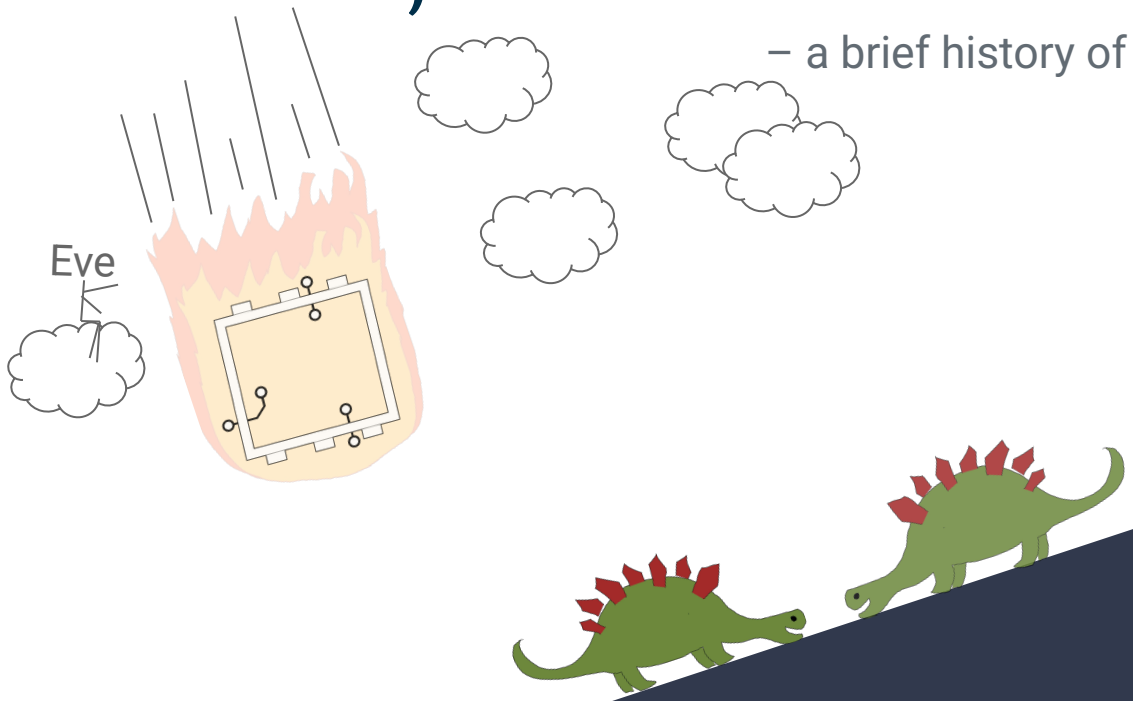
# Sharks, dinos and mammals

– a brief history of animals and cryptography

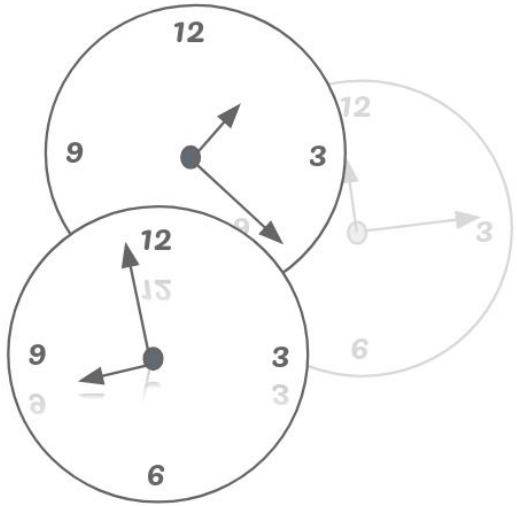


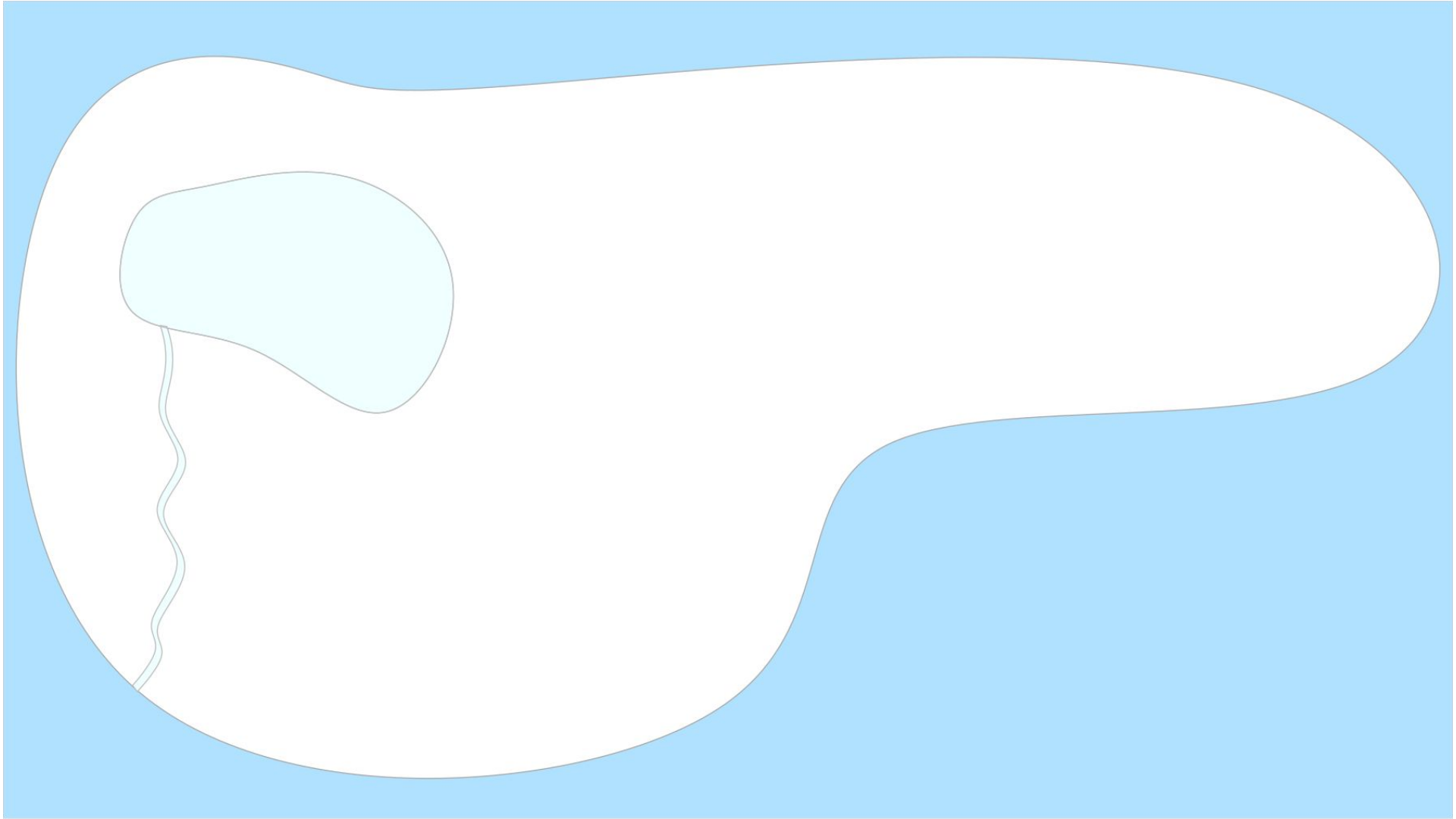

KASTEL Kolloquium 2025
– Marcel Tiepelt

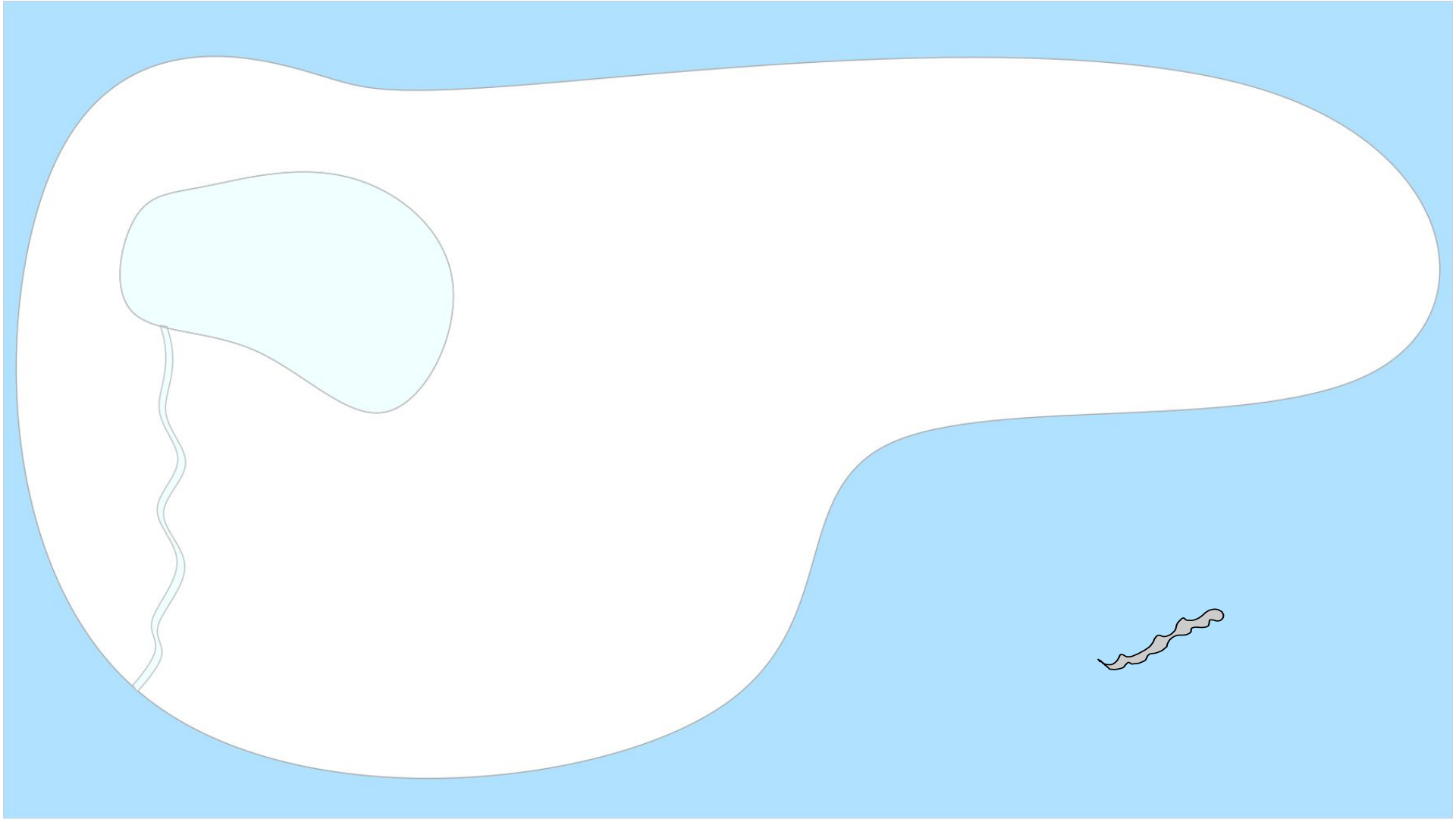
# Goals of this talks

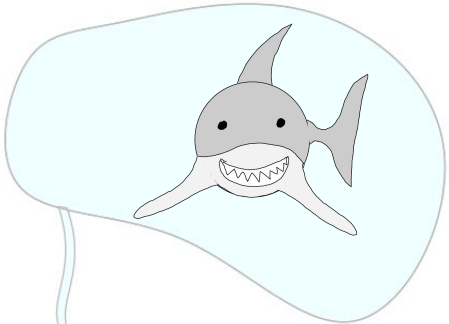
1. Oblivious learning about cryptography

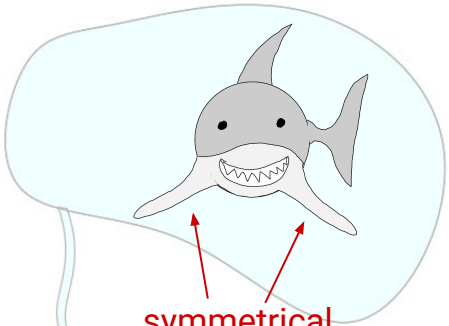
2. Explain the above with **bold** claims


