How to Compartment Secrets
Trust Everybody, but Cut the Cards

Gaëlle Candel[1,3], Rémi Géraud-Stewart[2,3(✉)], and David Naccache[3]

[1] Ingenico Labs, 28 boulevard Grenelle, 75015 Paris, France
gaelle.candel@ingenico.com
[2] Ingenico Labs, 9 Avenue de la Gare, 26300 Alixan, France
remi.geraud@ingenico.com
[3] DIENS, ENS, CNRS, PSL University, 45 rue d'Ulm, 75230 Paris cedex 05, France
{gaelle.candel,remi.geraud,david.naccache}@ens.fr

Abstract. Secret sharing splits a secret s into ℓ shares in such a way that $k \leq \ell$ shares suffice to reconstruct s. Let $\rho_{i,j}$ be the probability that shareholder i disclose their share to shareholder j, with $0 \leq i, j < n$.

Given $k \leq \ell \leq n$, to whom ℓ individuals should we hand shares, if we wish to minimize the probability that one of them reconstitutes s?

1 Introduction

Queen Elizabeth I stated *"Do not tell secrets to those whose faith and silence you have not already tested"*. Given the relative faith in the audience, how can we calculate the overall disclosure risk? This paper provides an answer to this question.

Secret sharing splits a secret s into "shares", distributed among n participants. Under certain conditions – the sharing scheme's *access structure* – this secret s can be reconstructed (e.g. from enough shares). In the simplest case, we may require all shares to be combined [Sha79, Mig82, Bla79]. A more interesting access structure requires that at least k shares among n are required. Constructions for different access structures are known [DD94].

In this paper we are given a table:

$$\boldsymbol{\rho} = \{\rho_{i,j}\}_{0 \leq i,j < n}$$

where $\rho_{i,j}$ is the probability that shareholder i will leak their share to shareholder j. Because of this leakage mechanism, it is possible that eventually one of the shareholders gets enough shares to reconstruct s. Our goal is to evaluate the probability p_{col} that any third party's reconstructs s, an event called "collapse".

The most general setting, where we can produce exactly ℓ shares, is motivated by a real-life data escrow scenario: given n data centers we wish to select ℓ of them to hold shares of a k-out-of-ℓ secret sharing, so that collapse probability is minimal. However, as we will discuss below this is a challenging problem, that is best approached in several steps. We therefore discuss two simplified versions of the problem first:

Published by Springer Nature Switzerland AG 2020
M. Laurent and T. Giannetsos (Eds.): WISTP 2019, LNCS 12024, pp. 3–11, 2020.
https://doi.org/10.1007/978-3-030-41702-4_1

- $\ell = k = n$ *and* $0 \leq \rho_{i,j} \leq 1$: Given $n, \boldsymbol{\rho}$, compute p_{col}.

The optimal strategy resides in the access structure. To see why, consider the simplest n-out-of-n setting in which all shares are necessary recover s.

The optimal choice in this situation (assuming $0 < \rho_{i,j} < 1$) is to give a share to everyone. The proof is immediate: adding a share to the game multiplies collapse probability by a factor < 1; p_{col} can be computed with the tools of Sect. 2.

- $\rho_{i,j} \in \{0, 1\}$: Given $n, \boldsymbol{\rho}$, compute p_{col} and potentially avoid giving shares to participants not affecting p_{col}.

The case $\rho_{i,j} \in \{0, 1\}$ allows a more parsimonious distribution of shares: we distribute a share per strongly-connected component of the graph defined by $\rho_{i,j}$. Thus the general problem can be solved by condensing the graph (this is done in linear time) and handing a share to each representative of a strongly-connected component. If there are fewer shares than connected components, we have $p_{\text{col}} = 0$. If there are more, then depending on k we can have $p_{\text{col}} = 0$ or 1. This process is detailed in Sect. 3.

