus all t-coalitions $(3 \leq t \leq n-2)$ release *distinct* sets of partitions.

Thus,

$$N_{dist(n,3,n)} = \binom{n}{3} + \binom{n}{4} + .. + \binom{n}{n-2} + 1 \quad \text{.....(24)}$$

$$= 2^n - 2n - \frac{n(n-1)}{2} - 1 \quad \text{.....(25)}$$

Hence the granularity of this

$$GRAN_{(n,3,n)} = \frac{N_{dist(n,3,n)}}{2^n - 1} \quad \text{.....(26)}$$

$$= 1 - \frac{(n^2 + 1.5n)}{2^n - 1} \quad \text{.....(27)}$$

It may observed that $GRAN_{n,3,n} \rightarrow 1$ as n becomes large.

5.2 Share size and comparison with Shamir's ramp

To compare efficiency of the MIX-SPLIT proposition, we examine another secret sharing scheme, which can be designed to provide the same degree of granularity. It is known, that Shamir's secret sharing scheme based on polynomial interpolation [1] originally designed as a *(k,n)* threshold scheme can be extended towards more general access structures [3]. The polynomial based construction lends itself to changes and expansions in the access structure and can be adapted to produce efficient share sizes for each participant. In contrast with the MIX-SPLIT construction, Shamir's scheme is a perfect secret sharing scheme which provides information theoretic secrecy, while the former relies on computational security gained by the strength of the sequence mixing process. However, to address the problem of optimizing the share size of each user while sharing multiple secrets, Shamir's scheme is deployed as a ramp scheme [7],[8] for the same multi-access structure.

5.3 Access structure of the linear access code

The linear access code protects exactly $v=n$ partitions of the block $\bar{X}$. To release a certain partition say P_i, the share of $\bar{S}_i$ must be included in the coalition of size $t \geq 3$. Let $\mathbf{R_i}$ be the set of all coalitions of size $t = 3$ or larger involving share $\bar{S}_i$ with $i = 1,2,3,...n$. All the access sets in $\mathbf{R_i}$ will have access to fragment (or partition) P_i. The total access space (*ASP*) is,

$$ASP_{n,3,n} = \mathbf{R}_1 \cup \mathbf{R}_2 \cup ... \cup \mathbf{R}_n \quad \text{.....(28)}$$

Since there are several overlapping combinations where the same coalition may be allowed to access multiple secrets, e.g. the 3-coalition $\bar{S}_i, \bar{S}_j, \bar{S}_k$ is allowed access to partitions P_i, P_j and P_k. The main challenge in Shamir's extension is to design a splitting mechanism that will allow the sub-secrets contained in the partitions P_i, $i=1,2,...,n$ to be shared efficiently amongst several users, while permitting fine grained access.

5.4 Extension of Shamir's scheme

To draw an equivalence in the analysis we observe that each partition P_i in the MIX-SPLIT construction is a L_p bit binary string which can be mapped to an integer K_i in the range $[0,1,\dots,(2^{Lp}-1)]$. For certain values of L_p, $2^{Lp}-1$ becomes a prime number, i.e. examples of primes are,

$$(2^7 - 1),\ (2^{13} - 1),\ (2^{17} - 1),\ (2^{19} - 1),\ (2^{31} - 1), \dots \quad \text{.....(29)}$$

Thus each one of the partitions can be represented by integer sub-secrets $K_1, K_2, \dots, K_n \in \{0,1, \dots, (2^{Lp}-1)\}$ uniformly distributed over the entire range since the strings were derived from I. I. D. binary random variables. Now consider a coalition of 3 share-holders $\{\bar{S}_i, \bar{S}_j, \bar{S}_k\}$. The sub-secrets accessible are, K_i, K_j and K_k. An efficient way to extend Shamir's scheme for sharing these three secrets is, by choosing a polynomial,