12
9
3
6



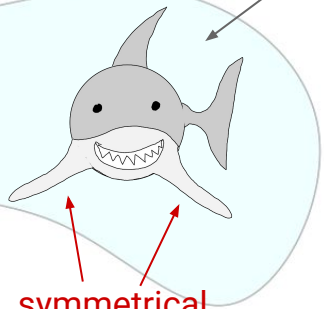




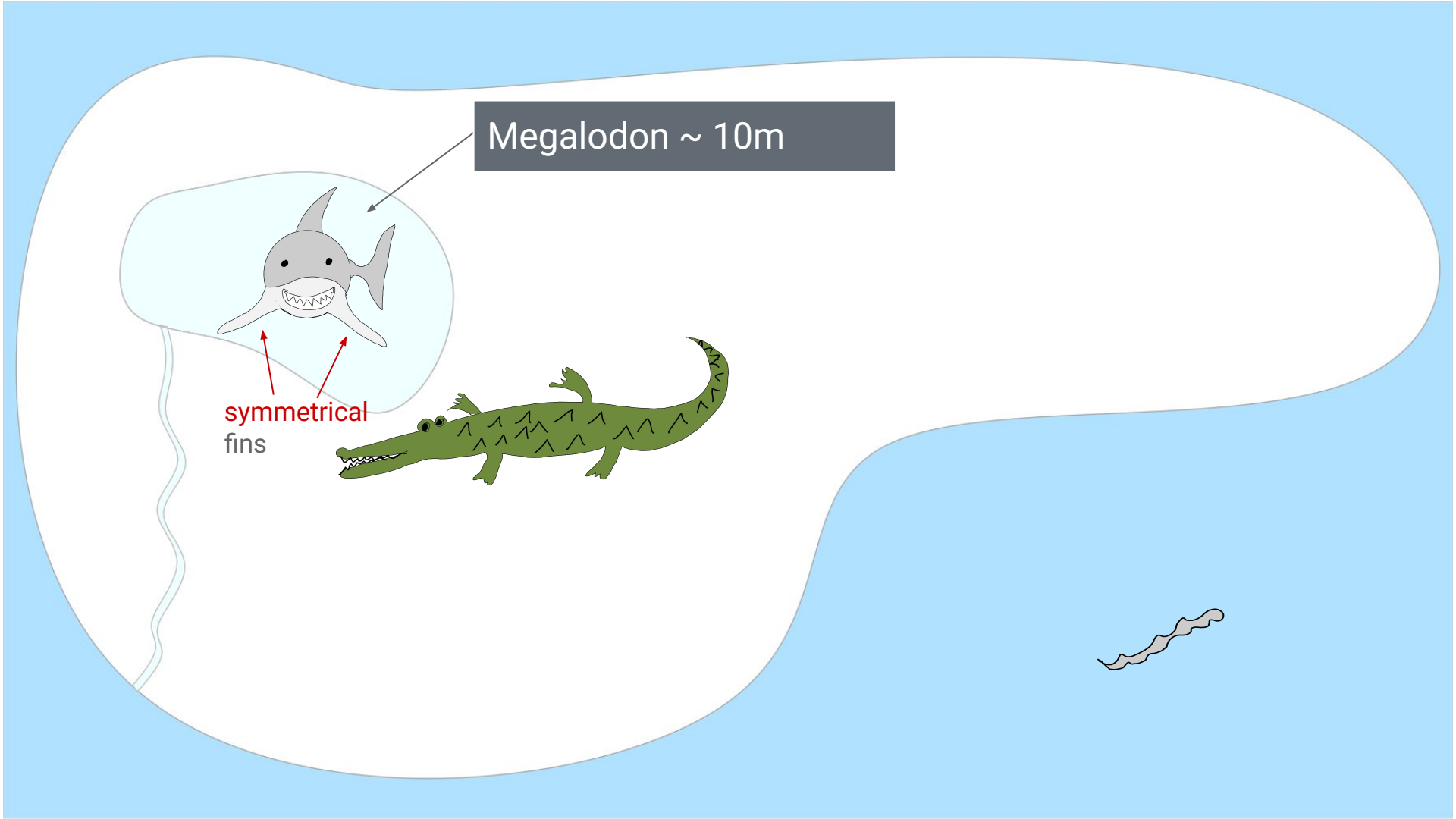


symmetrical
fins

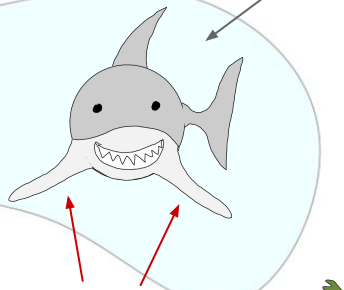
Megalodon ~ 10m



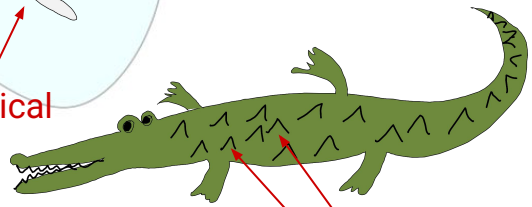
symmetrical
fins


Megalodon ~ 10m
symmetrical
fins

Megalodon ~ 10m



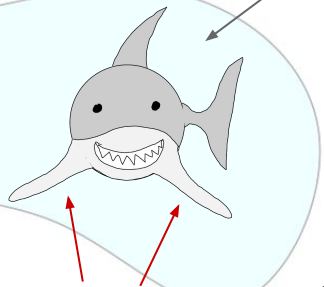
symmetrical
fins



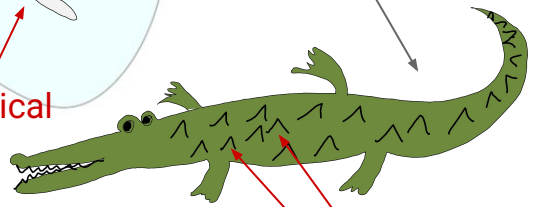
asymmetrical
scales

Megalodon ~ 10m

Sarcosuchus Imperator
~ 12m



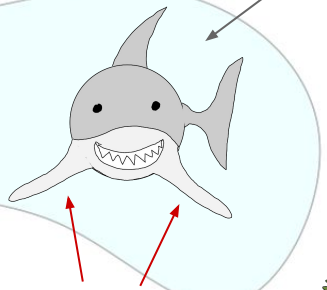
symmetrical
fins



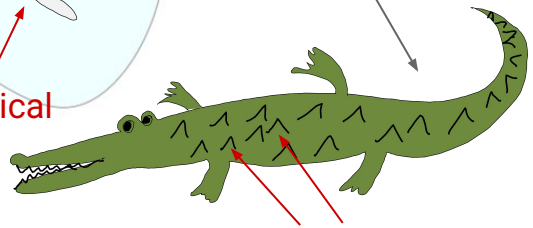
asymmetrical
scales

