Introduction to Cryptography and Security Mechanisms:

Unit 12

Entity authentication

Learning Outcomes

- Discuss a number of different mechanisms for randomly generating values suitable for use in cryptography
- Compare different techniques of providing freshness
- Recognise a number of different mechanisms for providing entity authentication
- Explain the principle behind dynamic password schemes

Sections

1. Random number generation
2. Providing freshness
3. Entity authentication fundamentals
4. Entity authentication mechanisms

1. Random number generation

Need for randomness

- Cryptographic keys
- Variable parameters (salts, IVs etc)
- Probabilistic encryption (e.g. El Gamal)
- Nonces
- etc etc

Non-deterministic v deterministic generators

Non deterministic

- Truly random
- Randomness comes from physical source
- Input conditions very hard to replicate
- Security depends on protection of source

Deterministic

- Pseudorandom
- Randomness comes from a seed value cryptographic key
- Same input always results in same output
- Security depends on protection of key

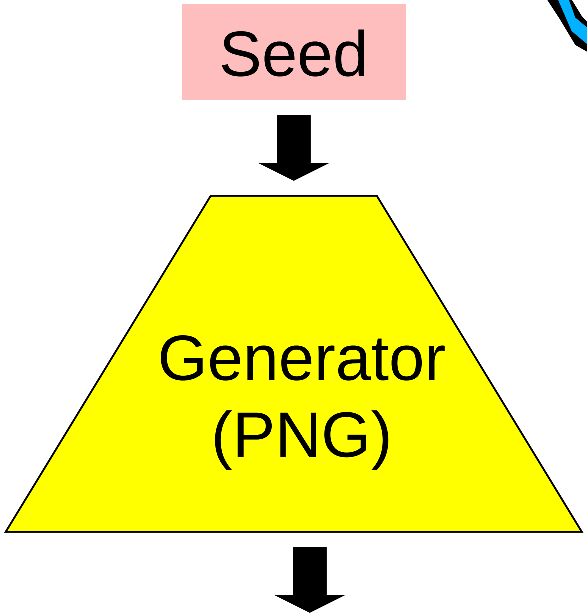
Truly random sources

- Non-deterministic generators
- Normally hardware-based
  - Radioactive decay
  - Semiconductor thermal noise
  - Free-running oscillators
  - White noise
- Expensive, potentially awkward processes
- Slow continuous output

Software-based random sources

- Non-deterministic generators
- Software based generators use physical phenomena that can be easily captured by a computer
  - System clock
  - Hard drive seek times
  - Keystroke timing
  - Time between interrupts
  - Network statistics
- Cheaper and easier than hardware sources
- Less secure but need to ensure attacker cannot access source

Deterministic generators



1. Have we seen this diagram before?
2. What happens if the seed is compromised?
3. How do you generate the seed?

Pseudorandom output 11001011010100100011101001….

Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin… John von Neumann

Designing deterministic generators

How can you tell if a deterministic generator is a good simulation of a truly random generator?

2. Providing freshness

Freshness mechanisms

- **Time-based**
  - A process that generates some data that identifies the “time” that the data was created using either:
    - conventional clock-based time
    - some notion of logical time (**sequence numbers**).
- **Nonce-based**
  - A **nonce** (“number used once”) is a number that is introduced into a cryptographic protocol as a one-time identification of that protocol message or protocol run.
  - Normally a (pseudo)randomly generated number

Clock-based mechanisms

- Alice includes the time when she sends the message to Bob.
- Bob checks the time on his clock on receipt.
- If they “match” then Bob accepts the message as fresh.

1. Identify potential problems with the simple clock-based freshness mechanism described above.
2. How might you address these problems?

