Physical Zero-Knowledge Proof: From Sudoku to Nonogram

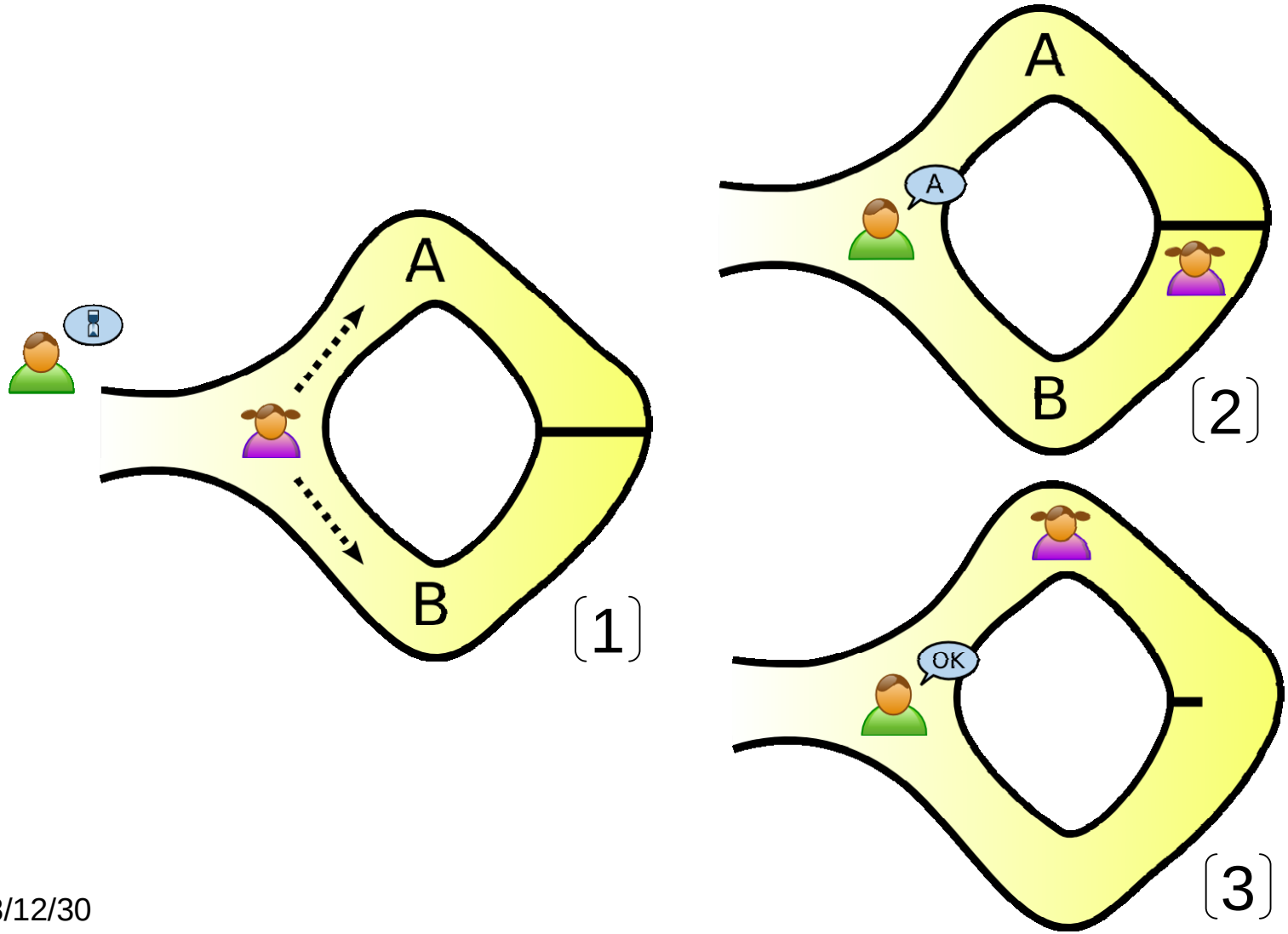
Wing-Kai Hon

(a joint work with YF Chien)

Outline

- Zero-Knowledge Proof (ZKP)
  1. Cave Story
  2. 3-Coloring
  3. Sudoku
  4. Nonogram (our work)

Cave story



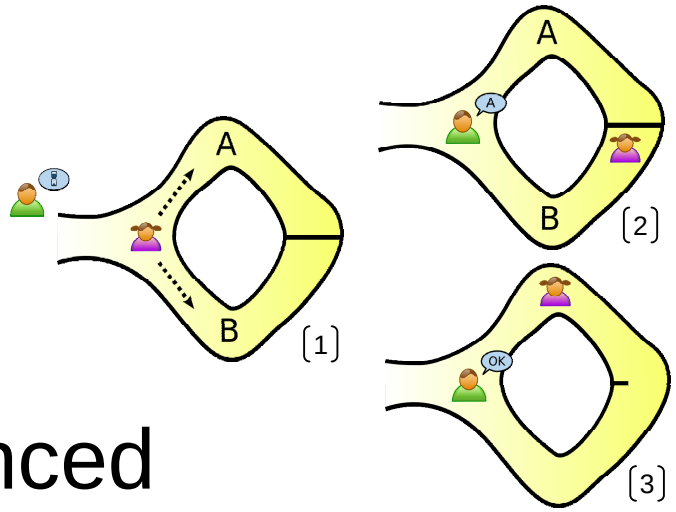
Cave story

If Peggy really knows

the magical password

➔ always pass the test

➔ Victor should be convinced



If Peggy does not know the password

➔ very unlikely to pass all the tests

➔ Victor should be able to catch Peggy lying

Zero-Knowledge Proof

It is an interactive method :

allows a Prover to prove to a Verifier

that a statement is true

Extra requirement :