1.1 Notations and Hypotheses

Let $n > 2$ be the number of possible shareholders. Let $[n]$ denote the set $\{0, \ldots, n-1\}$. The cardinality of a set X is denoted $|X|$. If p is a probability, we write $\overline{p} = 1 - p$. U_i will denote shareholder i.

For any $i, j \in [n]$ we denote by $\rho_{i,j}$ the probability that U_i shares all he knows with U_j. We say in that case that U_i and U_j *collude*. Collusion is transitive: U_i can send its share x_i to U_j, who then transmits it (along with x_j) to U_ℓ — even if U_i and U_ℓ abstain from direct interactions. Note that $\rho_{i,j}$ may differ from $\rho_{j,i}$.

2 Collapse Probability

In this section we explain how to compute the collapse probability, defined as the probability that *at least one* shareholder can reconstruct the secret (e.g. by gathering enough shares).

Definition 1 (Saturation, $G^{\pm}$). *A labeled directed graph G is* saturated *if all labels are equal to* 1*. Saturated graphs are in one-to-one correspondence with unlabeled directed graphs. We say that an edge is* saturated *if it is labeled 1, and is* unsaturated *otherwise. Unsaturated edges of G can be ordered (e.g. by lexicographic order), and we denote by $G^{+(i \to j)}$ (resp. $G^{-(i \to j)}$) the graph obtained by saturating (resp. removing) the first unsaturated edge of G, denoted $i \to j$.*

Definition 2 (Evaluation at a Random Graph). *Let f be a function taking as input a directed graph and returning an element in some (fixed) vector space.*

Let G be a labeled *directed graph, with edges labels in* $[0,1]$. *We define*

$$\widehat{f}(G) = \begin{cases} f(G) \\ \qquad \textit{if } G \textit{ is saturated} \\ \\ \ell_{i,j}\widehat{f}\left(G^{+(i\to j)}\right) + \overline{\ell_{i,j}}\widehat{f}\left(G^{-(i\to j)}\right) \\ \qquad \textit{for the unsaturated edge } i \to j \textit{ labeled } \ell_{i,j} \end{cases}$$

which we call the evaluation of f at G.

Definition 3 (Root-Directed Spanning Tree). *Let G be a directed graph. G has a* Root-Directed Spanning Tree *(RDST) if there is a vertex R such that, for every other vertex S, G has a directed path from S vertex to R.*

Remark 1. Note that if G has an RDST, then G is connected, so that connectivity is a necessary condition. Note also that if G is strongly connected, then it has an RDST, so that strong-connectivity is a sufficient condition. Both properties can be established in linear time.

Note that G has this property if and only if $G^\bullet$ has it, so that without loss of generality we may assume we are working on a condensed graph. Such a graph is acyclic which makes it possible to logarithmically check efficiently the presence of an RDST.

Example 1 ((n,n)-secret sharing). The (n,n)-collapse function $L^{(n,n)}$ takes a directed acyclic graph G as input, and returns 1 if G has a root-directed spanning tree, and 0 otherwise. Therefore, collapse probability for this access structure only depends on the graph ρ defined by the values of $\rho_{i,j}$, and is $p_{\text{col}}^{(n,n)} = \widehat{L}^{(n,n)}(G)$.

Example 2 ((n,n)-secret sharing, $n = 3$). Consider $n = 3$ and write $a = \rho_{0,1}$, $b = \rho_{1,2}$, $c = \rho_{2,0}$ all other probabilities being 0. The collapse probability is $p_{\text{col}} = ab + bc + ca - 2abc$. Note that the expression is symmetric in a, b, c, and $0 \le p_{\text{col}} \le 1$. A worked-out computation is given in Appendix A.