Logical time stamps

Alice

1. Alice looks up her database to find the latest value N of the sequence number N_{AB}
2. Alice sends her message to Bob along with N
3. Alice increments her value of N_{AB} and stores the new value on her database

Bob

4. Bob compares the sequence number N sent by Alice with the latest value of the sequence number N_{AB} on his database
5. If $N > N_{AB}$ then Bob accepts the latest message as fresh and he sets his stored value of N_{AB} to N
6. If $N \leq N_{AB}$ then Bob rejects the latest message

What happens if the N that Bob receives is much larger than N_{AB} ?

Nonces

Suppose:

1. Alice generates a nonce using a random number generator and then sends it to Bob.
2. Bob sends it straight back.

1. Precisely what can Alice deduce about the message that she receives back from Bob?
2. If this nonce appears in a later protocol message, can Bob be sure that the messages is fresh?

Properties of freshness mechanisms

	Clock-based	Sequence numbers	Nonce-based
Synchronisation needed?	Yes	Yes	No
Communication delays?	Window needed	Window needed	Window needed
Integrity required?	Yes	Yes	No
Minimum passes needed	1	1	2
Special requirements	Clock	Sequence database	Random generator

3. Entity authentication fundamentals

Entity authentication

Since entity authentication provides identity assurance in “real time”, it can only truly be achieved for an instant in time!

1. In what ways could the fact that entity authentication is an instantaneous process be exploited by an attacker?
2. How might we try to counter these attacks?

Typical uses of entity authentication

Entity authentication is either provided **unilaterally** or **mutually** as part of:

- Access control
- A cryptographic protocol

Basis for entity authentication

The most common basis for providing entity authentication is by using one (**one-factor**) or a combination of (typically **two-factor**) the following:

- something that you **have**
- something that you **are**
- something that you **know**

Can you think of any other basis for providing entity authentication?

Something you have

- **Dumb tokens**
  - physical device with no memory (e.g magnetic striped card)

- **Smart tokens**
  - Can take form of a smart card, “dongle” or resemble a calculator
  - can store some data securely and conduct (limited) cryptographic processes
  - require an interface to a computer system
  - often implemented as part of a two-factor process

Something you are

Biometrics are techniques for human user entity authentication that are based on physical characteristics of the human body.

- Typically convert a physical characteristic into a digital template that is stored on a database.
- Physical characteristic is measured by a reader, digitally encoded, and then compared with the template.
- **Static** (unchanging) measurements include: **fingerprints**, **hand geometry**, **face recognition**, **retina scans** and **iris scans**.
- **Dynamic** (changing) measurements include **handwriting measurements**, **typing patterns** and **voice recognition**.
- There are many implementation issues

Something you know

Many of the previous techniques are implemented in two-factor combination with a technique based on something you know.

1. From a security perspective, what advantages does basing an entity authentication technique on something you know have over the techniques already covered?
2. But… what often happens?

