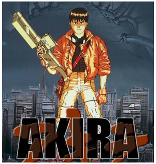
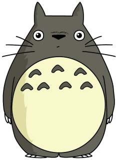
Physical Zero-Knowledge Proofs for Akari, Takuzu, Kakuro and KenKen





X. Bultel[1] J. Dreier[2] J-G. Dumas[3] P. Lafourcade[1]

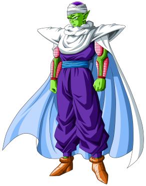
[1]LIMOS, University Clermont Auvergne, France

[2]Université de Lorraine, LORIA, Nancy, France

[3]LJK, Université Grenoble Alpes, Grenoble, France

FUN'16, 9th June 2016, Sardinia

Zero-Knowledge proof of knowledge



Prover knows
a solution s of P

Verifier knows
the problem P

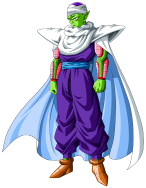
$$\xrightarrow{\text{bla bla...}}$$

$$\xleftarrow{\text{bla bla?}}$$

$$\xrightarrow{\text{bla bla!}}$$

accept or reject
s as a solution of P

Completeness



Prover knows
a solution s of P

Verifier knows
the problem P

$$\xrightarrow{\text{bla bla...}}$$

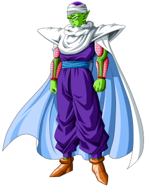
$$\xleftarrow{\text{bla bla?}}$$

$$\xrightarrow{\text{bla bla!}}$$

Hum, ok...

I'm convinced!
s is a solution of P

Soundness



Prover does not know
a solution s of P

Verifier knows
the problem P

$$\xrightarrow{\text{bla bla...}}$$

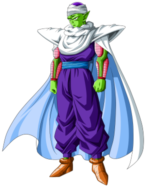
$$\xleftarrow{\text{bla bla?}}$$

$$\xrightarrow{\text{bla bla!}}$$

Hum, ...

I detect a problem!
s is not a solution

Zero-Knowledge



Prover knows
a solution s of P

Verifier knows
the problem P

$\xrightarrow{\text{bla bla...}}$

$\xleftarrow{\text{bla bla?}}$

$\xrightarrow{\text{bla bla!}}$

I do not learn
anything about s

Origins of ZKP

- Introduced by S. Goldwasser, S. Micali, and C. Rackoff in 1985.



- O. Goldreich, S. Micali, and A. Wigderson 1991: Polynomial ZKP for every problem in NP under the existence of one-way functions.



Related Works

R. Gradwohl, M. Naor, B. Pinkas, and G. N. Rothblum **(FUN'07)**
Physical (using cards) ZKP for Sudoku.

Prover

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

Verifier

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

$\xrightarrow{\text{bla bla bla...}}$

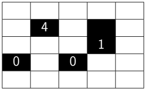
$\xleftarrow{\text{bla bla bla?}}$

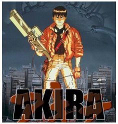
$\xrightarrow{\text{bla bla bla!}}$ accept or reject

Contributions

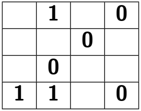
Physical Zero-Knowledge Proofs for 4 NP-complete games:

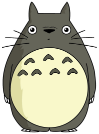
- Akari

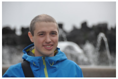




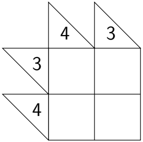
- Takuzu



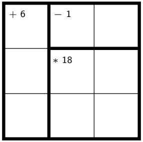




- Kakuro

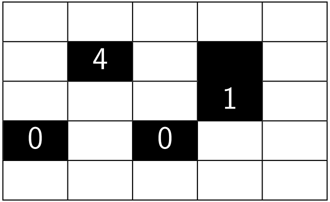


- KenKen



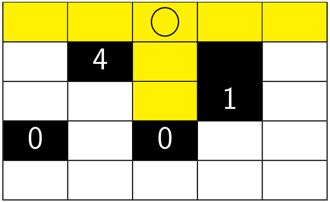
Akari

GOAL: Place lights on the white cells on the grid such that 3 constraints are respected



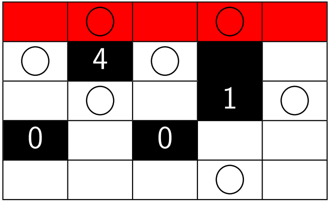
Akari

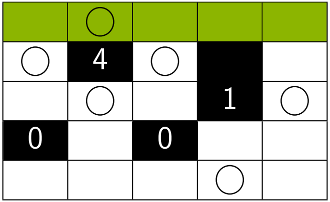
A light ◯ illuminates the whole row and column up to a black cell.