Megalodon ~ 10m

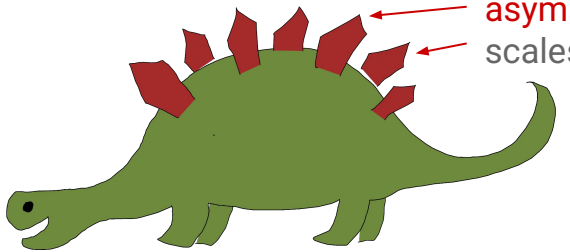
Sarcosuchus Imperator
~ 12m

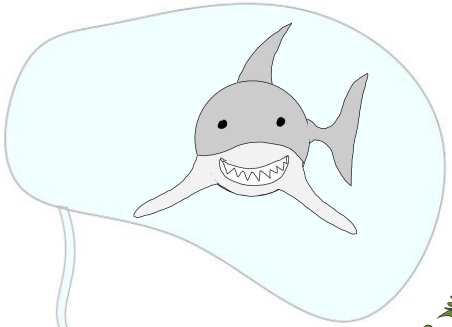


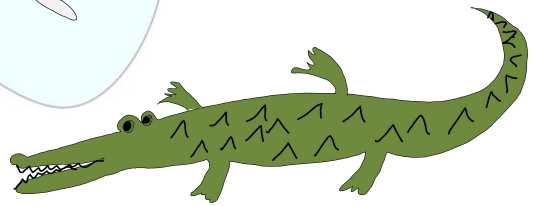
symmetrical
fins

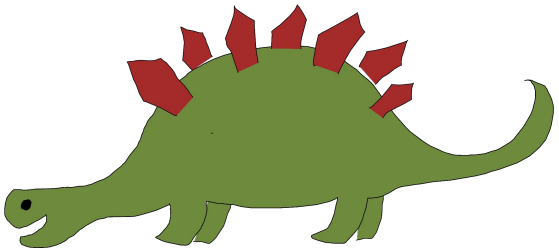


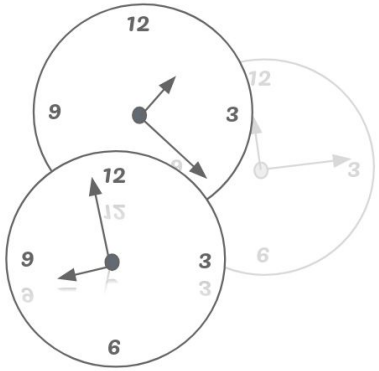
asymmetrical
scales

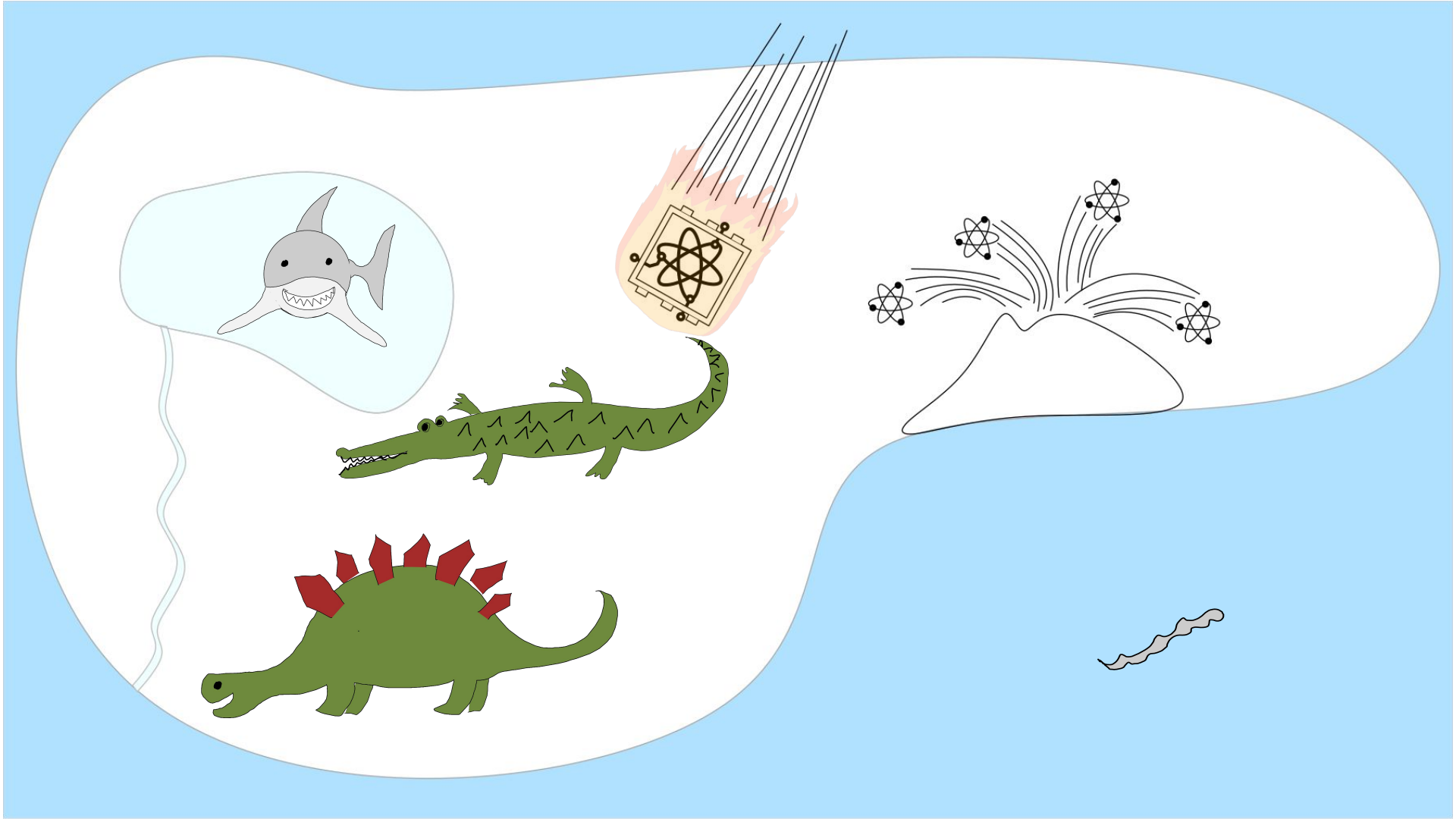


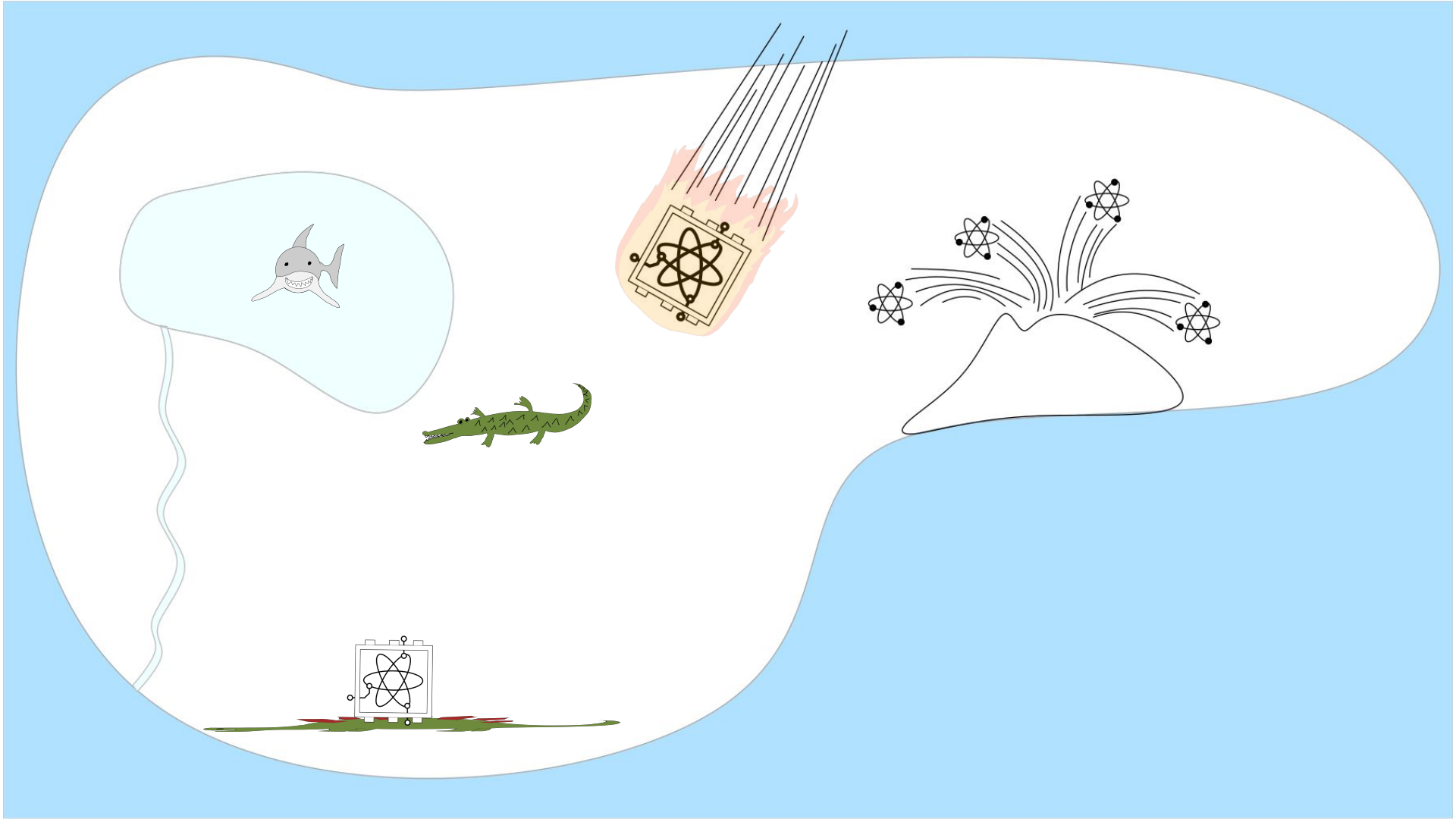




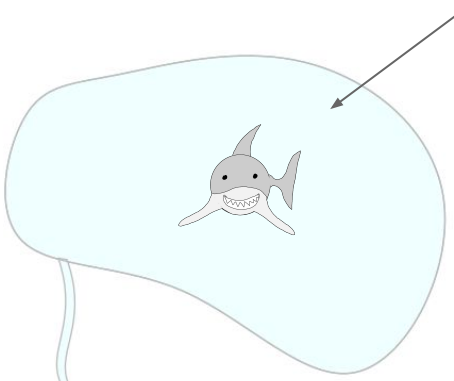


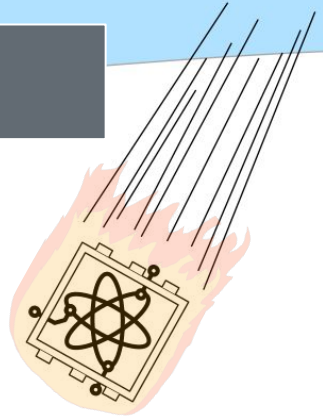

12
9
3
12
9
3
6

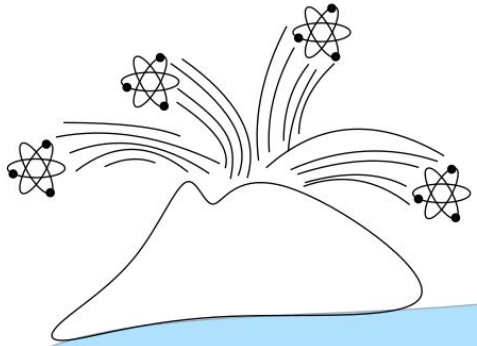


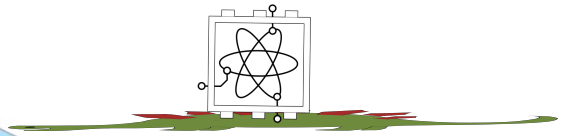