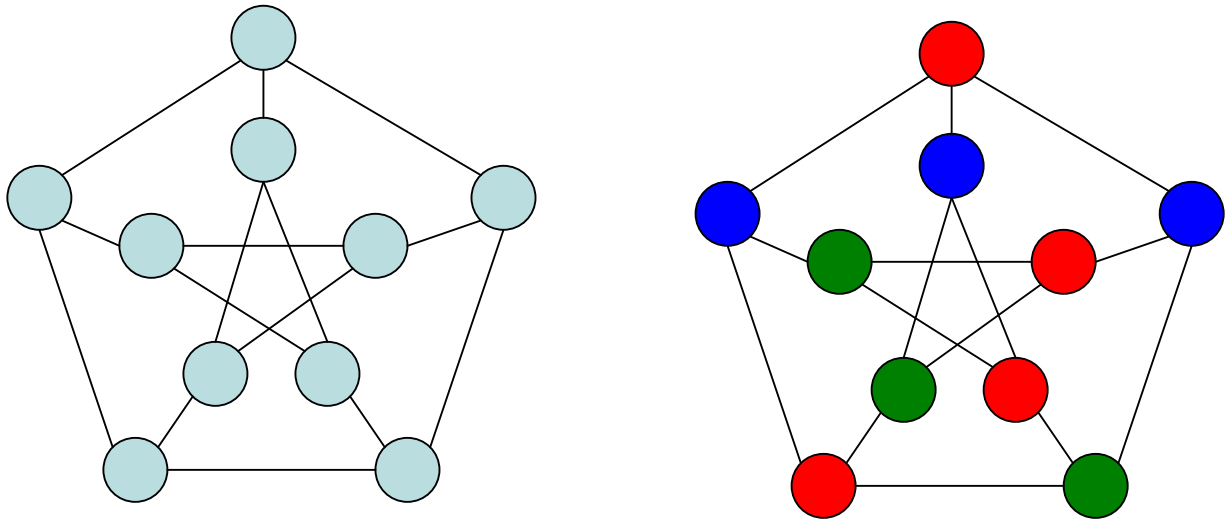
Zero-Knowledge :

- Verifier *cannot* learn anything

3-Coloring of a Graph

Given a graph *G*,

can we color the vertices with 3 colors so that no adjacent vertices have same color?

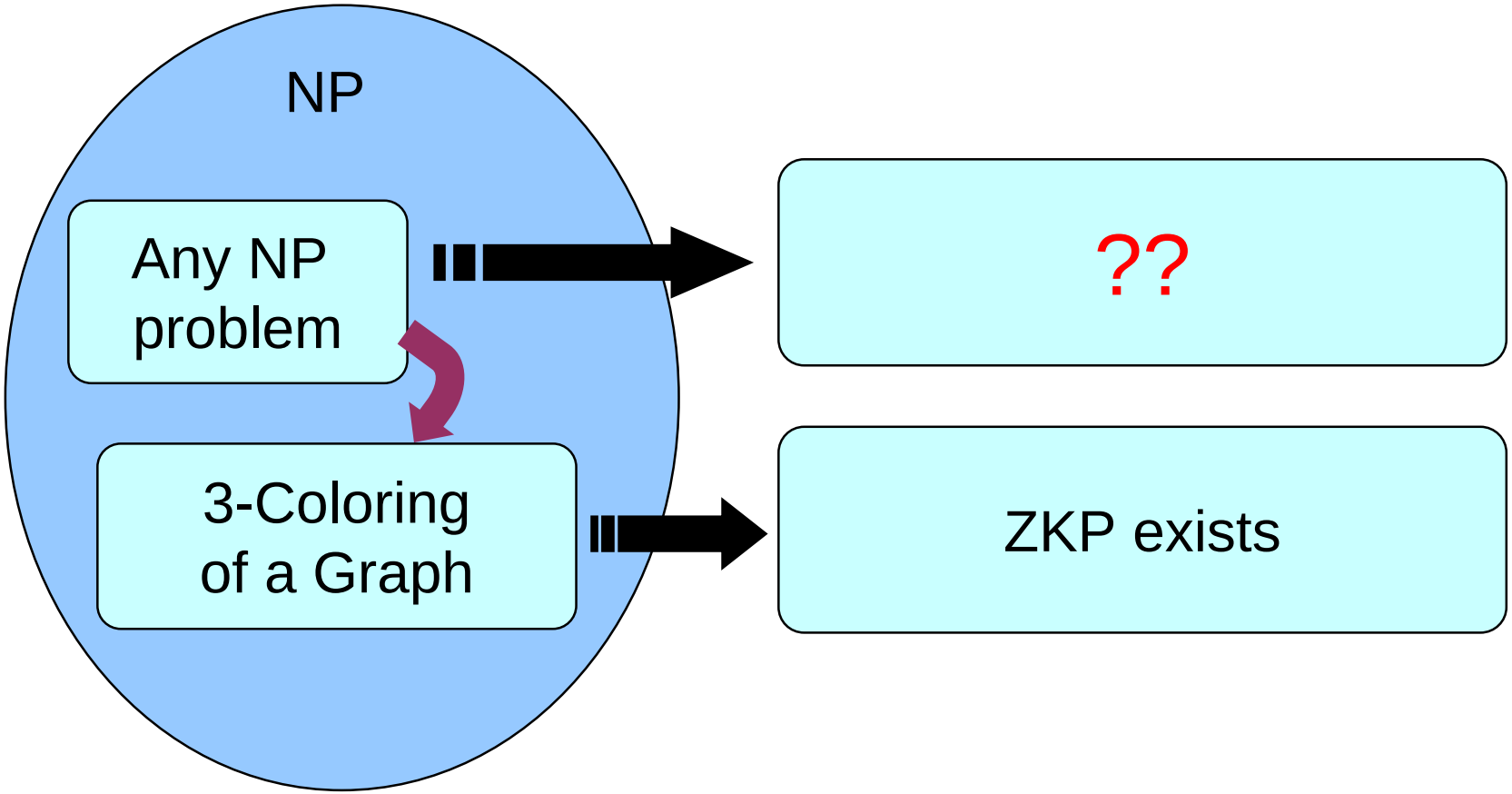


3-Coloring of a Graph

One possible method:

