COSC1107 Computing Theory

(We will commence soon. We are just allowing a few minutes for people to join and set up. *Please mute your microphone unless you are speaking.* You can raise your hand or use the chat at any time.)

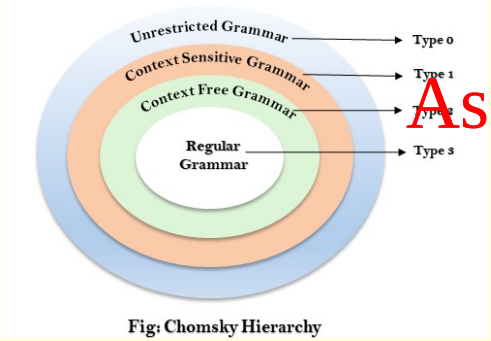
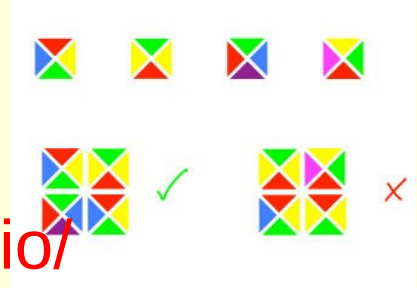
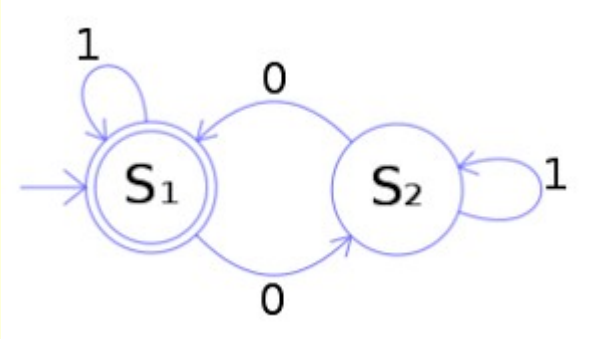
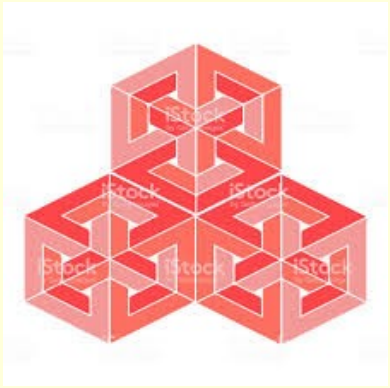
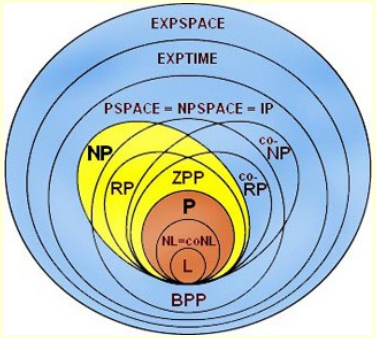


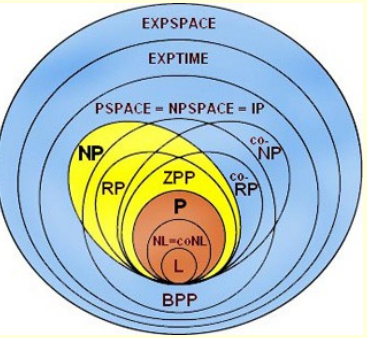
Fig: Chomsky Hierarchy

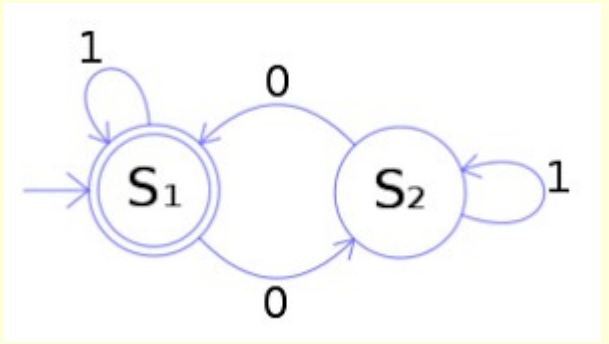












COSC1107

Week

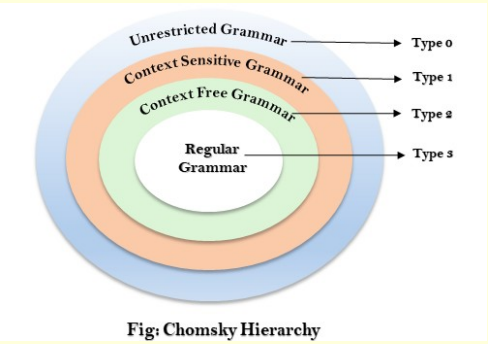


Fig: Chomsky Hierarchy

James Harland

james.harland@rmit.edu.au

* With thanks to Sebastian Sardina

Intro music 'Far Over' playing now …

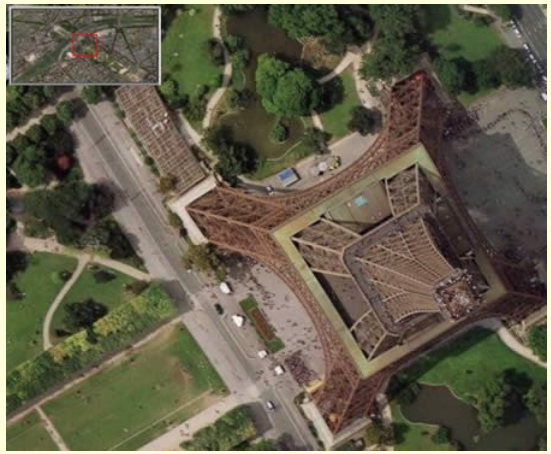
Acknowledgement

RMIT University acknowledges the people of the Woi wurrung and Boon wurrung language groups of the eastern Kulin Nations on whose unceded lands we conduct the business o University respectfully ac stors and Elders, past and present.