The following remark gives a computer-friendly representation of p_{col}:

Remark 2. Let $e_1, e_2, \ldots, e_\ell$ be the edges' labels in G, and for any ℓ-bit string $x = (x_1, \ldots, x_\ell)$, let $u_i(x_i) = (1 - x_i)e_i + (1 - e_i)x_i$ and $u(x) = \prod_i u_i(e_i)$. Then any collapse probability is of the form

$$p_{\text{col}} = \sum_{x \in X} u(x)$$

where X is a set of ℓ-bit strings, corresponding to edge saturations associated with an RDST graph. In Eq. 2, with $e_i = (a, b, c)$, we have $X = \{\texttt{011}, \texttt{101}, \texttt{110}, \texttt{111}\}$.

Example 3 ((n,n)-secret sharing, $n = 4$). (Same as above, but with an additional edge in the reverse direction). $X = \{\texttt{1111}, \texttt{1101}, \texttt{1011}, \texttt{1010}, \texttt{1001}, \texttt{0111}, \texttt{0101}, \texttt{0011}\}$.

3 Optimal Solution When $\rho \in \{0, 1\}$

The minimal collapse probability is achieved when every participant has a share. It is possible to be slightly more efficient using the following notion:

Definition 4 (Condensation). *Let G be a directed graph. A* strongly connected component *of G is a sub-graph in which there is a path in each direction between each pair of vertices. The* condensation *of G is the directed acyclic graph $G^\bullet$ obtained by contracting strongly connected components.*

We apply condensation to the graph G whose vertices are $[n]$ and whose edges are those edges $(i \to j)$ such that $\rho_{i,j} = 1$. Tarjan's algorithm [Tar72] computes the condensation of a graph in $O(n + e)$, where n is the number of vertices and e the number of edges, i.e. $e = |\{i, j \in [n] \mid i \neq j, \rho_{i,j} = 1\}|$. By design, all elements in an equivalence class have exactly the same knowledge (they share their knowledge with probability 1).

Thus it suffices to give a share per representative of each equivalence class, and the optimal solution is attained by giving a share to every vertex in $G^\bullet$ (which is at most n). Henceforth, we denote by n the number of vertices in the condensed graph, unless stated otherwise.

If there are fewer shares than connected components, we have $p_{\text{col}} = 0$. If there are more, then we attempt to distribute them uniformly. Let g denote the number of strongly-connected components in $G^\bullet$, then if $\ell/g < k$ we have $p_{\text{col}} = 0$. Otherwise $p_{\text{col}} = 1$.

4 Optimal Solution for Monotone Secret Sharing

In threshold secret sharing, the secret is recovered as soon as any k shares among n are known. Recent work [BDIR18] shows that high-threshold instances of Shamir's secret sharing scheme are secure against local leakage when the underlying field is of a large prime order and the number of parties is sufficiently large.[1]

As mentioned in the introduction, the minimal collapse probability is achieved when $k = n$; in some settings there exists a value $k < n$ with the same leakage probability.

Definition 5 (Access Structure). *An* access structure *on $[n]$ is a predicate on subsets of $[n]$.*

A secret sharing scheme has access structure P if it allows reconstructing s for any subset of $[n]$ satisfying P, and does not allow reconstruction for any subset not satisfying P. This generalizes the threshold construction discussed previously, which corresponds to $P : S \mapsto (|S| \leq k)$.

[1] Contrast this with the fact that for some protocols full recovery of a multi-bit secret is possible by leaking only one bit from each share [GW17].

Definition 6 (Monotone Access Structure). *An access structure P is* monotone *[Toc15] if* $\forall A \geq B$, $P(A) = 1 \Rightarrow P(B) = 1$.

In particular, threshold access structures are monotone; the argument made that a minimal leakage probability is achieved by handing shares to all participants applies immediately to monotone access structures.

5 Finding Optimal Strategies

Exact computation of p_{col}. The algorithmic complexity of computing $\widehat{L}$ on a graph with n vertices is $O\left(2^{n^2}\right)$ using a direct implementation of the recursive algorithm corresponding to Definition 2. Indeed there are $n(n-1)/2$ edges to consider and each of them leads to a twofold branch in the evaluation of $\widehat{L}$. Using classical memorization techniques is it possible to reduce this cost. This is illustrated in the explicit computation of Appendix A.

