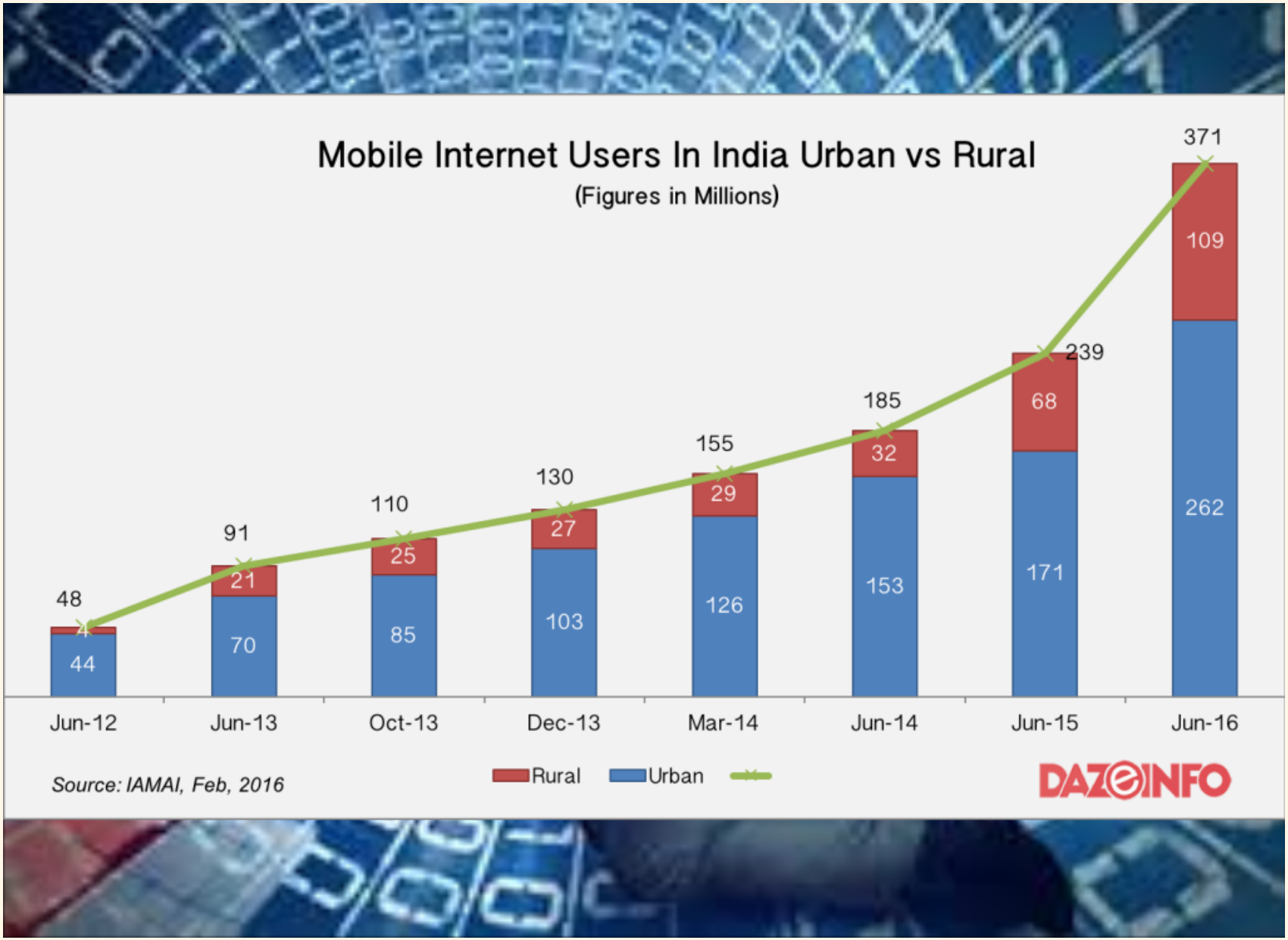
Practical Attacks on a Class of Secret Image Sharing Schemes: Few Open Issues

Avishek Adhikari

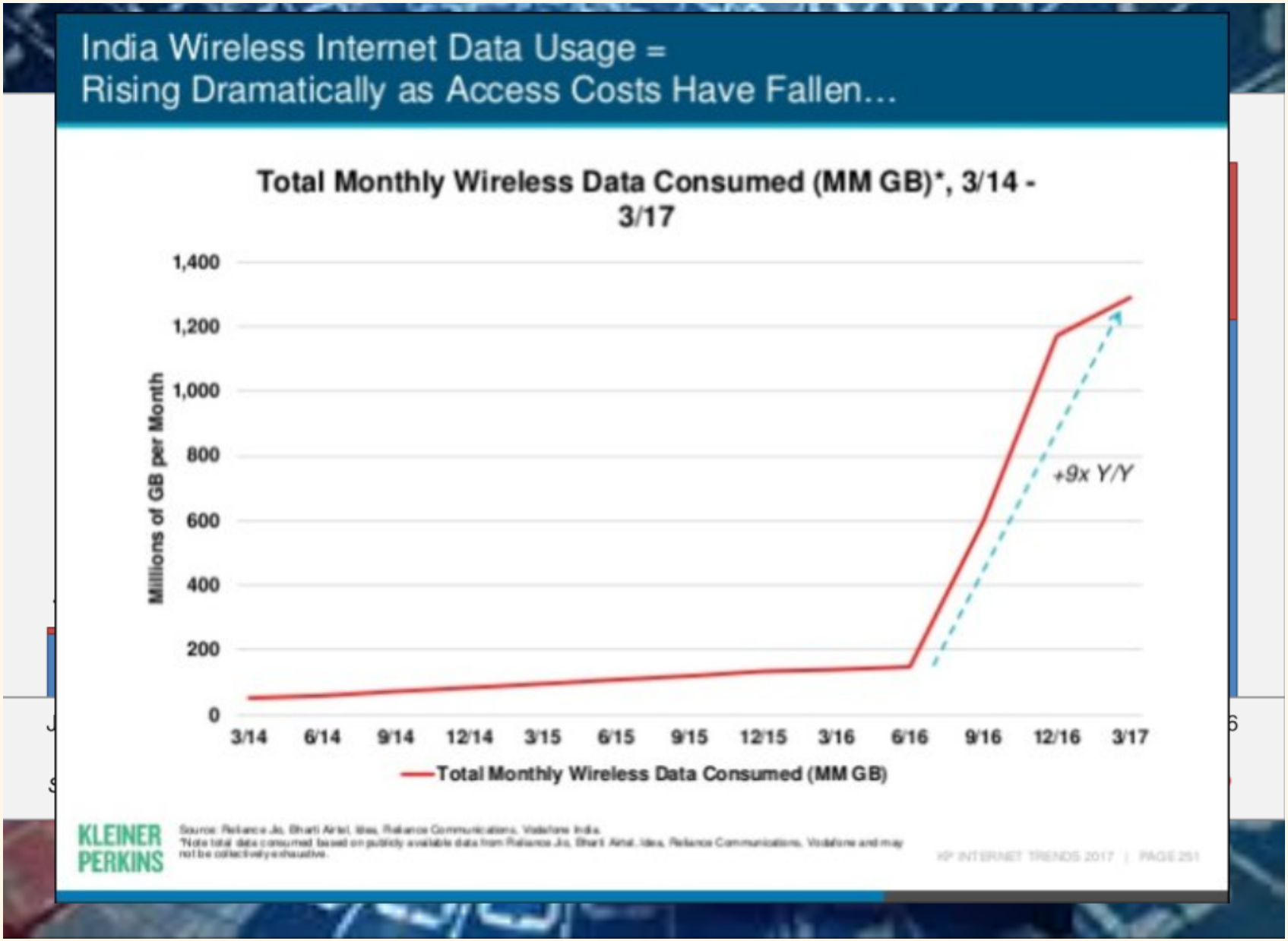
Professor, Department of Mathematics
Presidency University, Kolkata.
Founder Secretary of IMBIC, India (Branches: Sweden, Japan)
Treasurer, Cryptology Research Society of India (CRSI)

Research Team Members
Jyotirmoy Pramanik, Md. Kutubuddin Sardar
Subarsha Banerjee, Sandip Kumar Mondal
Chandan Goswami and Dr Prakash Dey

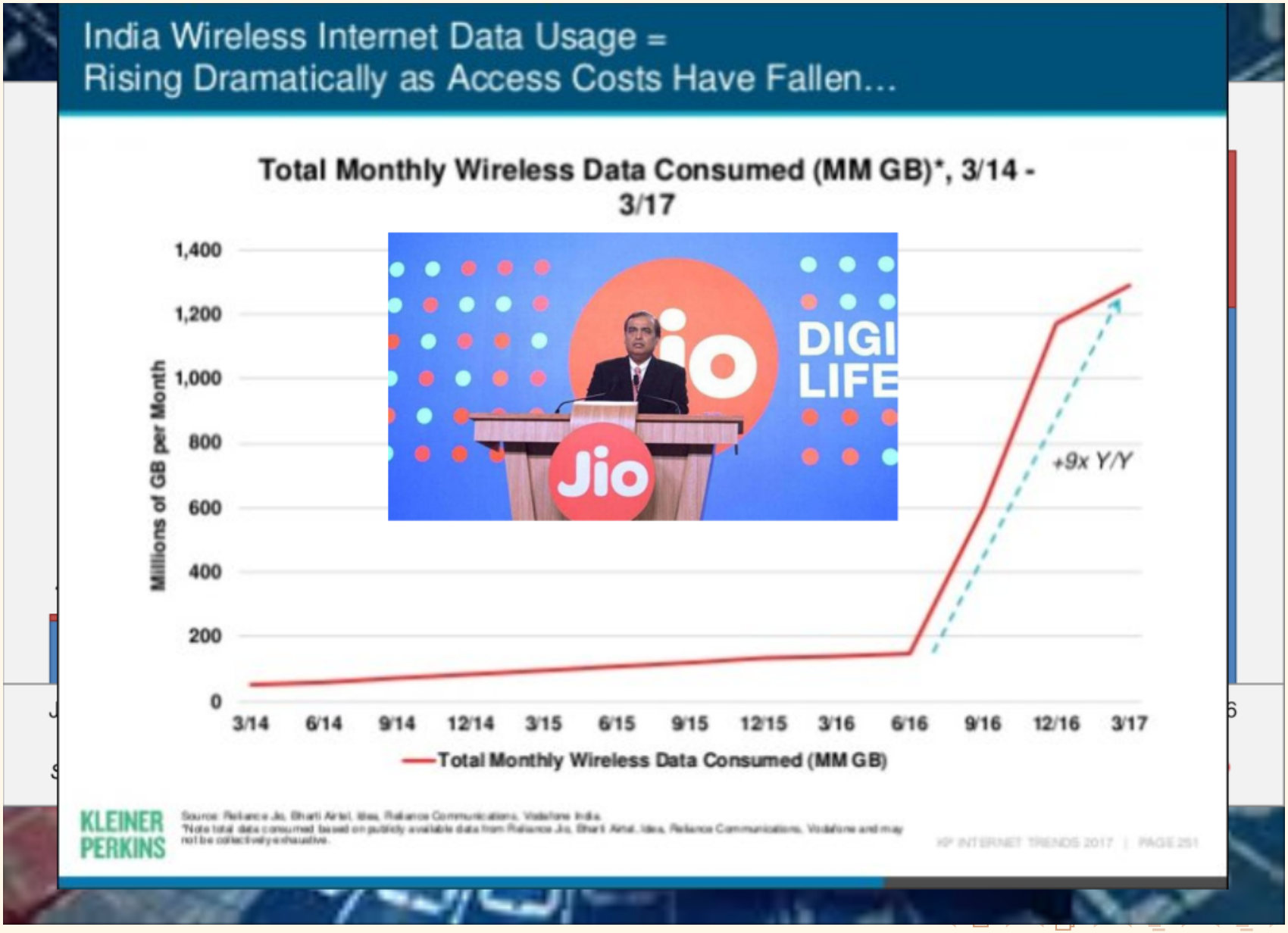
Internet in Day-to-Day Life



Internet in Day-to-Day Life



Internet in Day-to-Day Life



Big Data Usage: 2019

Google processes over 5 exabytes of data per day.
1 Exabytes = $1,000,000,000 = 10^9$ **GB**.
Google handles 1.2 trillion searches every year.
1 trillion=10^{12}.

Facebook generates 4 petabytes of new data per day, where **1 petabytes =** 10^6 **GB.**
350 million photos are uploaded per day. Users generate 4 million likes every minute.

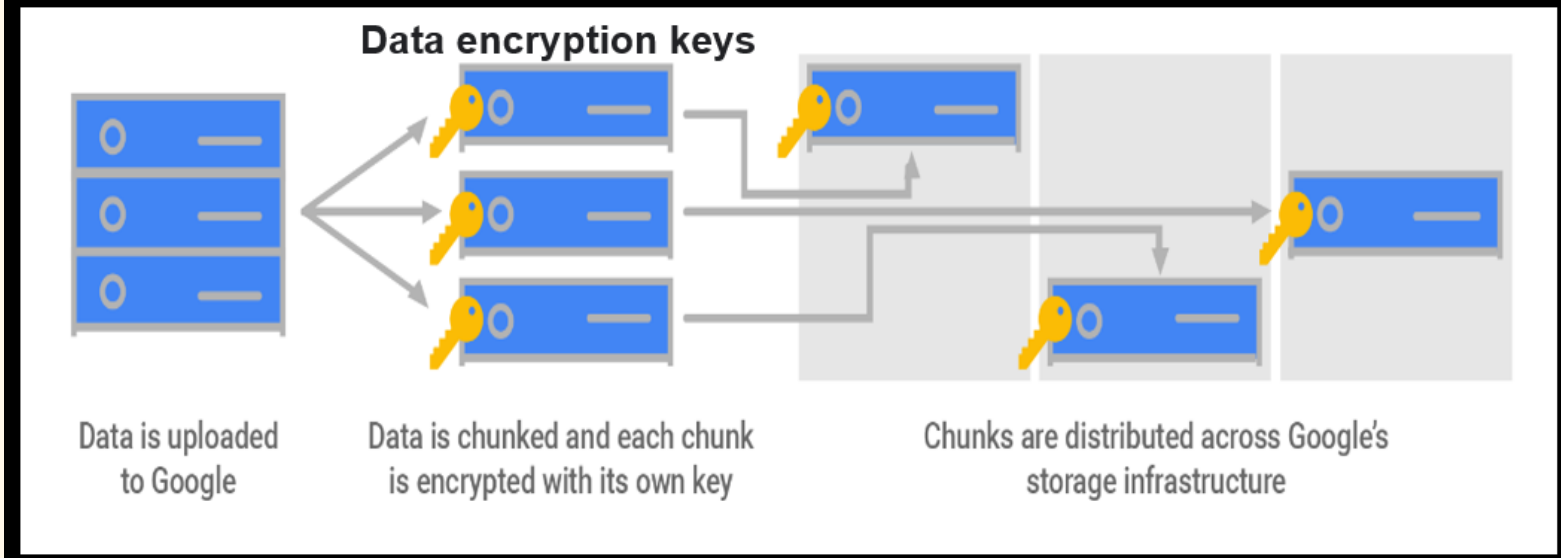
Whatsapp has nearly 500 Million Users. It processes, more than 70 Million Messages a Second.

Nicely said!

Application of Secret Sharing in Digital World

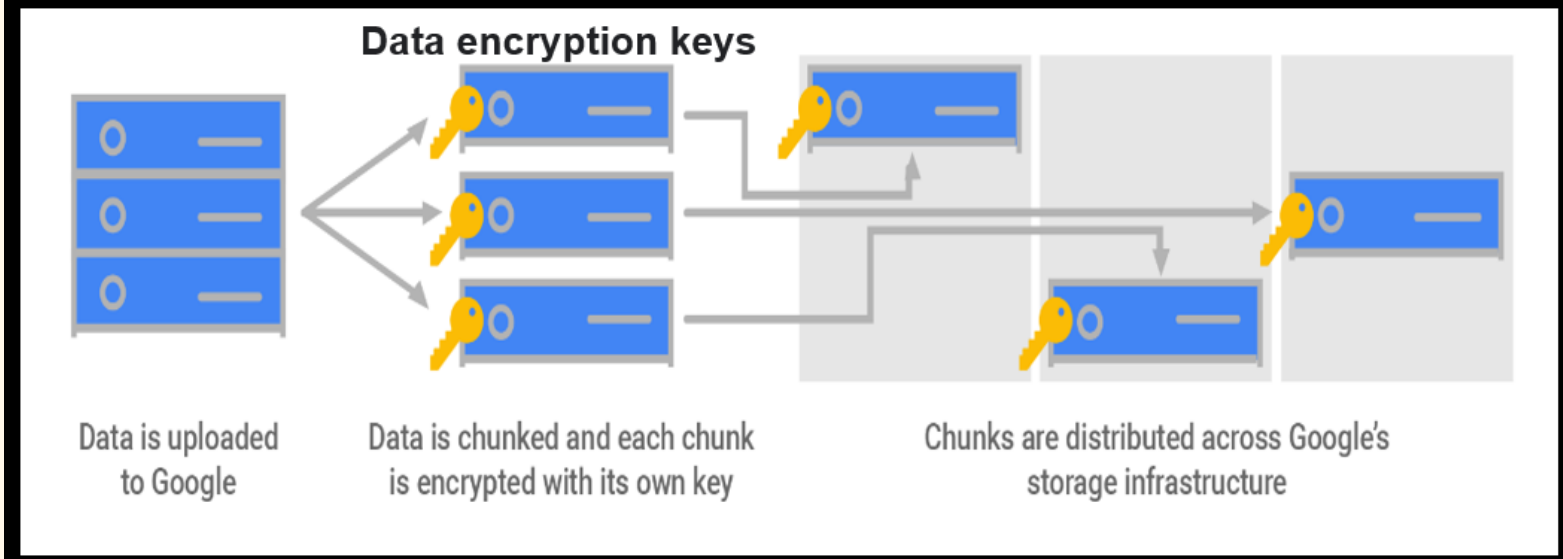
Application of Secret Sharing in Digital World

How Google Cloud Stores Data at Rest



Each chunk is encrypted at the storage level with an individual encryption key: two chunks will not have the same encryption key, even if they are part of the same Cloud Storage object, owned by the same customer, or stored on the same machine.