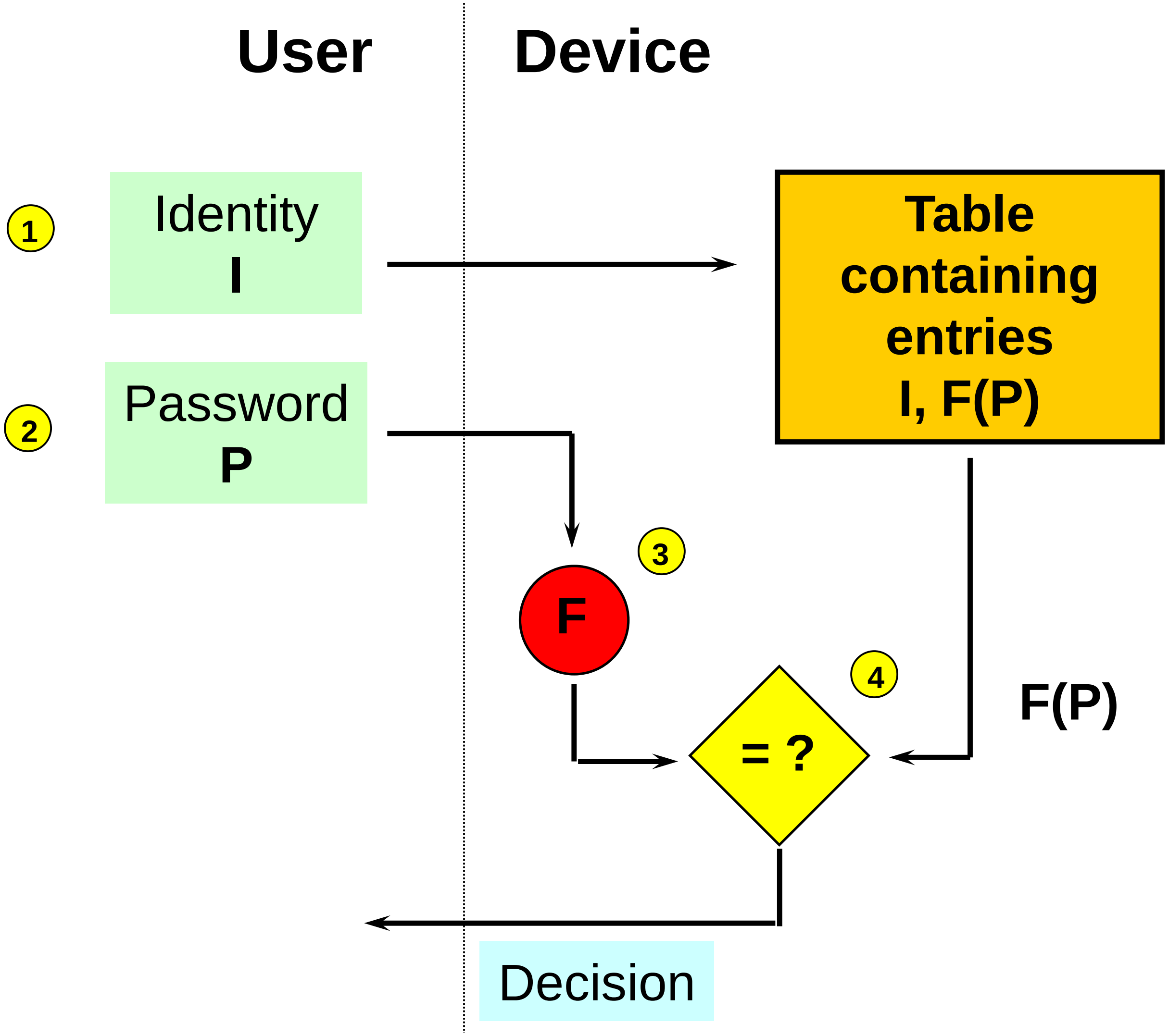
4. Entity authentication mechanisms

Passwords

Passwords are probably the most popular basis for providing entity authentication.

What are the main problems with a password as a basis for entity authentication?