Optimal Strategy. The previous algorithm gives a computable (if inefficient) way to find optimal strategies for small graphs, by exhausting subsets of $\{1, \ldots, n\}$ of size ℓ and computing the collapse probability for each of them. In the simplest case, $k = \ell = n$, there are n such subsets to be tested, whereas in the worst case, $k \approx n/2$ there are of the order of 2^{2n}. As a result, in practice it becomes intractable to compute general solutions for $n > 6$.

6 Heuristic Solutions

Sampling heuristic. One heuristic argument consists in replacing all probabilities in $\boldsymbol{\rho}$ by the nearest integer (0 or 1), so that the efficient algorithm in that case can be used. We propose a slightly more refined approach: let $0 < \eta < 1$ and h_η the function that sends x to 0 if $x \leq \eta$ and 1 otherwise. Applying h_η entrywise to $\boldsymbol{\rho}$ with $\eta = 0.5$ we fall back on the previous heuristic. For every value of η, we get a certain partition of the resulting graph G into strongly connected components. We are then interested in those components that are *stable* as we vary η. Indeed, any chose of η corresponds to an over- or an underestimation of the true leakage probabilities. We may assume $\eta \in \{i/m \mid i = 1, 2, \ldots, m-1\}$ for some integer m, consider the graphs G_η resulting from applying h_η to $\boldsymbol{\rho}$, and rank the vertices in G by the *number of strongly-connected components* that they belong to as we vary η. A vertex that remained in a single component all along ("stable") will be given many shares, whereas a vertex that often switched components ("unstable") will be given fewer shares, if any.

Furthermore, for every value of η, we get a collapse probability $p_\eta \in \{0, 1\}$: by averaging these values we can hope to obtain an approximation $p_{\approx} = \mathbb{E}_\eta[p_\eta]$ of the true collapse probability p_{col}.

7 Numerical Example

Consider the following (completely arbitrary) leakage matrix:

$$\rho_{i,j} = \begin{cases} 1 & \text{if } i = j \\ \left|\cos(1 - 2i + 3j^2)\sin(-4 + 5i^2 - 6j^3)\right| & \text{otherwise} \end{cases}$$

We can compute optimal strategies exactly for $n = \ell = 4$ and $k = 1, 2, 3, 4$ shares. The results are given in Table 1 and confirm that the scenario minimizing p_{col} corresponds to $k = \ell = n$. The heuristic algorithm finds that U_3 is the most stable vertex, followed by U_1, U_2, and U_0 in that order. It therefore produces the same strategy on this example.

Table 1. Optimal strategies, given as a list of i such that U_i gets a share, for $n = \ell = 4$ as a function of k.

k	Winning strategy	p_{col}	$p_{\approx}$	Number of strategies
1	None	1.0	1.0	4
2	[1, 3]	0.79695	0.79695	10
3	[1, 2, 3]	0.73182	0.73182	20
4	[0, 1, 2, 3]	0.71852	0.71852	35

However, it seems out of reach to perform exact computations for graphs larger than $n = 6$. Instead, we turn to the heuristic algorithm discussed in the previous section to address larger scenarios. Note that for small values of n, the exact and heuristic algorithms produce identical results.

Taking $n = 1000$, $\ell = 100$, $k = 15$ and the same $\boldsymbol{\rho}$ as above, we get $p_{\approx} = 0.9859$. Using $k = \ell = 100$ instead, we get $p_{\approx} = 0.9839$. The heuristic algorithm's running time grows roughly quadratically with respect to n, so we expect instances of size $n \approx 100{,}000$ to be within reach.