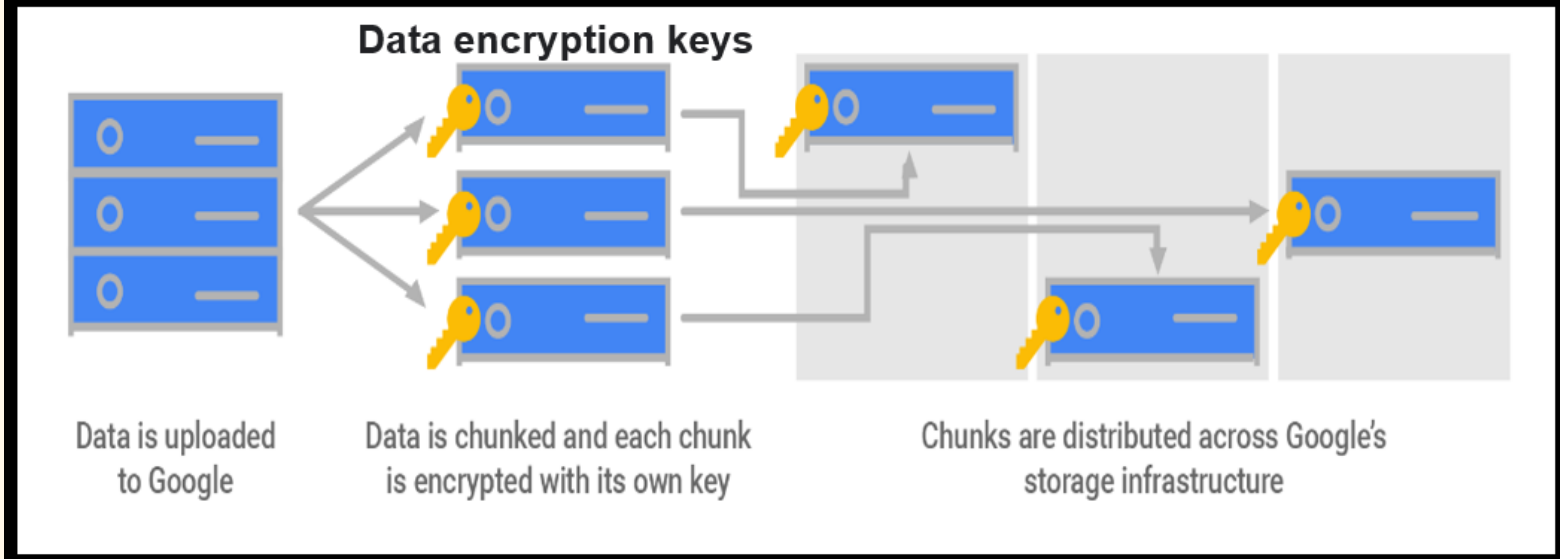
Application of Secret Sharing in Digital World



The DEKs are encrypted with (or "wrapped" by) a *key encryption key (KEK)*. One or more KEKs exist for each Google Cloud service.

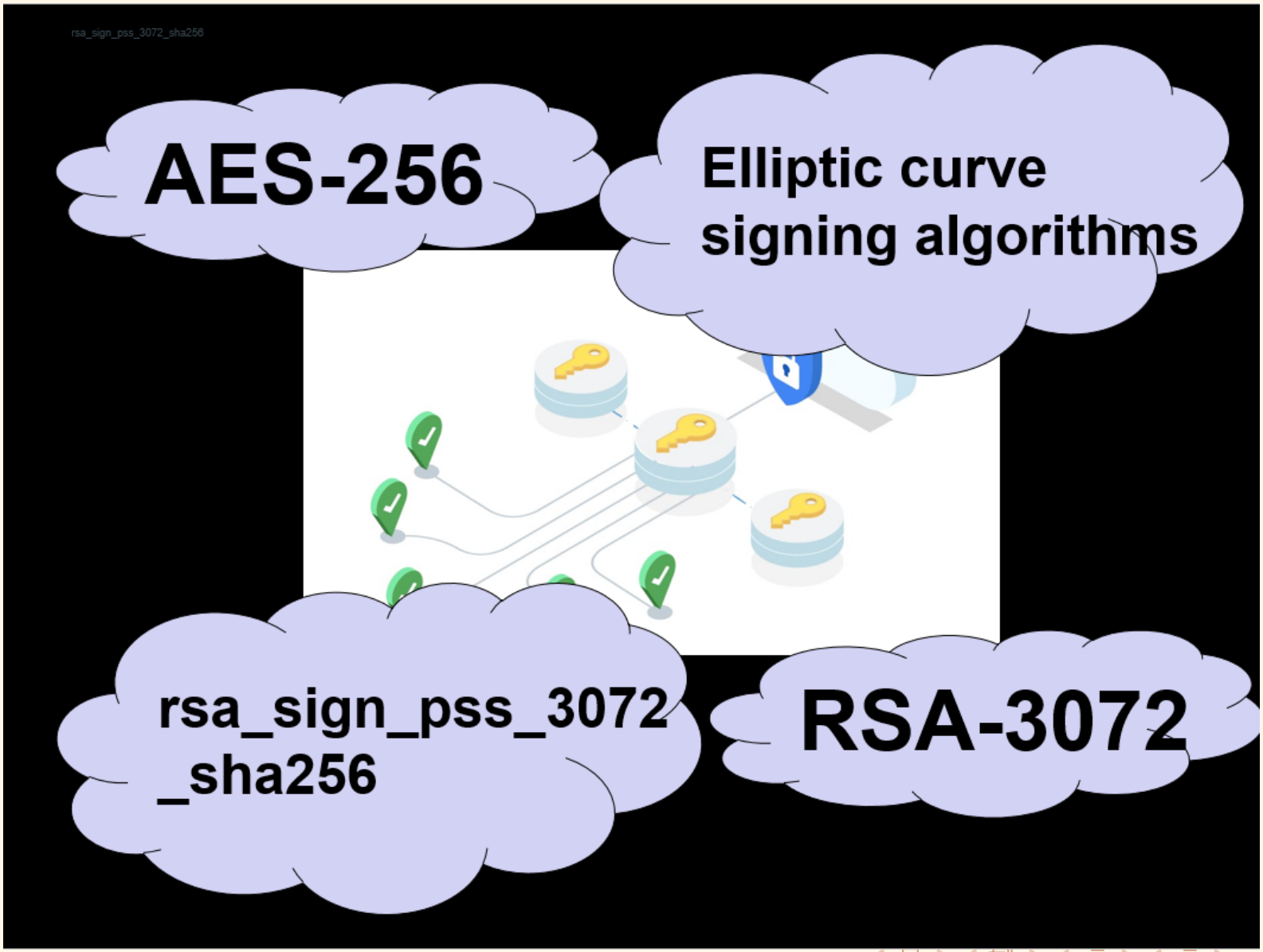
These KEKs are stored centrally in Google's *Key Management Service (KMS)*, a repository built specifically for storing keys.

How Google Cloud Stores Data at Rest



When a storage system needs to retrieve encrypted data, it retrieves the wrapped DEK and passes it to KMS. KMS then verifies that this service is authorized to use the KEK, and if so, unwraps and returns the plaintext DEK to the service. The service then uses the DEK to decrypt the data chunk into plaintext and verify its integrity.

Key Management Services: Cryptographic Tools Used

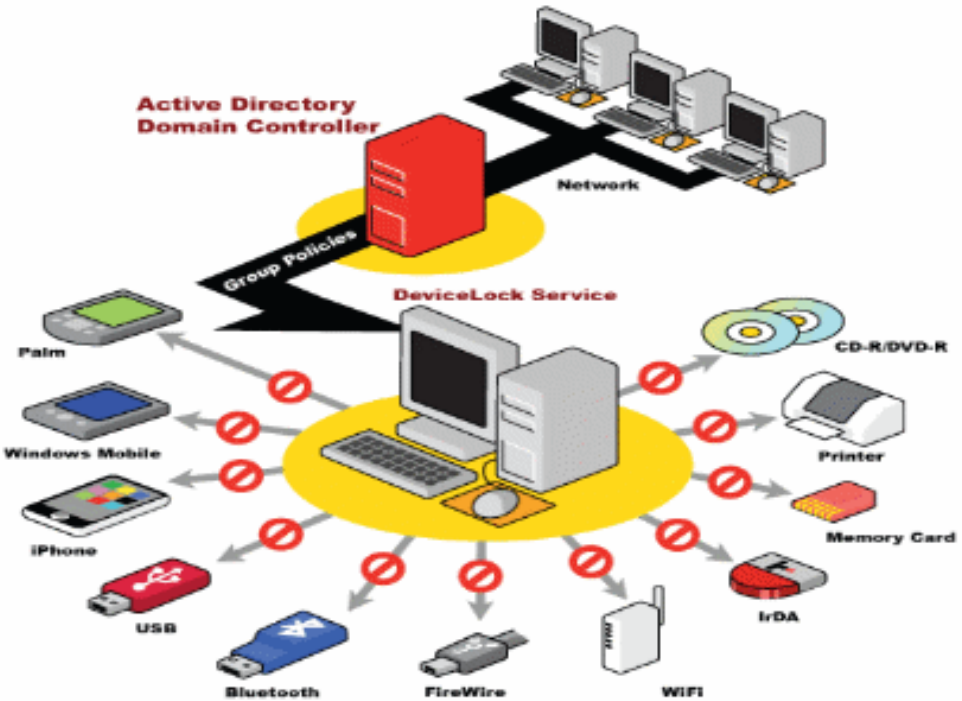


Prime Numbers in our day to day life (308 digits)

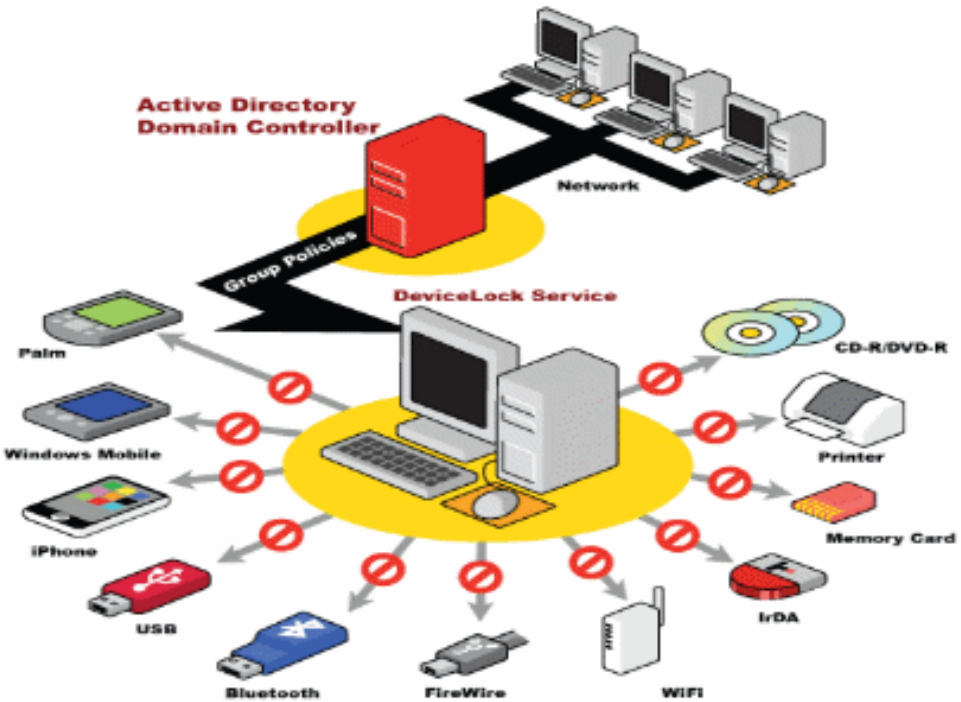
1230231922161117176931558813276752514640713895736833
7157661180291600588006146729487753600678385934595824
2964925405180490851288418089823682358508248206534833
1234959350355845017413023320111360666922624728239756
8804164344783156936750134130907572086903767932966588
1066294182449348845172650530371291600534674790862370
2673480919353936813105736620402352744776903840477883
6511003224093019834883638029305404824879097634840982
5394072868513204440886373475427121259247177864394948
6688511721051561970432780747454823776808464180697103
0838618121843485655227401957966826222055118455120805
5201031005025580158934964592800113374547422071501368
3413907542779063759833876101354235184245096670042160
7206294115815023712480084304471848420986103205804179
9220666224732872212208851364368390767036020916265367
0641130936997002170500675501374723998766005827579300
7232534748906122501351718891748990799112915123997738
72178519018229993369

All Have Secrets in Life

Storing Secrets in Different Places



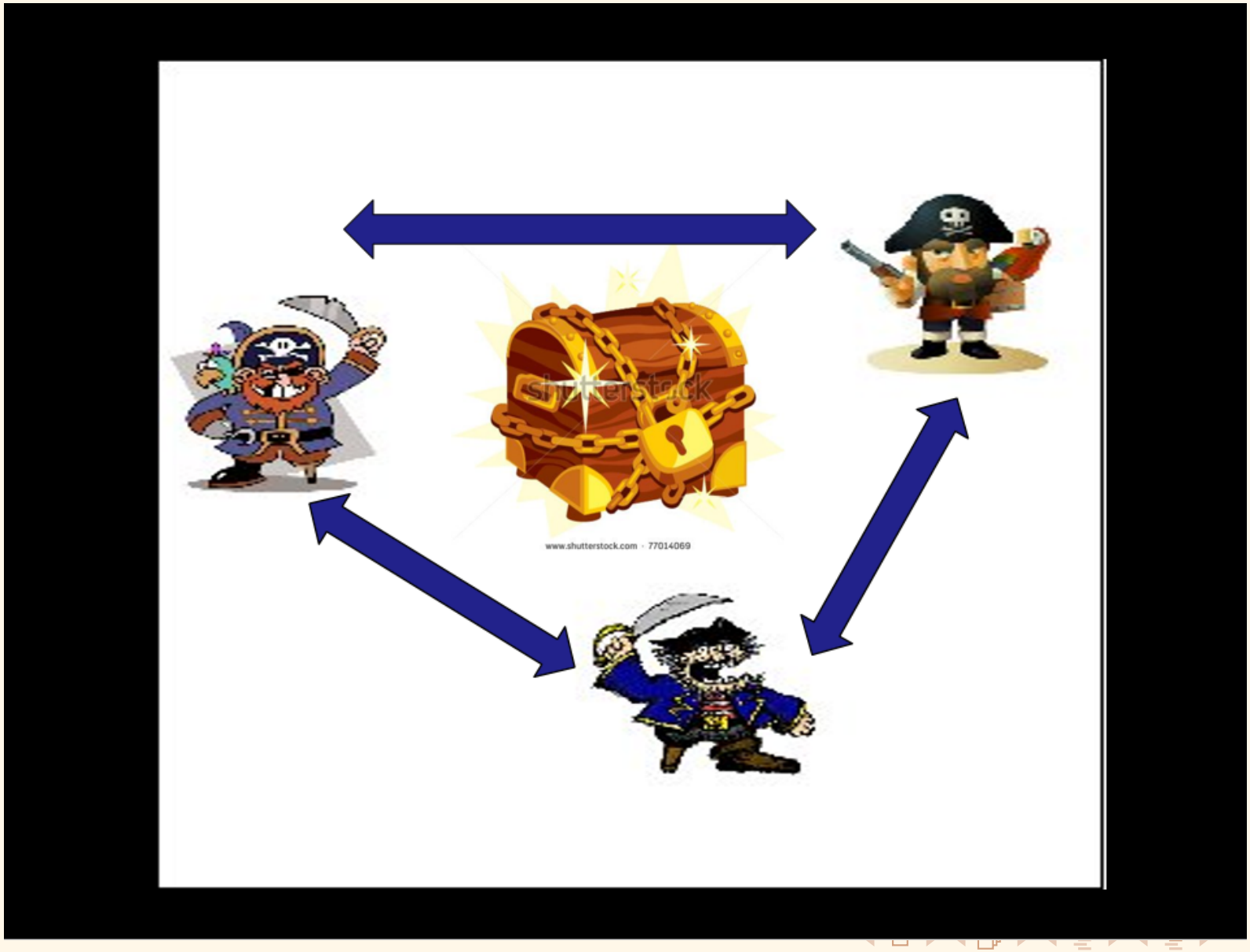
Storing Secrets in Different Places



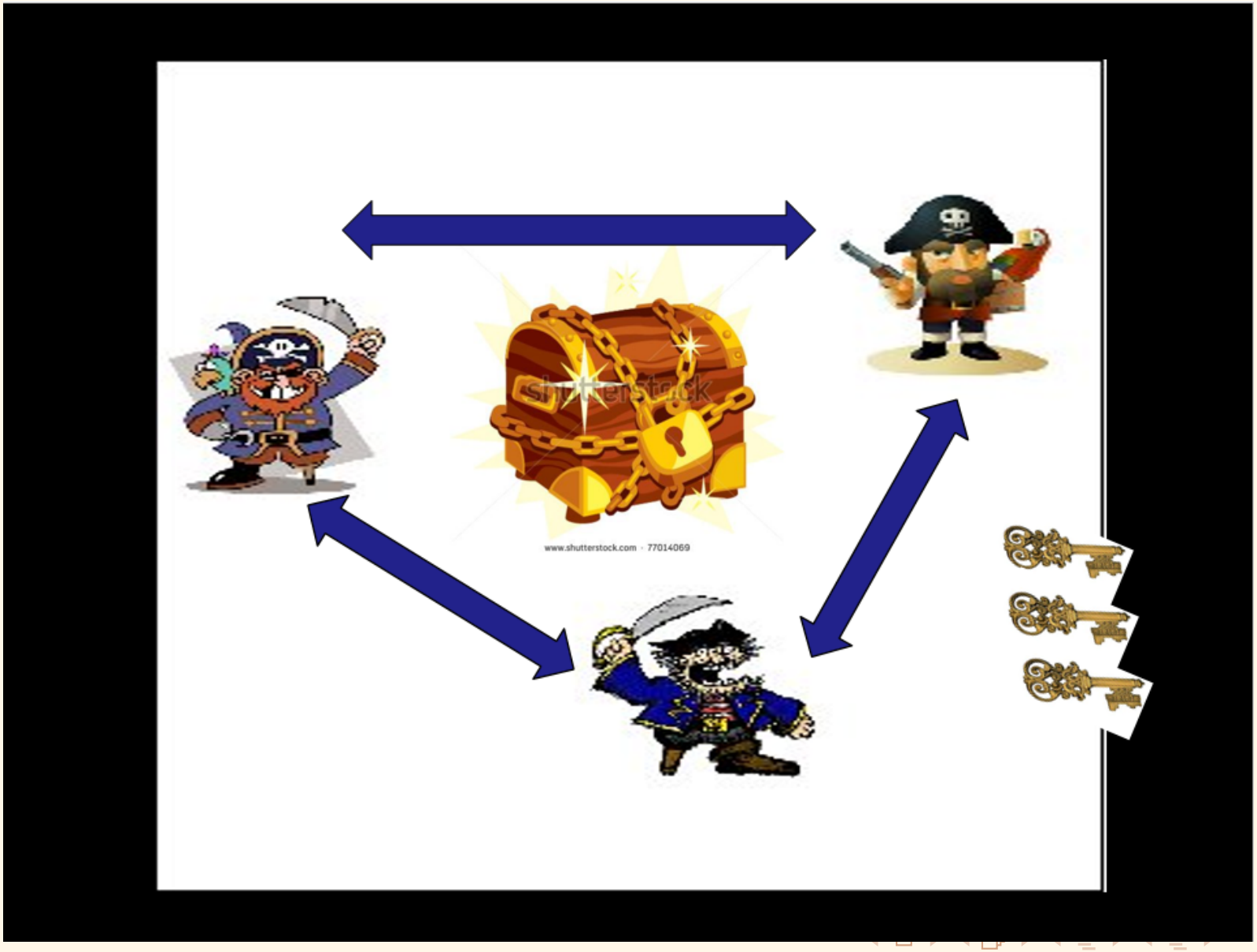
What is Secret Sharing?