Cryptographic password protection



User

Device

1 Identity I

2 Password P

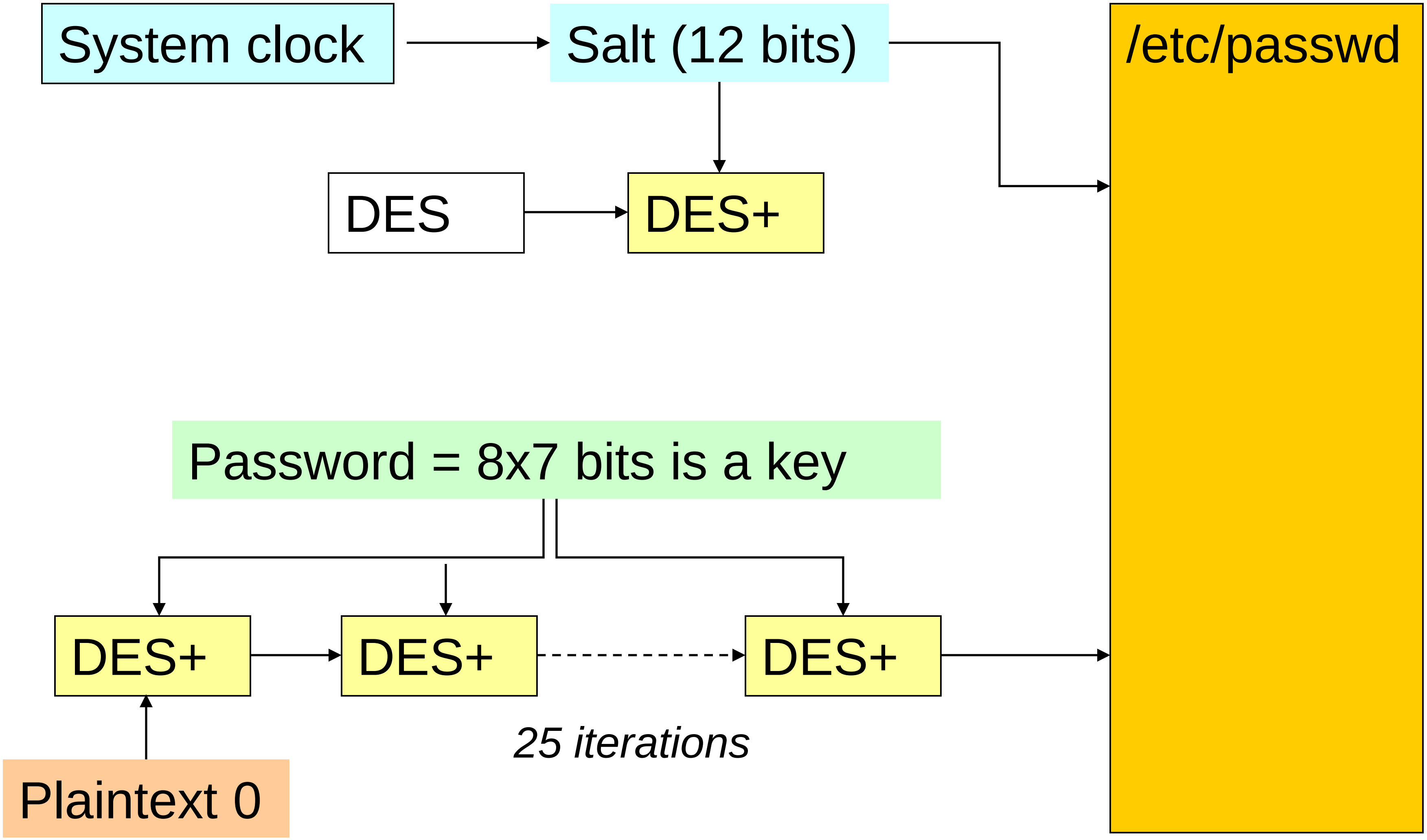
Table containing entries I, F(P)