RMIT also acknowledges the Traditional Custodians and their Ancestors of the lands and waters across Australia where we conduct our business.

(add your name here to volunteer for this or email me)
(my personal Acknowledgement of Country is here)

Overview



- Questions?
- Authentication
- Questions?
- Secure dealing
- Questions?
- Zero-knowledge proofs
- Questions?
- Platypus Game
- Questions?

Weekly Schedule

	Lecture/Lectorial	Tutorial	Assessment
1	Formal languages, grammars	Motivations & Mathematical preliminaries	
2	Finite State Machines	Grammars	Quiz 1
3	Pushdown Automata, nondeterminism	NFAs and DFAs	Quiz 2
4	Turing machines	Pushdown automata	Quiz 3
5	Computability, universali		Quiz 4
6	Pumping Lemma, NFA->DFA conversion	Comp lity	Assignment 1, Quiz 5
7	Chomsky Hierarchy	Nond ing Lemma	Quiz 6
8	Unrestricted grammars		Quiz 7
9	Complexity and intractability	Unrestricted grammars	Quiz 8
10	NP-completeness	Complexity and intractability	Quiz 9
11	**Zero-knowledge proofs**	**NP-completeness**	Quiz 10
12	Research and requests	Sample exercise	Assignment 2
14-16	--	--	Final exercise

Foundations

elationships

Analysis

Assessment

Weekly Schedule

	Lectorial	Tutorial	Assessment
11	Zero-knowledge proofs	NP-completeness	Quiz 10
12	**Research and requests**	Sample exercise	Assignment 2
14-16	--	--	Final exercise

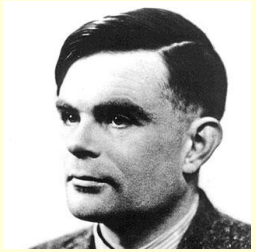
- Send me request **riday 8th October**
- Some parts of Assignment 2 w itted **online**
- Sample exercise will be in tutorials next week
- Information about all of these will be announced on Ed

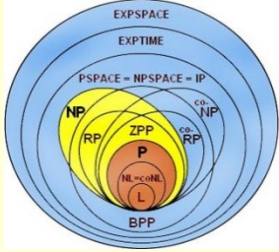
Questions?

Questions?

Questions?

Encryption





"Bob sucks!
From Alice"

Encrypt

xsgf
hasf
gedg

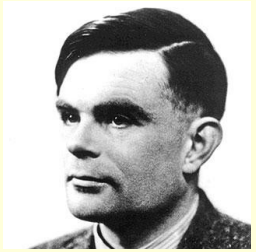
Send

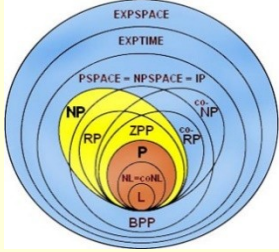
"Bob sucks!
From Alice"