8 Conclusion

The problem of distributing a secret amongst leaking shareholders is defined, along with the "collapse probability" which measures how likely it is that at least one shareholder reconstructs the secret. We show that this probability measures the likelihood of having a root-directed spanning tree (RDST) in a realisation of the underlying graph's condensation, and provide an algorithm to compute this probability. Unfortunately a direct implementation of this exact algorithm is computationally expensive; we therefore provide an efficient (but unproven) heuristic to find optimal distribution strategies (i.e. that minimises the collapse probability).

Given n potential shareholders, a leakage matrix $\boldsymbol{\rho}$, and ℓ shares of which k suffice for reconstruction, these algorithm tell us whom to hand the shares.

Future Work. The problem as stated is motivated by a distributed storage scenario; a "non-monotone" access structure [LMC15] may allow for a thriftier distribution of shares, assuming it is computationally difficult for the adversary to test all subsets of the shares they have collected.

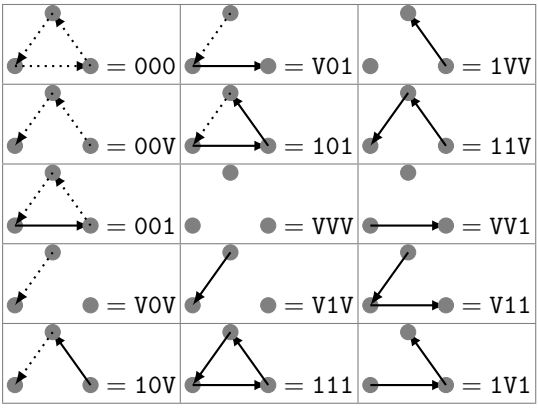
Complementary to our work, but beyond the scope considered here, is the question of obtaining a reasonable estimate for the matrix $\boldsymbol{\rho}$ — what does this matrix look like in the real world? — as well as the consequences of having imperfect knowledge of this matrix on the predicted results. For instance, in the explicit computation of Appendix A, an uncertainty δ in the values of (a, b, c) results in an uncertainty $O(\delta^2)$ in p_{col}. Can this be treated in more generality?

There are also several interesting directions in which it would make sense to extend our model, which may lead to simplifications. For instance, restrictions to some families of structured graphs (e.g. grids) may allow for more efficient algorithms (or even closed-form expressions, although that seems unlikely). Alternatively, rather than minimising secret reconstruction, we may wish to *maximise* it, maybe only for a selected subset of shareholders.

Finally, precise bounds on the heuristic algorithm's errors (or maybe, better approximation algorithms) are needed.

A Detailed Computation For $(3, 3)$

We denote `[XYZ]` $= \widehat{L}($`graph[XYZ]`$)$ for the following graphs:



The collapse probability is:

$$\begin{aligned}
p = \widehat{L}(\rho) &= \texttt{[000]} = \overline{a}\,\texttt{[00V]} + a\,\texttt{[001]} \\
&= \overline{a}(\overline{b}\,\texttt{[V0V]} + b\,\texttt{[10V]}) + a(\overline{b}\,\texttt{[V01]} + b\,\texttt{[101]}) \\
&= \overline{a}(\overline{b}(\overline{c}\,\texttt{[VVV]} + c\,\texttt{[V1V]}) + b\,\texttt{[10V]}) + a(\overline{b}\,\texttt{[V01]} + b\,\texttt{[101]}) \\
&= \overline{a}b\,\texttt{[10V]} + a\overline{b}\,\texttt{[V01]} + ab\,\texttt{[101]}) \\
&= \overline{a}b(\overline{c}\,\texttt{[1VV]} + c\,\texttt{[11V]}) + a\overline{b}\,\texttt{[V01]} + ab\,\texttt{[101]} \\
&= \overline{a}bc + a\overline{b}(\overline{c}\,\texttt{[VV1]} + c\,\texttt{[V11]})) + ab\,\texttt{[101]} \\
&= \overline{a}bc + a\overline{b}c + ab\,\texttt{[101]} \\
&= \overline{a}bc + a\overline{b}c + ab(\overline{c}\,\texttt{[1V1]} + c\,\texttt{[111]}) \\
&= \overline{a}bc + a\overline{b}c + ab \\
&= ab + bc + ca - 2abc
\end{aligned}$$

B Detailed Computation With The Heuristic Algorithm

We apply the heuristic algorithm to the same graph as in the previous section.

