

Cryptographic applications of combinatorial ranking for integer compositions

Gustavo Zambonin Larissa Gremelmaier Rosa
Ricardo Custódio Daniel Panario

Universidade Federal de Santa Catarina & Carleton University

June 9, 2026

IWOCA 2026

The target combinatorial object

Definition

A d -bounded k -composition of n is a sequence $(a_1, \dots, a_k)$ with $\sum_i a_i = n$ and each part $a_i \in \{0, \dots, d\}$.

We write $\mathcal{C}_d(n, k)$ for this set, with size

$$\mathfrak{C}_d(n, k) = [x^n] \left(\frac{1-x^{d+1}}{1-x} \right)^k \quad [\text{Moi30}]$$

$$= \sum_{i=0}^k (-1)^i \binom{k}{i} \binom{n-(d+1)i+k-1}{k-1}. \quad [\text{And76}]$$

Used across cryptography: hash-based [KKW25], sum-preserving encryption [MY22], code-based [Kra+23].

In each case the task reduces to (un)ranking a composition.

Combinatorial (un)ranking

Definition

Let S be a finite set, $N = |S|$. A *ranking function* is a bijection $\mathbf{R} : S \rightarrow [N]$ that defines a total order $\prec_{\text{ord}}$ on S . The *unranking function* is its inverse $\mathbf{U} = \mathbf{R}^{-1}$.

Used for *uniform sampling*: draw $\lceil \lg N \rceil$ random bits and evaluate $\mathbf{U}$ (a bijection maps the uniform law to the uniform law).

And for *lossless compression*: $\mathbf{R}$ packs a structured message into a minimal index, reducing transmission overhead.

However, algorithms are frequently rediscovered [Per+21; HK24; MY22], each tailored to a particular order, and with its own correctness proof.

Existing orders and their algorithms

The design of an (un)ranking algorithm is dictated by the total order $\prec_{\text{ord}}$ chosen on the combinatorial class. The literature offers several:

- *lex* and *colex* ($\prec_{\text{colex}}$), aligned with Pascal's identity [Knu11];
- a reflected *Gray* order ($\prec_{\text{gray}}$) [VV11; Arn11];
- the *recursive block* order ($\prec_{\text{rb}}$) [RR02; MY22];
- the *boustrophedonic* order ($\prec_{\text{bous}}$) [FZC94; MM01].

We depict these by enumerating $\mathcal{C}_4(6, 5)$, which gives 185 elements.

Each column is one composition: a part of value v is v black cells, and consecutive parts are separated by a gray cell.

Visualizing distinct orders

A regular “staircase” pattern appears in $\prec_{\text{colex}}$ [Knu11].



Compare it with the “saw-tooth” shifts of $\prec_{\text{gray}}$ [VV11; Arn11].



Nested blocks show up in $\prec_{\text{rb}}$ [RR02; MY22].



A more disconnected pattern is the result of $\prec_{\text{bous}}$ [FZC94; MM01].



Our approach: fix the object, vary the strategy

Order choice significantly impacts performance and storage [MM01], yet alternatives remain underexplored.

We propose a unified treatment of these algorithms, and use it to derive a new order suited to cryptographic (uniform random) inputs.

Prior frameworks [MM01; Mol05] fix the order and derive an algorithm for a generic class.

We take the complementary route: we fix the class (bounded compositions) and treat the order, or *strategy*, as a parameter.

Components of a strategy

A strategy is a tuple $\Sigma = (D, P, S)$ with *decomposition*, *path* and *split* functions.

The D function shapes the recursion given the number of parts. For instance, $D_r(k) = (k - 1, 1)$ and $D_s(k) = (\lfloor \frac{k}{2} \rfloor, \lceil \frac{k}{2} \rceil)$.

The P function decides the order in which to visit the sub-sums $w_l \in [n]$. For instance, $P_{\text{inc}}(n, k_l, k_r) = (w_1, w_2, \dots, w_m)$ and $P_{\text{alt}}(n, k_l, k_r) = (w_1, w_m, w_2, w_{m-1}, \dots)$ for $m \in [n + 1]$.

The S function splits a rank into sub-ranks. For instance, $S_{\text{id}}^{(1, v, w)}(r) = (0, r)$ and $S_{\text{div}}^{(u, v, w)}(r) = (\lfloor \frac{r}{v} \rfloor \pmod{u}, r \pmod{v})$.

Classical algorithms as instances of the framework

In the constructive language of [Arn11], we can express, for $(k_l, k_r) = D(k)$, the generic sequence as

$$\mathbf{Z}_d(n, k) = \bigoplus_{w_l \in P(n, k_l, k_r)} \left(\mathbf{Z}_d(w_l, k_l) \odot_S \mathbf{Z}_d(n - w_l, k_r) \right).$$

Here, P fixes the order of the blocks concatenated by the operator $\bigoplus$, and S fixes the order of the pairs joined by $\odot$.

We give generic, recursive ranking and unranking algorithms (resp. $\mathbf{R}_\Sigma$ and $\mathbf{U}_\Sigma$), and implement them in C and Lean.

