On the Complexity of Generalized Roller Splat!

Sebastian Zhu, William Yue, Vincent Fan

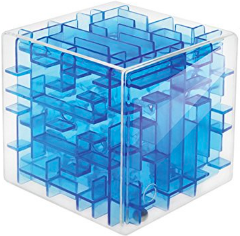
October 26, 2019
S.-T. Yau High School Science Award

Agenda

Motivation

- Wanted to study the computational complexity of recreational games
- Mainly inspired by the work of Erik Demaine
- Chose roller-splat, a game that exemplifies the motif of a maximal sliding agent
- Maximal Sliding agents have wide ranging applications in both robotics and video game design itself

Examples



Definitions

Definition (P)

The complexity class P contains all problems such that there exists a DTM L which decides P in time that is bounded by a polynomial function of the input size.

Definition (NP)

When a decision problem Π has a succinct certificate which can be used to check that a given instance is true in polynomial time then we say that the associated language, $L_\Pi = \{x | x$ is a natural encoding of an instance of $\Pi\}$ is accepted in non-deterministic polynomial time, or it is in NP. In other words, given a proposed "solution" to a problem in NP, there exists a polynomial time algorithm to check the validity of this solution.

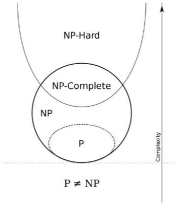
Definition

Definition (NP-hard)

We define a language L to be NP-hard if $A \in \text{NP}$ implies that $A \leq_m L$. Such a language L is at least as difficult as any other language in NP.

Definition (NP-complete)

We define a language L to be NP-complete if L is both in NP and is NP-hard.



Agenda

Stopping Point and Movement Graph Representation

- Stopping Point graph G consists of vertices where the paintball call stop, and directed edges if you can go from one point to another.
- Movement graph H consists of vertices as back-and-forth movements for the paintball, and directed edges if one movement can go to another by turning.

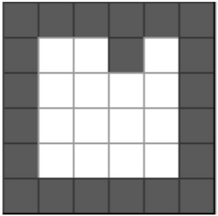


Figure: Original Grid

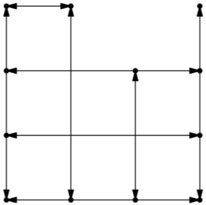


Figure: Graph G

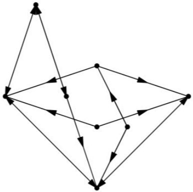


Figure: Graph H

Stop Reachability

Theorem (Stop Reachability)

Given a board, a starting square, and an ending square, the problem of determining whether there exists a path from the starting square that stops on the ending square is in P.

Proof.

- Stopping Point Graph Representation G in $\mathcal{O}(N^4)$ time.
- Add starting point to graph with indegree 0, outdegree ≤ 4.
- Perform *BFS* in $\mathcal{O}(V + E) = \mathcal{O}(N^4)$ to find path.

□

Pass Reachability

Theorem (Pass Reachability)

Given a board, a starting square, and an ending square, the problem of determining whether there exists a path from the starting square that passes through the ending square is in P*.*

Proof.

- Similar to above proof.
- Add special red vertex v corresponding to the ending square into G.
- Connect directed edges into v.
- Perform BFS in $\mathcal{O}(N^4)$ time to find path to red vertex.

$\square$

Stop Coverage

Theorem

Given a board and a starting square, the problem of determining whether it is possible to stop on every square in the board is in P*.*

Proof.

- Stopping Point Graph Representation G.
- Check if every vertex is in G. If not, done. Find a traversal through this graph.
- Use Tarjan's algorithm to reduce graph to G' by condensing Strongly Connected Components (SCCs) into vertices.
- Returns new vertices in reverse topological ordering.
- Check if path exists from each vertex in the topological ordering to the next.

$\square$

Pass Coverage

Theorem (Pass Coverage)

Given a board and a starting square, the problem of determining whether it is possible to pass through every square in the board is in P.

Theorem (Tejada)

Given a board, of which some (specified) squares have a collectible object in them, and a starting square, the problem of determining whether it is possible to collect every object by passing through its squares is in P.

- Simply fill every square with collectible object.

His proof involved using the Movement Graph Representation H of the board and contracting all SCCs to the reduced graph H'. Then, we can associate every pearl with two movements, vertices in H', and construct a 2-SAT formula that is satisfiable if and only if all the pearls can be collected. Further details can be found in section 2 of his paper.