Great White Shark,
~~10m~~ ~6m

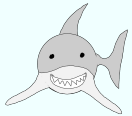


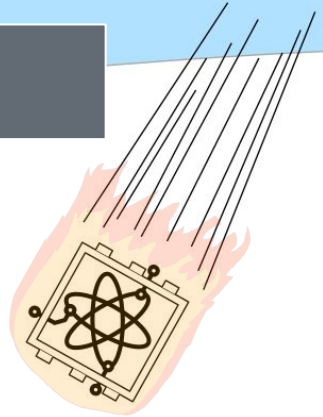


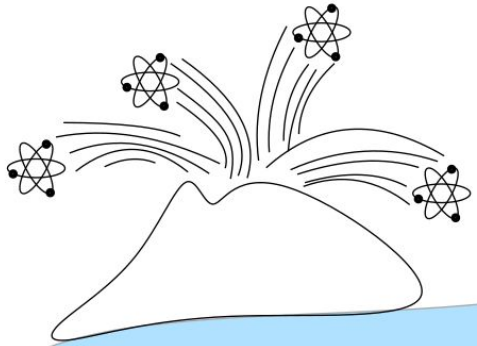




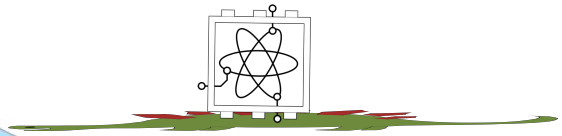
Great White Shark,
~~10m~~ ~6m



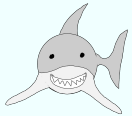


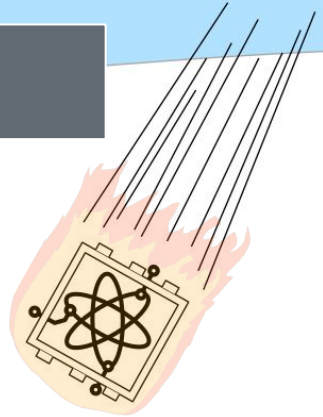


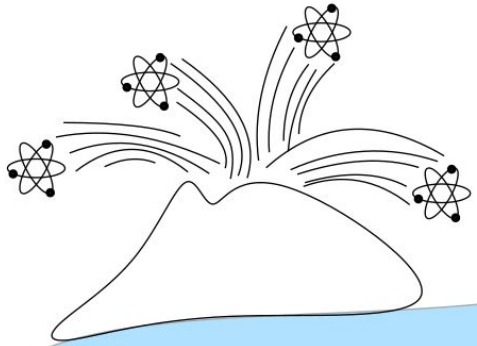
Saltwater Crocodile,
~~12m~~ ~6m



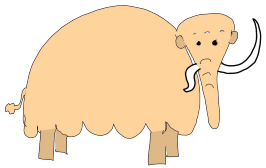
Great White Shark,
~~10m~~ ~6m

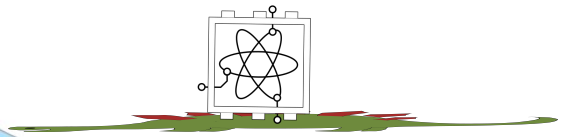




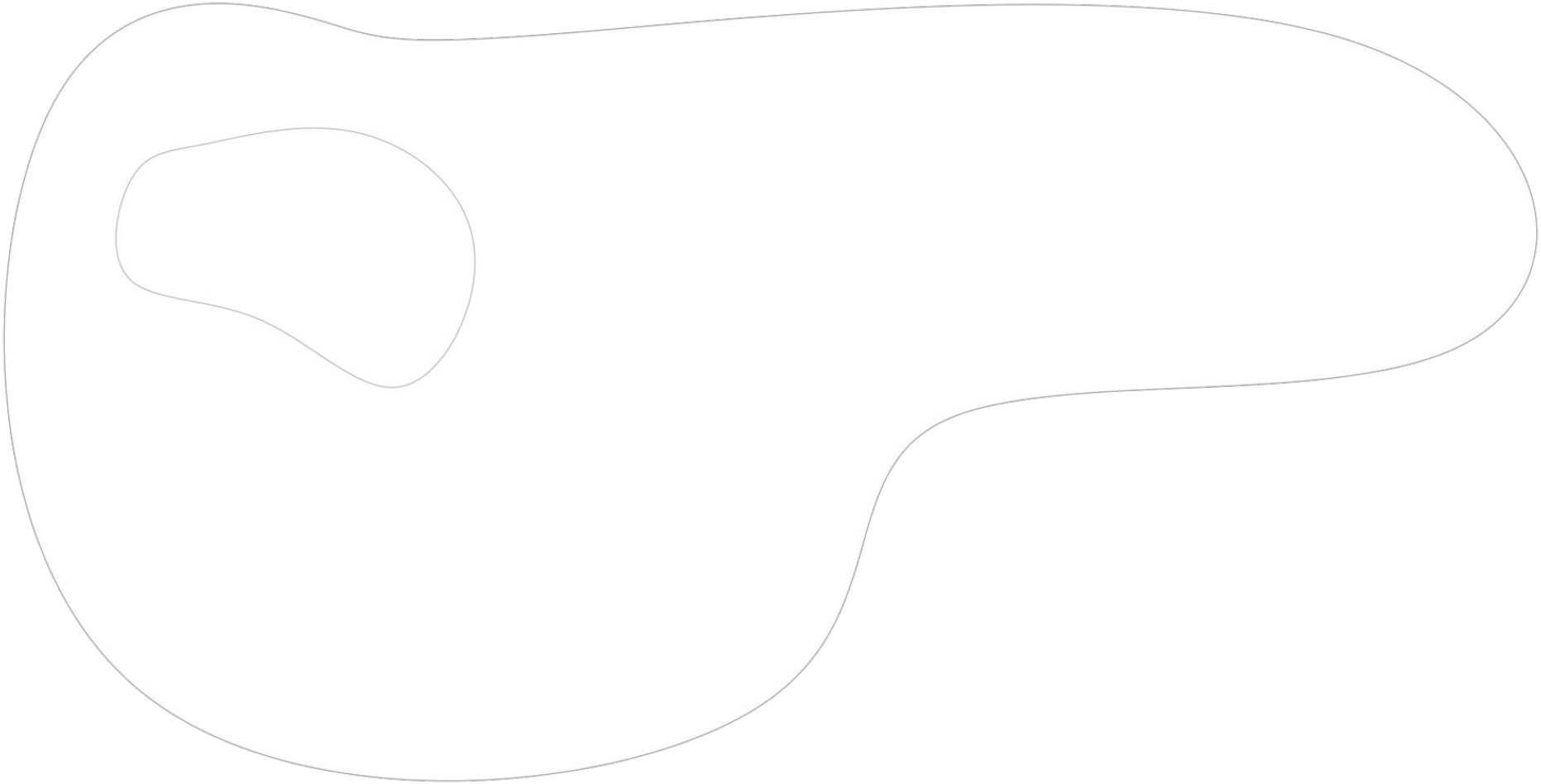


Saltwater Crocodile,
~~12m~~ ~6m