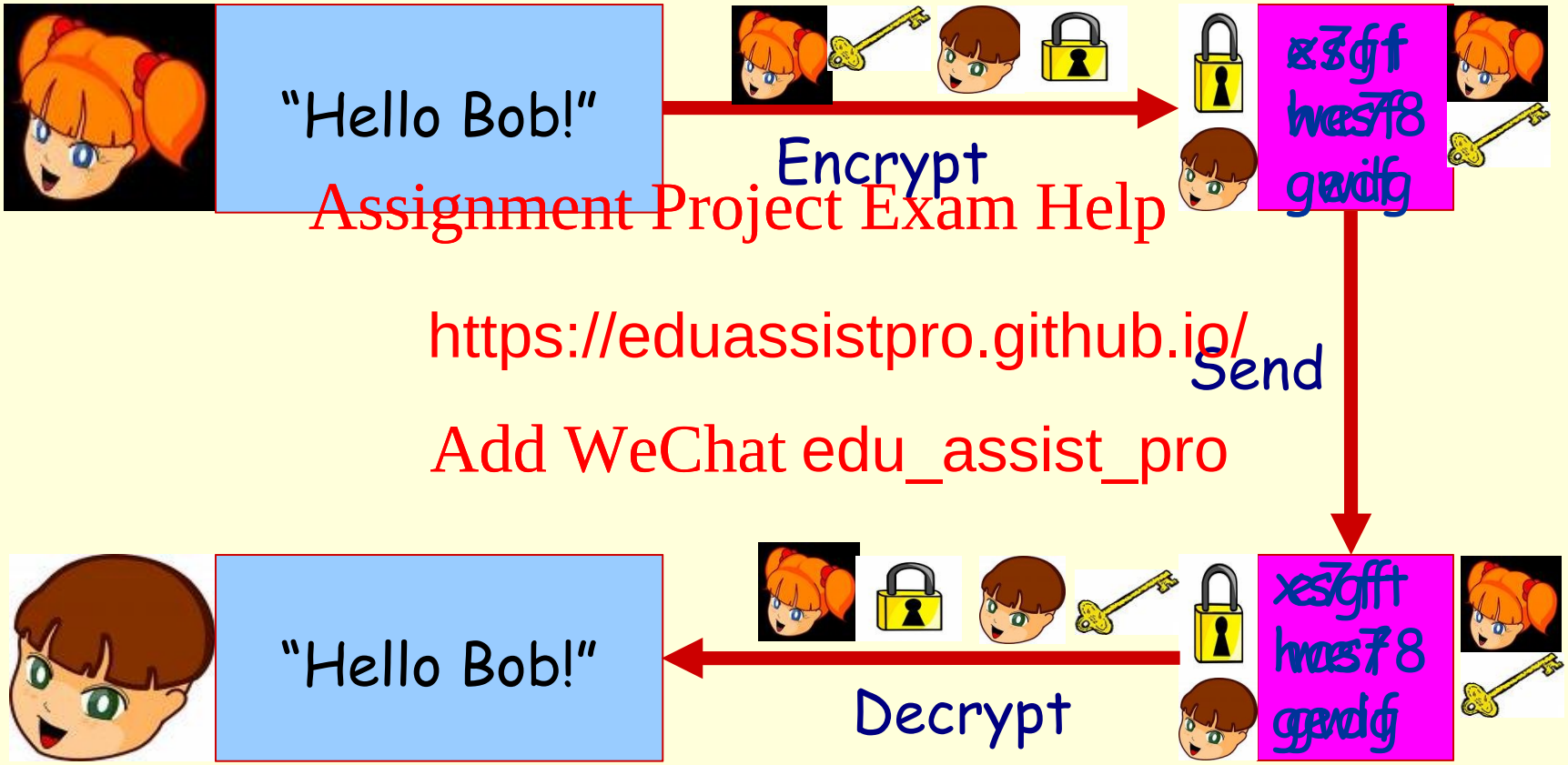
Decrypt

xsgf
hasf
gedg

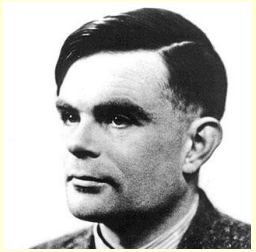
Encryption

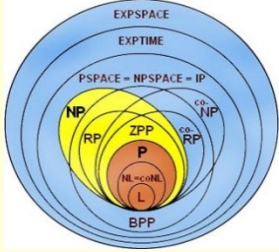






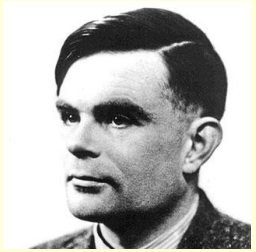
Authentication

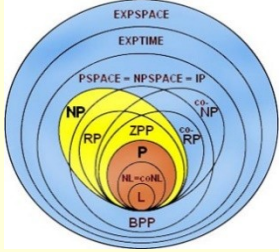




- Alice sends $m' = E_B(D_A(m))$ to Bob
- Bob decodes message with $E_A(D_B(m'))$
- Bob sees $E_A(D_B(E_B(D_A(m)))) = E_A(D_A(m)) = m$
- Spy message a
- Bob decodes with $E_A(D_B(m')) = E_A(m)$
- Alice's messages could only have been sent by Alice (as only Alice knows D_A
- Spy message could have been sent by anyone (including Alice)

Authentication





- This can be done with any public-key scheme for which
  - D(E(m)) = E(D(m))
  - RSA satisfies this (ie order of decryption and encryption
- Can also be use
  - $E_A(E_B(m)) = E_B(E_A(m))$ (an _ decryption)
  - RSA satisfies this too (ie order of encryption and/or decryption doesn't matter)

Questions?

Questions?

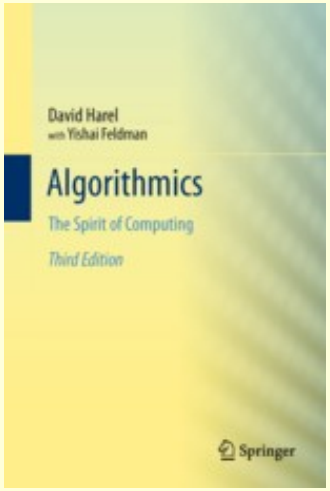
Questions?

Secure Dealings

Who does the dealing?

- No central 'umpire'
- No-one trusts anyone
- Need to distr hidden
- Need to be able to verify distribution
- Online equivalent of 'you cut, I'll shuffle'

Great reference: David Harel & Yishai Feldman, Algorithmics (3rd edition), Springer, 2012. (especially Chapter 12)



Secure Dealings

Alice shuffles

Bob discards 2, chooses 2 others, sends rest to Alice

Shuffles

discards 2, chooses 2

How do we implement 'face down'?

Secure Dealings

1,2,3,4,5,6

Shuffles, encrypts

$E_A(4,1,2,5,3,6)$

tmds 2, encrypts 2

$E_A(4,5)$, $E_B(E$

decrypts all

4,5

$E_B(2,3)$

decrypts

2,3

Secure Dealings

- Deal 2 cards each from a deck of 6 to 2 players A and B
- **All keys kept secret (until after game)**
- 1,2,3,4,5,6 original
- $E_A(4,1,2,5,3,6)$ rypts all cards
- $E_A(4,5)$ $E_B(E_A(2,3))$ B chooses s 2
- 4,5, $E_B(2,3)$ A decrypts all 4
- $E_B(2,3)$ A sends encrypted cards to B
- 2,3 B decrypts

Secure Dealings

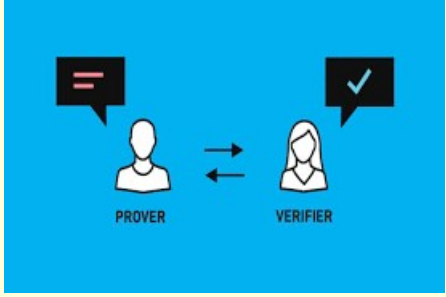
- Bob doesn't know which cards Alice has
- Alice doesn't know which cards Bob has
- Afterwards, t yption keys
- Bob can check
- Alice can check $E_B(E_A(2,3))$
- No-one gets shot :-)

Questions?