Agenda

Originality of Our Problem

Tejada proved the problem of collecting objects in a 3D grid is NP-complete.

We prove a better claim in three dimensions, that the problem of pass covering the whole grid is NP-complete.

Definitions

Definition

- The **grid** is the entire 3D object constructed in this section.
- A **board** is one of the $m \times n$ layers.
- The **particle** is the $1 \times 1 \times 1$ "maximal sliding agent."
- A **block** is a $1 \times 1 \times 1$ occupied space.
- A **square** is a $1 \times 1 \times 1$ unoccupied space.

Definition (Polynomial Time Reduction)

We say there is a *polynomial time reduction* from A to B if

$$A \overset{f}{\Longrightarrow} B.$$

Then write $A \leq_m B$. If B is 'easy' then so is A. Therefore, if A is hard then so is B.

Membership in NP

- Movement graph representation H, constructed in $\mathcal{O}(N^{2d})$ time.
- From any location, particle can reach $v = \mathcal{O}(N^d)$ possible vertices in H.
- Takes $\mathcal{O}(v)$ moves to get to any vertex from any vertex.
- If every square can be reached, takes $\mathcal{O}(v^2) = \mathcal{O}(N^{2d})$ time to check it. Thus game is in NP.

3-SAT Reduction

The 3-SAT problem asks whether or not it is possible to find a set of assignments true or false for each variable v_i. This problem is NP-hard.

$$\underbrace{(v_1 \vee v_2 \vee v_3)}_{\text{clause 1}} \wedge \underbrace{(\overline{v_1} \vee \overline{v_2} \vee \overline{v_3})}_{\text{clause 2}} \implies \text{some } \textit{Roller Splat!} \text{ grid.}$$

This gives

$$\text{3-SAT} \leq \textit{Roller Splat!}$$

- Idea for construction is that for each variable, the paintball goes down two selection paths of true or false.
- Each clause is associated with a special *clause square* that can only be reached by certain variable paths.

VTA and PC Traversal

Definition (VTA Traversal)

The *variable truth assignment* (VTA) traversal is the process of assigning truths (either true or false, both not both) to each variable in the 3-SAT instance.

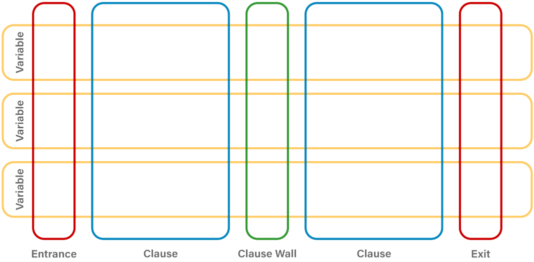
Definition (PC Traversal)

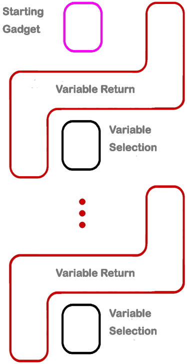
The *pass coverage* (PC) traversal is is the process of passing through all non-essential (specifically non-clause) squares after the VTA traversal.

- The first three gadgets we will show later are for the VTA traversal, while the later three are for details in the PC traversal.

High Level Overview

For 3-SAT instance $(v_1 \vee v_2 \vee v_3) \wedge (\overline{v_1} \vee \overline{v_2} \vee \overline{v_3})$.





Layers Intro

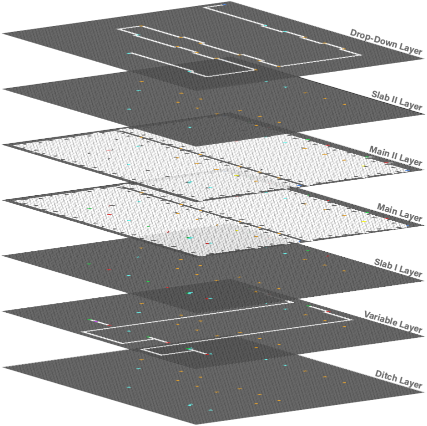


Figure: The order/orientation of the layers. Top and bottom cover layers not shown.