Contraints (1/3)

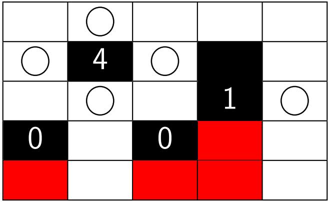
- Two lights cannot illuminate each other

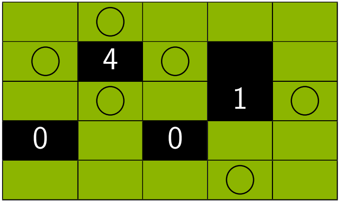




Contraints (2/3)

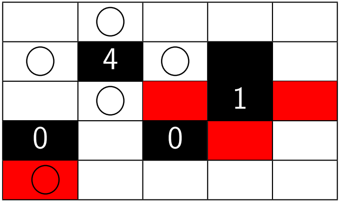
- All cells are illuminated !

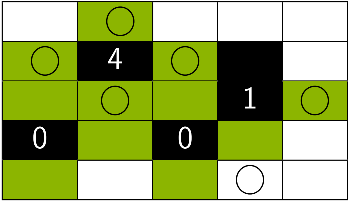




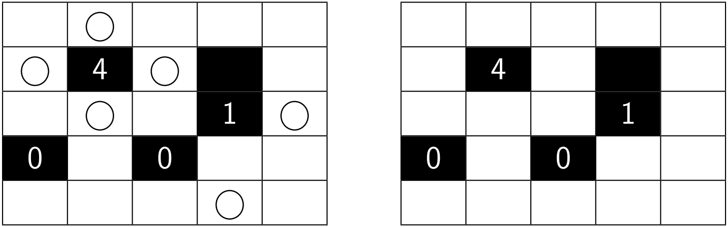
Contraints (3/3)

- Numbers in black cells = adjacent lights





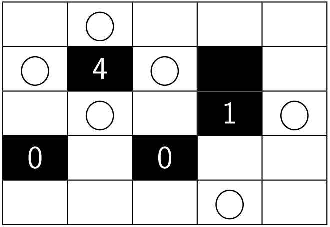
Prover Commitment

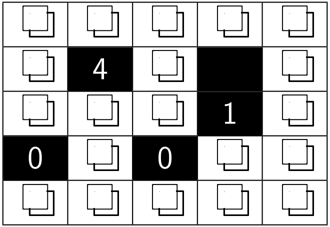


Prover commitment:

- use the empty grid, empty cards and ◯ cards.

Prover Commitment

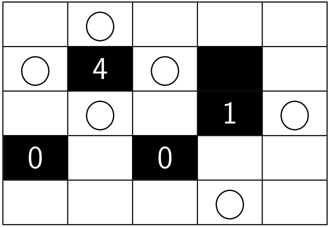


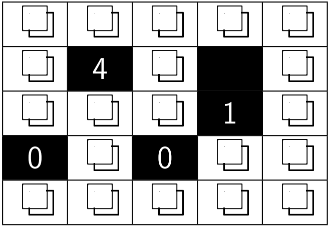


Prover commitment:

- use the empty grid, empty cards and ◯ cards.
- put a packet of **identical** cards on each white cell according to the solution.

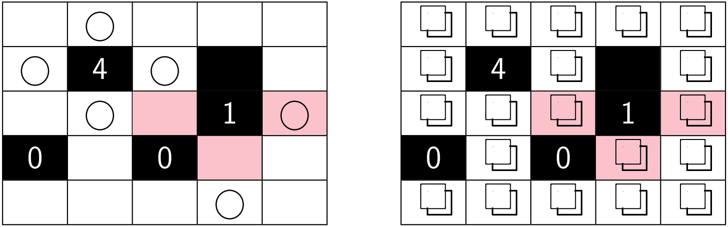
Verification (1/3)





Numbers in black cells = adjacent lights

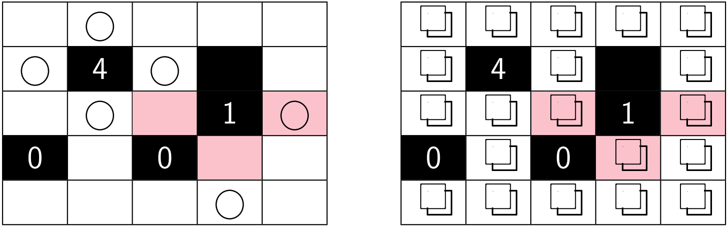
Verification (1/3)



Numbers in black cells = adjacent lights

For each black cell with number x:
pick one card in all adjacent white cells and shuffle them.