Questions?

Questions?

Zero-knowledge Proofs



Question: How can you prove that you know a secret without revealing it?

What is your PIN?
Password:
First pet?

Answer: Proba

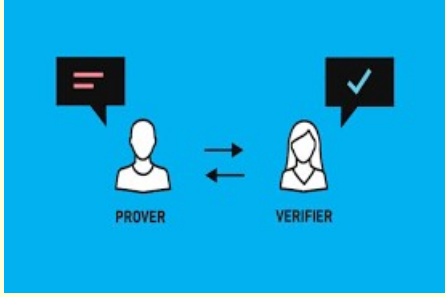
"I know the secret!"

prover

"Convince me!

verifier or doubter

Zero-knowledge Proofs

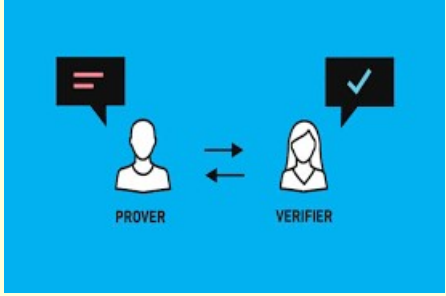


- NP-complete problems are useful here:
- Solution can be **checked** in polynomial time
- (Almost certainly) solution can't be **found** in polynomial tim

We use a **probabilistic** version **g a solution**

Usually **interactive** as well

Zero-knowledge proofs



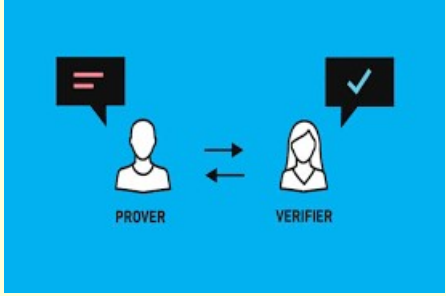
Prover is often known as 'Peggy'

Verifier is often known as 'Victor'

Assumptions:

- Peggy and Victor are both rational
- Peggy only needs to convince Victor
- Victor does not know the secret
- Victor will only allow access if he is convinced
- Peggy and Victor act independently and don't collude
- Peggy reacts to input from Victor
- Process continues until Victor is convinced or claim is disproved
- Peggy's 'proof' **must not allow Victor to know the secret**

Zero-knowledge proofs



" I know the secret!"

"Do this then!"

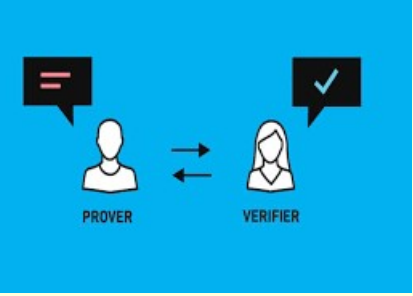
(perfor)

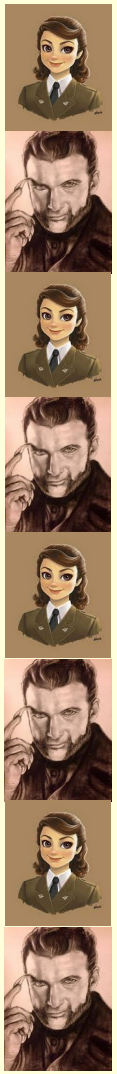
"Okay. Now do this!"

(performs action unsuccessfully)

"GOTCHA!"

Zero-knowledge proofs





"I know the secret!

"Do this then!"

(performs action successfully)

"Okay.

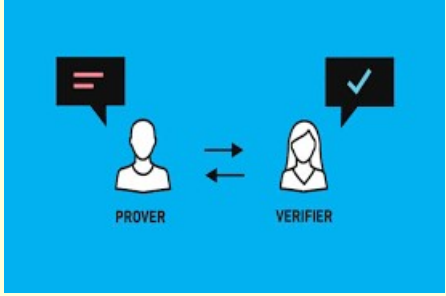
(performs action successfully)

"Okay. Now do this!"

(performs action successfully)

"Ok! I am convinced."

Ali Baba example



"?"