Variable + Clause Gadget

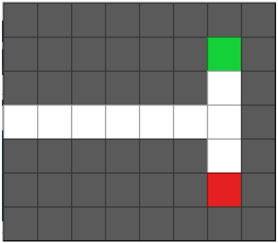


Figure: The true/false selections match the true/false selections in the layer above (right diagram).

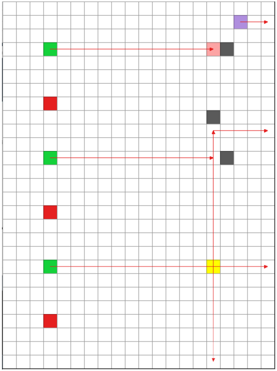


Figure: The clause gadget.

Clause Gadget

Gadget for clause $(v_1 \vee v_2 \vee v_3)$.

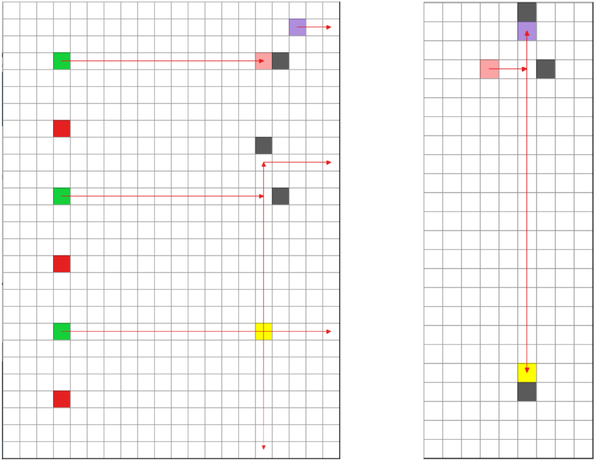


Figure: Both layers of the clause gadget. By design, after passing through the yellow clause square the particle will continue on the designated path.

Clause Wall Gadget

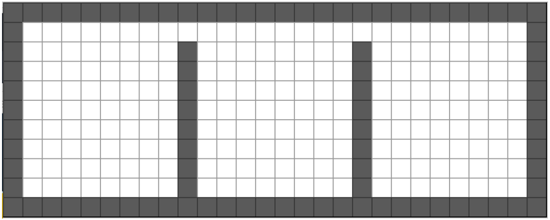


Figure: The clause wall gadget. Each cell is a clause gadget, and the wall is solid except for discrete holes for traversal.

Traversal Gadget

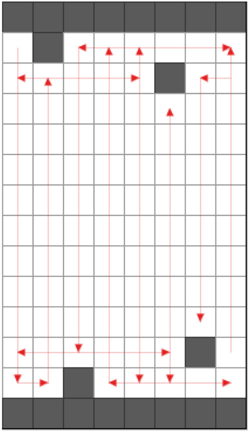


Figure: The traversal gadget, with the particle beginning in the top left square. The arrows show that every square in an 8-square-long section can be traversed by placing four blocks as shown in the figure.

Drop-Down Gadget

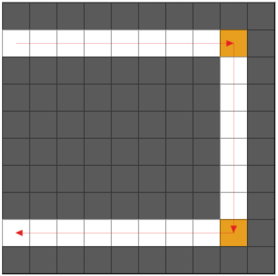


Figure: Example of drop-down gadgets within a layer.

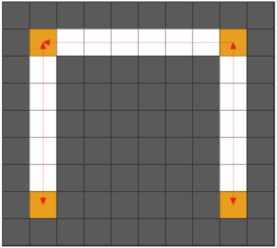


Figure: Cross section of drop-down gadgets, showing all nine layers.

Wall Gadgets

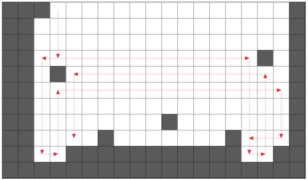


Figure: Type I wall gadget, for transferring between vertical and horizontal traversal gadgets.

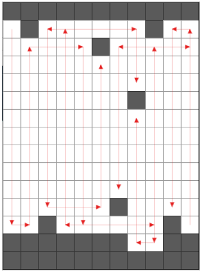


Figure: Type II wall gadget, for passing through squares that are blocked off by blocks in the middle of the board.