Verification (1/3)

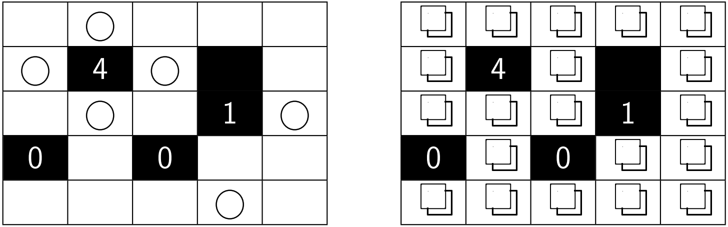


Numbers in black cells = adjacent lights

For each black cell with number x:
pick one card in all adjacent white cells and shuffle them.

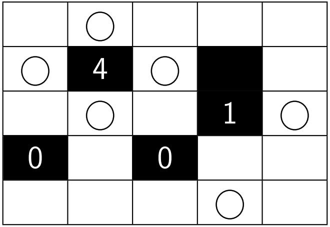
V checks that there is exactly x ◯ cards.

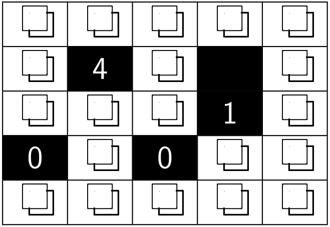
Verification (2/3)



No two lights see each other ⇔ At most one ◯ by row/column.

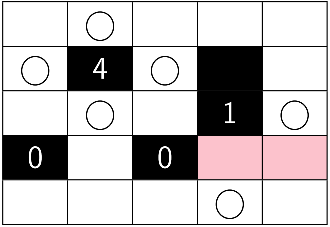
Verification (2/3)

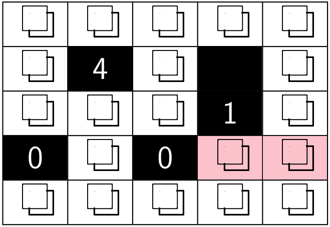




No two lights see each other ⇔ At most one ◯ by row/column.
For each row/column, take one card per cell and shuffle them.

Verification (2/3)

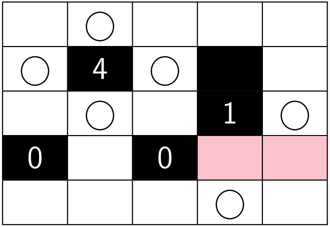


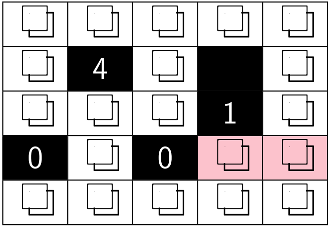


No two lights see each other ⇔ <u>At most</u> one ◯ by row/column.
For each row/column, take one card per cell and shuffle them.

- **case 1**, empty cards: P adds a ◯ card

Verification (2/3)

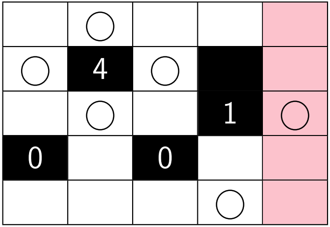


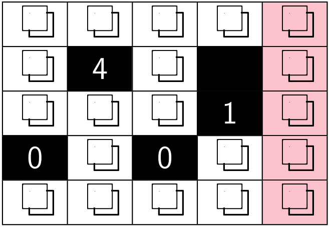


No two lights see each other ⇔ At most one ◯ by row/column.
For each row/column, take one card per cell and shuffle them.

- **case 1**, empty cards: P adds a ◯ card → exactly 1 ◯

Verification (2/3)

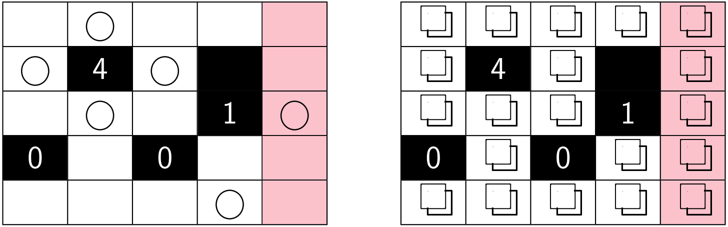




No two lights see each other ⇔ At most one ◯ by row/column.
For each row/column, take one card per cell and shuffle them.