Exactly one of Door A and Door B open at any time

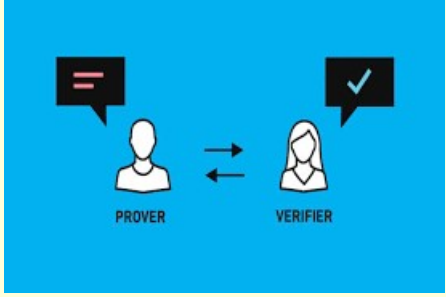
Action:

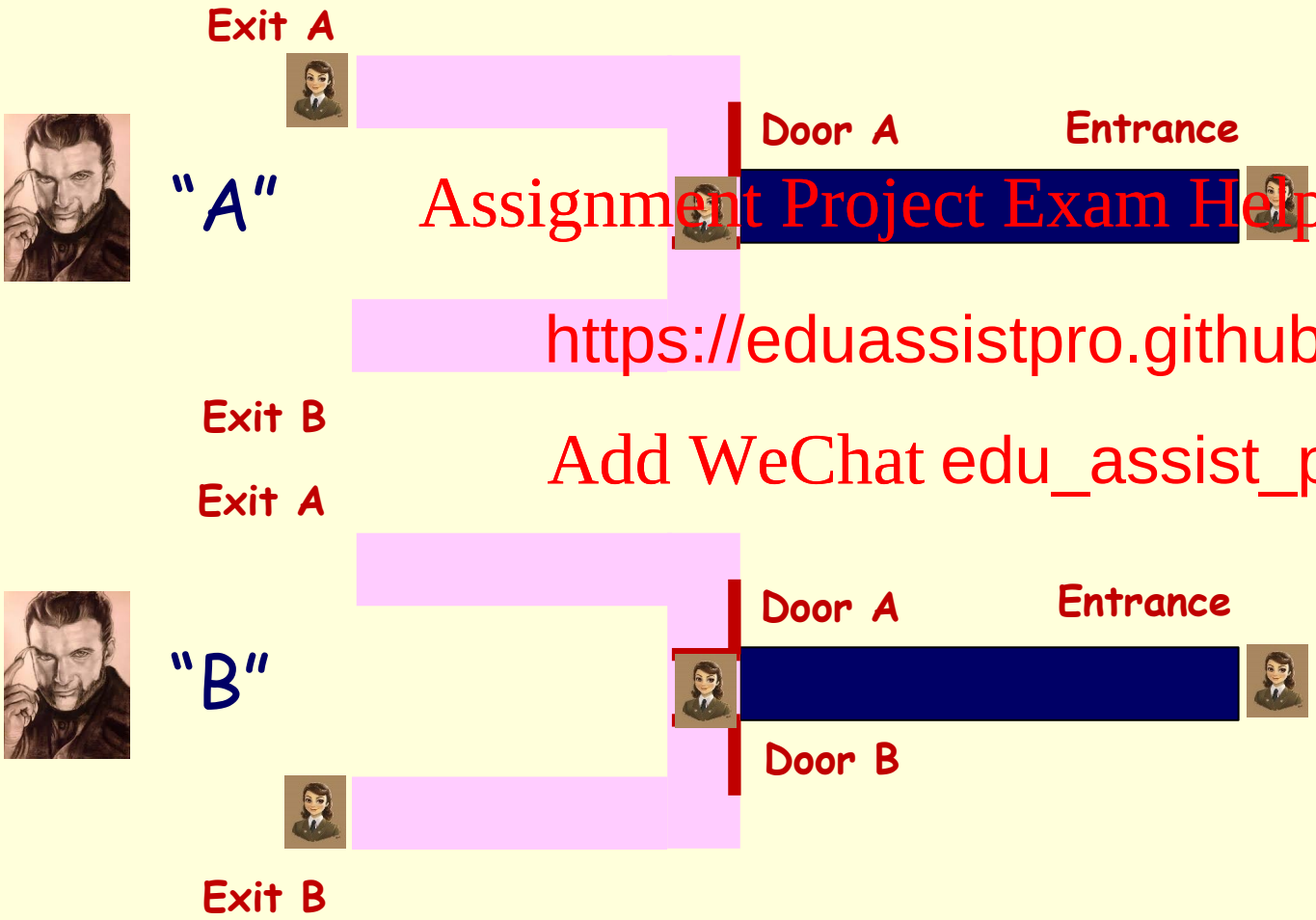
- Peggy goes into the tunnel
- Victor randomly chooses an exit
- Peggy then comes out Exit A or B
- Successful if **Peggy is at the exit Victor chose**

How to open either door
(or how to 'flip' the doors' status)

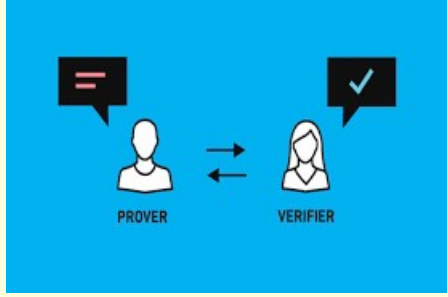
From: Jean-Jacques Quisquater, Louis Guillou & Thomas Berson, "How to Explain Zero-Knowledge Protocols to Your Children", Advances in Cryptology - CRYPTO'09, LNCS 435 628-631, 1990.