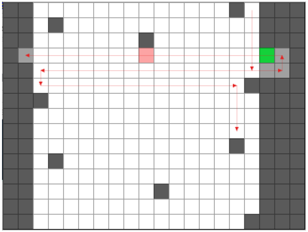
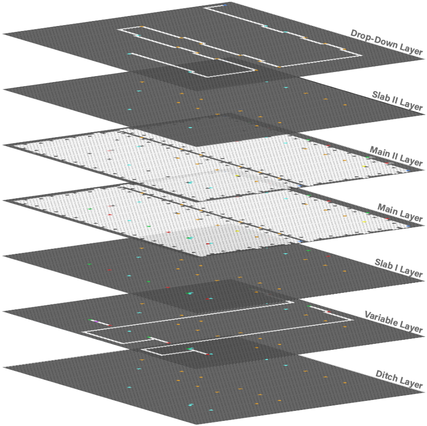


Figure: Type III wall gadget, for passing through squares that lie at the intersection of an upward between-layers movement and downward movement in VTA traversal.

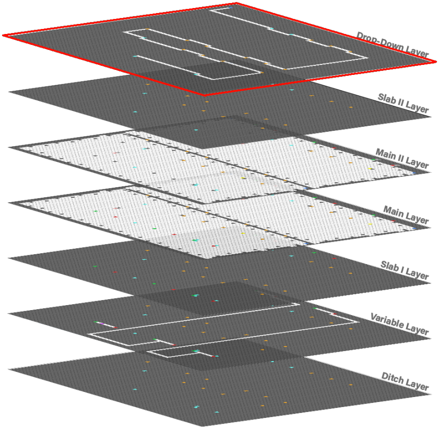
Construction for $(v_1 \vee v_2 \vee v_3) \wedge (\overline{v_1} \vee \overline{v_2} \vee \overline{v_3})$.

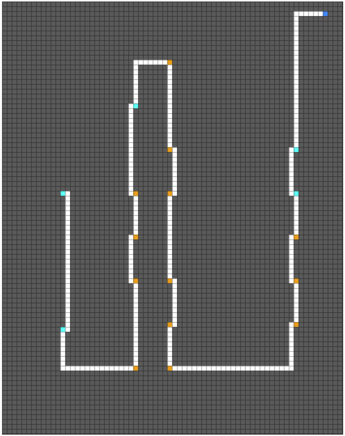


Cover Layers

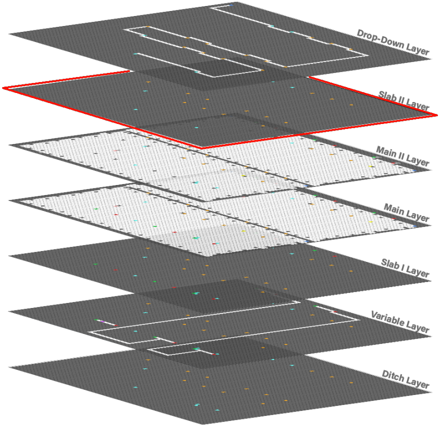
The cover layers sandwich the layers just shown and prevent the particle from escaping the grid.