- **case 1**, empty cards: P adds a ◯ card → exactly 1 ◯
- **case 2**, one ◯: P adds an empty card

Verification (2/3)

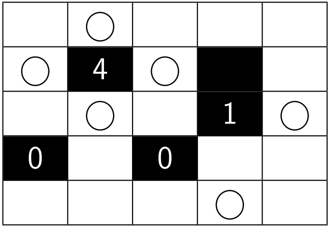


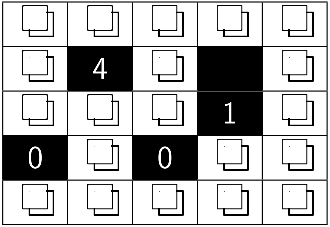
No two lights see each other ⇔ <u>At most</u> one ◯ by row/column.
For each row/column, take one card per cell and shuffle them.

- **case 1**, empty cards: P adds a ◯ card → exactly 1 ◯
- **case 2**, one ◯: P adds an empty card → exactly 1 ◯

V checks that there is exactly one ◯ card.

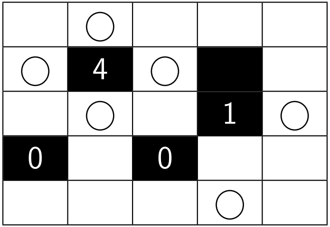
Verification (3/3)

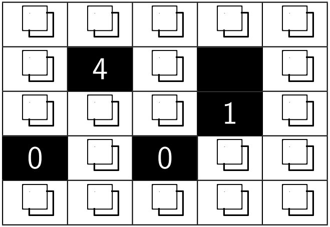




All cells are illuminated ⇔ For each cell, at least one ◯ in its row and column.

Verification (3/3)

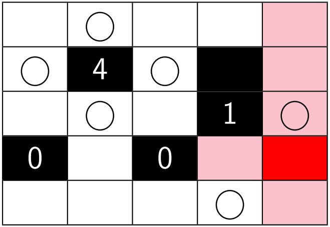


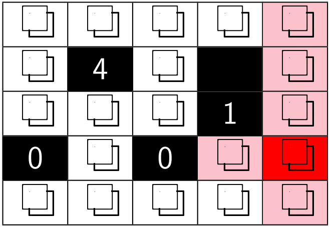


All cells are illuminated ⇔ For each cell, at least one ◯ in its row and column.

For each cell, take one card per cell in the same row and column and shuffle them.

Verification (3/3)



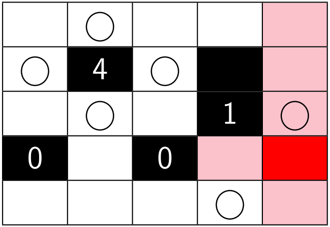


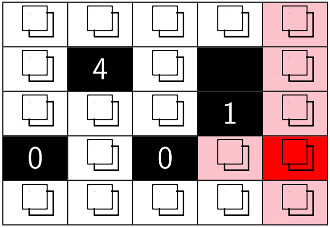
All cells are illuminated ⇔ For each cell, <u>at least</u> one ○ in its row and column.

For each cell, take one card per cell in the same row and column and shuffle them.

- **case 1**, one ○: *P* adds a ○ card

Verification (3/3)



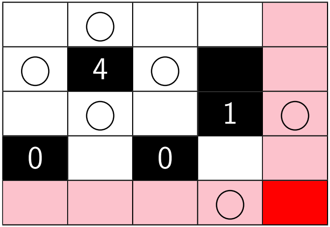


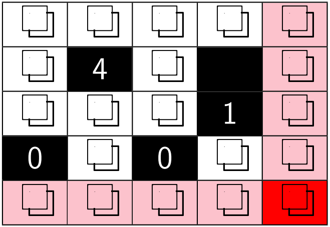
All cells are illuminated $\Leftrightarrow$ For each cell, at least one ◯ in its row and column.

For each cell, take one card per cell in the same row and column and shuffle them.