1. Verifier leaves the room
2. Prover colors vertices properly (in R, G, B)
   and then covers each vertex with a cup lid
3. Verifier comes back to the room
   and select any edge (u,v)
4. Verifier removes cup lids of u and v
   and checks if they have different colors

Consequence



Sudoku Puzzle

A numeric logic puzzle

Target:

Fill each entry with an integer from {1,2, …, 9}

such that

each row, each column, and each 3 x 3 block contains 1 to 9

5					1			8
						6		
				6	2	5	7	
	9		2		5	1		
		4		1		3		
		8	3		9		2	
	7	6	9	8				
		5						
8			1					3

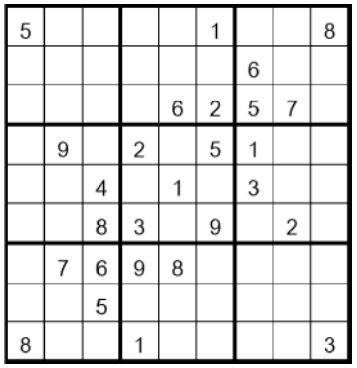
Created by David Eppstein

Sudoku Puzzle

Some Facts:

- Invented in 1979 by Howard Garns
- Named “Sudoku” in 1986
- Made popular by Wayne Gould
- NP-Complete (Takayuki Yato, 2003)
- A Simple ZKP
  (Gradwohl and others, 2007)

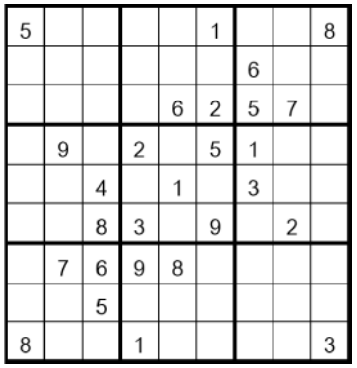
Sudoku Puzzle



One possible method:

1. Verifier leaves the room
2. Prover prepares 3 decks of cards
   and uses the cards to write out the solution
3. Prover flips the cards in originally empty entries
4. Verifier chooses one of the tests:
   Row test, Column test, Block test
5. (Next Page)

Sudoku Puzzle



5(a) For Row Test :

Prover collects the cards in each row into a pile (thus, in total 9 piles) ; Next, shuffle the cards in each pile ; Finally, Verifier examines if each pile contains 1 to 9

5(b) For Column Test : Similar

5(c) For Block Test : Similar

Nonogram Puzzle

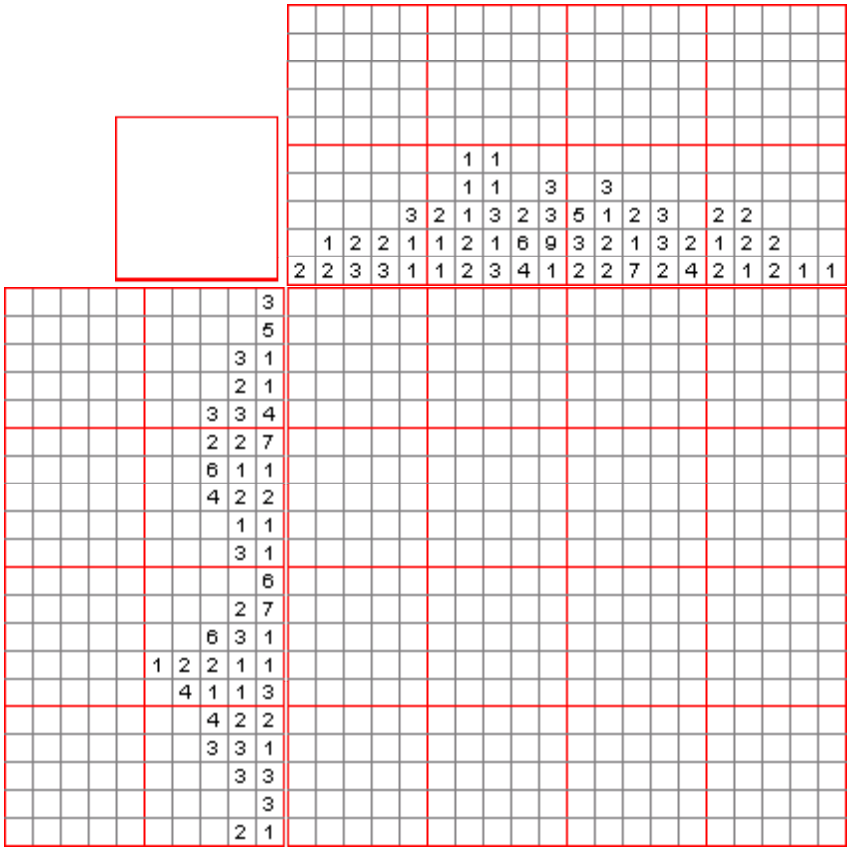
A picture logic puzzle

Target:

Each cell to be colored black or white

such that

occurrence of black blocks in each row and each column follows the corresponding hint