The collapse probability is 1 whenever any two edges are saturated, and 0 otherwise. In other terms, using that $\Pr[A \vee B] = \Pr[A] + \Pr[B] - \Pr[A \wedge B]$ and $\Pr[A \wedge B] = \Pr[A]\Pr[B \mid A]$, we have:

$$\begin{aligned}
p_\eta &= \Pr[(a < \eta \wedge b < \eta) \vee (a < \eta \wedge c < \eta) \vee (c < \eta \wedge b < \eta)] \\
&= \Pr[a < \eta]\Pr[b < \eta] + \Pr[a < \eta]\Pr[c < \eta] + \Pr[c < \eta]\Pr[b < \eta] \\
&\quad - \Pr[a < \eta]\Pr[b < \eta]\Pr[c < \eta] \\
&\quad - \Pr[(a < \eta \wedge b < \eta) \wedge ((a < \eta \wedge c < \eta) \vee (c < \eta \wedge b < \eta))] \\
&= \Pr[a < \eta]\Pr[b < \eta] + \Pr[a < \eta]\Pr[c < \eta] + \Pr[c < \eta]\Pr[b < \eta] \\
&\quad - \Pr[a < \eta]\Pr[b < \eta]\Pr[c < \eta] \\
&\quad - \Pr[a < \eta \wedge b < \eta]\Pr[(a < \eta \wedge c < \eta) \vee (c < \eta \wedge b < \eta) \mid a < \eta \wedge b < \eta] \\
&= \Pr[a < \eta]\Pr[b < \eta] + \Pr[a < \eta]\Pr[c < \eta] + \Pr[c < \eta]\Pr[b < \eta] \\
&\quad - 2\Pr[a < \eta]\Pr[b < \eta]\Pr[c < \eta]
\end{aligned}$$

Sampling over η this gives:

$$p_{\approx} = \mathbb{E}_\eta[p_\eta] = ab + ac + bc - 2abc$$

matching the result obtained in the previous section.

References

[BDIR18] Benhamouda, F., Degwekar, A., Ishai, Y., Rabin, T.: On the local leakage resilience of linear secret sharing schemes. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018. LNCS, vol. 10991, pp. 531–561. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-96884-1_18

[Bla79] Blakley, G.R.: Safeguarding cryptographic keys. In: Proceedings of the National Computer Conference, vol. 48, (1979)

[DD94] Dawson, Ed., Donovan, D.M.: The breadth of Shamir's secret-sharing scheme. Comput. Secur. **13**(1), 69–78 (1994)

[GW17] Guruswami, V., Wootters, M.: Repairing Reed-Solomon codes. IEEE Trans. Inf. Theory **63**(9), 5684–5698 (2017)

[LMC15] Liu, J., Mesnager, S., Chen, L.: Secret sharing schemes with general access structures. In: Lin, D., Wang, X.F., Yung, M. (eds.) Inscrypt 2015. LNCS, vol. 9589, pp. 341–360. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-38898-4_20

[Mig82] Mignotte, M.: How to share a secret. In: Beth, T. (ed.) EUROCRYPT 1982. LNCS, vol. 149, pp. 371–375. Springer, Heidelberg (1983). https://doi.org/10.1007/3-540-39466-4_27

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

[Tar72] Robert Endre Tarjan: Depth-first search and linear graph algorithms. SIAM J. Comput. **1**(2), 146–160 (1972)

[Toc15] Tochikubo, K.: New secret sharing schemes realizing general access structures. JIP **23**(5), 570–578 (2015)