$$f_1(z) = K_i + K_j \cdot z + K_k \cdot z^2 \qquad \dots\dots\dots\dots(30)$$

where, $z \in \{0,1, \dots, (2^{Lp}-1)\}$. The shares given to users U_i, U_j *and* U_k are three different samples of the polynomial $f_1(z)$, i.e. $f_1(z = z_i)$, $f_1(z = z_j)$ and $f_1(z = z_k)$ respectively with $z_i, z_j, z_k \in \{0,1, \dots, (2^{Lp}-1)\}$ and available in public domain. When the three share holders pool their shares, they can construct three different equations which can be solved to reconstruct the polynomial (or solve for the coefficients K_i, K_j and K_k). If any one (or fewer) shares are missing then the system of equations is incomplete and none of the sub-secrets can be reconstructed. Thus the scheme is computationally secure if L_p is large but there is some information leakage regarding the parent polynomial. As a result this scheme becomes a ramp version of Shamir's original secret sharing scheme.

Now consider another 3-coalition where exactly one member of the previous coalition is displaced by another, i.e. the coalition is $\bar{S}_i, \bar{S}_j, \bar{S}_r$, where U_k has been displaced by U_r. The sub-secrets available to this coalition are K_i, K_j and K_r. The sharing is done by constructing a new polynomial,

$$f_2(z) = K_i + K_j \cdot z + K_r \cdot z^2 \qquad \dots\dots\dots\dots(31)$$

and distributing shares $f_2(z = z_i)$, $f_2(z = z_j)$ and $f_2(z = z_r)$ to users U_i, U_j and U_r respectively, ensuring that,

$$f_2(z_i) \neq f_1(z_i)$$
$$f_2(z_j) \neq f_1(z_j)$$

Each different *3*-coalition therefore requires a distribution of three distinct shares to exactly three users. Thus the total number of shares distributed per participant for this multi-access structure considering only legitimate *3*-coalitions is specified as the normalized share length per user, $LEN_{Sh-user(t=3)}$,

$$LEN_{\text{Sh-user}(t=3)} = \frac{\binom{n}{3} \times 3}{n} = \frac{(n-1)(2n-1)}{2} \qquad \dots\dots\dots\dots(32)$$

If the *3*-coalition expands to include one more user U_l, the access set becomes $\bar{S}_i, \bar{S}_j, \bar{S}_r, \bar{S}_l$. This 4-coalition has access to sub-secrets K_i, K_j, K_r and K_l. An efficient construction would be to generate the extended polynomial,

$$f_3(z) = f_2(z) + K_l \cdot z^3 = K_i + K_j \cdot z + K_r \cdot z^2 + K_l \cdot z^3 \qquad \dots\dots\dots\dots(33)$$

The only additional share required is $f_3(z = z_l)$, which is given to U_l. The other users U_i, U_j, U_r can use the older shares produced using the polynomial $f_2(z)$, i.e. $[f_2(z_i), f_2(z_j), f_2(z_r)]$ to

reconstruct $f_2(z)$ and consequently determine $f_2(z_l)$. Thus the size of the share required by Shamir's ramp scheme is atleast $O(n^2)$.

5.5 Share Size for MIX-SPLIT ramp construction

The ramp selective access code used for MIX-SPLIT has the structure shown in Eqn. 15. Each user is therefore given a share size of $v=n$ units, which correspond to a length of $n \cdot L_p$ bits. Hence the share size per user with a MIX-SPLIT ramp code is of order $O(n)$, which is considerably less as compared to Shamir's ramp.

6. FINE-GRAINED ACCESS OF INDUSTRIAL DESIGNS

An example of a protected bike design is shown in Figure.1. The overall design sketch has been encrypted using a block parametric transform [12]. Granularity in access is made possible by first partitioning the design into $v=5$ non-overlapping regions R_1, R_2, R_3, R_4, R_5 and encrypting them with different keys K_1, K_2, K_3, K_4 and K_5. The encryption process is a linear transformation with the help of a modified discrete Fourier transform (DFT) kernel. The following codebook is used as a $(v = 5, K = 3, n=5)$-SAC (ramp code) and is used to produce $n=5$ shares of the keys using the MIX-SPLIT algorithm,

$$\mathbf{C}_{(5,3,5)} = \begin{pmatrix} 0\,1\,1\,1\,1 \\ 1\,0\,1\,1\,1 \\ 1\,1\,0\,1\,1 \\ 1\,1\,1\,0\,1 \\ 1\,1\,1\,1\,0 \end{pmatrix} \qquad \ldots\ldots\ldots\ldots(34)$$