Copied from Wikipedia

Nonogram Puzzle

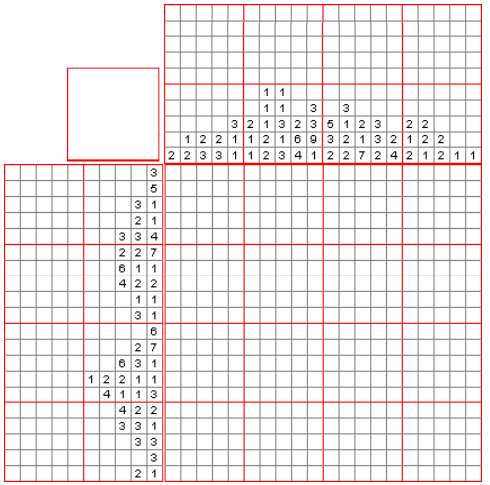
Some Facts:

- Invented separately in 1987 by
  Non Ishida and Tetsuya Nishio
- Made popular by Nintendo
- NP-Complete
  (Nobuhisa Ueda and Takaaki Nagao, 1996)
- A Simple ZKP ? (We are working on it)

Nonogram Puzzle

One idea: Reuse method of Sudoku

➔ Then, we can check if each row and each column has the correct number of black cells



However, we still need to check

1. the “blocking” of black cells
2. the ordering of the blocks

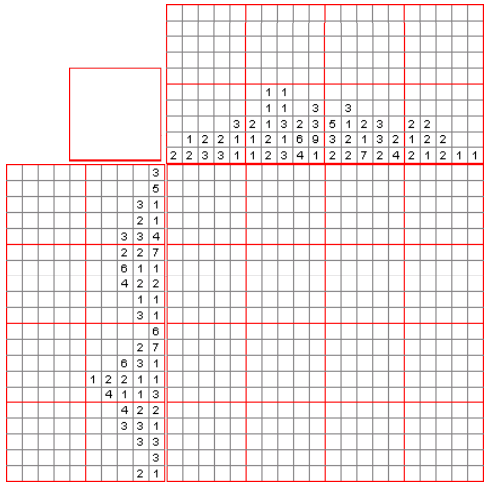
Nonogram Puzzle

Observation:

If we use one card for each cell,

then after shuffling, the position info

of the card disappears



➔ Links between adjacent cards disappear

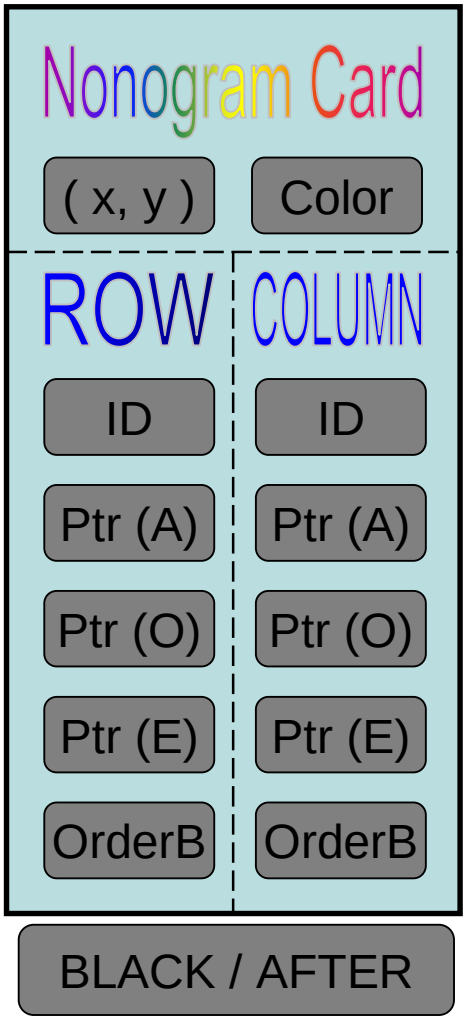
➔ Checking for the “blocking” seems impossible

One idea: Store hidden links in the cards

Scratch-Off Card

- We use 2 tools :
  - Scratch-off cards → Commitments
  - Bags → Shuffling

Our Scratch-off Cards



Framework of Our Protocols

Prover Phase:

1. Distributes blank cards to cells
2. Assigns IDs to each cell
3. Fills in the entries in the cards properly and seals each entry afterwards

Framework of Our Protocols

Verifier Phase:

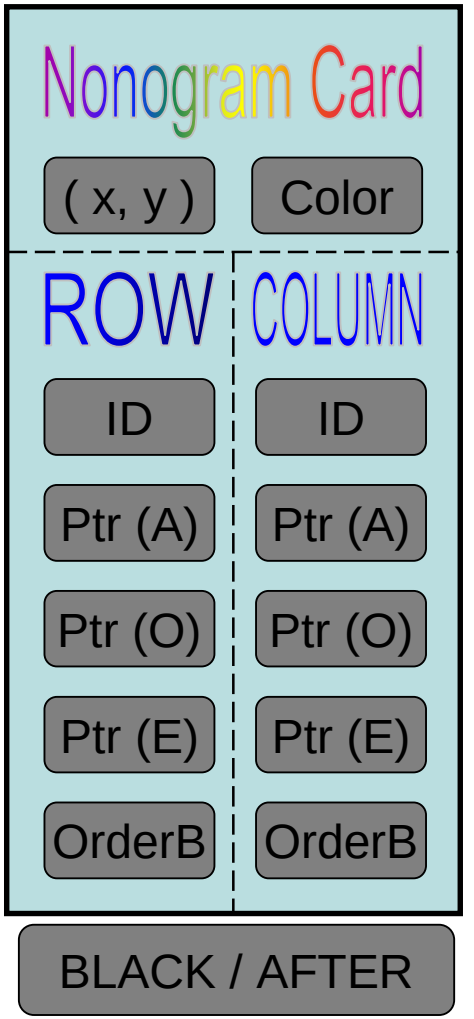
1. Selects a test (7 available tests)
2. Open *specific entries* of each card according to the selected test