- **case 1**, one ◯: *P* adds a ◯ card $\rightarrow$ exactly 2 ◯

Verification (3/3)



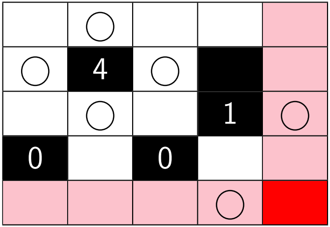


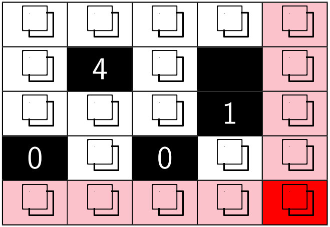
All cells are illuminated ⇔ For each cell, <u>at least</u> one ◯ in its row and column.

For each cell, take one card per cell in the same row and column and shuffle them.

- **case 1**, one ◯: P adds a ◯ card → exactly 2 ◯
- **case 2**, two ◯: P adds an empty card

Verification (3/3)



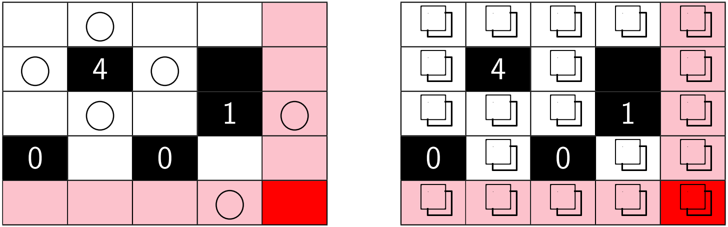


All cells are illuminated $\Leftrightarrow$ For each cell, at least one ◯ in its row and column.

For each cell, take one card per cell in the same row and column and shuffle them.

- **case 1**, one ◯: P adds a ◯ card $\rightarrow$ exactly 2 ◯
- **case 2**, two ◯: P adds an empty card $\rightarrow$ exactly 2 ◯

Verification (3/3)



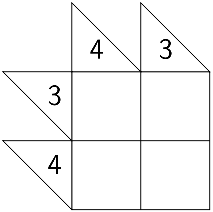
All cells are illuminated ⇔ For each cell, at least one ◯ in its row and column.

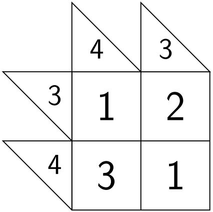
For each cell, take one card per cell in the same row and column and shuffle them.

- **case 1**, one ◯: *P* adds a ◯ card → exactly 2 ◯
- **case 2**, two ◯: *P* adds an empty card → exactly 2 ◯

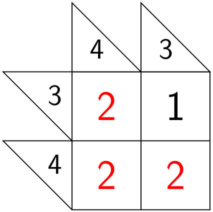
V checks that there is exactly two ◯ cards.

Kakuro: Cross Sums





- Digits from 1 to 9.
- Triangular cell = sum of digits in the row/column
- A number can appear only once per row/column.



Digit Encoding

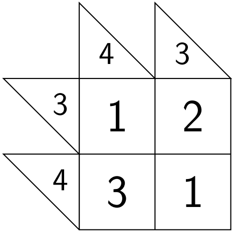
Using black and red cards.
To represent a number x put in an envelope:

- $9 - x$ black cards
- x red cards

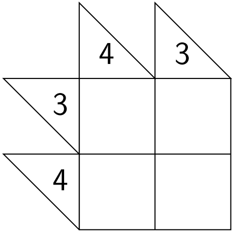
For 3: 

Prover Commitment

- **Draw an empty grid**

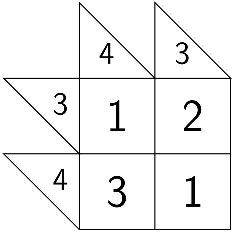


Commitment:

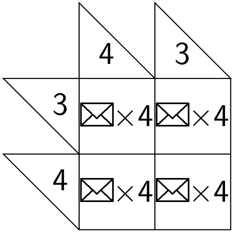


Prover Commitment

- **Draw an empty grid**
- **On each empty cell:** put 4 identical envelopes encoding the digit

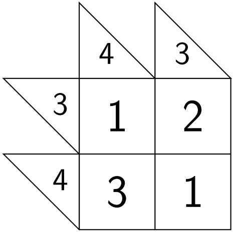


Commitment:



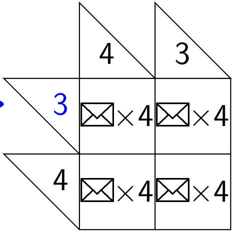
Prover Commitment

- **Draw an empty grid**
- **On each empty cell:** put 4 identical envelopes encoding the digit
- **On each triangular cell:** put envelopes encoding all missing digits in the row/column