What is Secret Sharing?

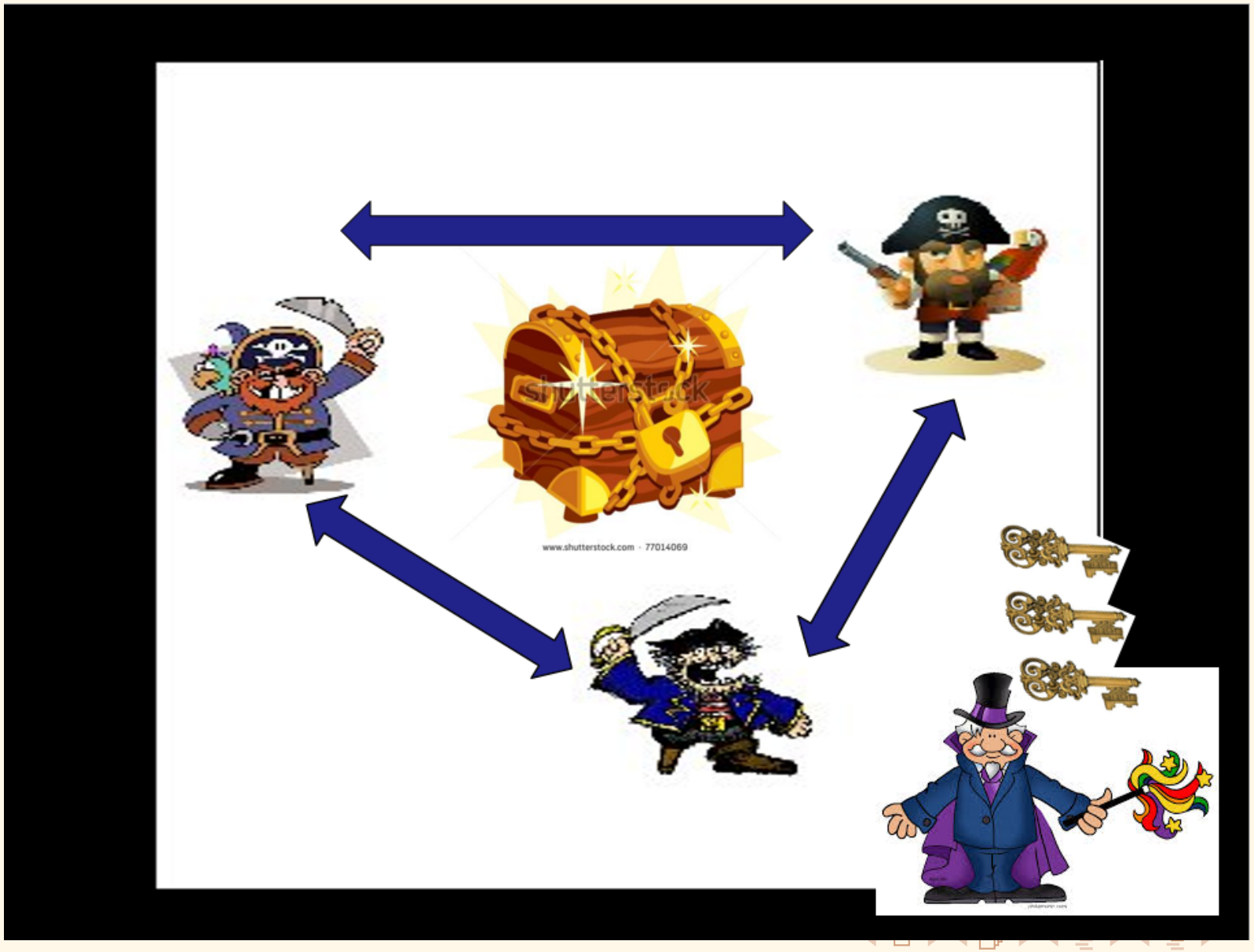
What is Secret Sharing?



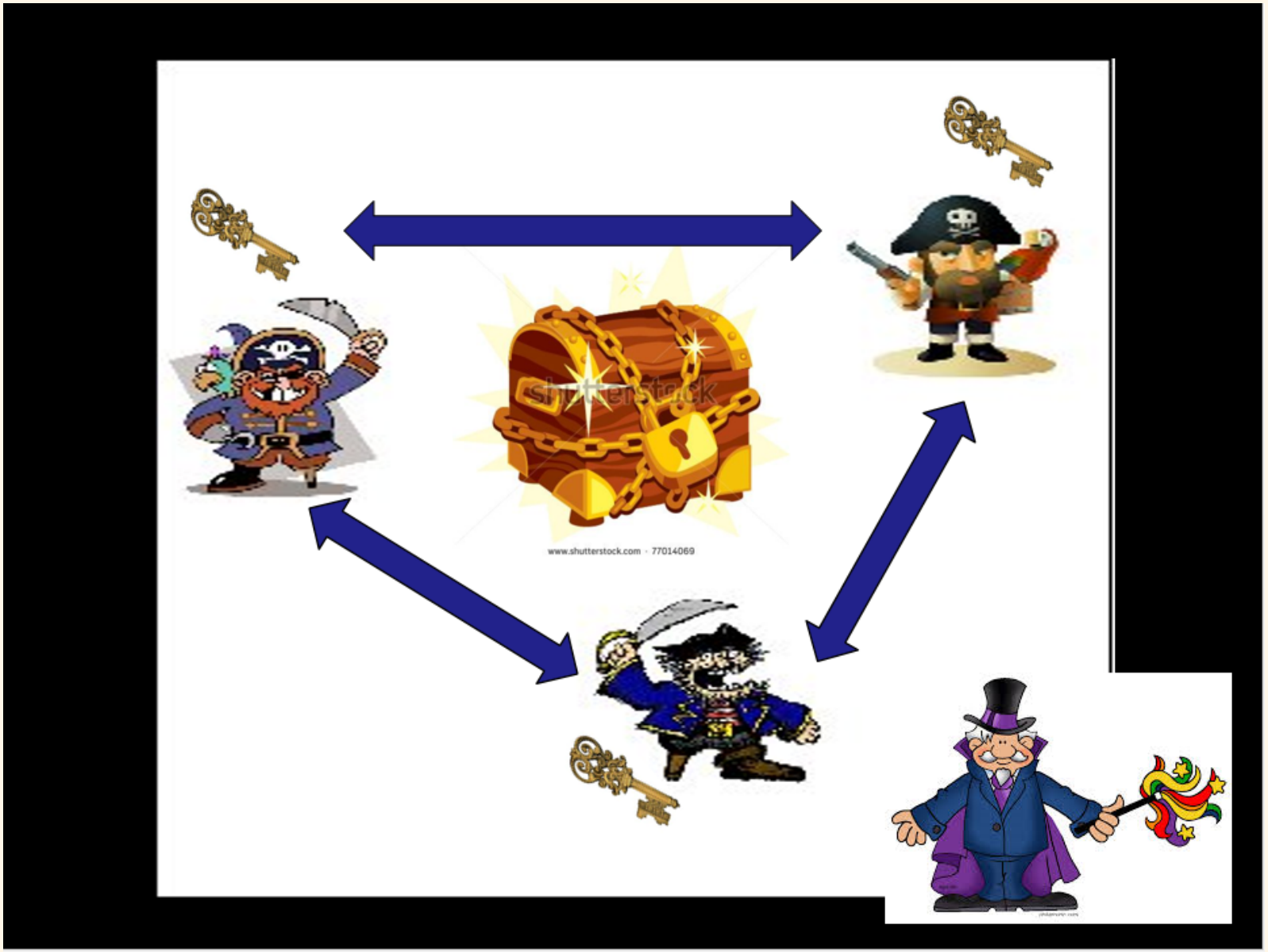
What is Secret Sharing?



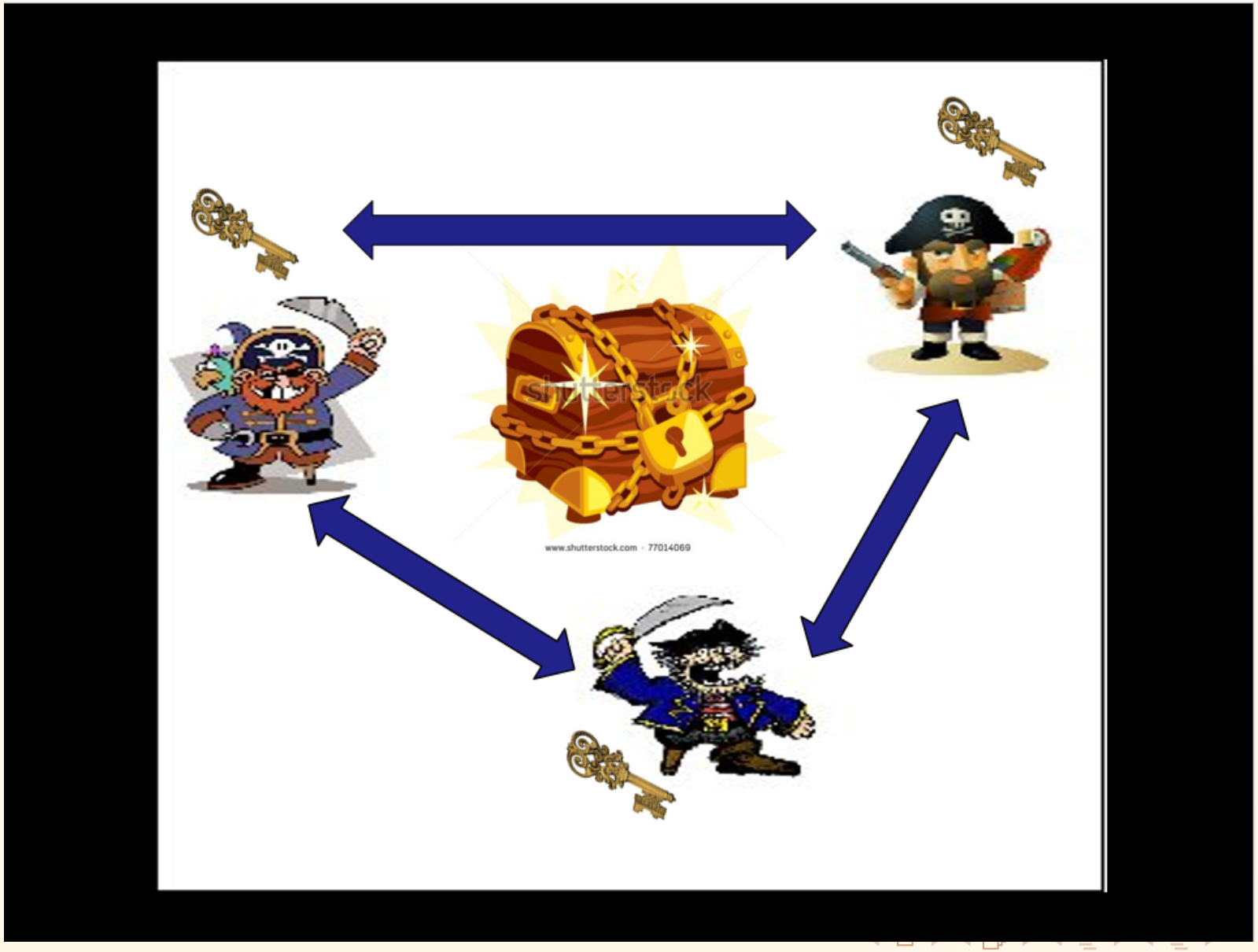
What is Secret Sharing?



What is Secret Sharing?



What is Secret Sharing?

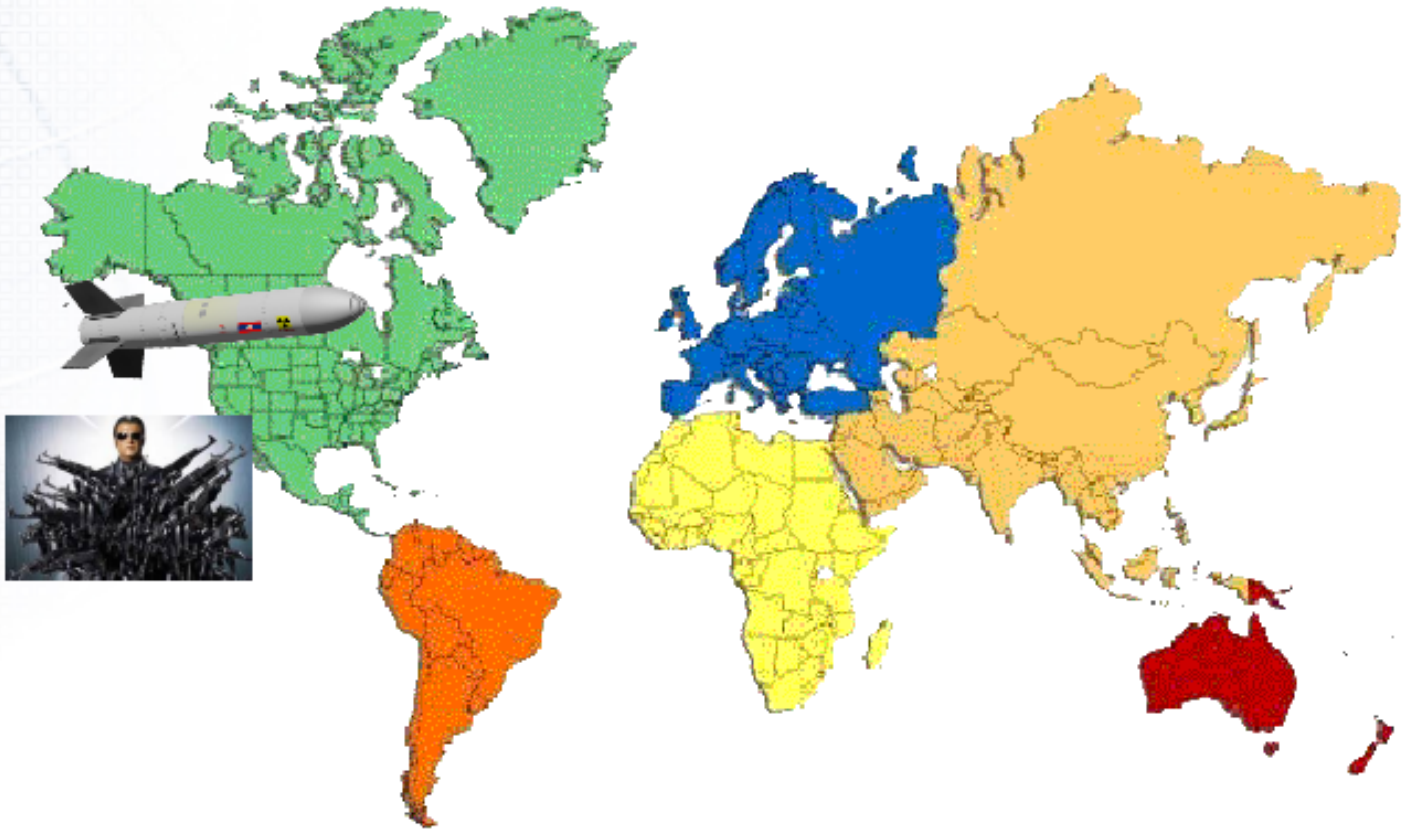


(t, w) threshold scheme

Let t and w be two positive integers, such that $t \leq w$. A (t, w) *threshold scheme* is a method of sharing a scheme key k among a set of w participants in such a way that any t participants can compute the value of k, but no group of $(t-1)$ participants can do so.