3

4

Decision

UNIX password protection

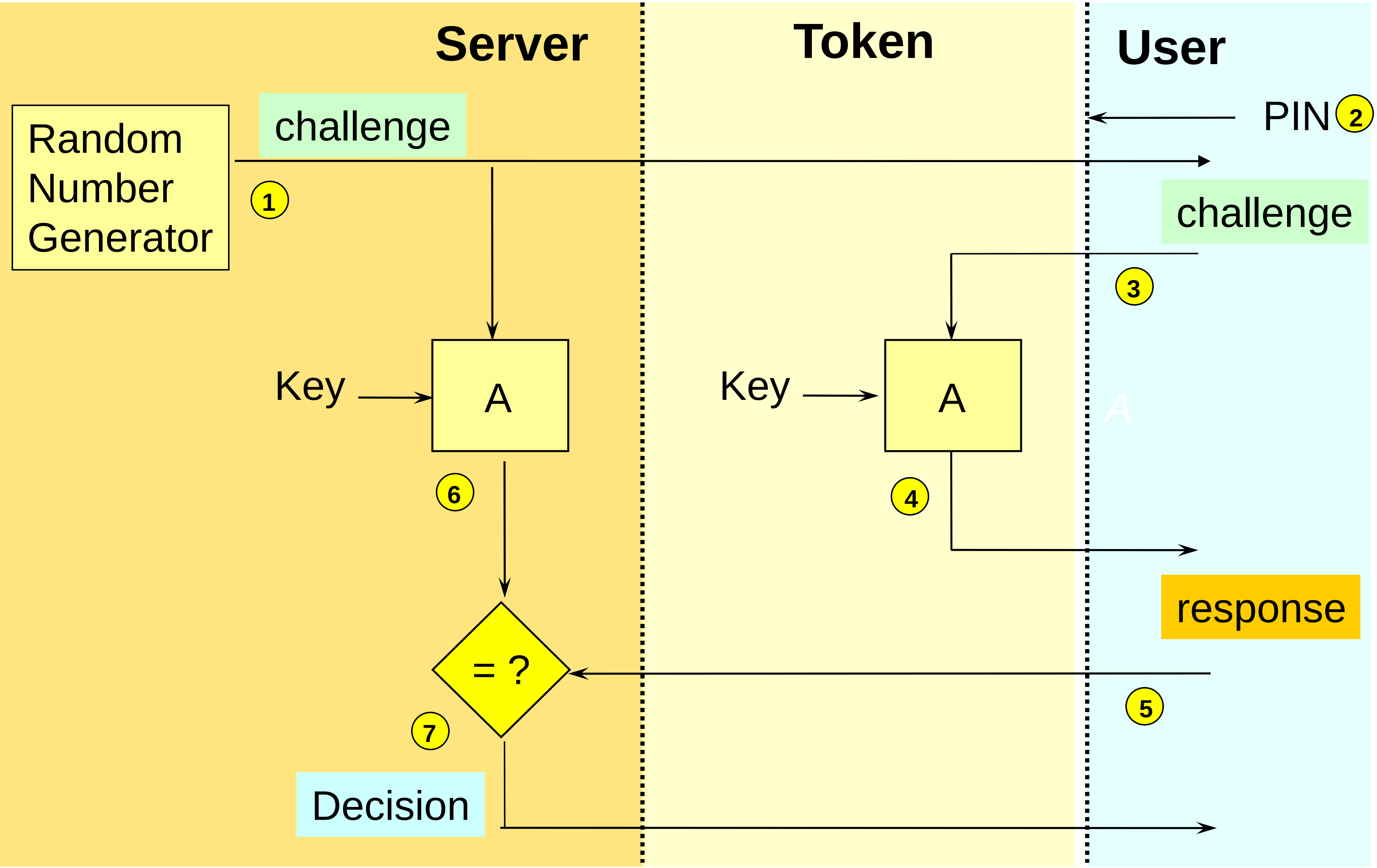


Improving use of passwords

Even when they are “protected”, passwords offer limited security.

What techniques could we employ to make password based entity authentication more secure?

Dynamic password schemes



Server

Token

User

PIN 2

Random Number Generator

challenge

1

challenge

3

Key

A

Key

A

A

6

4

response

= ?