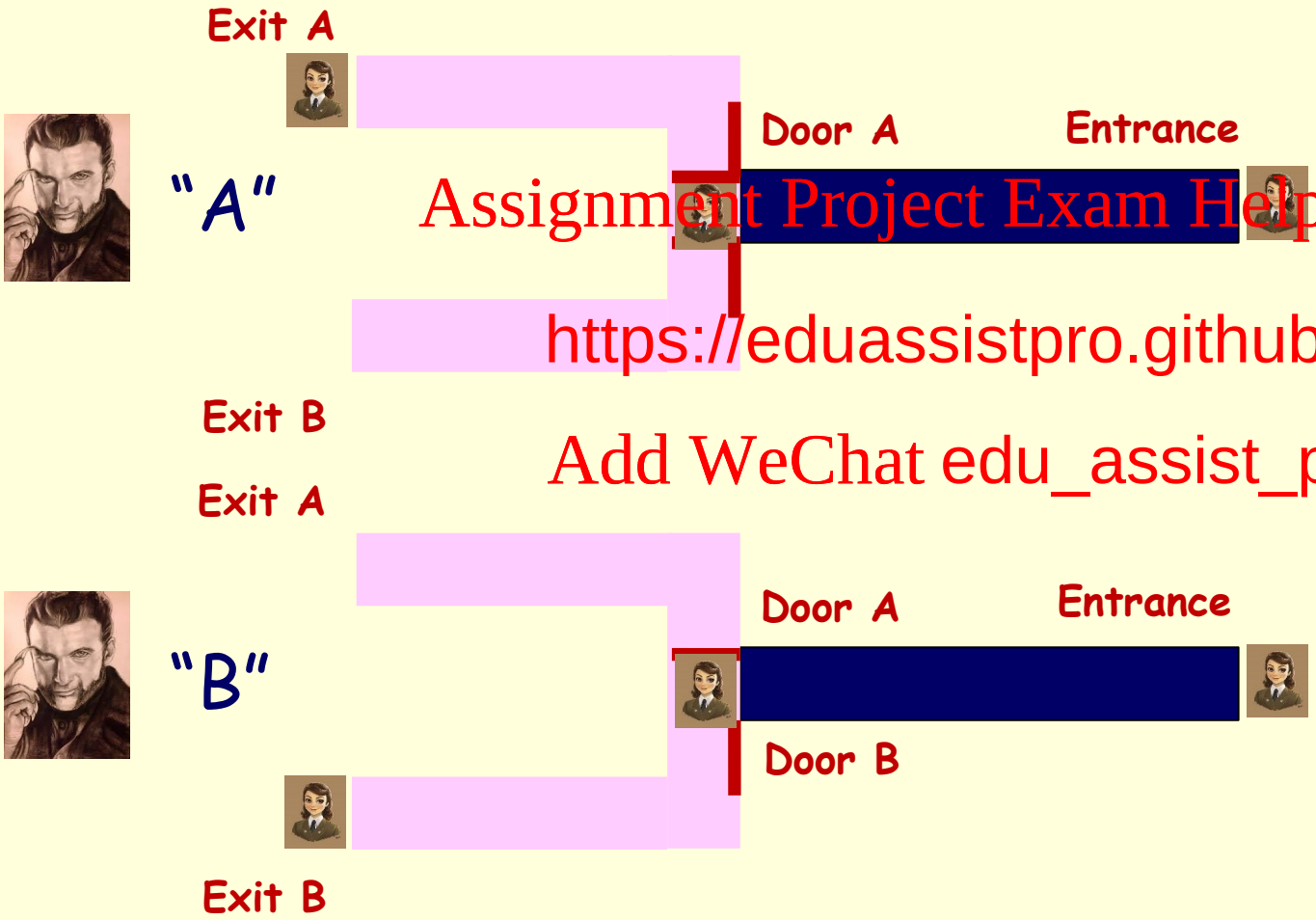
Ali Baba example



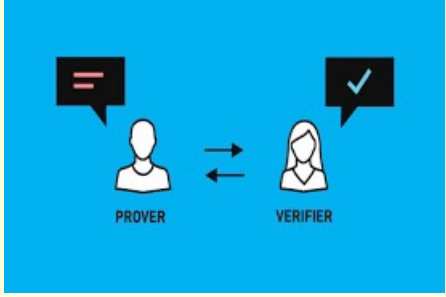


Ali Baba example

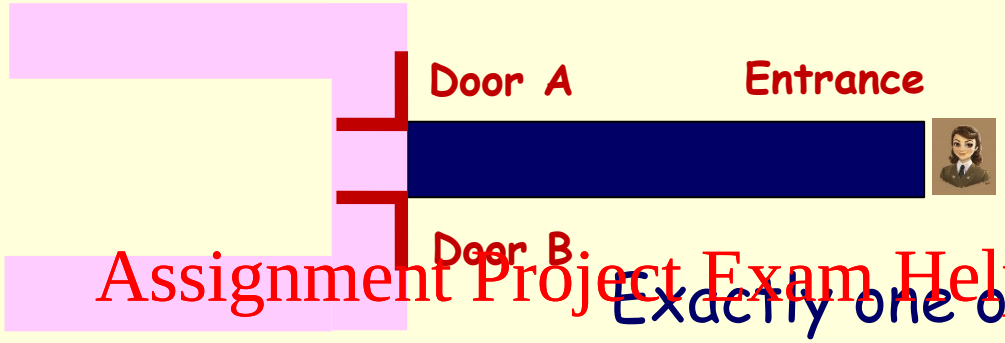




Ali Baba example



"?"



Exactly one of Door A and Door B open at any time

For n trials:

Probability of passing 1 test falsely is

Probability of passing 2 tests falsely is $(1/2)^2$

Probability of passing 3 tests falsely is $(1/2)^3$

...

Probability of passing n tests falsely is $(1/2)^n$

When n = 20, this is 0.0000009536 (!!) (99.9999046% correct)

Questions?

Questions?

Questions?

Quiz time!

Go to Canvas and find the quiz Lectorial 11 Question set

- Not worth any marks
- You can consult other students if you wish
- Time limit will b

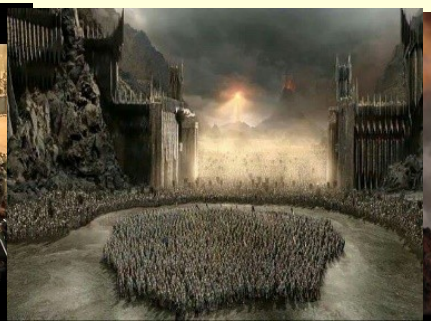
Are you ready?