We are able to express known orders from the literature, such as $\Sigma_{\text{colex}} = (D_r, P_{\text{inc}}, S_{\text{id}})$ and $\Sigma_{\text{bous}} = (D_s, P_{\text{alt}}, S_{\text{div}})$.

Correctness of the generic algorithms

Theorem

For any strategy Σ , $\mathbf{R}_\Sigma$ and $\mathbf{U}_\Sigma$ are mutual inverses.

Proof (sketch).

By strong induction. The framework and several strategy instances are mechanized in Lean 4 [MU21]. □

Consequently, the correctness of algorithms tied to particular orders follows as a corollary of our result.

We can also more easily reason about a pre-computed table χ_{comb} of counts $\mathfrak{C}_d(n, k)$, which increases the performance of the algorithms.

Decomposition controls storage

The entry (i, j) of χ_{comb} is $\mathfrak{C}_d(j, i)$, for $0 \leq i \leq k$ and $0 \leq j \leq n$, and the choice of D decides which rows the recursion reads.

A “linear” decomposition such as D_r solves subproblems of every size, so χ_{comb} must be stored in full.

A “balanced” decomposition such as D_s , in contrast, requires only the *iterated halving set* $H(k) = \bigcup_{p=1}^{\lceil \lg k \rceil} \{\lfloor \frac{k}{2^p} \rfloor, \lceil \frac{k}{2^p} \rceil\}$, with $H(1) = \emptyset$.

We also give a closed-form formula for $|H(k)|$ and show how it approximates $\lg k$.

Path enables search optimizations

When P is *monotonic*, unranking reduces to a search over a sorted sequence of partial sums

$$A(n', i, d) = \left(0, \mathfrak{C}_d(n', i), \dots, \sum_{j=0}^{\min(n', d)} \mathfrak{C}_d(n' - j, i) \right).$$

We characterize 2 ad hoc optimizations as properties of this search.

The *bisect* technique is a binary search over A in $\lg d$ steps. We show it is cheaper than computing terms of A directly via [Per+21].

The *partial sum* technique derives each count from the previous one by an exact ratio, reducing the number of binomial sums evaluated.

The “spiral” order for cryptographic applications

In this setting, the input to $\mathbf{U}_\Sigma$ is usually uniform random, so we exploit the structure of the resulting composition.

By the Gaussian limit law for coefficients of powers of polynomials [FS09, §VIII], $\mathfrak{C}_d(n, k)$ is asymptotically *normal* [Rat08].

Consequently, the partial sums S_j in the unranking path concentrate around their mean.

We give closed-form formulas for $\mathbb{E}[S_j]$ and $\mathbb{V}\text{ar}(S_j)$ and a new strategy $\Sigma_{\text{spiral}} = (D_r, P_s, S_{\text{id}})$ based on this fact.

The “spiral” order and a balanced variant

We center P_s on the expected next part $\mu = \frac{n'}{i}$, for i parts summing to n' , and visit outward, i.e.,

$$P_s = (\mu, \mu + 1, \mu - 1, \mu + 2, \mu - 2, \dots) \cap [d].$$

Continuing the previous example, the pattern below is the result of enumerating $\mathcal{C}_4(6, 5)$ with $\prec_{\text{spiral}}$.



Analogously, we define a “balanced” spiral $\Sigma_{\text{rb-spiral}} = (D_s, P_s, S_{\text{id}})$.



Practical considerations

Through this new strategy, we reduce storage requirements of χ_{comb} by storing only the high-probability band of elements.

We write $\chi_{\text{comb}-\sigma}$ for the new table, where σ denotes a number of standard deviations.

We provide a portable C library to test several parameters, strategies, techniques, and tabling mechanisms.

Empirically, χ_{comb} accelerates every strategy by about $100\times$, and $\chi_{\text{comb}-4}$ halves storage for a 5% performance penalty.

Practical considerations (cont.)

Other tabling mechanisms, such as storing all $A(n', i, d)$ or only binomial coefficients, are in general not as effective.

Without a pre-computed table, D decides the performance. For small k , the partial sum technique over $\prec_{\text{colex}}$ is very efficient.

The performance of $\prec_{\text{spiral}}$ is slightly better than $\prec_{\text{colex}}$, depending on the choice of n, k, d .

The C library is designed to allow expressive experiments on performance, storage access patterns, new orders and techniques.

Conclusion & next steps

We give a single algebraic definition $\Sigma = (D, P, S)$ that unifies (un)ranking algorithms for bounded compositions. We obtain

- classical orders and proofs recovered under one induction;
- structural characterization of ad hoc tricks;
- a new “spiral” order exploiting uniform inputs.

We expect to continue the characterization of further orders and techniques under this framework.

We are also working on the design of *constant-time* (un)ranking algorithms for cryptography.

Acknowledgements

Partially supported by FAPESC (61/2024, 62/2024), ON-RCPN of Brazil, and NSERC (RGPIN-2024-05341).

Thank you! Any questions?