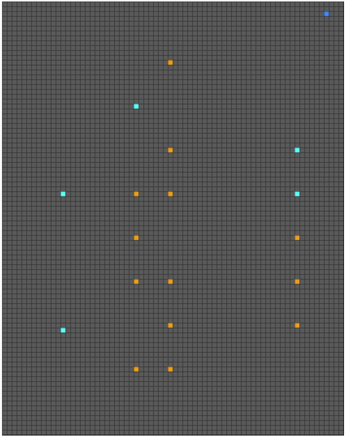
Drop-Down Layer



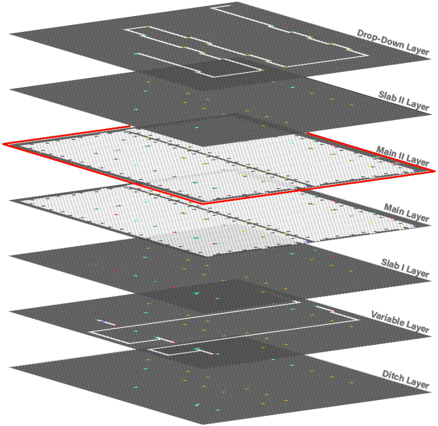


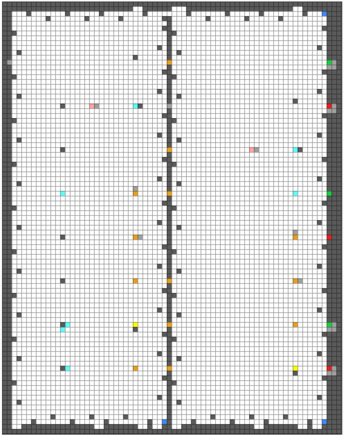
Slab II Layer



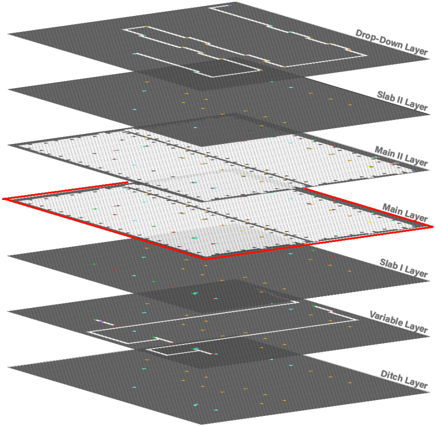


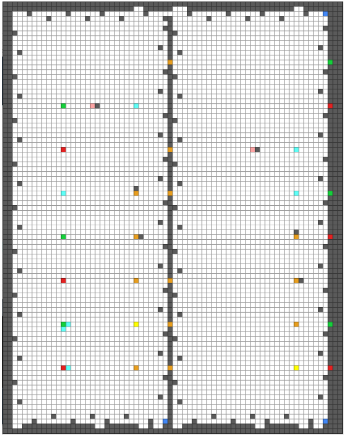
Main II Layer



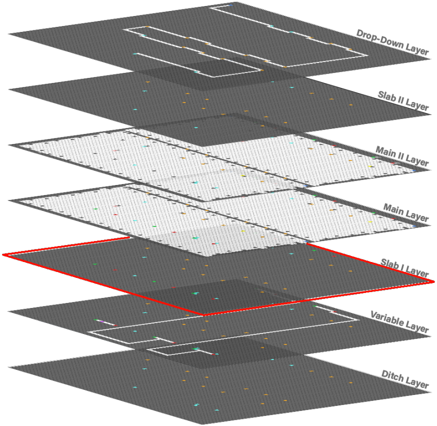


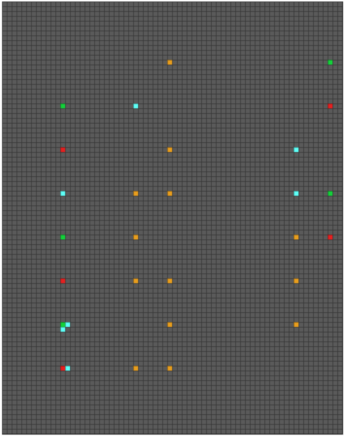
Main Layer


Drop-Down Layer
Slab II Layer
Main II Layer
Main Layer
Slab I Layer
Variable Layer
Ditch Layer

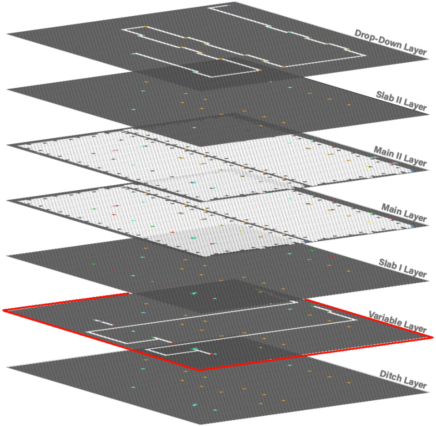


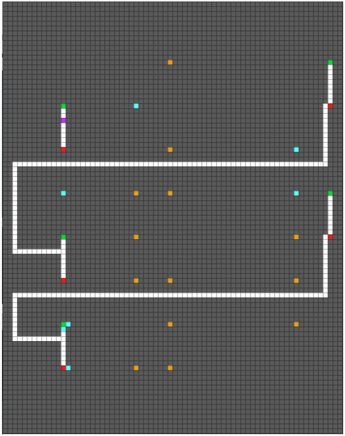
Slab I Layer


Drop-Down Layer
Slab II Layer
Main II Layer
Main Layer
Slab I Layer
Variable Layer
Ditch Layer