Each share $\bar{S}_i$, $i=1,2,..,5$ is a mixture of all five key-strings. None of the partitions can be resolved when two or fewer shares are combined. Figure.1(a) shows the complete bike design blueprint which is a union of all the five regions. When a coalition of any two designers fuse their shares (e.g. $[\bar{S}_1, \bar{S}_3]$, no part of the design is revealed (Figure.1 (b)). However when a coalition of three users combine their shares, depending on the type of coalition, a unique portion of the design is made partially visible (Figure.1(c,d,e)). When any four out of the five shares are combined (e.g. coalition $[\bar{S}_1, \bar{S}_2, \bar{S}_3, \bar{S}_5]$) the complete design is visible (Figure.1(f)).

Hence coalitions of size *2* are considered non-access subsets, coalitions of size *3* partial access subsets and coalitions of size greater than *3* complete access sets.

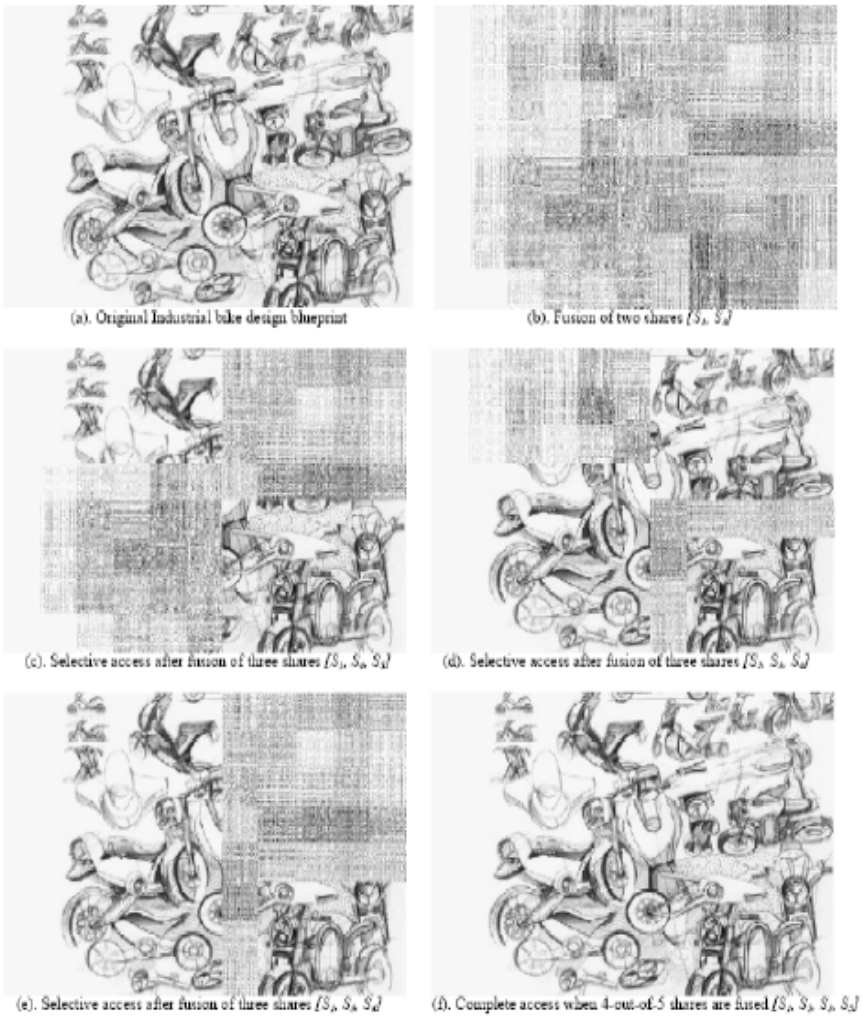

Figure 1. Application of the *(v=5, K = 3, n = 5)* selective access code towards the protection of Bike designs. No part of the design is revealed when any 2 out of 5 shares are fused. When any 3 shares are fused, depending on the type of coalition a unique portion of the design is revealed.When any 4 or more shares are fused then the complete bike design is revealed.