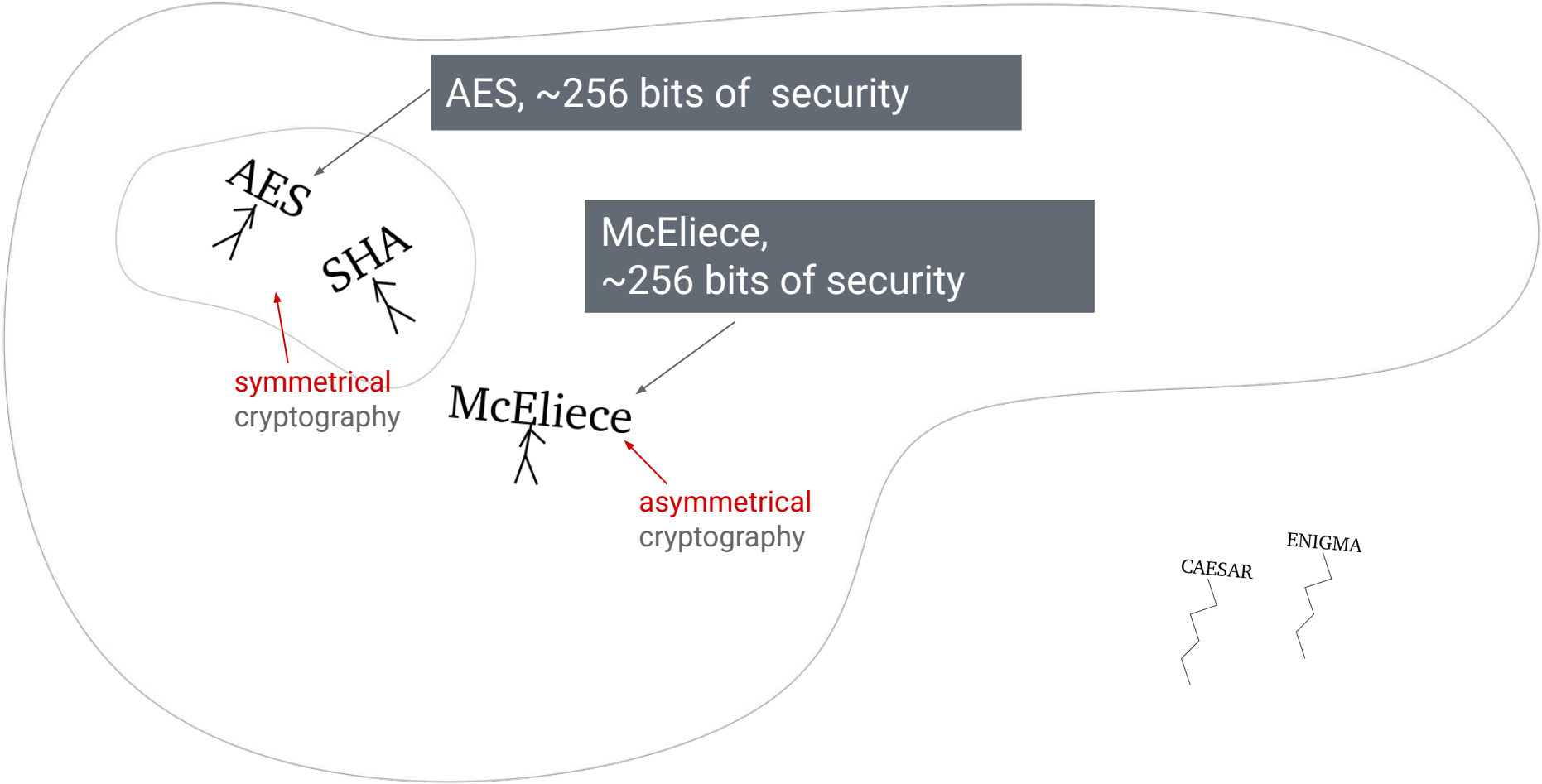
# Did we already learn something about cryptography?

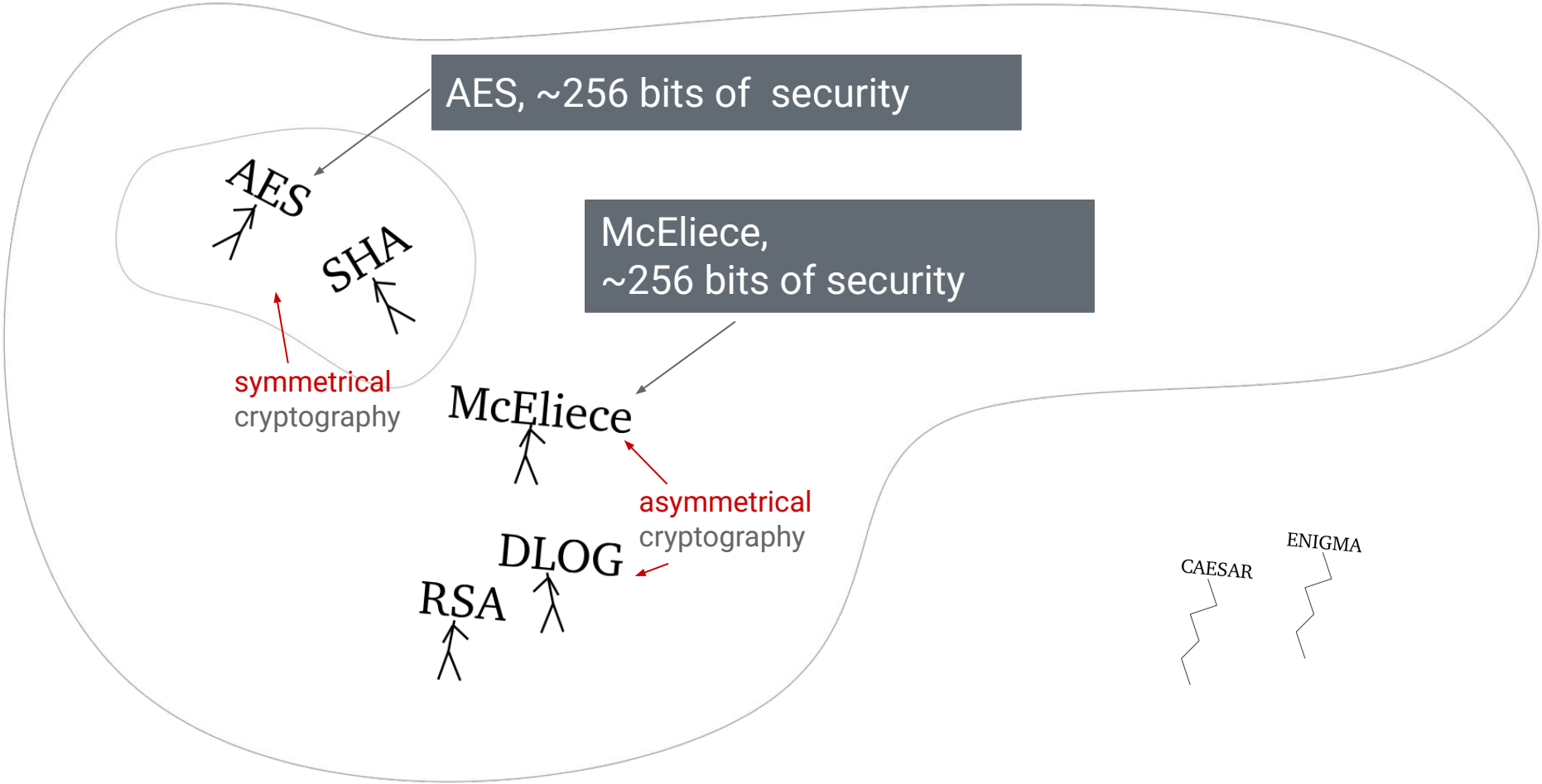



CAESAR
ENIGMA


AES
SHA
symmetrical
cryptography
CAESAR
ENIGMA


AES, ~256 bits of security
AES
SHA
symmetrical
cryptography
CAESAR
ENIGMA


AES, ~256 bits of security
AES
SHA
symmetrical
cryptography
McEliece,
~256 bits of security
McEliece
asymmetrical
cryptography
CAESAR
ENIGMA


AES, ~256 bits of security
AES
SHA
symmetrical
cryptography
McEliece,
~256 bits of security
McEliece
asymmetrical
cryptography
DLOG
RSA
CAESAR
ENIGMA