Are you sure?

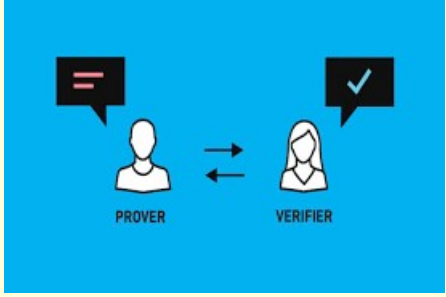
Go!

The pictures will take 5 minutes to disappear!

Thomas music means 1 minute left!



3-colouring example



Secret: 3-colouring of a given graph G

Action:

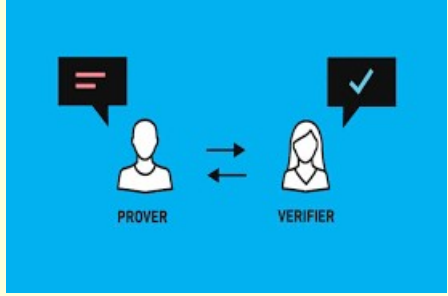
- Peggy shows Victor the graph G
- Victor randomly c
- Peggy shows Victo nodes
- Successful if **the** **ifferent**

(shows graph G)

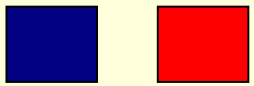
"Colours for these two nodes?" (chooses two adjacent nodes)

From: David Harel with Yishai Feldman, "Algorithmics: The Spirit of Computing" (3rd edition), Springer, 2012.

3-colouring example

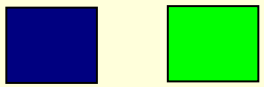


"What are the colours of these two nodes?"

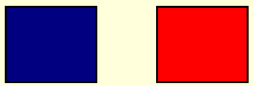


"What are the colours of these two nodes?"

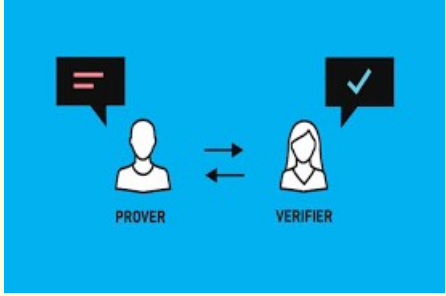
"What are the colours of thes "



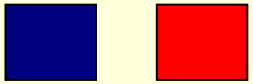
"What are the colours of these two nodes?"



3-colouring example



"What are the colours of these two nodes?"



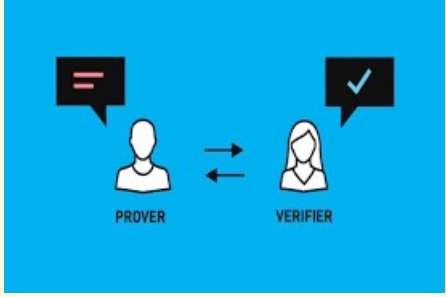
"What are the colours of these two nodes?"

"What are the colours of thes

"Gotcha!"

"Oops!"

3-colouring example



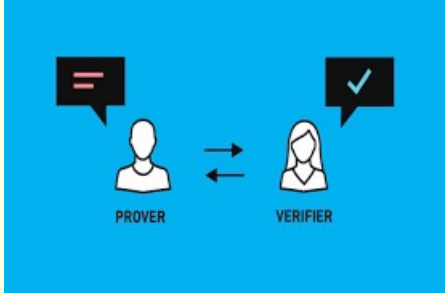
- Victor randomly chooses an edge from the graph
- Peggy only shows colouring for these two nodes

Process continues until either the claim is disproved or the Victor gives up

If the graph has n edges:

Probability of passing 1 test fal)/n
Probability of passing 2 tests falsely is $[(n-1)/n]^2$
Probability of passing 3 tests falsely is $[(n-1)/n]^3$
...
Probability of passing k tests falsely is $[(n-1)/n]^k$

3-colouring example



N = 100

Trials	Probability of error
10	90.44%
20	81.79
30	73.97
40	66.90%
50	60.50%
60	54.72%
70	49.48%
80	44.75%
90	40.47%
100	36.60%

N = 200

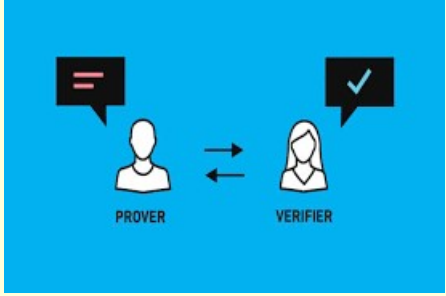
Trials	Probability of error
10	95.11%
20	90.46%
30	86.04%
4	81.83%
5	77.83%
60	74.03%
70	70.41%
80	66.96%
90	63.69%
100	60.58%

Questions?

Questions?

Questions?

3-colouring example



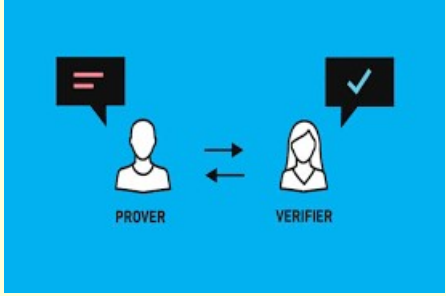
Issues

- Not very fast to an arbitrary probability
- How does Victor know Peggy is not just choosing randomly?