Our Scratch-Off Cards

- (x,y) : Position of the cell
- Color : the color of the cell *in the solution*

- BLACK / AFTER :

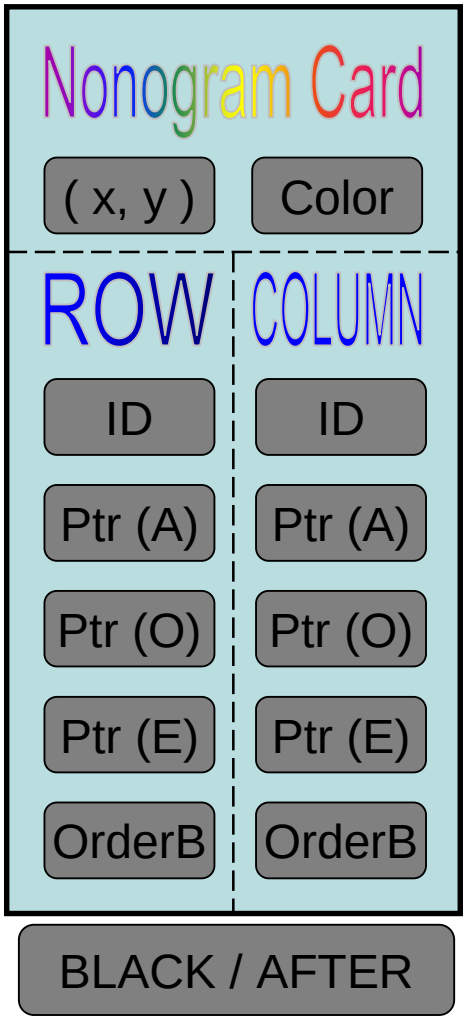
  (1) IDs of all subsequent cells
  (2) marking of black cells



Our Scratch-Off Cards

- Row & Column
  - ID : ID of the cell
  - Ptr (A) : ID of the next cell
  - OrderB :

    if color of the cell is black, record its order among black cells in same row / column



Example of filled cards

- How the card**s** represent

2,1,3,1														
ID	2	4	7	8	3	5	1	10	14	9	6	11	12	15
Ptr(A)	4	7	8	3	5	1	10	14	9	6	11	12	15	x
Ptr(O)	4	7	8	x			10	14	9	6	x			
Ptr(E)				3	5	x						12	15	x

BLACK / AFTER of a card

ROW	15		11	9	14	6		12
				■	■			■
COLUMN	21	24	22	27	23	28	29	
		■	■	■	■			

Theorem 1

- If Prover can fill the cards properly,
  such that
  (S1): The *(x,y) entry* of each card is correct;
  (S2): All *IDs* are different and all *IDs, Ptr(A),*
  and *(x,y)* entries, are consistent;
  (S3): The Black / After of each card contains the *correct set of IDs*;

Theorem 1

(S4): Each row and each column contains the *correct number of black* cards;

(S5): For each row and each column,

IDs and OrderB entries of all cards with black color are consistent with the set of IDs marked by ■ in Black / After entries;

Theorem 1

(S6): For each row and each column,
each odd-ranked block has correct # of black cells with correct OrderB entries;

(S7): For each row and each column,
each even-ranked block has correct # of black cells with correct OrderB entries;

then, Prover must know the solution.

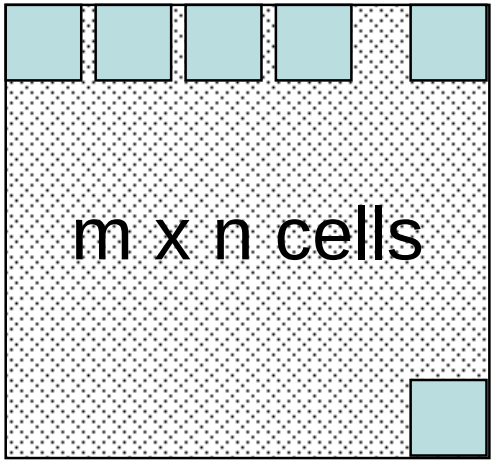
Theorem 2
(Converse of Theorem 1)

- If a person knows the solution,
  then the cards can be filled such that
  all statements S1 to S7 are satisfied

Physical ZKP

Prover Phase

- Writes all information on m*n cards and seals them.
  - Prover cannot change the position after all cards are put on cells