Real Life Example of Secret Sharing

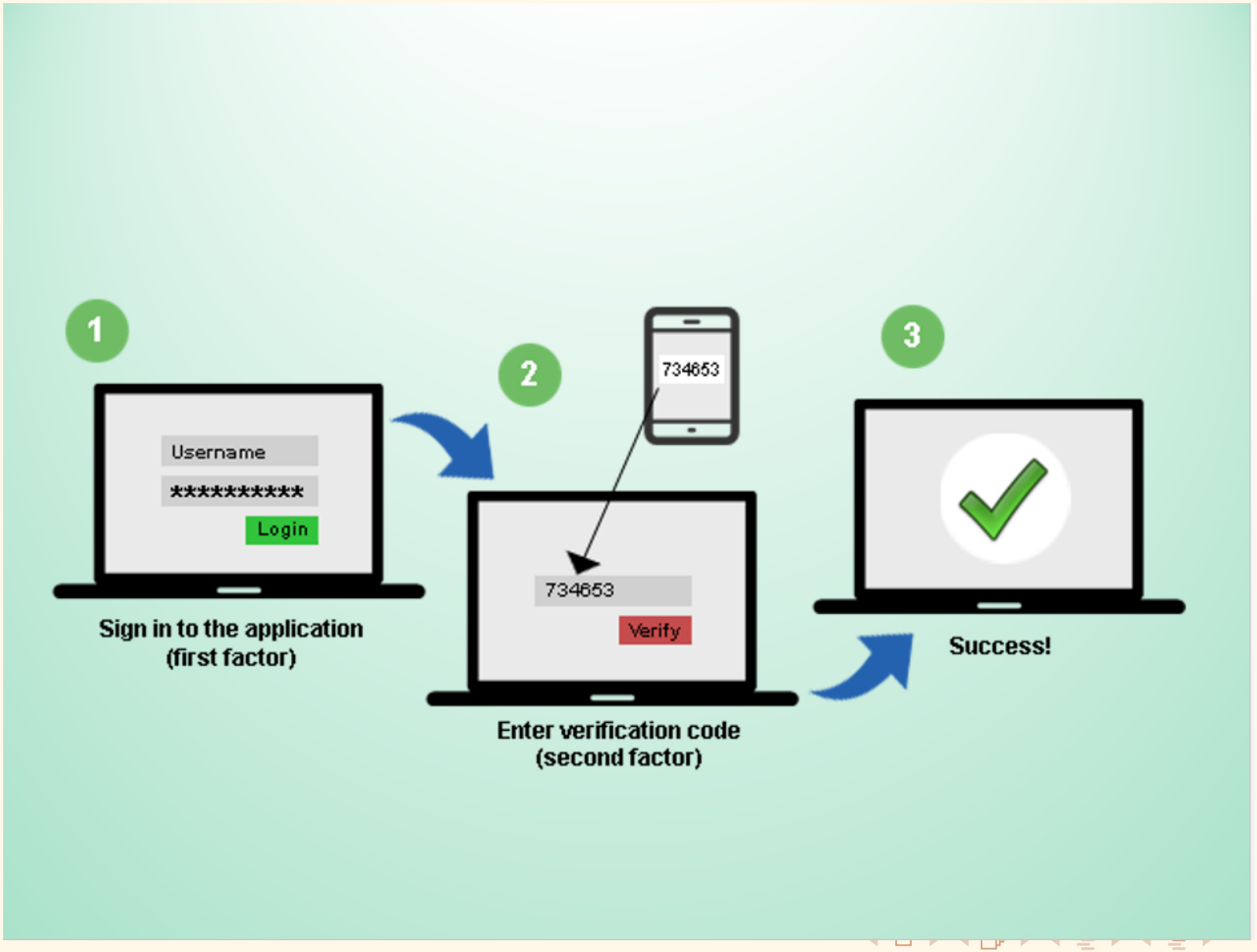
Real Life Example of Secret Sharing



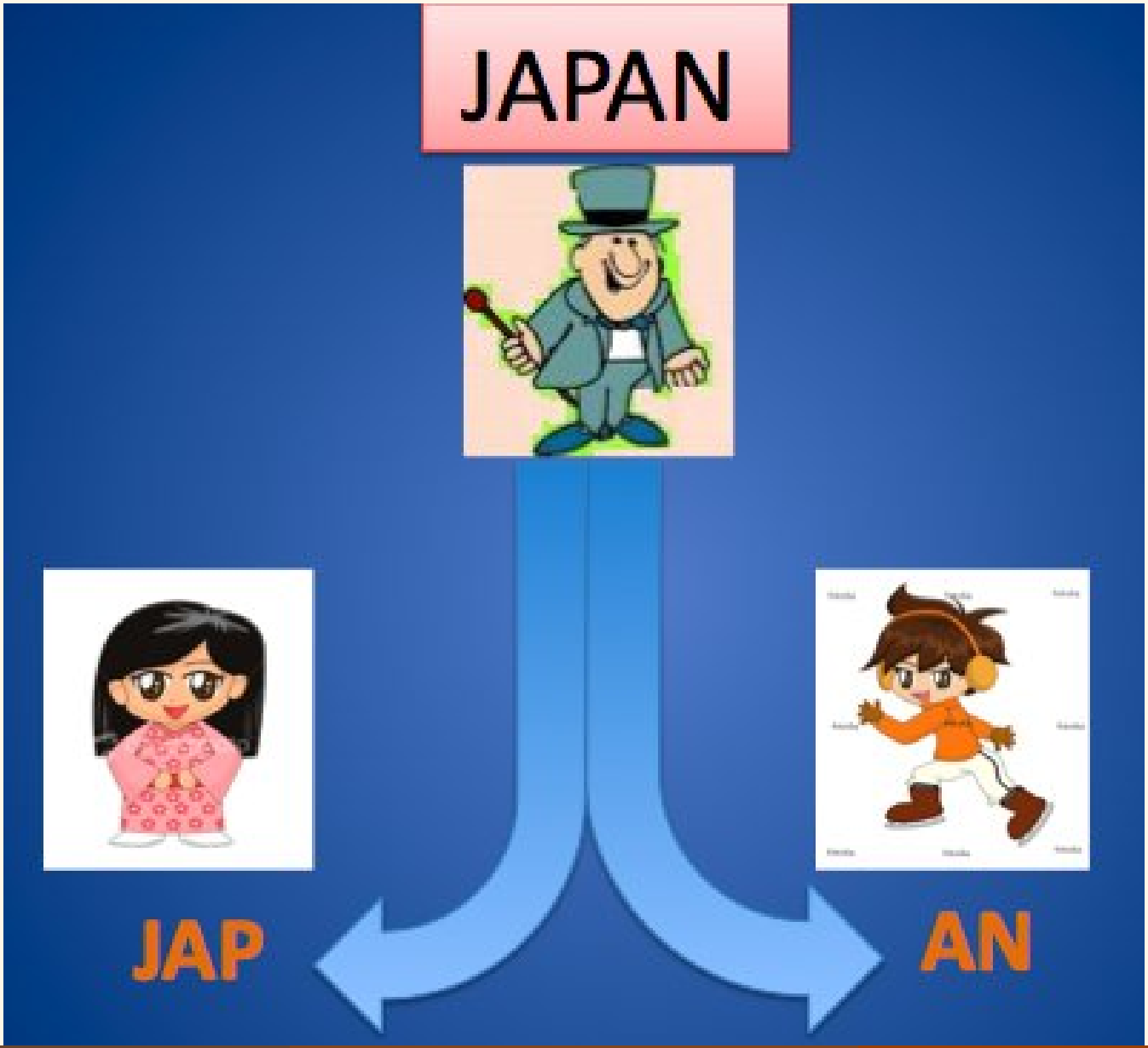
Credit Card and its Pin

Credit Card and its Pin

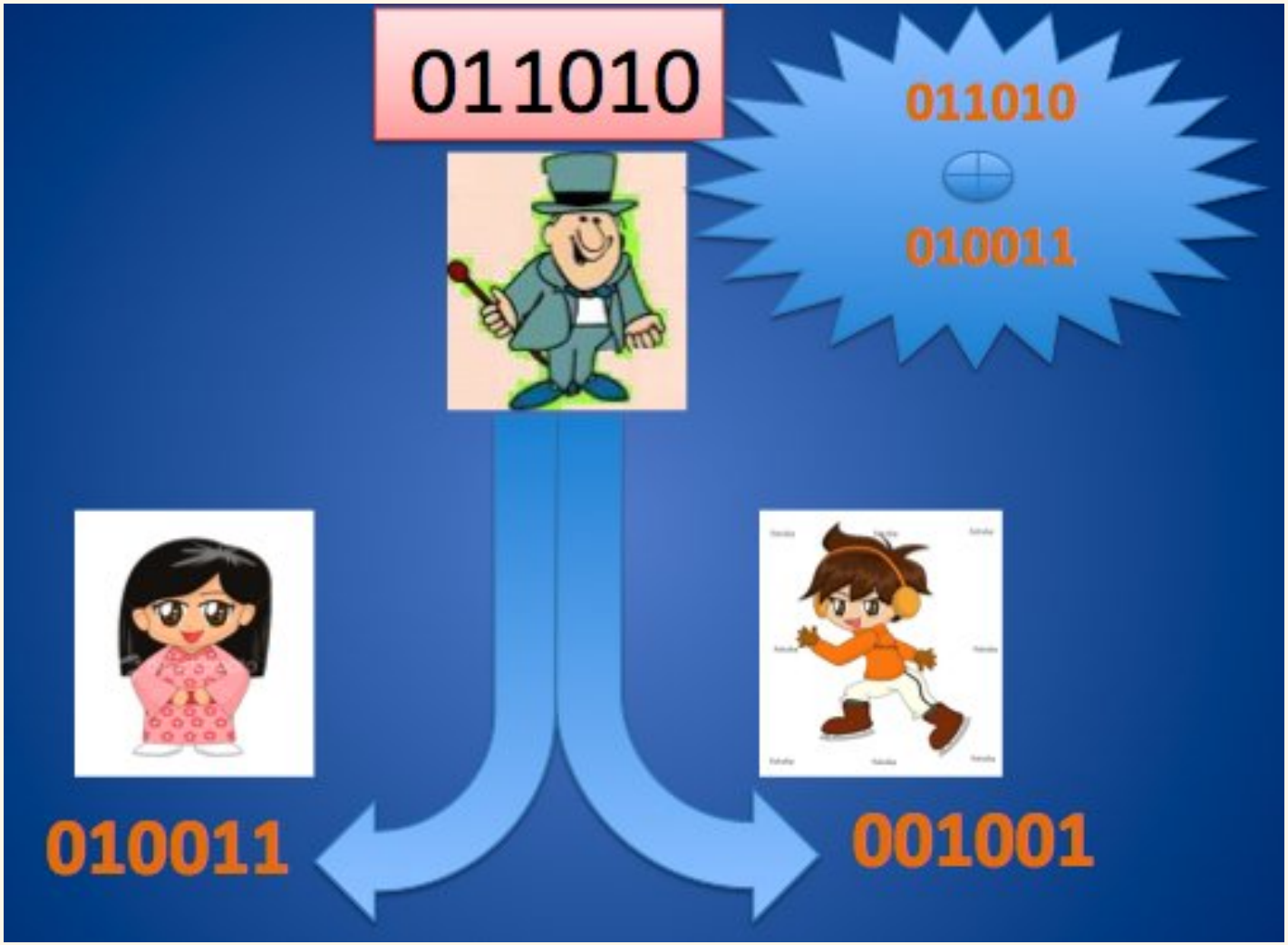
Multi-Level Authentications



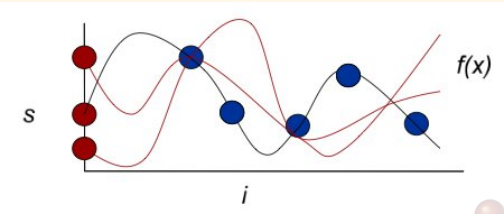
Simple Way!



Perfectly Secure (2,2)-SSS

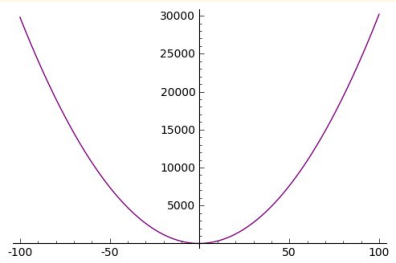


Shamir's (k, n)-Secret Sharing Scheme



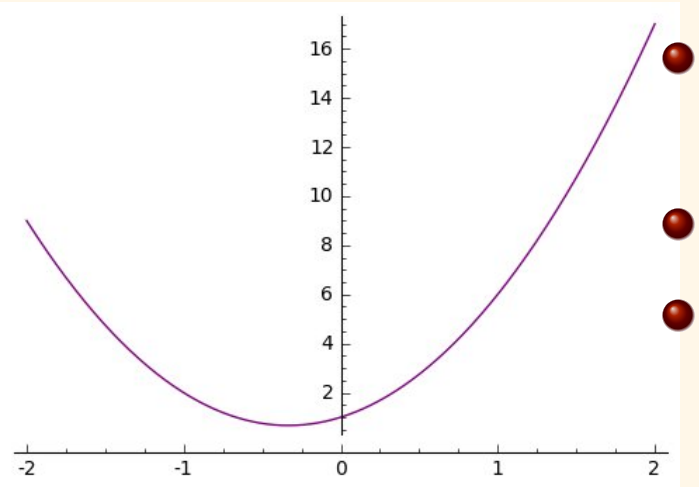
- It takes two points to define a straight line, three points to fully define a quadratic, four points to define a cubic, and so on.
- One can fit a unique polynomial of degree $(k-1)$ to any set of k points that lie on the polynomial.

Shamir's (k, n)-Secret Sharing Scheme



- It takes two points to define a straight line, three points to fully define a quadratic, four points to define a cubic, and so on.
- One can fit a unique polynomial of degree $(k-1)$ to any set of k points that lie on the polynomial.

Shamir's (3, 4) threshold scheme



- Let $\mathcal{P}=\{P_1, P_2, P_3, P_4\}$ be a set of 4 participants.
- The key set, $\mathcal{K}=\mathbb{Z}_p$, where $p = 5$ is a prime & $p > n$. Let the secret be 1.
- The set of all possible shares, $\mathcal{S}=\mathbb{Z}_5$.
- The dealer constructs a random polynomial $f(x) \in \mathbb{Z}_5[x]$ of degree $t - 1 = 3 - 1 = 2$, in which the constant term is the secret $K = 1$.

$$f(x) = 1 + 2x + 3x^2$$

Shamir's (3, 4) threshold scheme

- Every participant P_i obtains a point (x_i, y_i) on this polynomial, where $y_i = f(x_i)$ and distinct $x_i \in \mathbb{Z}_p$.
- P_1 gets (1,a(1)=6=1), (P_2) gets (2,2), P_3 gets $(3, 4)$ and P_4 gets (4,2).

Recovery of Secret

- Suppose a subset B of $t = 3$ participants wants to recollect the secret.
- Let the participants P_1, P_2, P_3 want to determine $K = 1$.
- They know that $1 = f(1)$, $2 = f(2)$ and $4 = f(3)$.
- They will assume the form of the secret polynomial as $y = f(x) = a_0 + a_1 x + a_2 x^2$, where a_0, a_1 and a_2 are unknown and belong to $\mathbb{Z}$.
- Thus, these participants can obtain 3 linear equations in the 3 unknowns a_0, a_1, a_2.

Shamir's (t, n) threshold scheme

- $$\begin{bmatrix} 1 & 1 & 1^2 \\ 1 & 2 & 2^2 \\ 1 & 3 & 3^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} f(1) \\ f(2) \\ f(3) \end{bmatrix}$$
  - Now, the coefficient matrix A is the so called Vandermonde's matrix.

$$detA = \prod_{1 \le j < k \le t} (x_{i_k} - x_{i_j}) \mod p = (1-2)(2-3)(3-1) = 4*4*2 = 2 \neq 0$$

Thus multiplying both sides by the inverse of A, we can find the $a_0 = 1$.