Commitment:

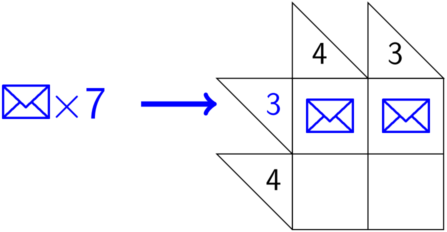
✉×7 for 3, 4, 5, 6, 7, 8 and 9 ⟶



Verification (1/2)

A number appears only once per row/column

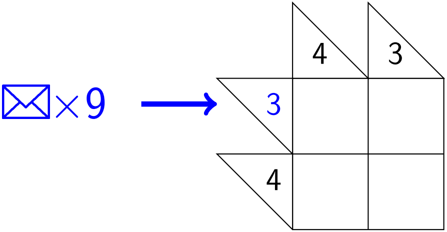
- For each row/column, pick an envelope per cell plus the envelopes on the triangular cell.



Verification (1/2)

A number appears only once per row/column

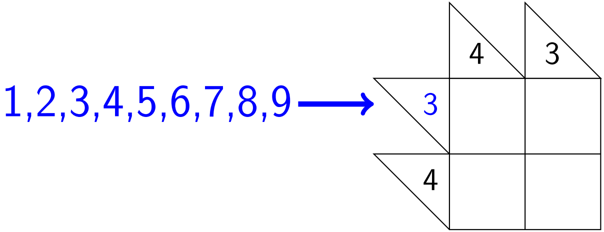
- For each row/column, pick an envelope per cell plus the envelopes on the triangular cell.
- Shuffle and open them.



Verification (1/2)

A number appears only once per row/column

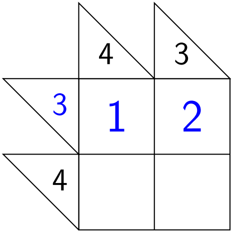
- For each row/column, pick an envelope per cell plus the envelopes on the triangular cell.
- Shuffle and open them.
- Verify that all numbers between 1 and 9 appear exactly once.

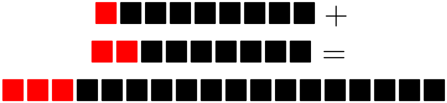


Verification (2/2)

The sum per row and per column corresponds to the number in the triangular cell

- Randomly picks one envelope per cell in the row/column.
- Opens (face down) the content of each envelope and shuffle it.
- Check that red cards corresponds to the number given in the triangular cell.





KenKen

+ 6	− 1	
	∗ 18	

+ 6 3	− 1 1	2
1	∗ 18 2	3
2	3	1

- **Addition:** similar to Kakuro.
- **Multiplication:** addition of the exponent of each prime factors.

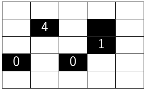
$$9 \times 6 = (2^0 3^2) \times (2^1 3^1) = 2^{0+1} 3^{2+1} = 54$$

- **Substraction/division:** finding the maximum.

Conclusion

Physical Zero-Knowledge Proofs for:

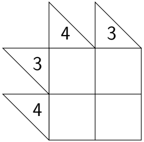
- Akari



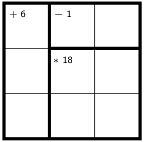
- Takuzu

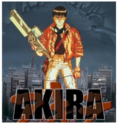
	1		0
		0	
	0		
1	1		0

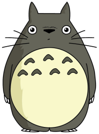
- Kakuro



- KenKen







More Games !

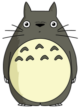
Conclusion

Physical zero-knowledge mechanisms for several constraints:

- At least/most one occurence of a symbol in a row/column.
- Equality of the number of 1 and 0 per row/column.
- Result of the addition/substraction of cells.
- Result of the multiplication/division of cells.
- Number of adjacent symbol.
- All rows/columns are different.
- No k consecutive identical symbols.

Thank you for your attention.

Questions?



Takuzu Rules: Binary Puzzle

Goal: fill the grid with 0's and 1's

	1		**0**
		0	
	0		
1	**1**		**0**

- Each row/column has exactly the same number of 1's and 0's
- Each row/column is unique
- In each row/column there can be no more than 2 identical numbers next to each other: 110010, but 110001

0	**1**	1	**0**
1	0	**0**	1
0	**0**	1	1
1	**1**	0	**0**