Physical ZKP

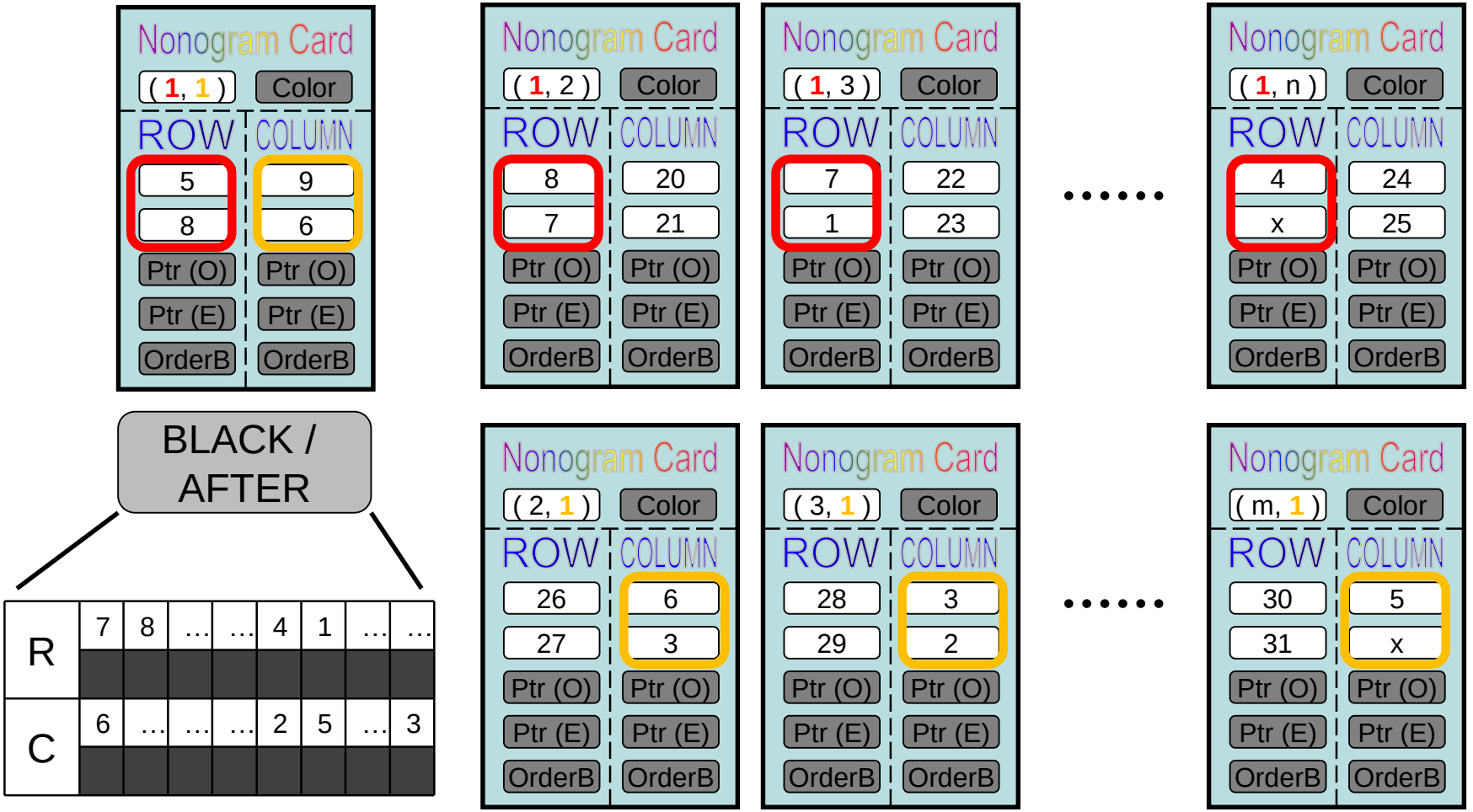
Verifier Phase

- Chooses one of the ***7*** tests
- Open specific entries in each card according to the selected test

Test 1: Pointer System

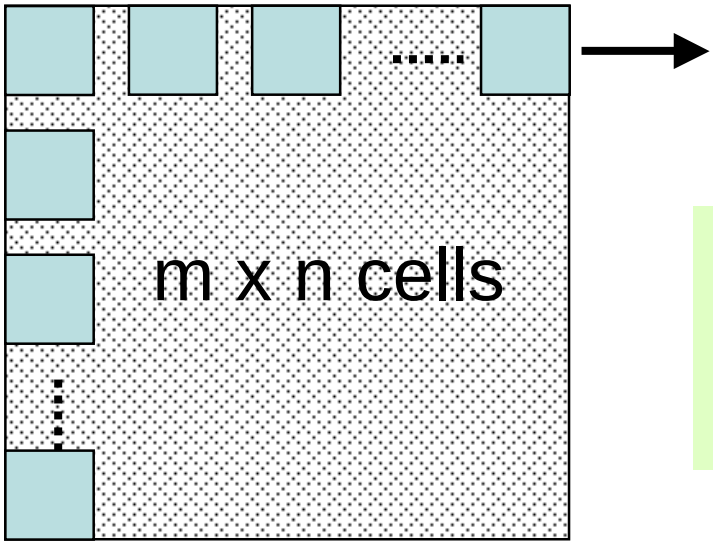
- Simultaneously test S1, S2, and S3
- Opens the following entries of each card:
  - (*x, y)*
  - *ID*
  - *Ptr(A)*
  - *the ID part of Black/After*

Physical ZKP (Test 1)