AES

SHA

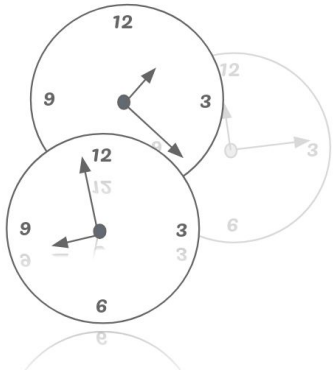
symmetrical
cryptography

McEliece

asymmetrical
cryptography

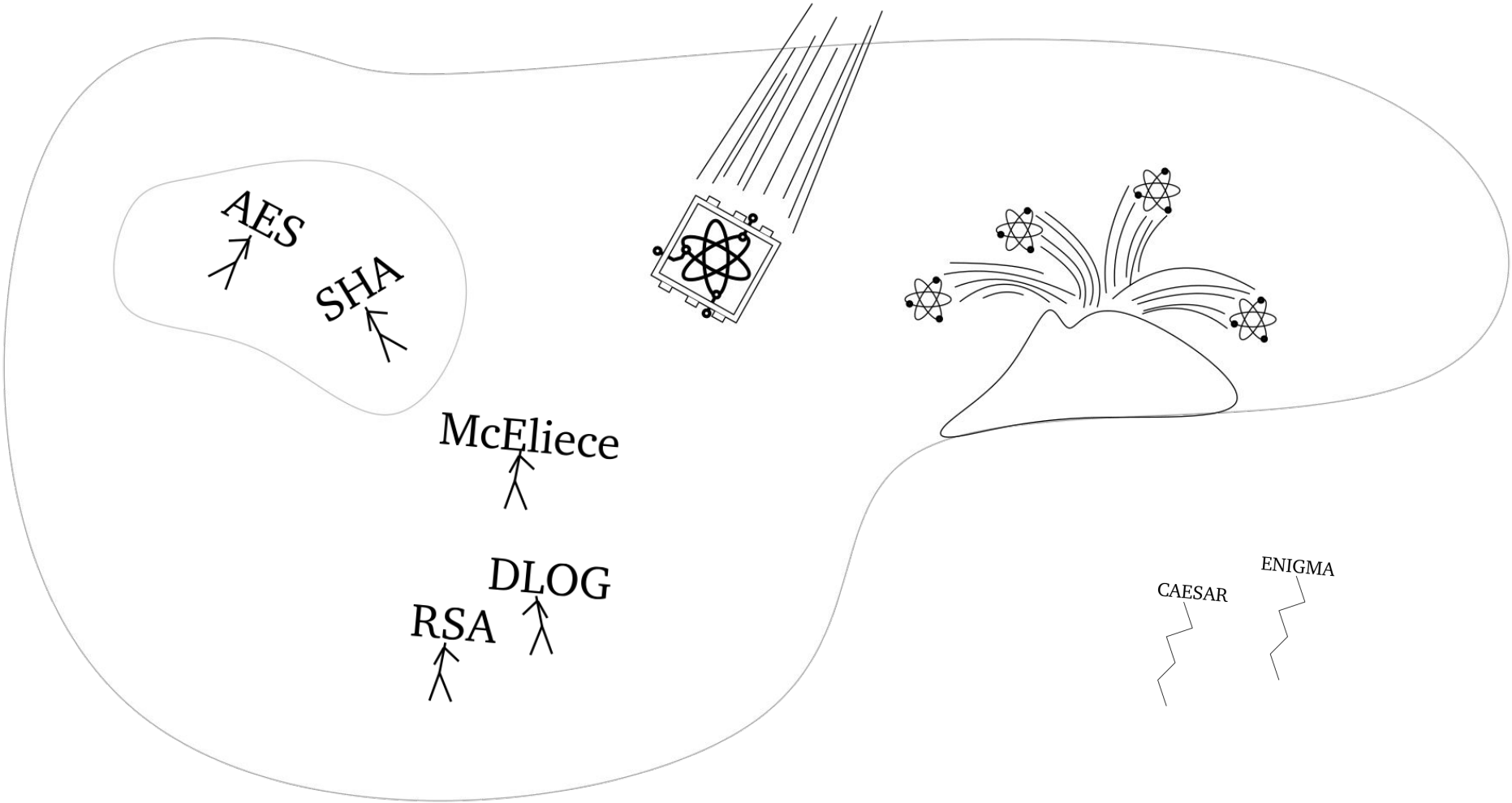
DLOG

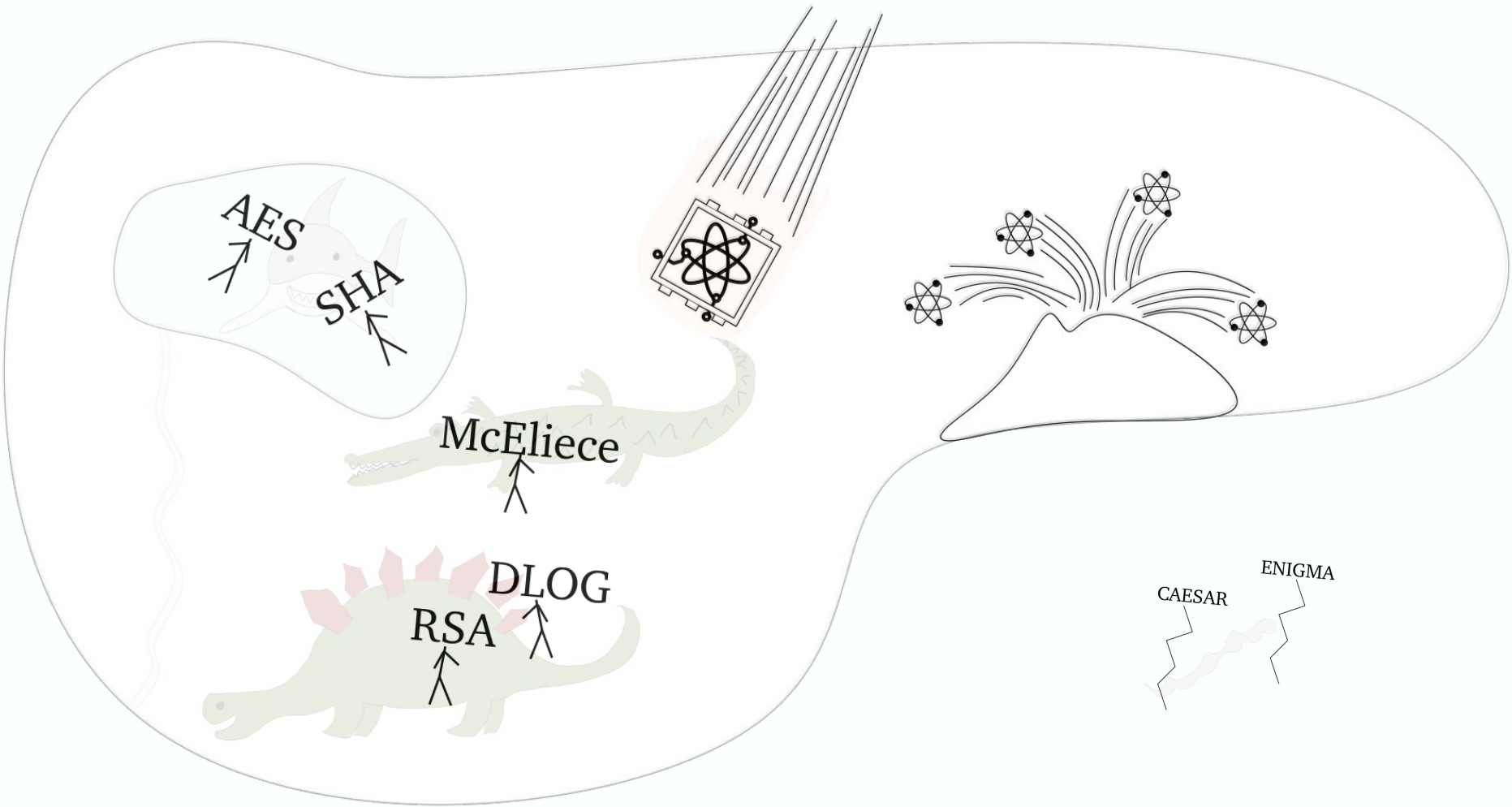
RSA

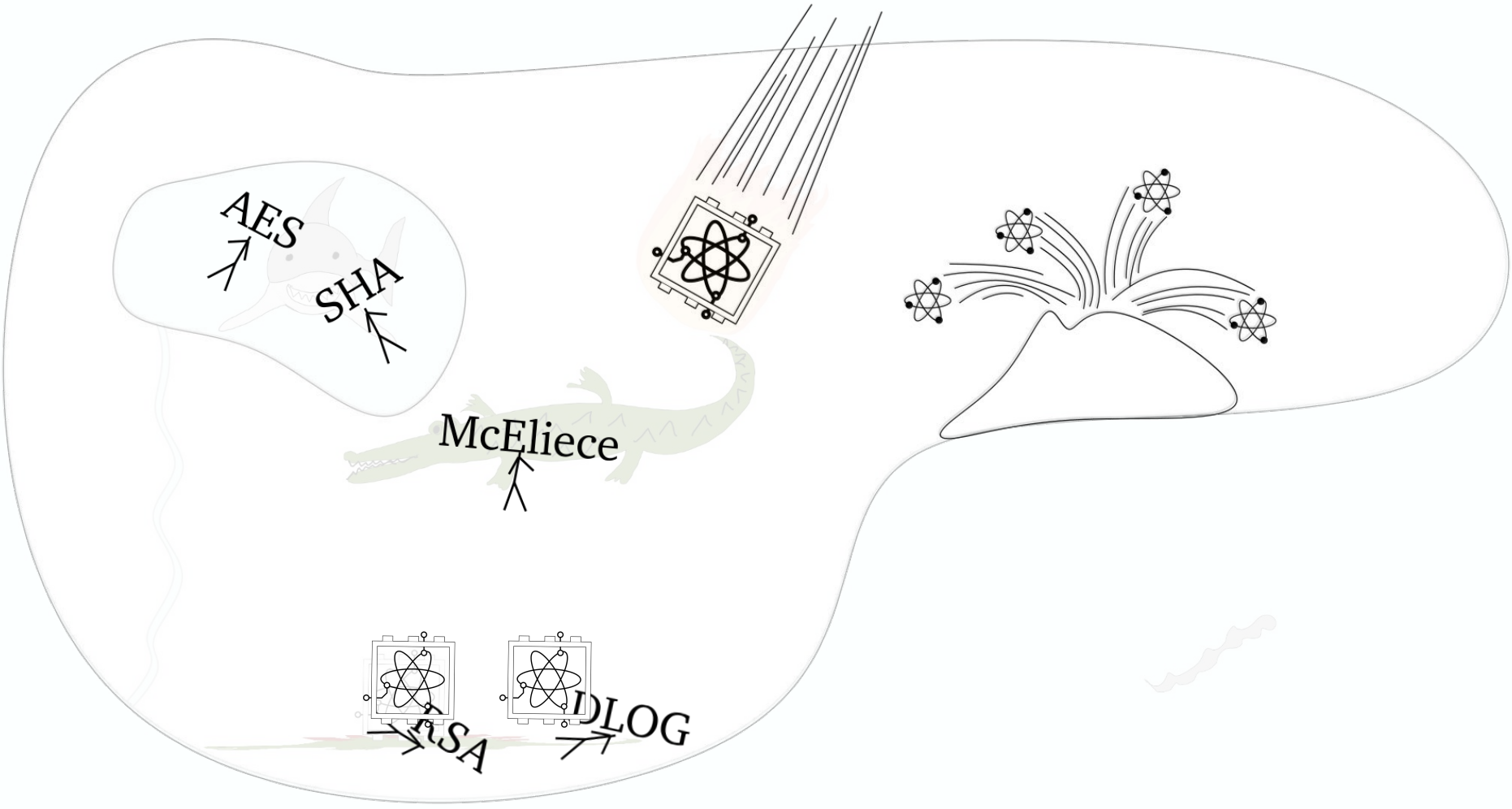


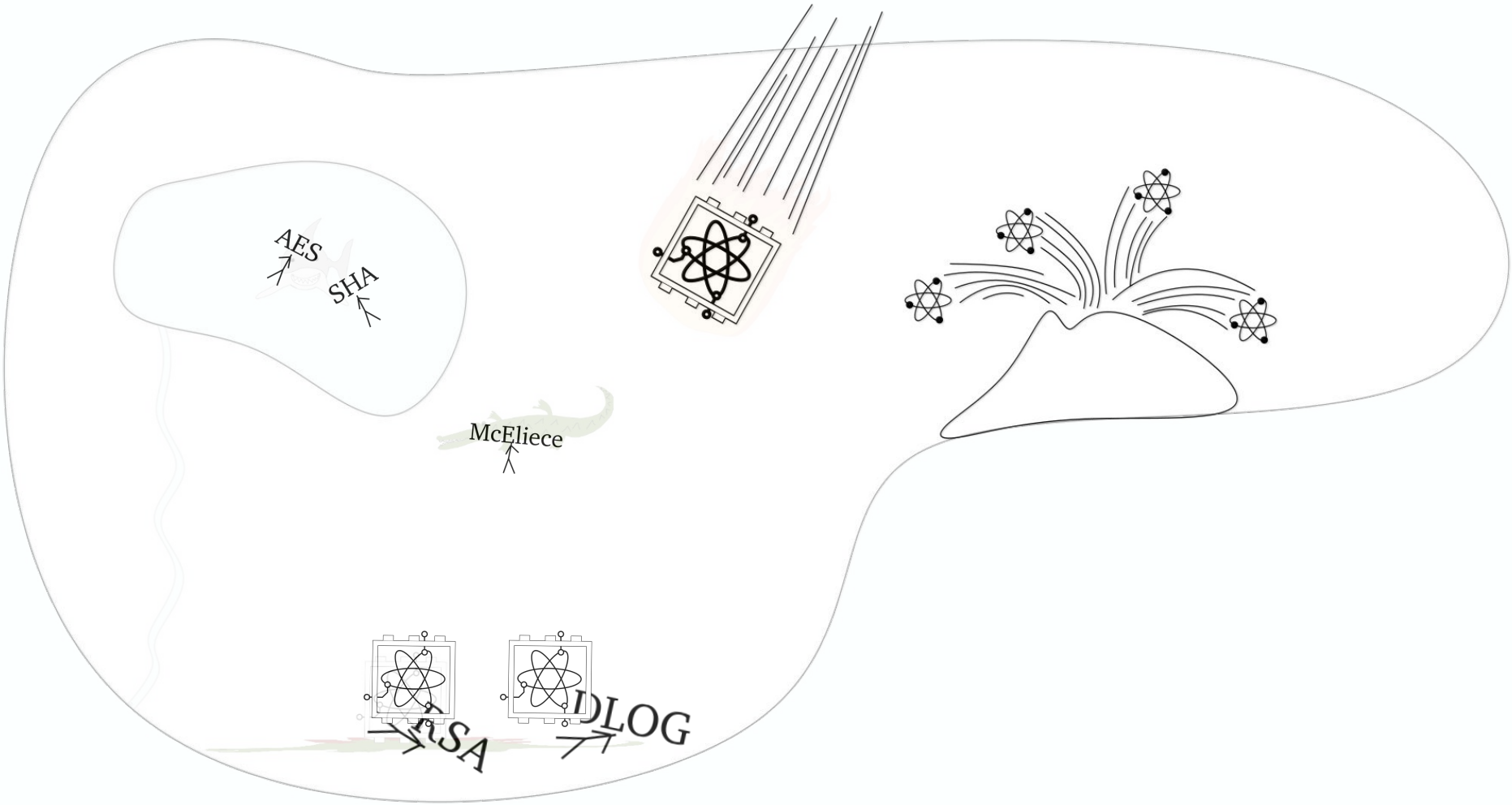

CAESAR