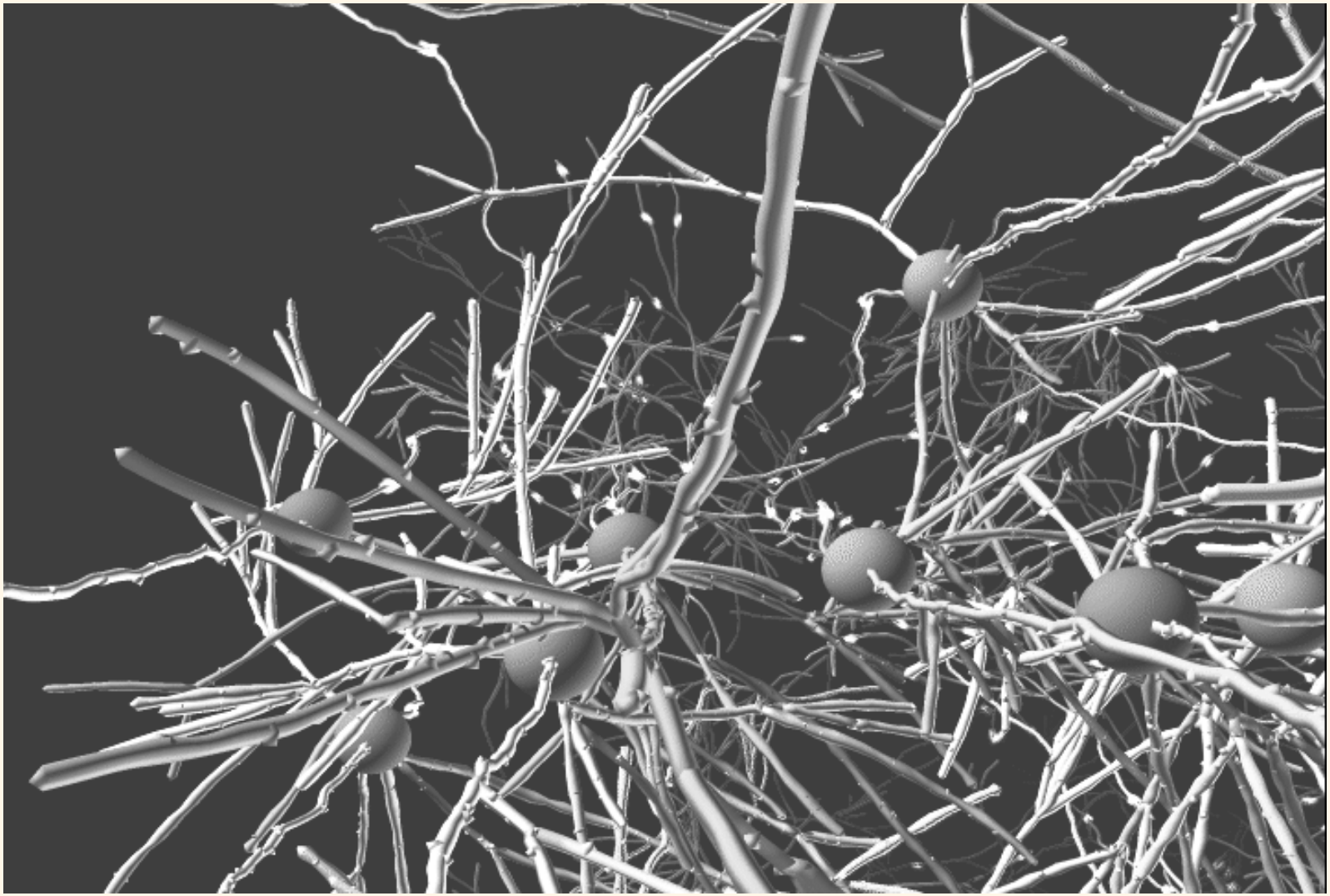
Shamir's (t, n) threshold scheme

- $$\begin{bmatrix} 1 & 1 & 1^2 \\ 1 & 2 & 2^2 \\ 1 & 3 & 3^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} f(1) \\ f(2) \\ f(3) \end{bmatrix}$$
- Now, the coefficient matrix A is the so called Vandermonde's matrix.

$$detA = \prod_{1 \le j < k \le t} (x_{i_k} - x_{i_j}) \mod p = (1-2)(2-3)(3-1) = 4*4*2 = 2 \neq 0$$

Thus multiplying both sides by the inverse of A, we can find the $a_0 = 1$.

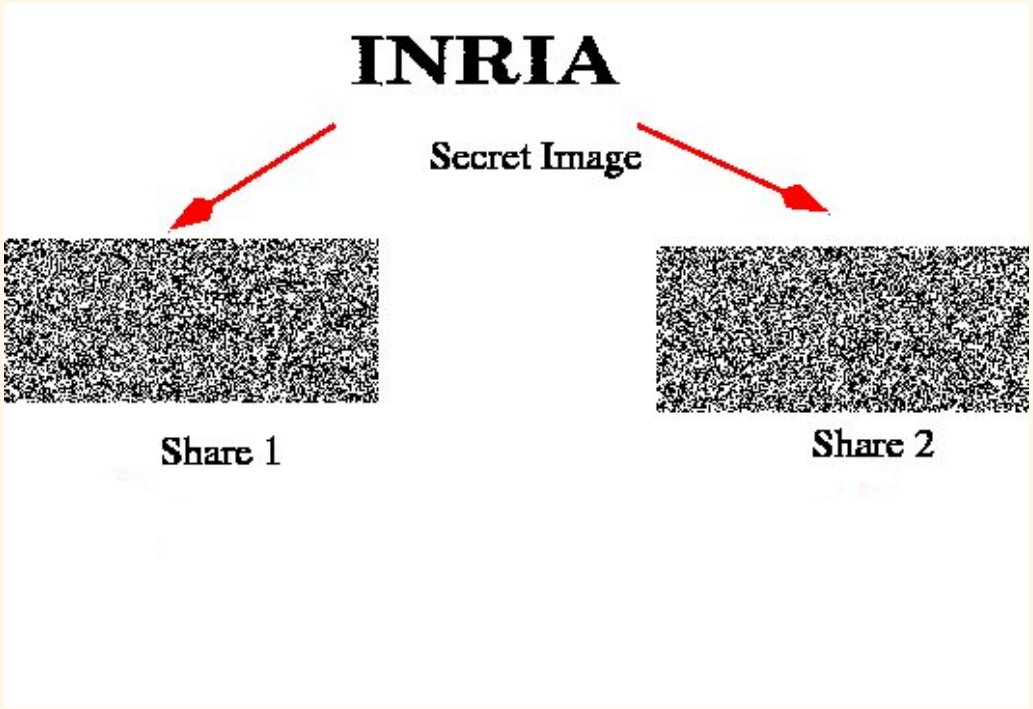
When the Image is a Secret: pgm Extension



Text Version of the Image: pgm Extension

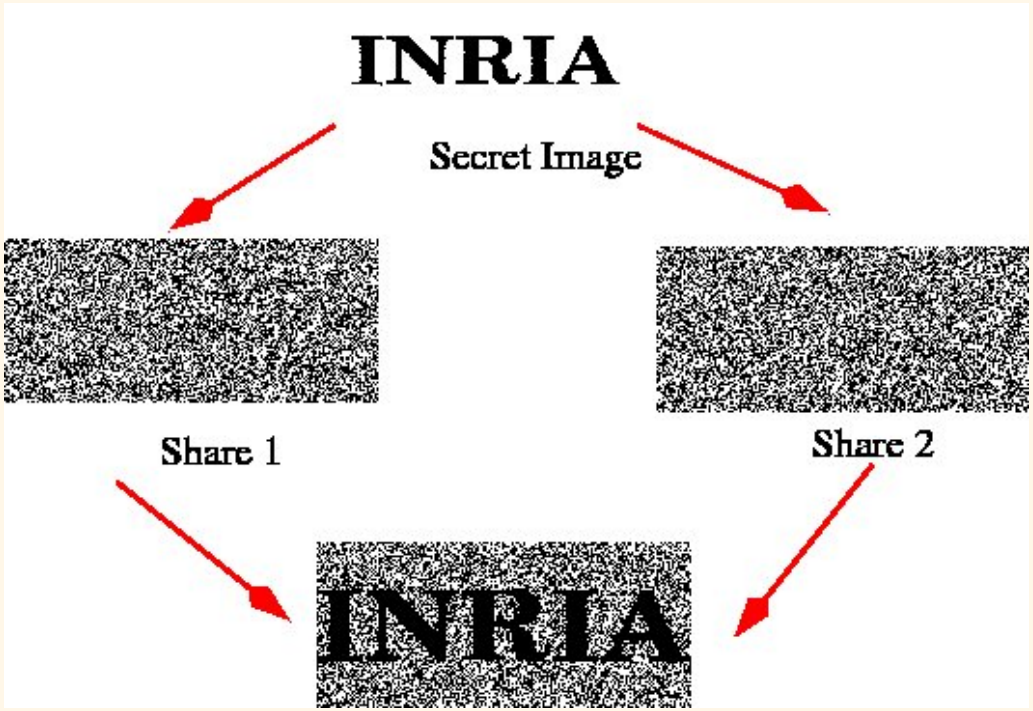
```
P2
# created by 'xv brain_604.tif'
720 486
255
 60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60
 60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60
 60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60
 60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60
 60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60
 60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60
 60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60
 60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60
 60  60  60  60  60  60  60  60  80 107 107 110 120 110 123 123 129
129 139 140 139 153 140 153 153 165 165 165 165 182 182 182 189 182
139  66  60  60  60  72 139 236 247 255 255 171 104  66  60  60  60
 60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60
 60  60  60  60  60  60  60  60  60  60  60  90 153 214 223 218 214
193 123 193 193 190 168 153 126 103  71  60  60  60  60  60  60  60
 60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60
 60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60
 60  60  60  60  60  87 106 145 163 214 212 212 193 190 168 157 145
134 123 116 113 116 113 116 123 116 123 123 116 113 106 106  97  97
 97  90  87  87  87  87  87  87  87  94  97  97 103 103 106 113 116
123 134 134 126 123 113 106 103  97  97  97  94 113 138 134 145 174
212 225 223 180 116  66  60  60  60  60  60  60  60  60  60  60  60
 60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60
 60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60
 60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60
 60  60  60  60  60  60  60  60  60  60  60  60 161 255 255 225 143
 90 118 215 255 218  80 106 214 214 193 193 180 159 145 113  97  97
 97 103  99 106 126 153 180 193 214 225 225 225  87  60  60  60  60
 60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60
 60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60
 60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60
 60  60  60  60  60  60  60  60  60  60  60  60  60  60  66  87 100
115 140 153 165 153 153 120  66  60  60  60  60  60  60  60  60  60
 60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60
 60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60
 60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60
 60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60
 60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60
```

Example of $(2, 2)$-Visual Cryptographic Scheme (VCS)



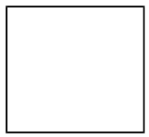
'x'

Example of (2, 2)-Visual Cryptographic Scheme (VCS)



'x'