7. CONCLUSIONS

This paper proposes a ramp code using the MIX-SPLIT construction for fine-grained access control of a block of I.I.D. binary strings which could represent a collection of encryption / decryption keys in the mixed form, where each key corresponds to a specific fragment (or partition) of the parent secret string. Conditional visibility of these partitions can be enforced by controlled macro-mixing of the parent secret with a bit-complementary counterpart. This macro-mixing is done with the help of a codebook which is designed based on some basic rules. One such code is the proposed ramp code.

The proposed ramp code designed for fine-grained access has a high granularity and requires a much smaller share size of $O(n)$ as compared to Shamir's ramp adaptation which requires a share size of atleast $O(n^2)$. An industrial design access control application using the MIX-SPLIT ramp code for securing the key distribution, has been demonstrated.

REFERENCES

[1] A. Shamir, "How to share a secret", Communications of the ACM, Vol. 22, pp 612—613, 1979.

[2] G. Blakley, "Safeguarding cryptographic keys", Proceedings of AFIPS 1979 National Computer Conference, Vol. 48, pp 313-317, 1979.

[3] M. Itoh, A. Saito and T. Nishizeki,, "Secret sharing scheme realizing general access structure", Proceedings of IEEE Globecom'87, pp 99-102, 1987.

[4] L. Xiao and M. Liu, "Linear multi-secret sharing schemes", Science in China Series F: Information Sciences, Vol. 48, No. 1, pp 125-136, Feb 2005.

[5] G. R. Blakley and C. Meadows, "Security of ramp schemes", Proceedings of CRYPTO 84 on Advances in cryptology, Lecture notes in Comouter Science, Vol. 196, pp 242-268, 1985.

[6] E. Karnin and J. Green and M. Hellman, "On secret sharing systems", IEEE Transactions on Information Theory, Vol. 29, No. 1, pp 35-41, 1983.

[7] L. Csirmaz, "Ramp secret sharing and secure information storage", Lecture slides, 2009, www.renyi.hu/~csirmaz/intellisec/ramplecture.pdf.

[8] C. C. Thien and J. C. Line, "Secret Image Sharing", Journal of Computers and Graphics, Vol. 26, No. 5, pp 765-770, 2002.

[9] K. Karthik and D. Hatzinakos, "Multimedia Encoding for Access Control with Traitor Tracing: Balancing Secrecy, Privacy and Traceability," VDM Verlag Dr. Muller, 2008, ISBN: 978-3-8364-3638-0, 2008.

[10] K. Karthik and D. Hatzinakos, "A Unified Approach To Construct Non-perfect Secret Sharing And Traitor Tracing Schemes", International Conference on Security and Management (SAM), pp83-89, 2007.

[11] K. Karthik and D. Hatzinakos, "Secure group authentication using a non-perfect secret sharing scheme based on controlled mixing," in Proceedings of INDICON'09, pp 1-4, 2009.

[12] N. Suresh and K. Karthik, "Secure fingerprint embedding based on modified GDFT based parametric transform," International Conference on Image Information Processing (ICIIP 2011), pp. 1 –6, Nov. 2011.

Authors

Kannan Karthik

Received Ph. D. degree from University of Toronto in 2006. Currently he is an Assistant Professor in the Department of Electronics and Electrical Engineering, Indian Institute of Technology Guwahati. Some of his research interests include, Multimedia Security, Privacy preserving retrieval, Biometric Hashing, Adaptation of non-perfect secret sharing schemes for fine-grained access control, Traitor tracing and Digital Watermarking.