ENIGMA

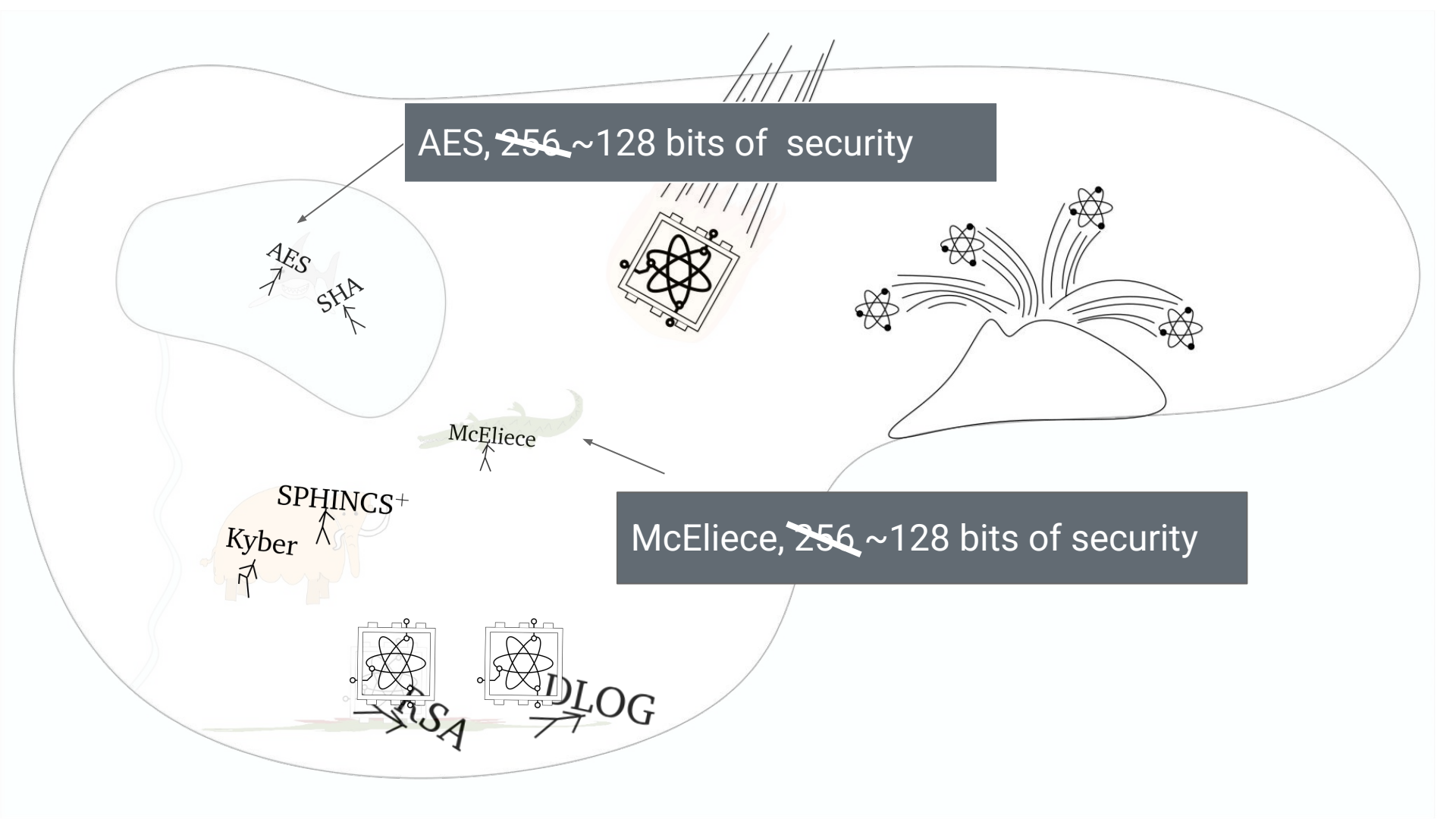

AES
SHA
McEliece
DLOG
RSA
CAESAR
ENIGMA


AES
SHA
McEliece
DLOG
RSA
CAESAR
ENIGMA


AES
SHA
McEliece
RSA
DLOG


AES
SHA
McEliece
RSA
DLOG


AES, ~~256~~ ~128 bits of security
AES
SHA
McEliece
McEliece, ~~256~~ ~128 bits of security
RSA
DLOG