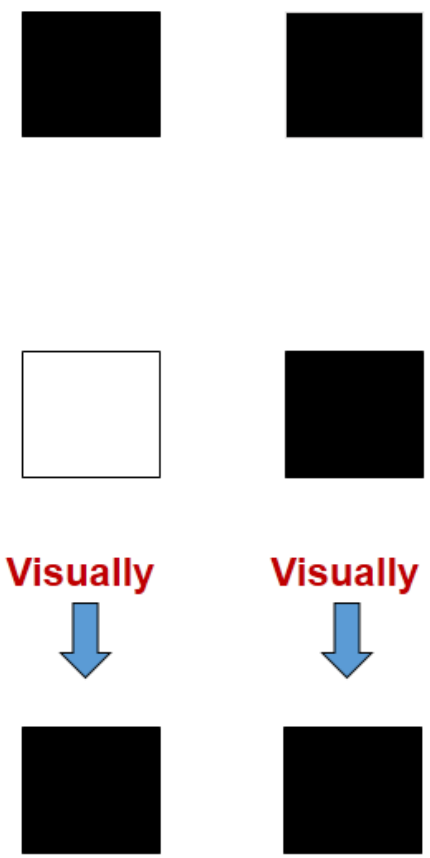
Superimposition of pixels



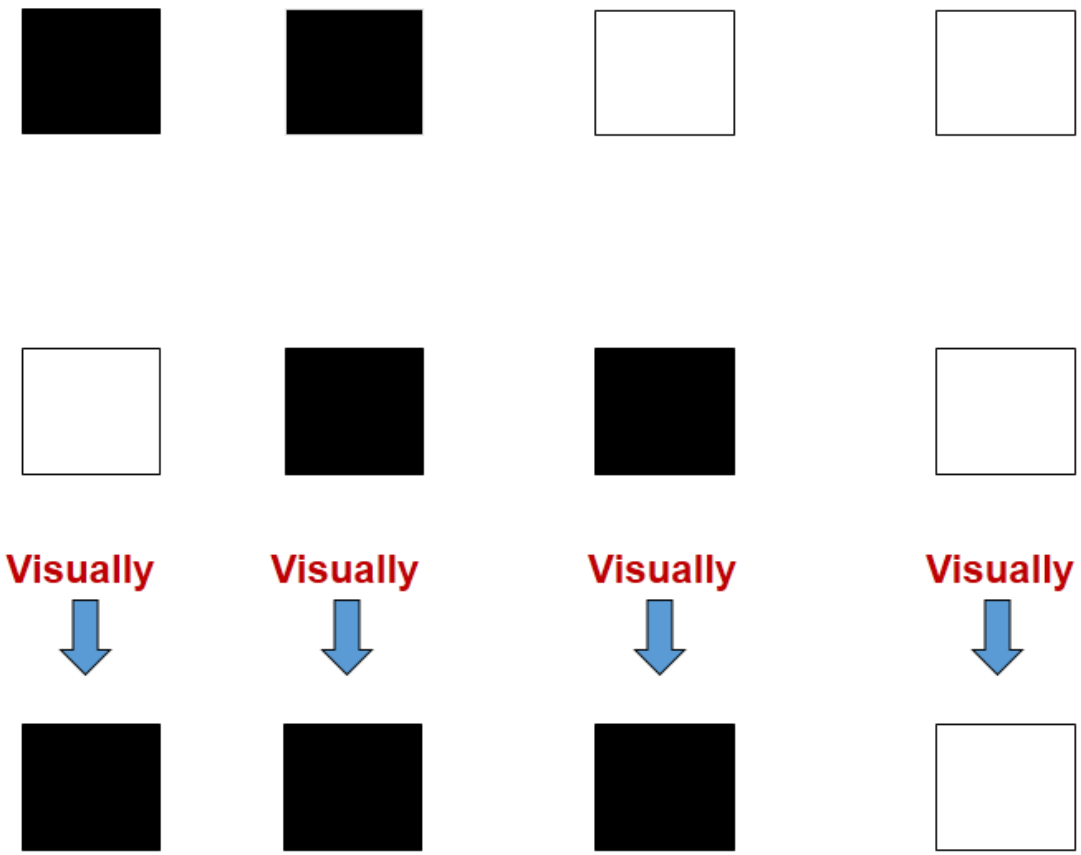
Superimposition of pixels

Visually

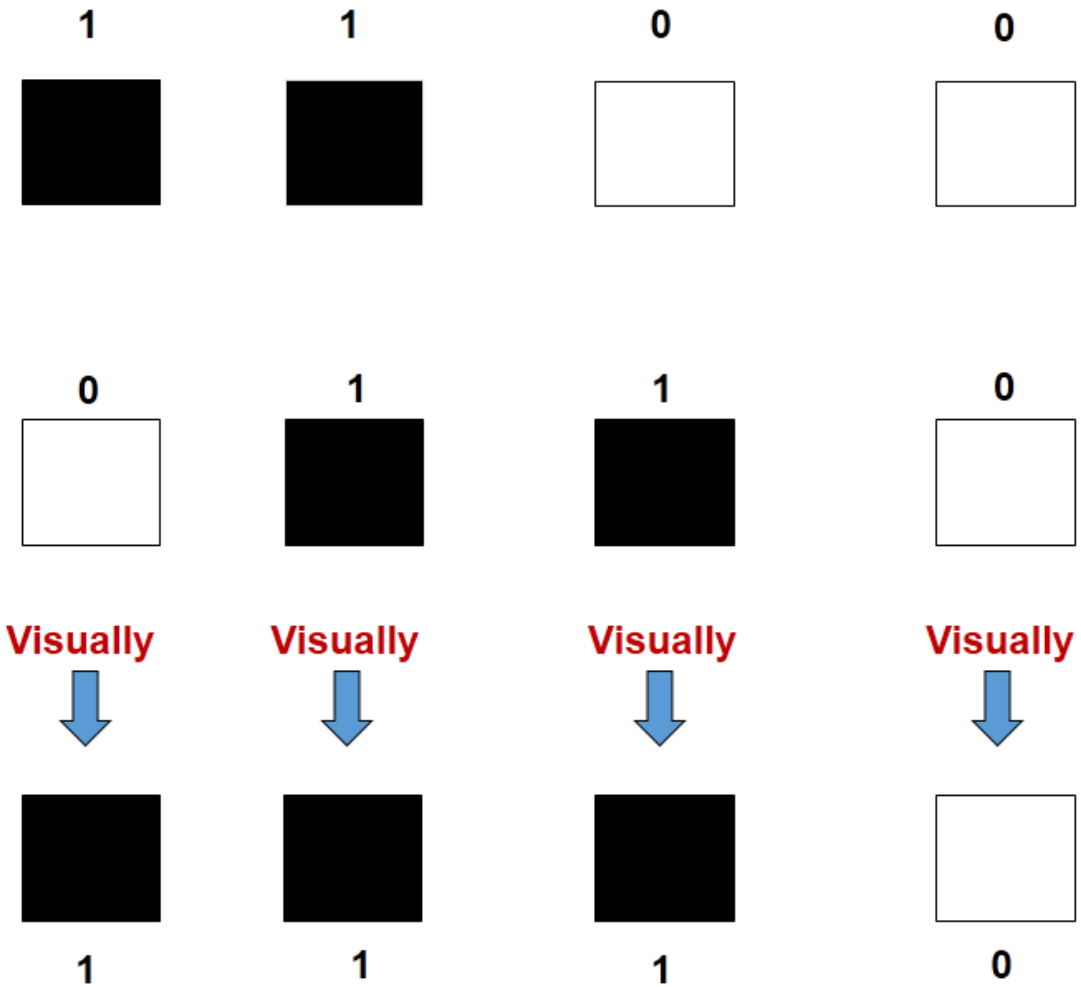
Superimposition of pixels



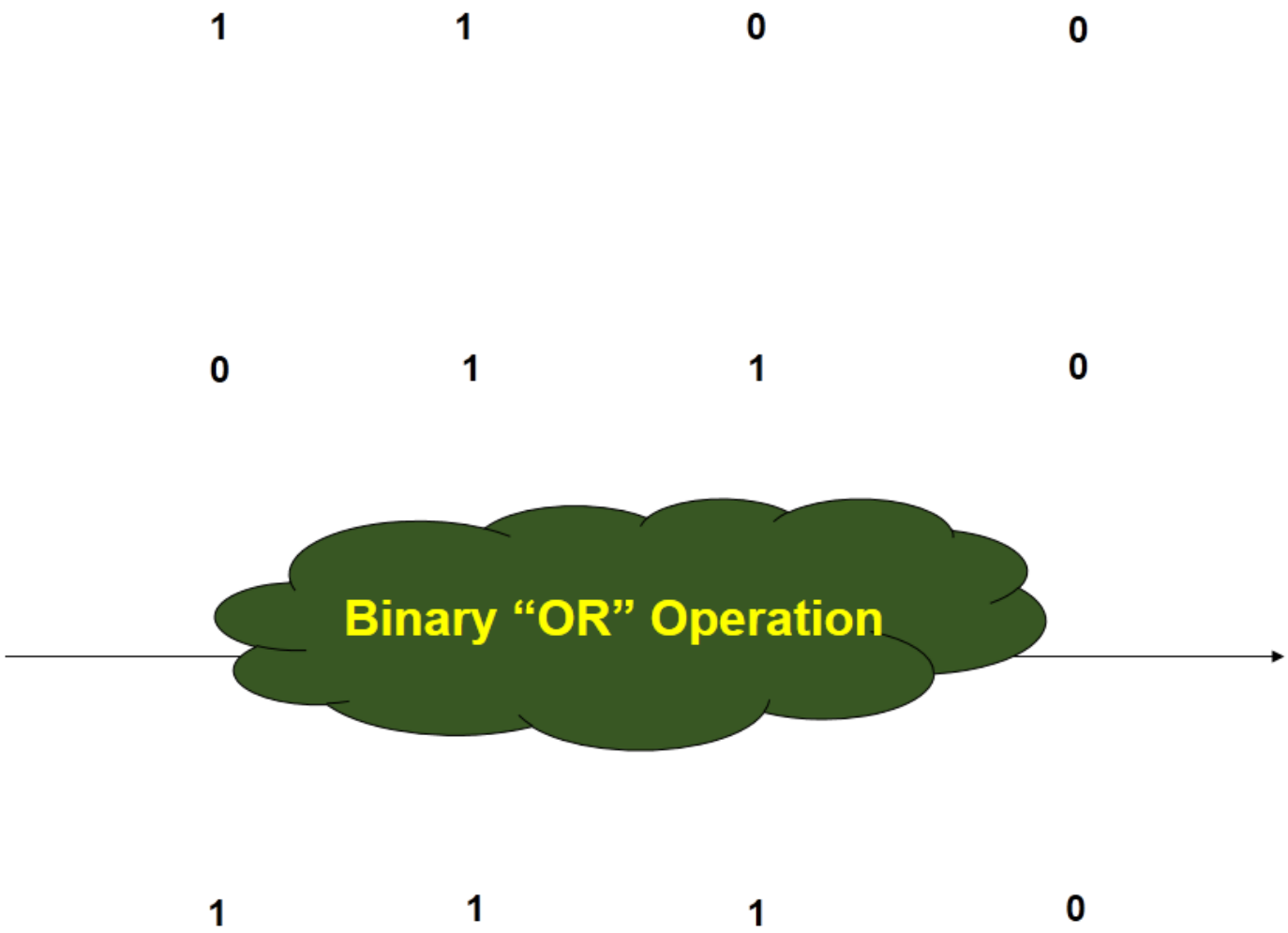
Superimposition of pixels



Superimposition of pixels

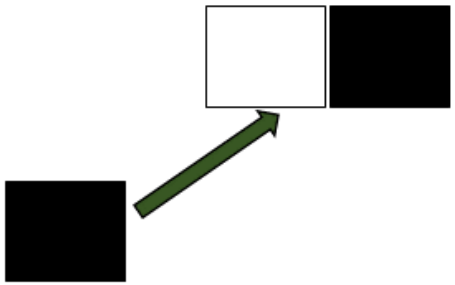


Superimposition of pixels

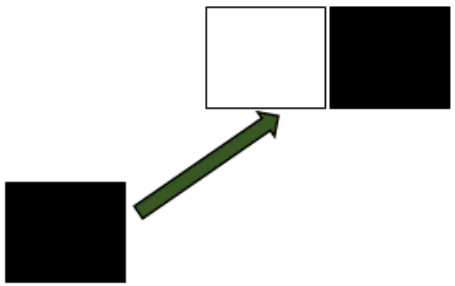


$(2, 2)$-VCS

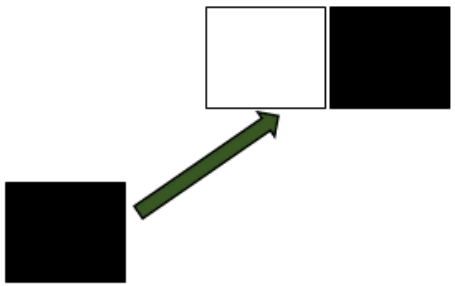
(2, 2)-VCS



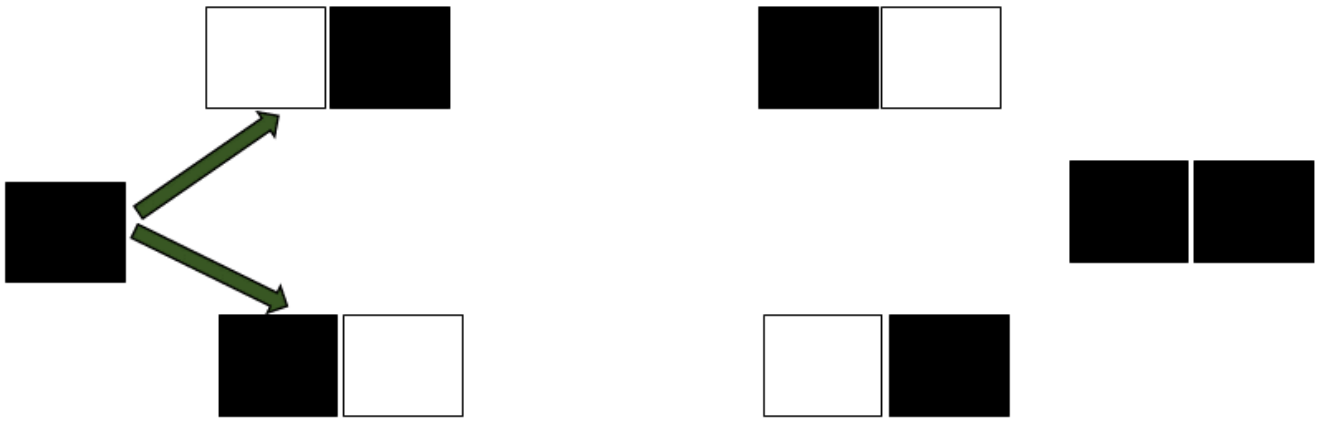
$(2, 2)$-VCS



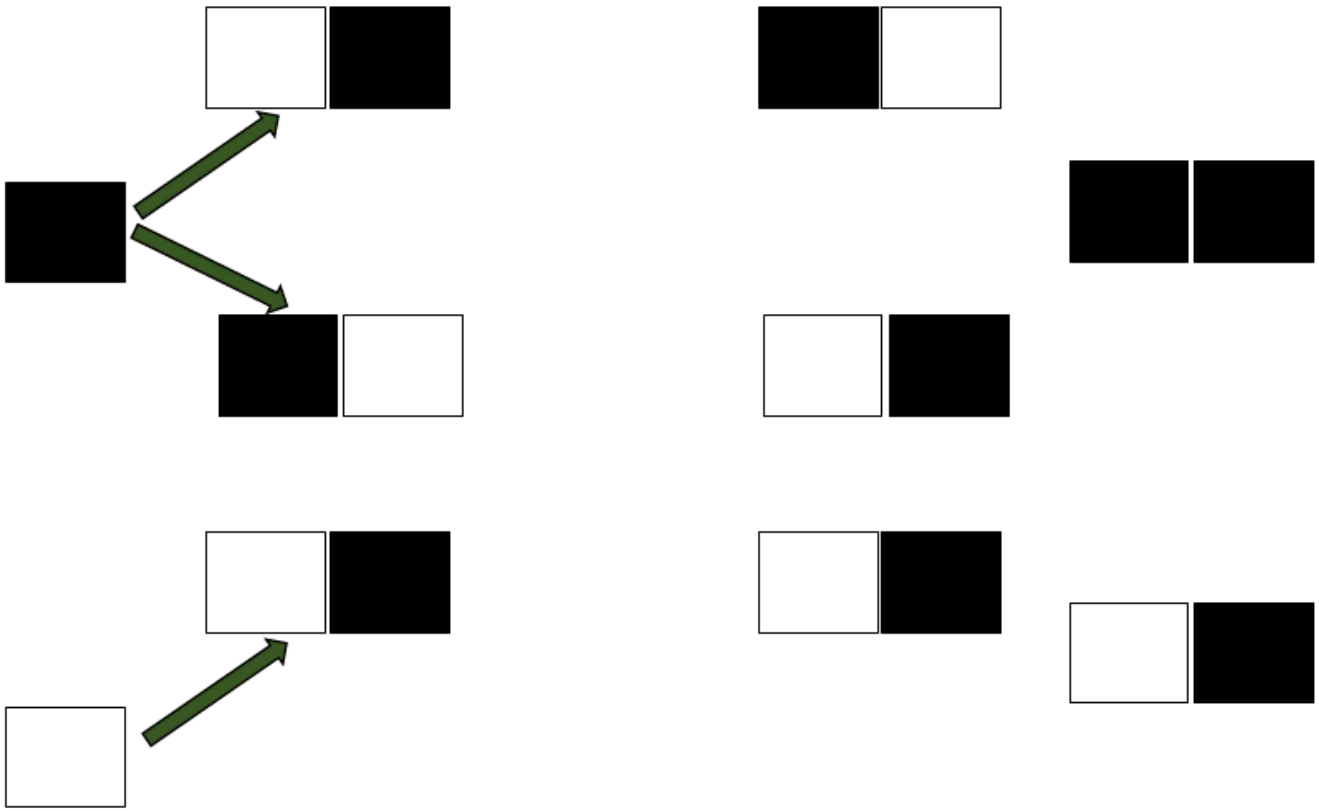
(2, 2)-VCS



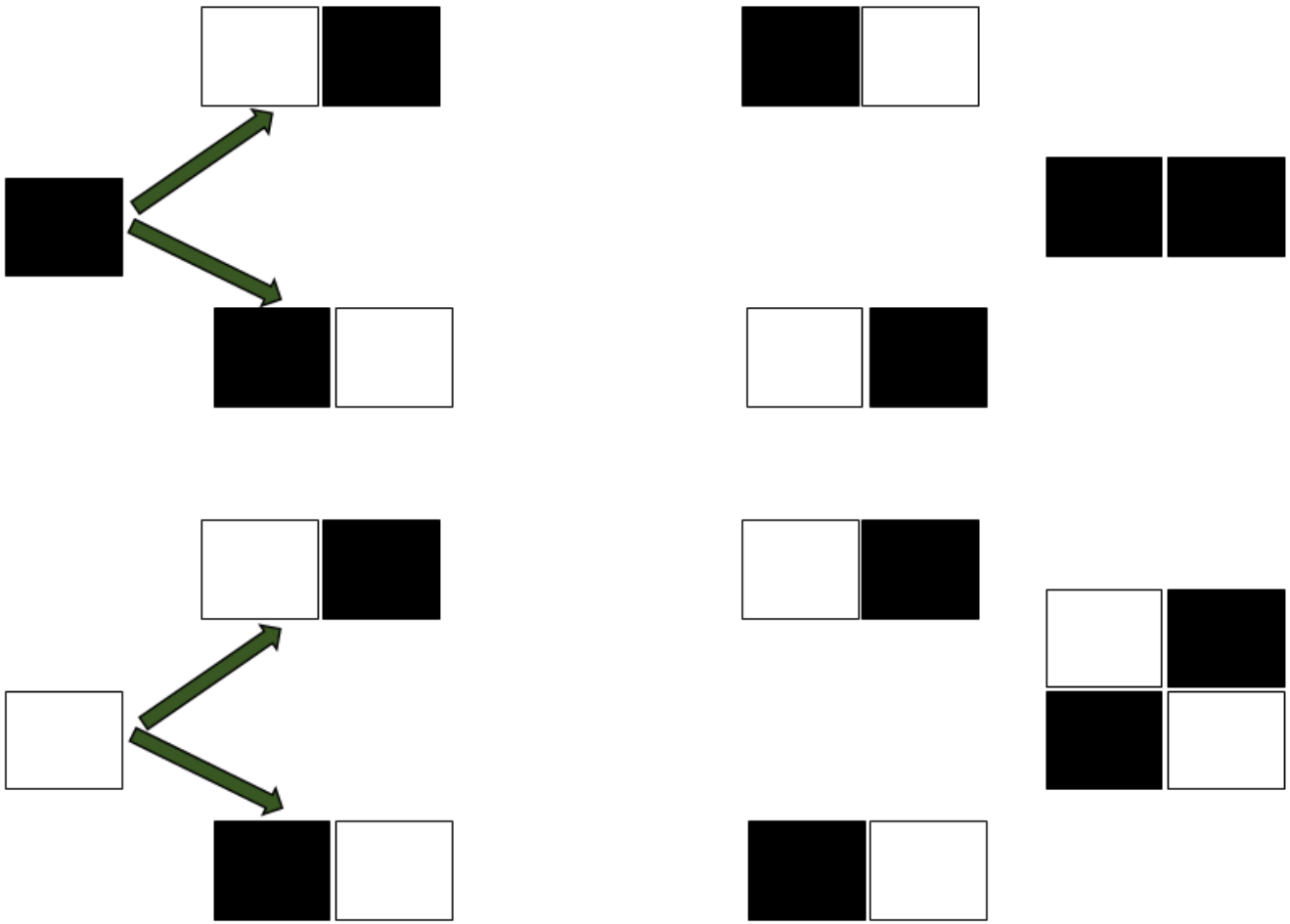
(2, 2)-VCS



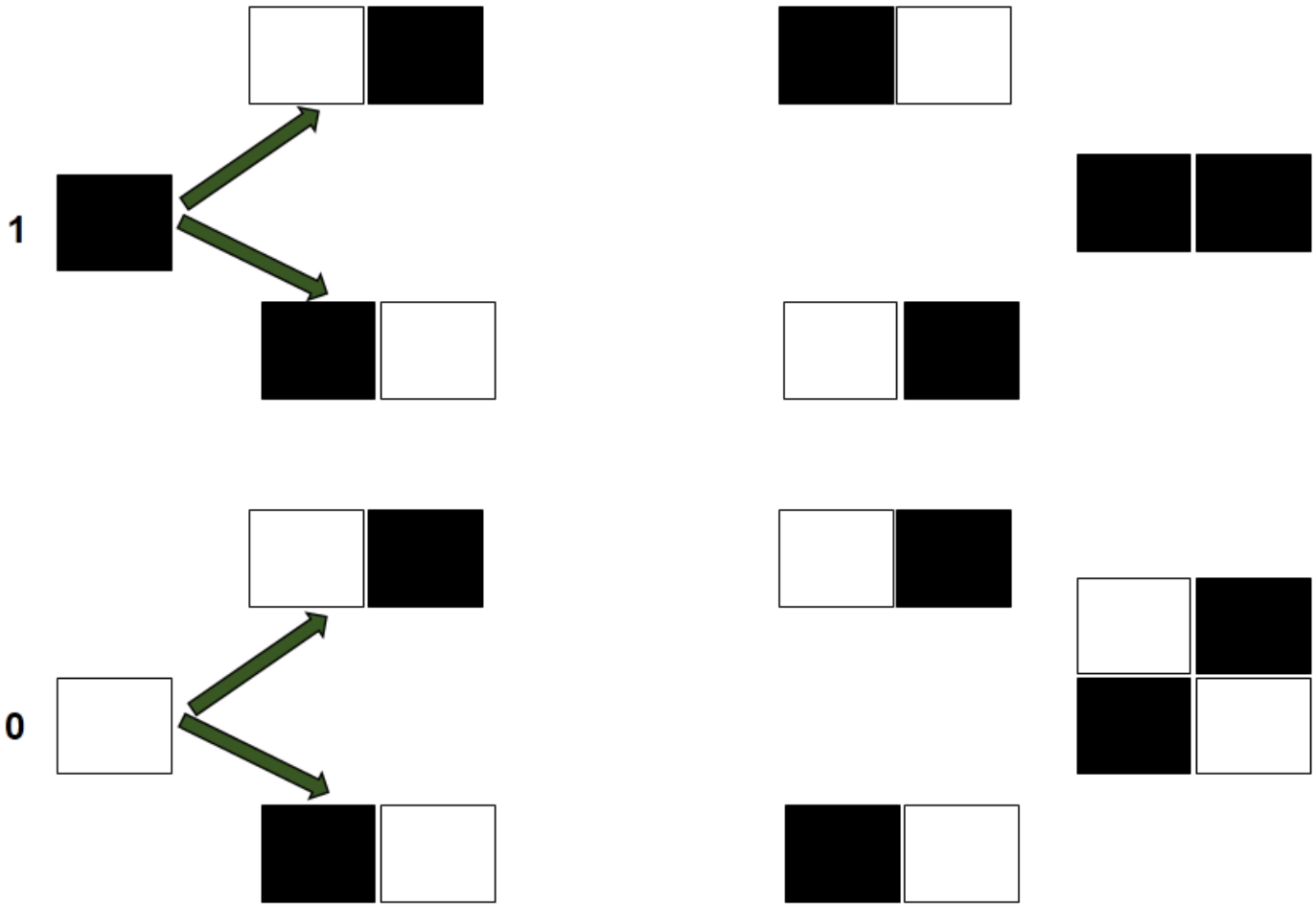
(2, 2)-VCS



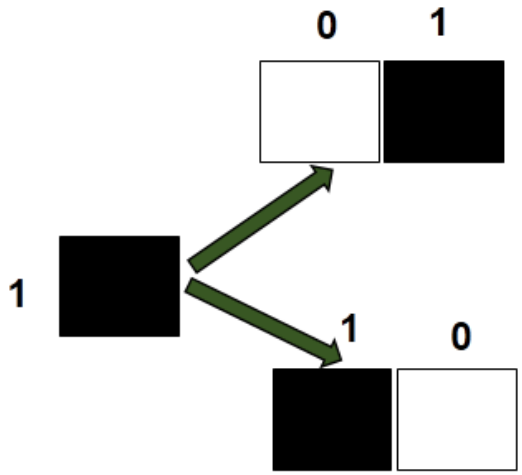
(2, 2)-VCS

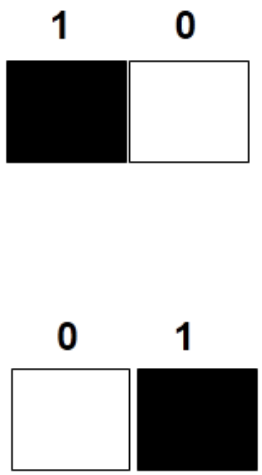


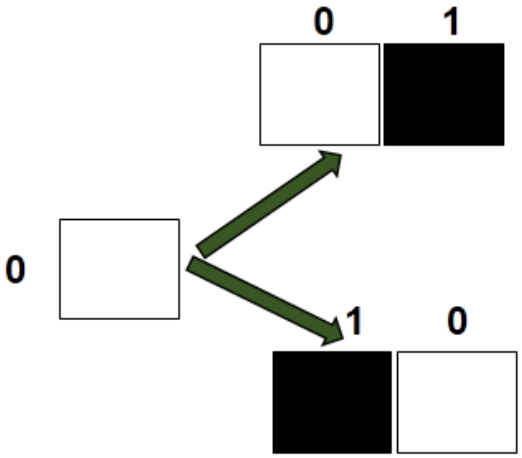
(2, 2)-VCS

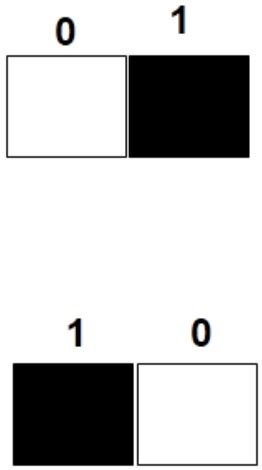


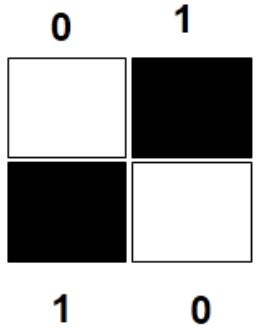
(2, 2)-VCS



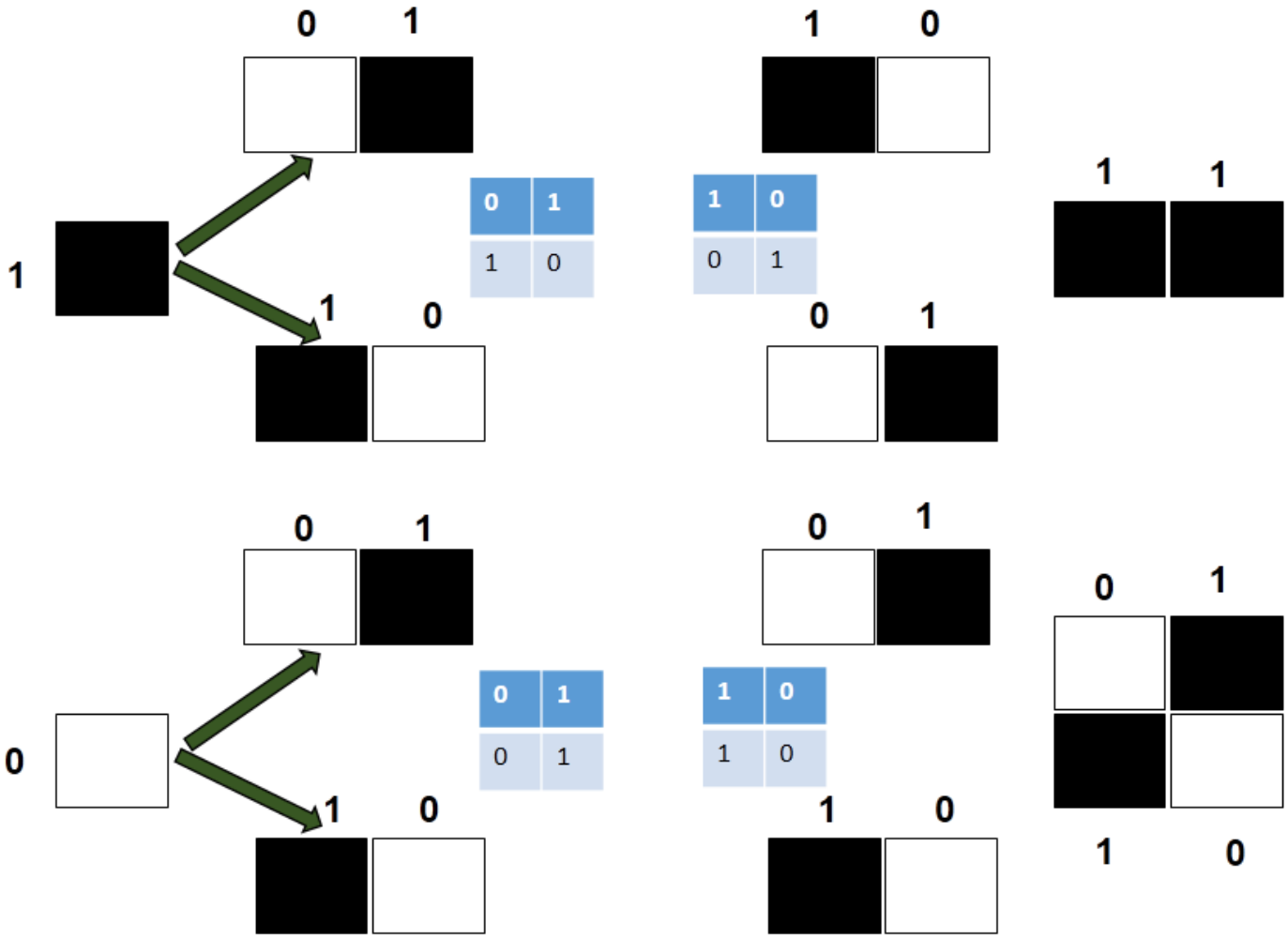




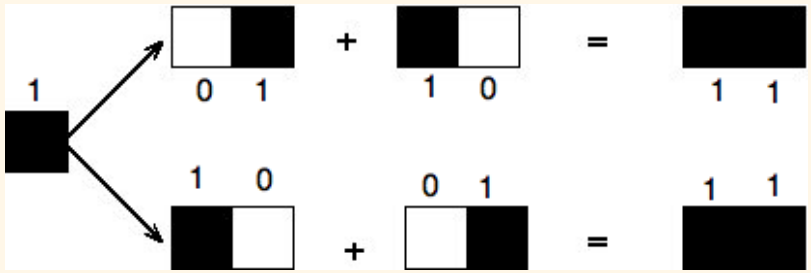




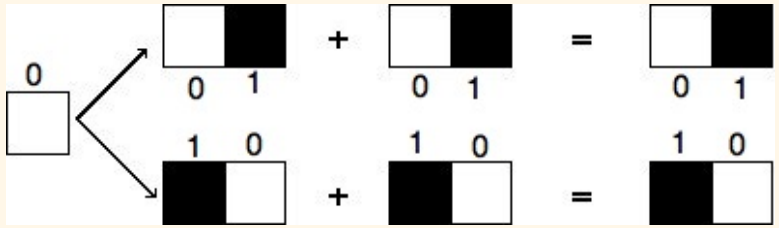
(2, 2)-VCS



(2, 2)-VCS



For Black pixel



For White pixel

$$S^1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \text{ and } S^0 = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}.$$

Relative contrast

Let us consider a $(2, n)$-VCS on a set $\mathcal{P} = \{1, 2, \ldots, n\}$ of n participants with basis matrices S^0 and S^1 and having *pixel expansion m* which is the number of columns of the basis matrices. Then the *relative contrast* for the participants corresponding to X, $X \subseteq \mathcal{P}$, is denoted by $\alpha_X(m)$ and is defined as

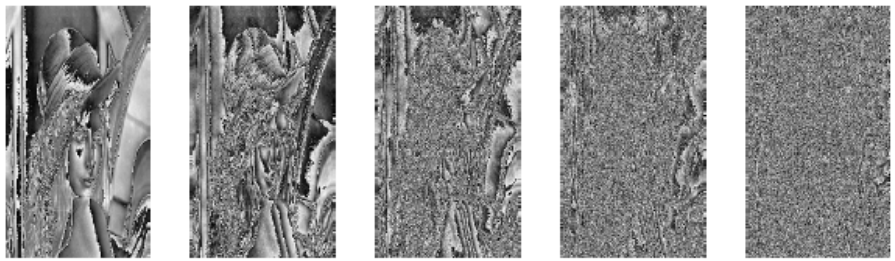
$$\frac{w(S^1_X) - w(S^0_X)}{m}.$$

Few Significant Contributions on VCS

1. Sabyasachi Dutta, Avishek Adhikari, Sushmita Ruj: Maximal contrast color visual secret sharing schemes. **Des. Codes Cryptogr. 87(7): 1699-1711 (2019)**.