References I

- [And76] Désiré André. “Mémoire sur les combinaisons régulières et leurs applications”. In: *Annales scientifiques de l'École Normale Supérieure* 5 (1876). 2e série, pp. 155–198. DOI: 10.24033/asens.136.
- [Arn11] Jörg Arndt. *Matters Computational: Ideas, Algorithms, Source Code*. 1st. Springer Berlin, Heidelberg, 2011. ISBN: 9783642147630.
- [FS09] Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. 1st. Cambridge University Press, 2009. ISBN: 9780521898065.
- [FZC94] Philippe Flajolet, Paul Zimmermann, and Bernard Van Cutsem. “A calculus for the random generation of labelled combinatorial structures”. In: *Theoretical Computer Science* 132.1 (Sept. 1994), pp. 1–35. DOI: 10.1016/0304-3975(94)90226-7.
- [HK24] Motonari Honda and Yuichi Kaji. “Optimum Fingerprinting Function for Winternitz One-Time Signature”. In: *2024 IEEE International Symposium on Information Theory (ISIT)*. July 2024, pp. 2898–2902. ISBN: 9798350382846. DOI: 10.1109/ISIT57864.2024.10619351.

References II

- [KKW25] Dmitry Khovratovich, Mikhail Kudinov, and Benedikt Wagner. “At the Top of the Hypercube – Better Size-Time Tradeoffs for Hash-Based Signatures”. In: *Advances in Cryptology – CRYPTO 2025*. Ed. by Yael Tauman Kalai and Seny F. Kamara. Vol. 16005. Lecture Notes in Computer Science. July 2025, pp. 93–123. DOI: 10.1007/978-3-032-01887-8_4.
- [Knu11] Donald E. Knuth. *The Art of Computer Programming, Volume 4A: Combinatorial Algorithms, Part 1*. 1st. Addison-Wesley Professional, 2011. ISBN: 9780201038040.
- [Kra+23] Markus Krausz, Georg Land, Jan Richter-Brockmann, and Tim Güneysu. “A Holistic Approach Towards Side-Channel Secure Fixed-Weight Polynomial Sampling”. In: *Public-Key Cryptography – PKC 2023*. Ed. by Alexandra Boldyreva and Vladimir Kolesnikov. Vol. 13941. Lecture Notes in Computer Science. May 2023, pp. 94–124. DOI: 10.1007/978-3-031-31371-4_4.
- [MM01] Conrado Martínez and Xavier Molinero. “A Generic Approach for the Unranking of Labeled Combinatorial Classes”. In: *Random Structures & Algorithms* 19.3-4 (Oct. 2001), pp. 472–497. DOI: 10.1002/rsa.10025.

References III

- [Moi30] Abraham de Moivre. *Miscellanea analytica de seriebus et quadraturis*. 1st. J. Tonson & J. Watts, 1730.
- [Mol05] Xavier Molinero. “Ordered generation of classes of combinatorial structures”. PhD thesis. Universitat Politècnica de Catalunya, July 2005.
- [MU21] Leonardo de Moura and Sebastian Ullrich. “The Lean 4 Theorem Prover and Programming Language”. In: *Automated Deduction – CADE 28 (CADE 2021)*. Ed. by André Platzer and Geoff Sutcliffe. Vol. 12699. Lecture Notes in Computer Science. July 2021, pp. 625–635. DOI: 10.1007/978-3-030-79876-5_37.
- [MY22] Sarah Miracle and Scott Yilek. “New Algorithms and Analyses for Sum-Preserving Encryption”. In: *Advances in Cryptology – ASIACRYPT 2022*. Ed. by Shweta Agrawal and Dongdai Lin. Vol. 13793. Lecture Notes in Computer Science. Dec. 2022, pp. 3–31. DOI: 10.1007/978-3-031-22969-5_1.
- [Per+21] Lucas Pandolfo Perin, Gustavo Zambonin, Ricardo Custódio, Lucia Moura, and Daniel Panario. “Improved constant-sum encodings for hash-based signatures”. In: *Journal of Cryptographic Engineering* 11.4 (June 2021), pp. 329–351. DOI: 10.1007/s13389-021-00264-9.

References IV

- [Rat08] Joel Ratsaby. “Estimate of the number of restricted integer-partitions”. In: *Applicable Analysis and Discrete Mathematics* 2.2 (2008), pp. 222–233. DOI: 10.2298/AADM0802222R.
- [RR02] Leonid Reyzin and Natan Reyzin. “Better than BiBa: Short One-Time Signatures with Fast Signing and Verifying”. In: *Information Security and Privacy*. Ed. by Lynn Batten and Jennifer Seberry. Vol. 2384. Lecture Notes in Computer Science. July 2002, pp. 144–153. DOI: 10.1007/3-540-45450-0_11.
- [VV11] V. Vajnovszki and R. Vernay. “Restricted compositions and permutations: From old to new Gray codes”. In: *Information Processing Letters* 111.13 (July 2011), pp. 650–655. DOI: 10.1016/j.ipl.2011.03.022.