5

7

Decision

Dynamic password questions

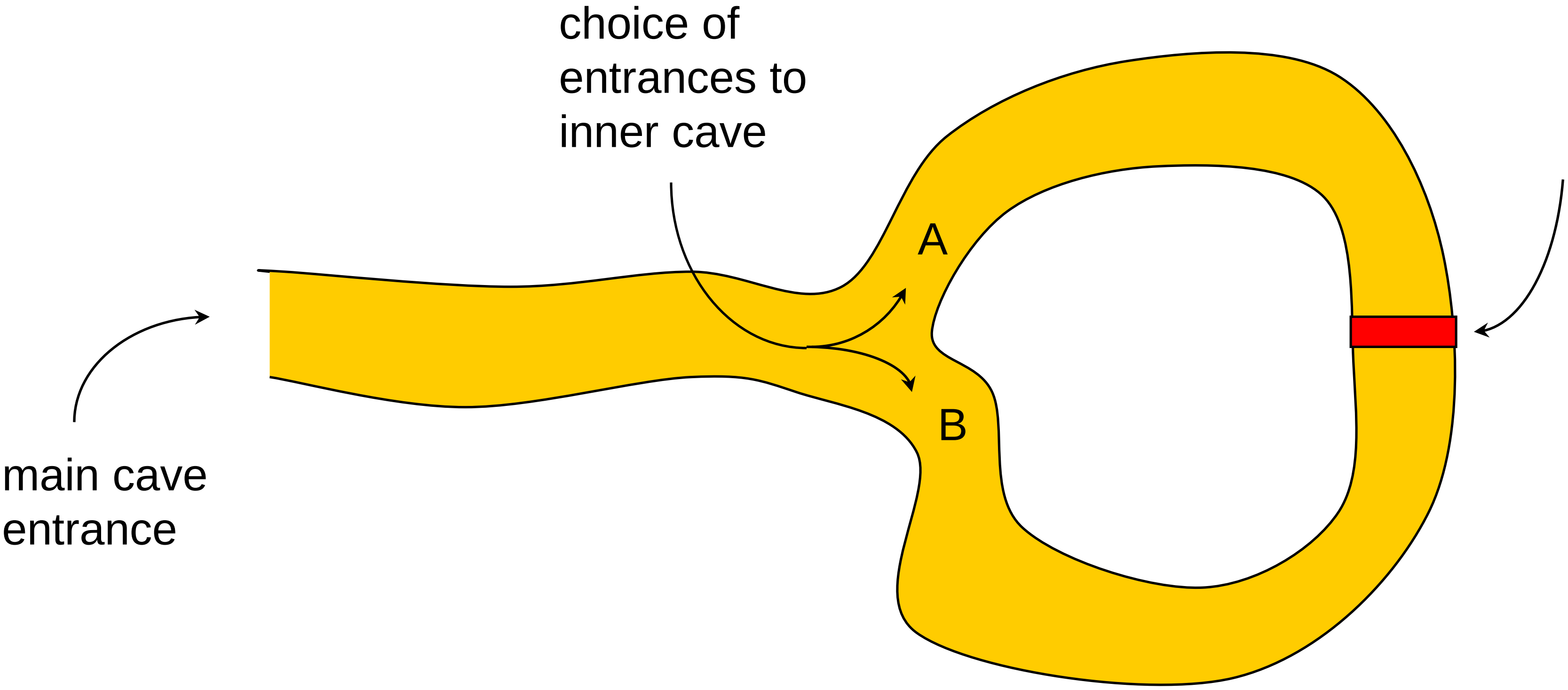
1. From the perspective of the server, what exactly is the basis for assurance that the user is who they claim to be?

2. To what extent is this better than a scheme based on “normal” passwords?

Zero knowledge mechanisms

- Our previous entity authentication techniques are all based on a degree of trust between entities.
- There are situations where Alice and Bob are potential adversaries and do not share any information.
- A requirement of a **zero knowledge mechanism** is that Alice can provide assurance of her identity to Bob:
  - without Alice and Bob sharing a key
  - in a way that makes it is impossible for Bob to later impersonate Alice, even after Bob has observed and verified many different successful authentication attempts.

Zero knowledge mechanisms



choice of entrances to inner cave

main cave entrance

We’re not finished yet!

AKE protocols are coming in Unit 9

Summary

- The source of any randomness in any cryptographic process is important and has often proved to be the weak link in a cryptographic system

- An important property of many cryptographic protocols is freshness, which can be achieved using a variety of different techniques

- There are wide variety of different mechanisms for establishing entity authentication (more in Unit 9)

- Cryptography can be used both to protect passwords and to implement stronger entity authentication mechanisms than passwords