2. Sabyasachi Dutta, Raghvendra Singh Rohit, Avishek Adhikari: Constructions and analysis of some efficient t-$(k, n)^*$-visual cryptographic schemes using linear algebraic techniques. **Des. Codes Cryptogr. 80(1): 165-196 (2016)**.
3. Avishek Adhikari: Linear algebraic techniques to construct monochrome visual cryptographic schemes for general access structure and its applications to color images. **Des. Codes Cryptogr.** 73(3): 865-895 (2014).

(k, n)-Image Sharing Scheme (Thien and Lin, '02)

Secret



S_1 S_2 S_3 S_4 S_5

$S_{\{1,2\}}$ $S_{\{1,2,3\}}$ $S_{\{1,2,3,4\}}$

Polynomial Based SIS: Same Share Size

- Hu, W., Wu, T., Chen, Y., Shen, Y., and Yuan, L. (2021), A lossless secret image sharing scheme using a larger finite field.
Multimedia Tools and Applications, 80.
- Gong, Q., Wang, Y., Yan, X., and Liu, L. (2019), Efficient and lossless polynomial-based secret image sharing for color images.
IEEE Access, 7:113216–113222.
- Ding, W., Liu, K., Yan, X., and Liu, L. (2018).
Polynomial-based secret image sharing scheme with fully lossless recovery, *Int. J. Digit. Crime For.*, 10(2):120?136.
- Kanso, A. and Ghebleh, M. (2017), An efficient (t, n)–threshold secret image sharing scheme.
Multimedia Tools Appl., 76(15):16369–16388.
- Kanso, A., Ghebleh, M., and Alazemi, A. (2020).
A lossless linear algebraic secret image sharing scheme, *Informatica*, pages 1–24.

Polynomial Based SIS: Reduced Share Size

Sardar, M. K. and Adhikari, A. (2020a).
A New Lossless Secret Color Image Sharing Scheme with Small Shadow Size.
Journal of Visual Communication and Image Representation, page 102768.

Sardar, M. K. and Adhikari, A. (2020b).
Essential secret image sharing scheme with small and equal sized shadows.
Signal Processing: Image Communication, 87:115923.

Thien, C.-C. and Lin, J.-C. (2002).
Secret Image Sharing.
Computers & Graphics, 26(5):765 – 770.

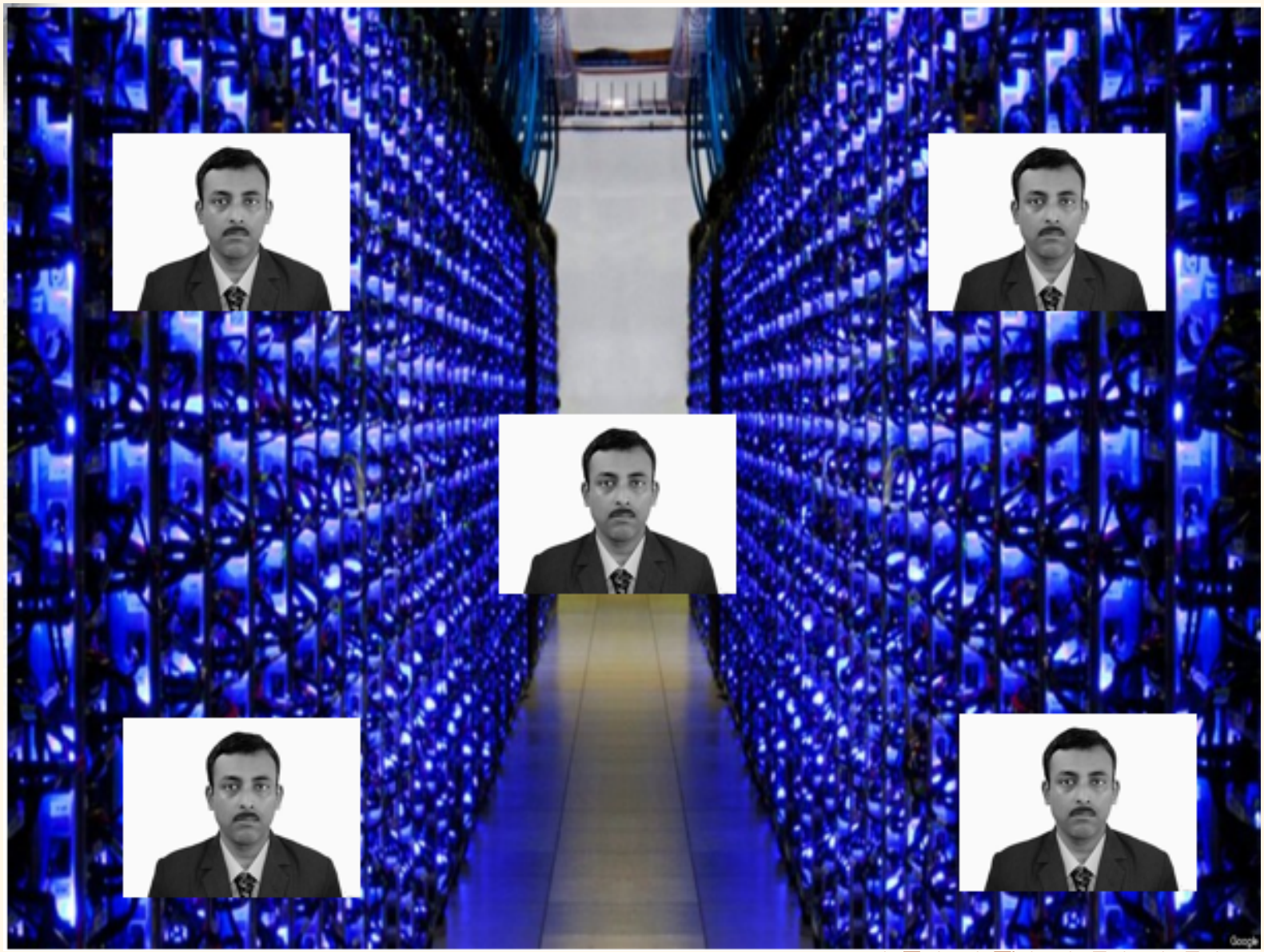
Li, P., Liu, Z., and Yang, C. (2018).
A construction method of (t, k, n)-essential secret image sharing scheme.

Signal Process. Image Commun., 65:210–220.