Test 2: Black Cells Order in Row

- Simultaneously test S4-row and S5-row



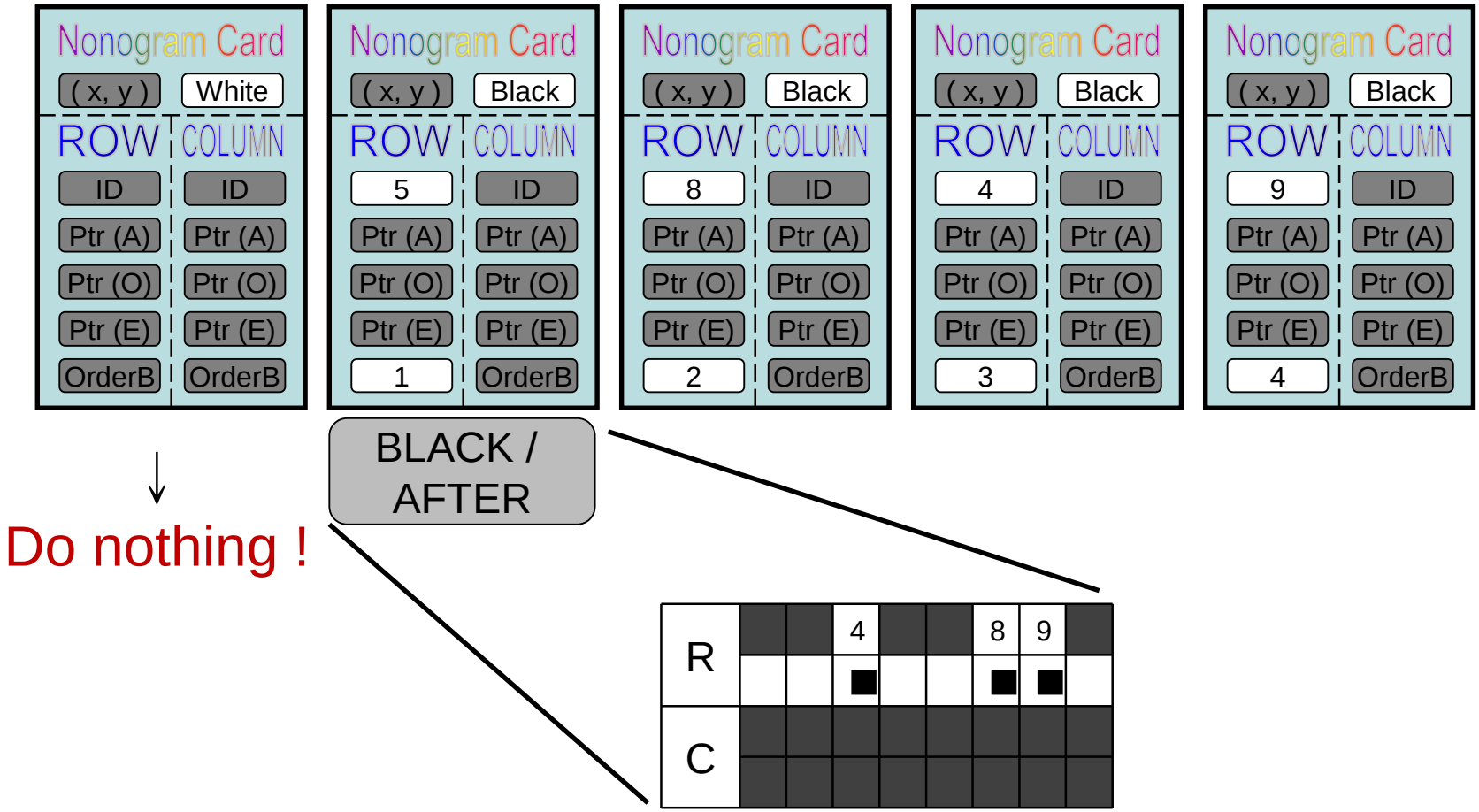
Take all cards

= m bags (row)

↓

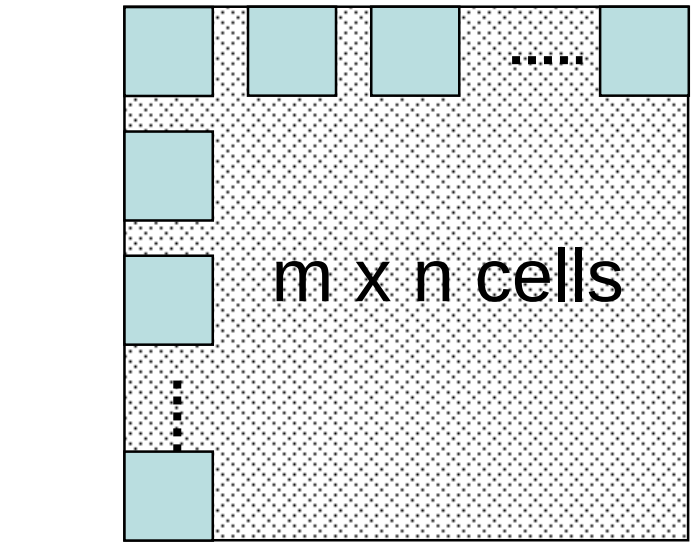
Shuffle each bag

Physical ZKP (Test 2)



Test 3: Black Cells Order in Col

- Simultaneously test S4-col and S5-col



Take all cards

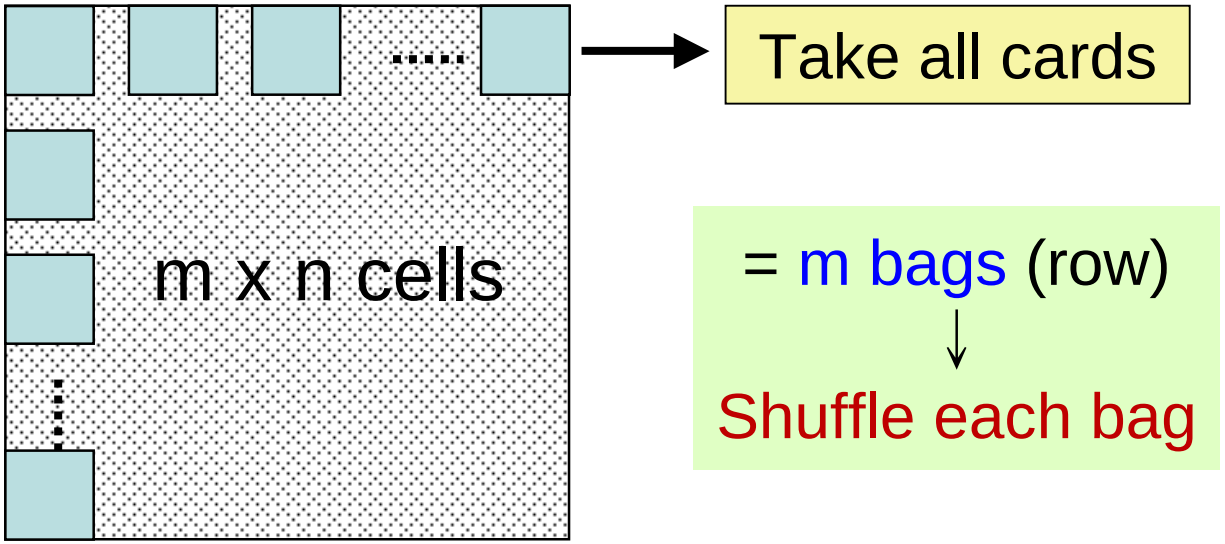
= n bags (col)