Hamiltonian Cyc

- Similar use of 'guess and ch ty of NP problems
- Requires cryptographic commitment from Peggy to the graph
- Victor can then check the cycle is really from the graph

Hamiltonian cycle version

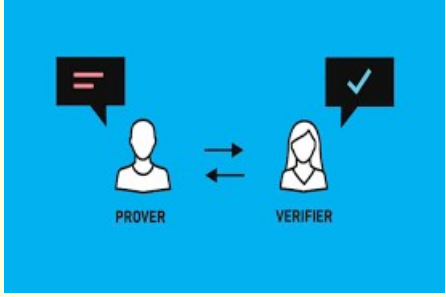


Secret: A Hamiltonian cycle of a given graph G

Action:

- Peggy creates a random copy (isomorph each time) of G
- Peggy commits to
- Victor randomly c
  1. The isomorph
  2. The Hamiltonian cycle in H
- Peggy then shows Victor as below
  1. The mapping between H and G
  2. The Hamiltonian cycle in H
- Victor then verifies
  1. H and G are isomorphic
  2. The cycle is really made from edges in H
- Successful if **Victor is satisfied with the verification**

Hamiltonian cycle version



- Victor randomly chooses between the two options
- Peggy can easily generate isomorphic copies of G
- Peggy can easily generate the cycle in H from the cycle in G
- Victor never finds out both H and the Hamiltonian cycle in H
- Victor needs both to work out the Hamiltonian cycle in G
- The randomness o t Peggy can't cheat by predicting request swers

Similar performance to Ali Bab

- Probability of passing 1 test falsel
- Probability of passing 2 tests falsely is $(1/2)^2$
- Probability of passing 3 tests falsely is $(1/2)^3$
- …
- Probability of passing n tests falsely is $(1/2)^n$

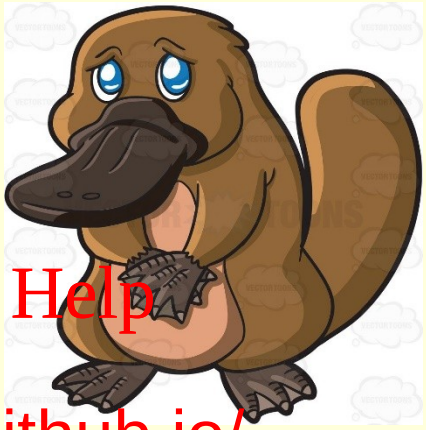
When $n = 20$, this is 0.0000009536 (!!) (99.9999046% correct)

Questions?

Questions?

Questions?

The Platypus Game



https://www.youtube.com/watch?reload=9&v=0gM5TjSOQ48
https://zoneringtones.com

The Platypus Game

Assignment 2

Adding 10 machines

- Are they legal?
- Are they different from the existing 2500?

extra.pl:


```
extra(1, [t(y,k,g,k,gg),t(g,k,g,w,g                    gg),t(g,w,g,e,wa),t(y,p,y,k,wa)]).
extra(2, [t(y,k,y,k,gg),t(g,k,y,e,gg),t(y,e,y,p,wa),t(g,e,y,p                t(g,w,y,w,gg),t(y,p,g,e,wa)]).
extra(3, [t(y,k,y,k,wa),t(g,k,y,p,gg),t(y,e,g,p,wa),t(g,e,g,                 t(g,w,g,e,gg),t(y,p,y,w,gg)]).
extra(4, [t(y,k,g,e,wa),t(g,k,g,k,gg),t(y,e,y,k,gg),t(g,e,g,p                t(g,w,y,p,wa),t(y,p,g,e,wa)]).
extra(5, [t(y,k,g,k,wa),t(g,k,y,k,wa),t(y,e,y,p,wa),t(g,e,y,                 ,t(g,w,y,k,gg),t(y,p,y,e,wa)]).
extra(6, [t(y,k,y,e,wa),t(g,k,g,p,gg),t(y,e,g,w,wa),t(g,e,y,k,gg),t(y,w,y,w,gg),t(g,w,y,p,wa),t(y,p,g,e,gg)]).
extra(7, [t(y,k,g,e,gg),t(g,k,g,w,gg),t(y,e,y,w,wa),t(g,e,g,e,wa),t(y,w,g,e,gg),t(g,w,y,w,gg),t(y,p,y,p,gg)]).
extra(8, [t(y,k,y,k,wa),t(g,k,y,w,gg),t(y,e,g,w,wa),t(g,e,y,k,wa),t(y,w,g,w,gg),t(g,w,y,k,gg),t(y,p,y,e,gg)]).
extra(9, [t(y,k,g,w,wa),t(g,k,g,e,wa),t(y,e,g,w,wa),t(g,e,g,w,wa),t(y,w,g,k,wa),t(g,w,g,w,gg),t(y,p,g,e,gg)]).
extra(10, [t(y,k,g,w,gg),t(g,k,y,w,wa),t(y,e,g,p,gg),t(g,e,y,w,gg),t(y,w,y,p,wa),t(g,w,g,k,wa),t(y,p,y,e,gg)]).
```

The Platypus Game

consult platypus7.pl

consult 2500 machines

consult new machines

run test

Questions?

Questions?

Questions?

That's it!

I am out of here!

Break time! (We resume when all the pictures are gone! This will take 3 minutes!)