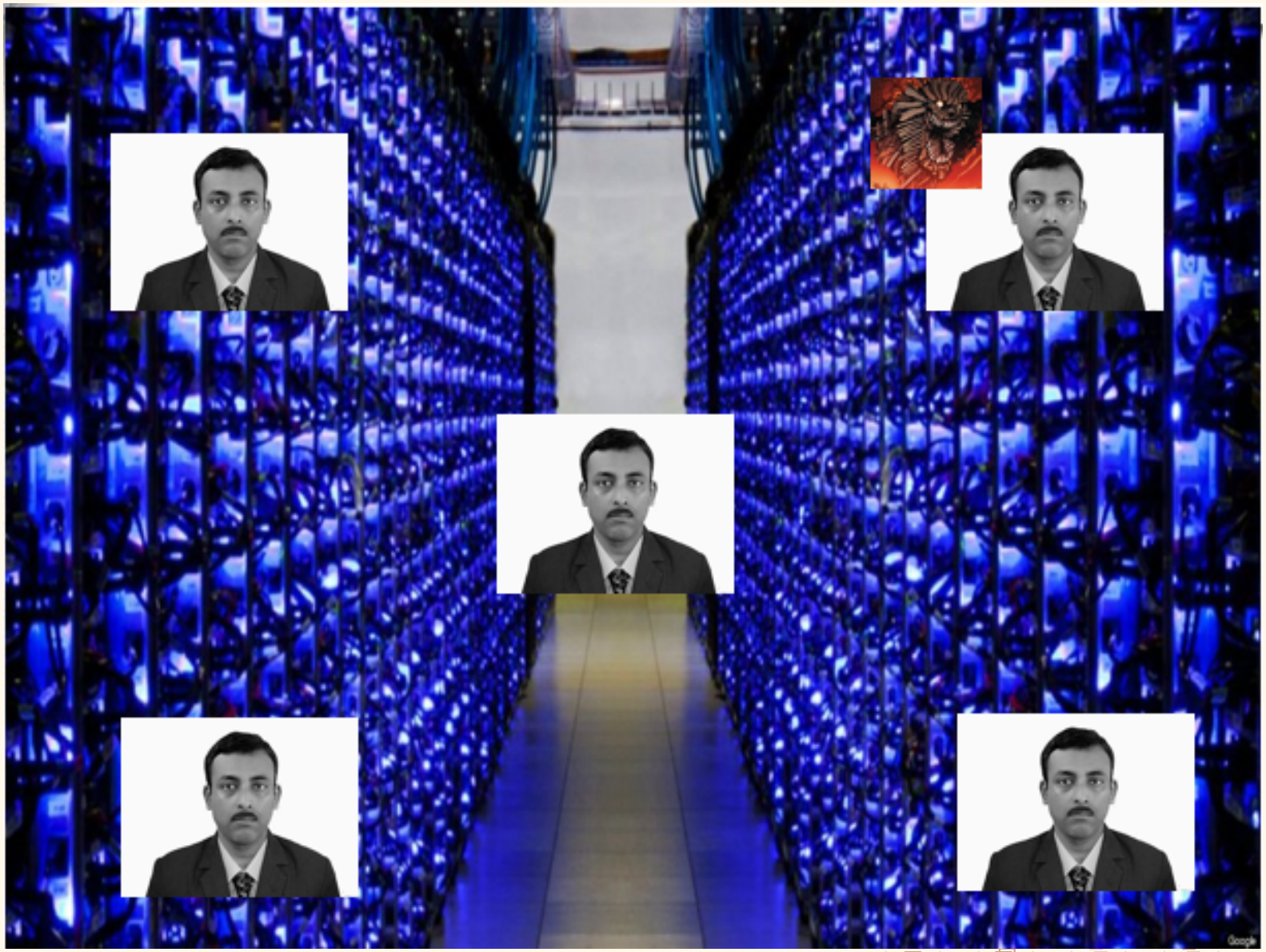
Attacks on (k, n)-SISS

Attacks on (k, n)-SISS

Attacks on (k, n)-SISS



Attacks on (k, n)-SISS



Attacks on (k, n)-SISS

Attacks on (k, n)-SISS

Attacks on (k, n)-SISS

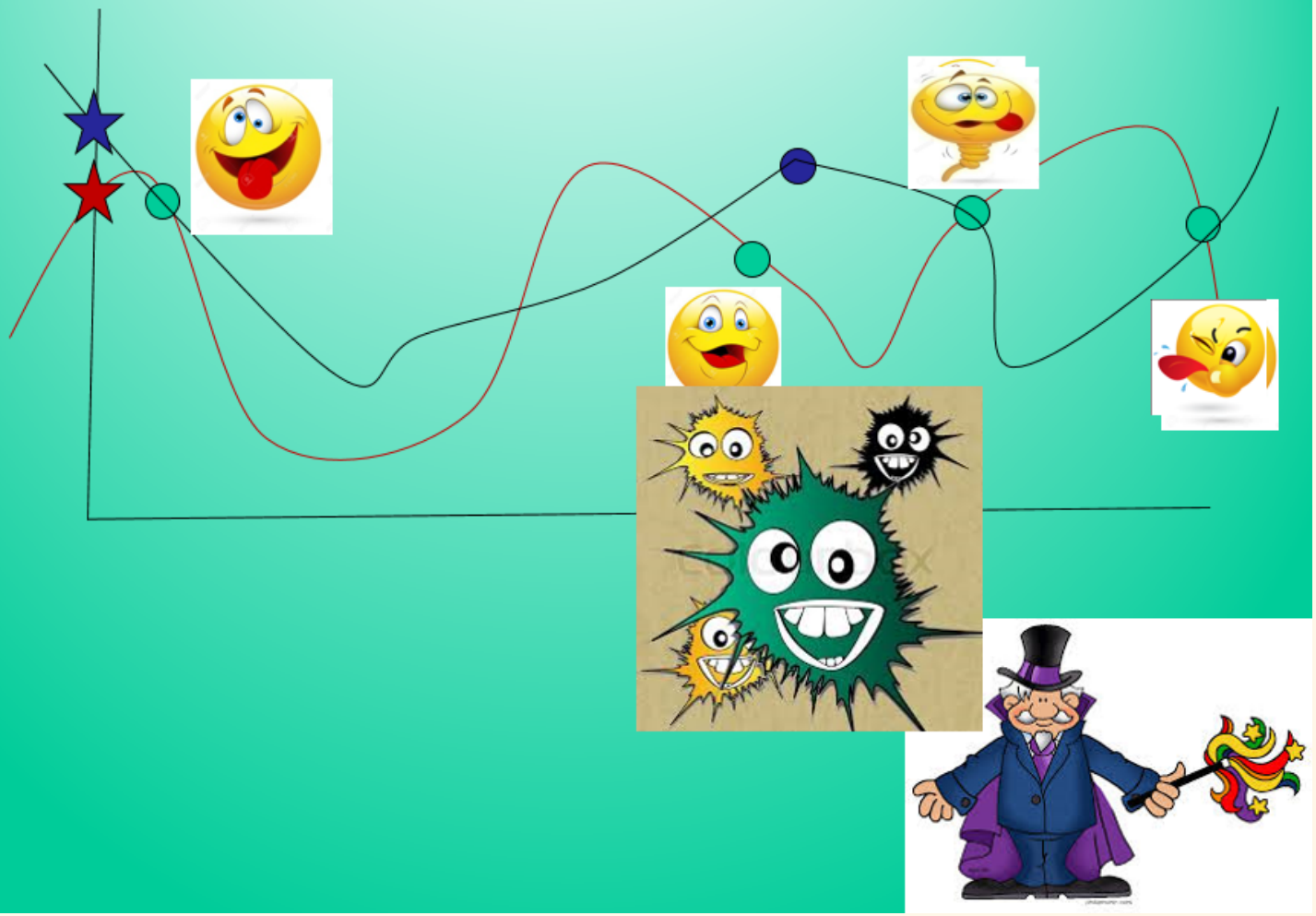
Attacks on (k, n)-SISS

Attacks on (k, n)-SISS

Attacks on (k, n)-SISS

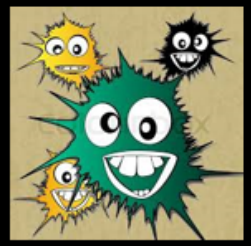
Attacks on (k, n)-SISS

Twist in the Story!



Characterization of Adversarial Activities

Adversary



Honest but Curious

Malicious

Non-Rushing

Rushing

Share Generation Algo for SIS using SSSS

Algorithm 1: Share Generation of Shamir's (k, n) - SIS scheme Using Public ID

Input: A secret image S of size $M \times N$, i.e., $S[i, j], 0 \leq i \leq M-1, 0 \leq j \leq N-1$, and three integers k and n, such that $1 < k \leq n$.

Output: n shadow images $S_1, S_2, \ldots, S_n$ with a proxy share S'_t each of size $M \times N$.

Assume a bijective mapping $\phi : \mathbb{Z}_{2^8} \rightarrow \mathrm{GF}(2^8)$.
Choose an irreducible polynomial $q(x) \in F_p[x]$ of degree 8.
for $(i = 0; i < M; i++)$ **do**
  for $(j = 0; j < N; j++)$ **do**
    Assign the coefficients $a_0, a_1, \ldots, a_{k-1} \in \mathrm{GF}(2^8)$ as follows:
    $a_0 \leftarrow \phi(S[i, j])$
    for $(q = 1; q < k; q++)$ **do**
      Generate $a_q \xleftarrow{\$} \mathrm{GF}(2^8)$ uniformly at random.
    end
    for $(r = 1; r \leq n; r++)$ **do**
      $f(\alpha_r) \leftarrow (a_0 + a_1 \cdot \alpha_r + a_2 \cdot \alpha_r^2 + \cdots + a_{k-1} \cdot \alpha_r^{k-1}) \pmod{q(x)}$
      $S_r[i, j] = \phi^{-1}(f(\alpha_r))$
    end
  end
end

Attack on SSSS

Algorithm 4: An Attack on Shamir's (k, n) - CSIS scheme Using Public ID

Input: A secret image S of size $M \times N$, i.e., $S[i, j], 0 \leq i \leq M-1, 0 \leq j \leq N-1$, and three integers k and n, such that $1 < k \leq n$.

Output: n shadow images $S_1, S_2, \ldots, S_n$ with a proxy share S'_t each of size $M \times N$.

1 **for** $(i = 0; i < M; i++)$ **do**
2 **for** $(j = 0; j < N; j++)$ **do**
3 $g(y) = (y - \alpha_1)(y - \alpha_2) \cdots (y - \alpha_{t-1})(y - \alpha_{t+1}) \ldots (y - \alpha_k) \pmod{q(x)}$
4 $S'_t[i, j] = \phi^{-1}(g(\alpha_t) + f(\alpha_t))$
5 **end**
6 **end**

Share Generation Algo for Meaningful Images

Algorithm 5: Share generation of Shamir's (k, n)-SIS scheme with Private ID.

Input: A secret image S and a meaningful proxy image P each of of size $M \times N$, i.e., $S[i, j], 0 \leq i \leq M-1, 0 \leq j \leq N-1$, and three integers k and n, such that $1 < k \leq n$.

Output: n shadow images $S_1, S_2, \ldots, S_n$ with a proxy share S'_t each of size $M \times N$.

Choose an irreducible polynomial $q(x) \in F_p[x]$ of degree 8.
for $(i = 0; i < M; i++)$ **do**
  for $(j = 0; j < N; j++)$ **do**
    Assign the coefficients $a_0, a_1, \ldots, a_{k-1} \in \mathrm{GF}(2^8)$ as follows:
    $a_0 \leftarrow \phi(S[i, j])$
    for $(q = 1; q < k; q++)$ **do**
      randomly choose $a_q \in \mathrm{GF}(2^8)$
    end
    for $(r = 1; r \leq n; r++)$ **do**
      $f(\alpha_r) \leftarrow (a_0 + a_1 \cdot \alpha_r + a_2 \cdot \alpha_r^2 + \cdots + a_{k-1} \cdot \alpha_r^{k-1}) \pmod{q(x)}$
    end
    $$g(y) = \frac{(-1)^{k-1}}{\prod_{v=1, v \neq t}^{k} \phi(\alpha_v)} (\phi(P[i, j]) - \phi(S[i, j]))\,(y - \alpha_1)(y - \alpha_1) \cdots (y - \alpha_{t-1})(y - \alpha_{t+1}) \ldots (y - \alpha_k) \pmod{q(x)}$$
    for $(r = 1; r \leq n; r++)$ **do**
      $S_r[i, j] = \phi^{-1}(f(\alpha_r))$
    end
    $S'_t[i, j] = \phi^{-1}(g(\alpha_t) + f(\alpha_t))$
  end
end

Recovery Algo for Meaningful Image

Algorithm 6: Recovery phase of Shamir's (k, n)-SIS scheme with Private ID

Input: Suppose k participants $P_1, \ldots, P_t, \ldots, P_k$ submit their shares $S_1, \ldots, S'_t, \ldots, S_k$ each of size $M \times N$.

Output: The meaningful proxy image P of size $M \times N$.

Choose an irreducible polynomial $q(x) \in \mathrm{F}_p[x]$ of degree 8.

for $(i = 0; i < M; i++)$ **do**

  for $(j = 0; j < N; j++)$ **do**

    Lagrange's interpolation formula as follows:

$$h(0) = \sum_{r=1}^{k} \phi(S_{\alpha_r}[i,j]) \prod_{t=1, t\neq r}^{k} \frac{-\alpha_t}{\alpha_r - \alpha_t} \pmod{Q(x)}$$

    $P[i,j] = \phi^{-1}(h(0))$

  end

end

Example to Show how the attack works

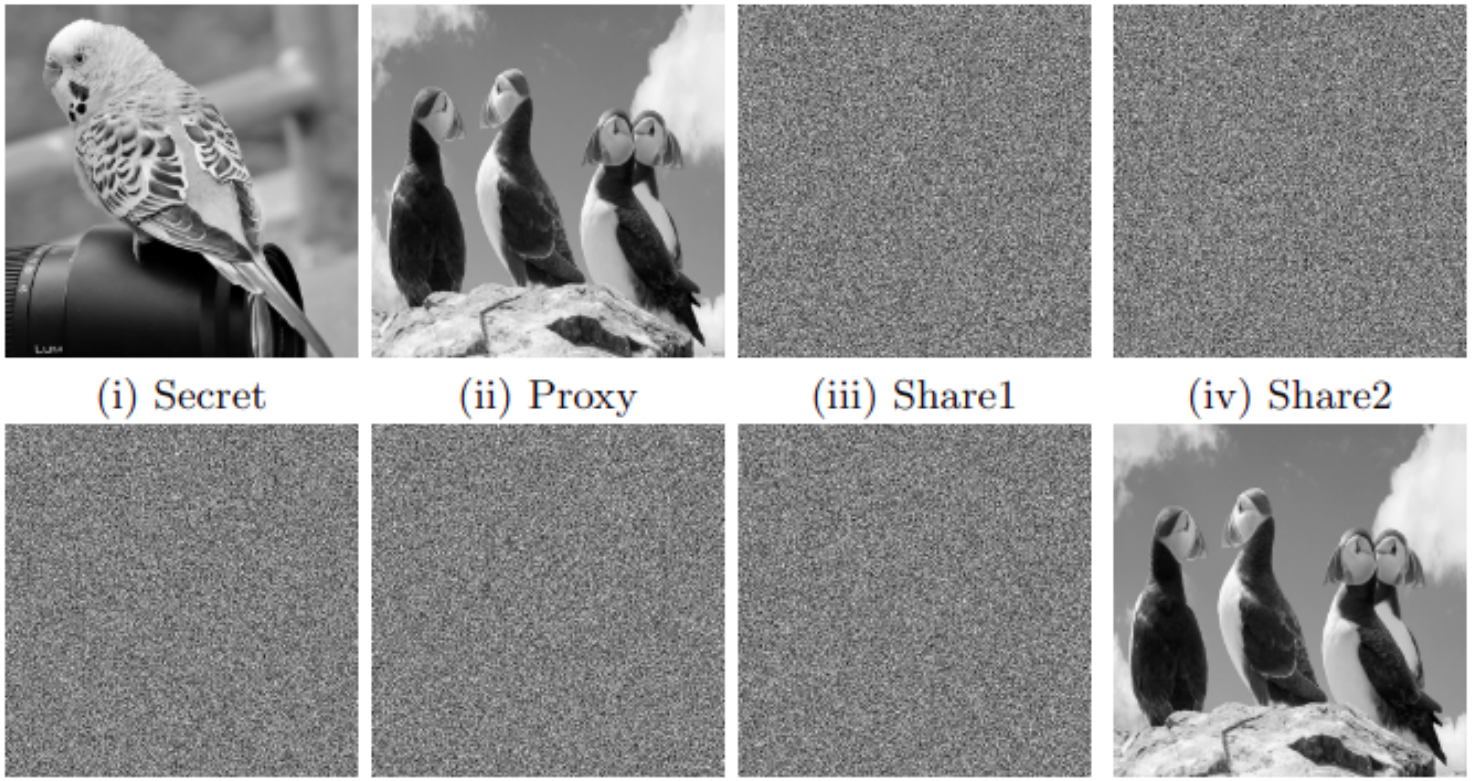


Fig. 1. Experimental results of our (3, 4)-CDSIS scheme. (i) Original secret image, (ii) Proxy image (iii)-(vii) shadow images, (viii) random proxy share generate by participant and (ix) secret recovered by 123 after cheating.

Acknowledgment and Bibliography

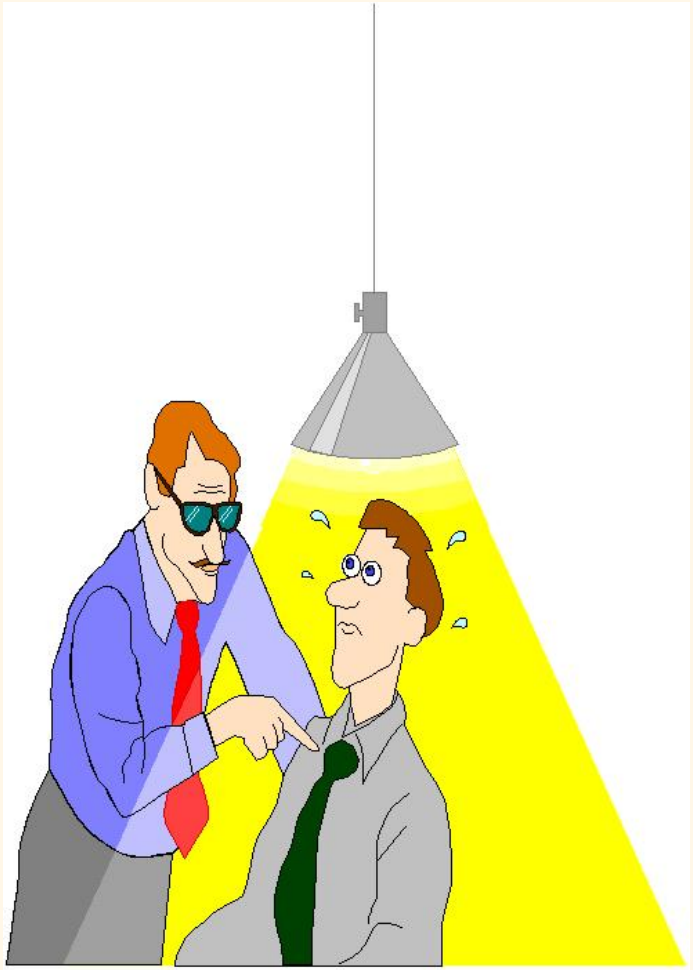
- Thankful to **Google** for helping in finding important information and necessary images.

- I am thankful to my PhD Scholars Jyotirmoy Pramanik, Md Kutubuddin Sardar and my student Anisha Dutta and my Masters Students for helping me in making the slide.
- Thanks to **Department of Information Technology, Government of India, DRDO and WESEE (Ministry of Defense), DST-SERB, DST-MATRICS, NBHM, JSPS and JST, Government of Japan** for providing me financial support towards my research.

Bioliography

- Avishek Adhikari, M R Adhikari, **Basic Modern Algebra with Applications**, **Springer Book**, 2014.
- Avishek Adhikari, *Linear Algebraic Techniques to Construct black and white Visual Cryptographic Schemes for General Access Structure and its Applications to Color Images*, **Design, Codes and Cryptography, Springer Journal**, December 2014, Volume 73, Issue 3, pp 865-895.
- Sabyasachi Dutta, Raghvendra Singh Rohit and Avishek Adhikari, *Constructions and Analysis of Some Efficient t-$(k, n)^*$-Visual Cryptographic Schemes Using Linear Algebraic Techniques*, **Design, Codes and Cryptography, Springer Journal**, 2015, DOI 10.1007/s10623-015-0075-5.
- Sabyasachi Dutta and Avishek Adhikari, *XOR Based Non-monotone t-$(k, n)^*$-Visual Cryptographic Schemes Using Linear Algebra*, (ICICS 2014), Lecture Notes in Computer Sciences, **Springer**, Volume 8958, pp 230-242.

Questions or Comments!



avishek.adh@gmail.com


Thank
You