↓

Shuffle each bag

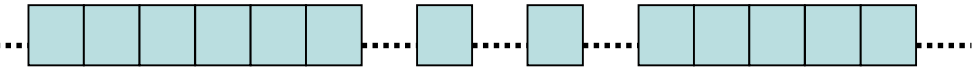
Test 4: Odd Blocks in Row

- (T4) Testing S6-row

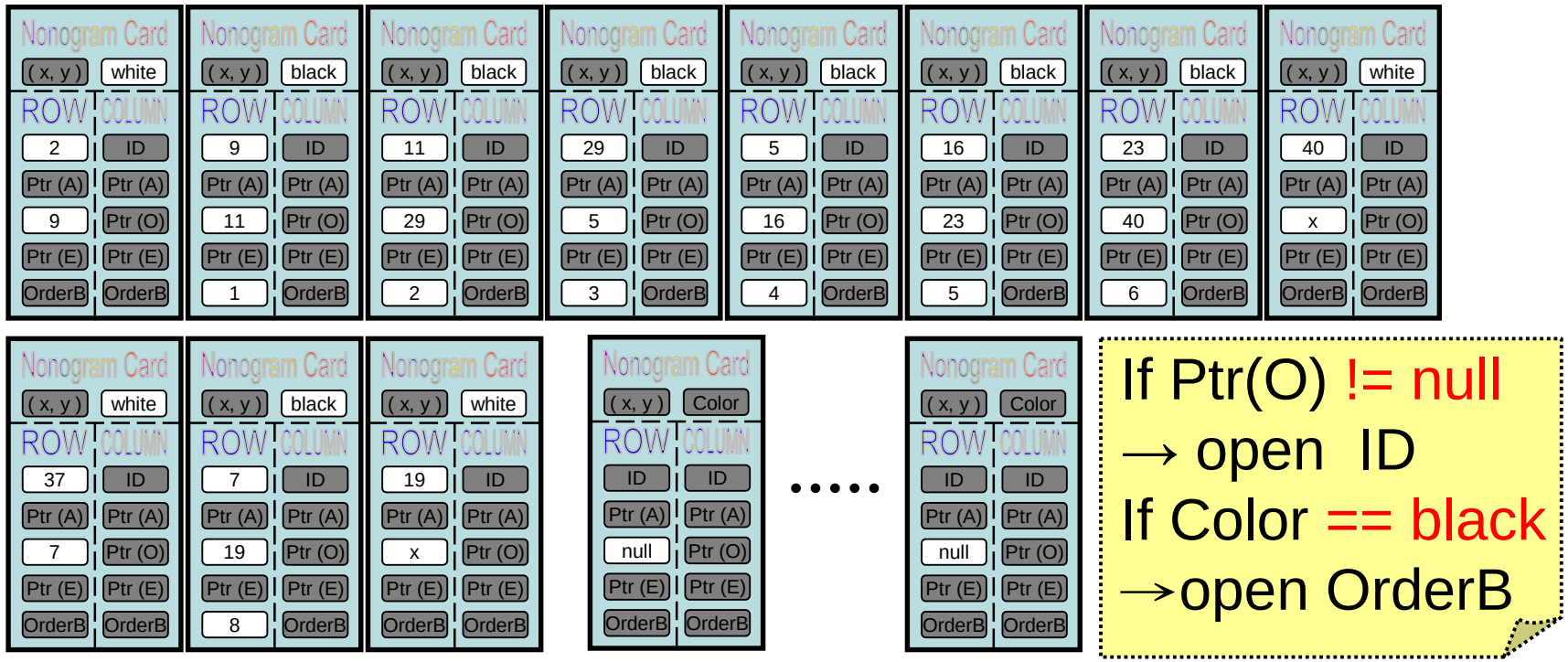


Physical ZKP (Test 4)

- Ex : 6115 →



  - Check 6?1?



If Ptr(O) != null
→ open ID
If Color == black
→open OrderB

Other Tests

- (T5) Testing S6-col
- (T6) Testing S7-row
- (T7) Testing S7-col

Analysis

Theorem:

Our method is zero-knowledge.

Also, whenever Prover knows the solution, he always pass the test.

However, if Prover does not know the solution, he will fail at least one test

➔ Verifier can catch him lying with probability at least 1/7.

Analysis

Zero-Knowledge *:*

After each test, Verifier gains no info except for some **trivial** facts, or some **unrelated** things to the solution

E.g.,

- number of black cells in each row
- number of black cells in each odd block
- IDs in each cell

Other Links

- To know more about ZKP
  - O. Goldreich' book

    Foundations of Cryptography: Basic Tools
- How to solve Sudoku ?
  - D. Eppstein's paper on Sudoku
- To know more about ZKP for Sudoku
  - R. Gradwohl et al's paper (FUN 2007)

Other Links

- To know more about Nonogram
  - Wikipedia

- To know more about ZKP for Nonogram
  - YF Chien's Master Thesis (2008)