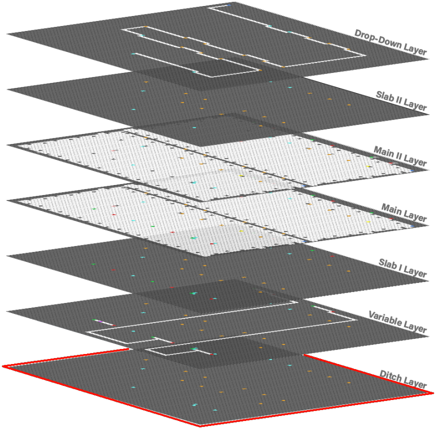


Variable Layer

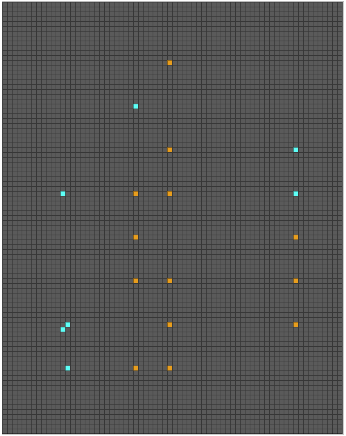




Ditch Layer



Drop-Down Layer
Slab II Layer
Main II Layer
Main Layer
Slab I Layer
Variable Layer
Ditch Layer



3-SAT Reduction Conclusion

- Note that this construction can be extended to more clauses and variables.
- Each clause will only have three variables so the construction will look similar.
- We have shown now that this problem is both in NP and NP-hard, so it is NP-complete.

Conclusion

- Explored the recurring ice-sliding motif
- Provided implementations to solve all 4 2-D variants in Polynomial Time
- Constructed a novel reduction from 3-SAT to show that the higher dimensional case of Pass Coverage is NP-complete.
- Future research may involve certain combinatorial problems, other variants of the game (with multiple agents or pushable blocks). More complex movement patterns can also be explored if the rules of the board are changed.

Acknowledgements

- Our mentor, Chun Hong Lo, for his invaluable feedback and guidance
- MIT PRIMES, for research opportunity
- Tanya Khovanova, for her editing feedback
- Our parents, for their continued love and support

Bibliography

[1] Greg Aloupis, Erik D Demaine, Alan Guo and Giovanni Viglietta. *Classic games are computationally hard. Theoretical Computer Science, 586:135-160, 2015.*

[2] W.W.Rouse Ball and H.S.M.Coxeter. *Mathematical Recreations and Essays. The Macmillan Company, 1939.*

[3] Stephan A. Cook. *The complexity of theorem-proving precedures. Proceedings of the third annual ACM symposium on Theory of computing - STOC 71, 1971.*

[4] Erik D Demaine, Michael Hoffmann, and Markus Holzer. *Pushpush-k is pspace-complete. In Proceedings of the 3rd International Conference on FUN with Algorithms, pages 159– 170. Citeseer, 2004.*

[5] Erik D Demaine, Susan Hohenberger, and David Liben-Nowell. *Tetris is hard, even to approximate. In International Computing and Combinatorics Conference, pages 351– 363. Springer, 2003.*

[6] Gary William Flake and Eric B Baum. *Rush hour is pspace-complete, or "why you should generously tip parking lot attendants". Theoretical Computer Science, 270(1- 2):895–911, 2002.*

[7] Paul Harrison. *Robust topological sorting and tarjan's algorithm in python, 2011.*

[8] Arthur M Jaffe. *The millennium grand challenge in mathematics. Notices of the AMS, 53(6), 2006.*

[9] Andre Fabbri Julien Moncel Aline Parreau et al. Paul Dorbec, Eric Duchene. *Ice sliding games. International Journal of Game Theory, 47:487–508, 2018.*

[10] Daniel Ratner and Manfred Warmuth. *The* $(n^2 - 1)$-puzzle and related relocation problems. Journal of Symbolic Computation, 10(2):111–137, 1990.

[11] Robert Tarjan. *Depth-first search and linear graph algorithms. SIAM journal on computing, 1(2):146–160, 1972.*

[12] Pedro J Tejada. *On the complexity of collecting items with a maximal sliding agent. 2014.*

[13] John Talbot and Dominic Welsh. *Complexity and Cryptography. Cambridge University Press, first edition, 2006.*