AES, ~~256~~ ~128 bits of security
AES
SHA
McEliece
McEliece, ~~256~~ ~128 bits of security
SPHINCS+
Kyber
RSA
DLOG

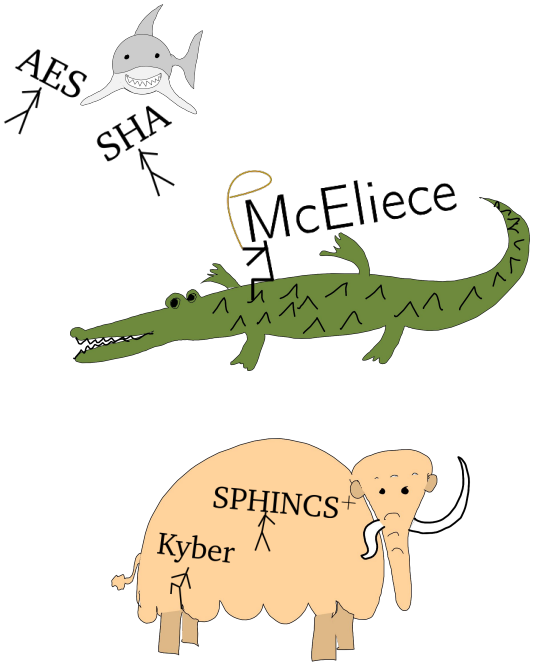
# Goals of this talks

1. Oblivious learning about cryptography 
2. Explain the above with **bold** claims

# What did we learn so far?

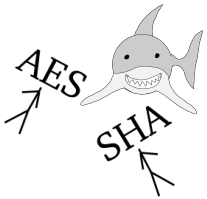
# What did we learn so far?

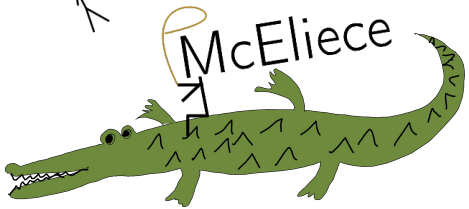
Quantum-secure cryptography exists,
but security is halved.



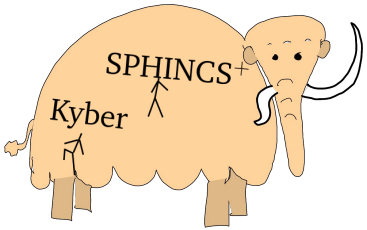
# What did we learn so far?

Quantum-secure cryptography exists,
but security is roughly halved.




The Why and the How

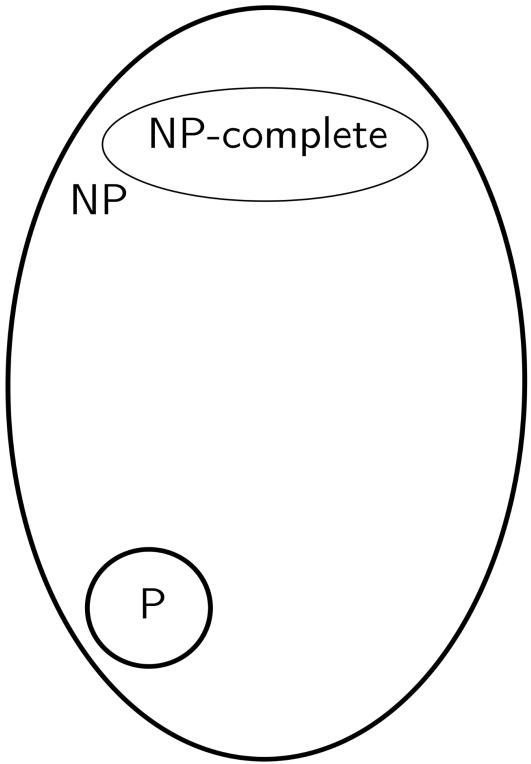


# The Why and the How

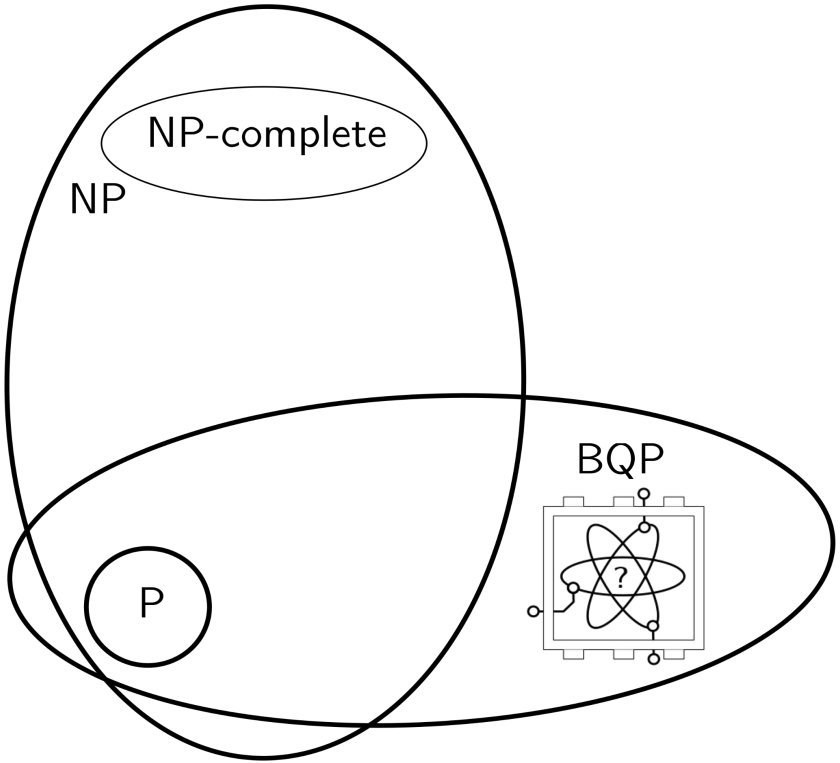
# The Why and the How

P

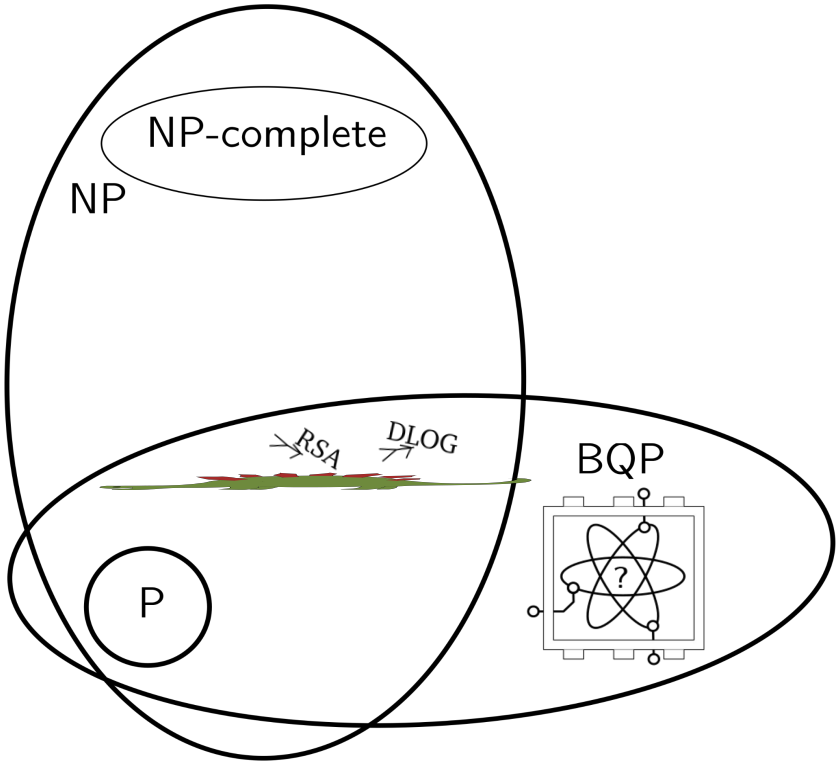
# The Why and the How



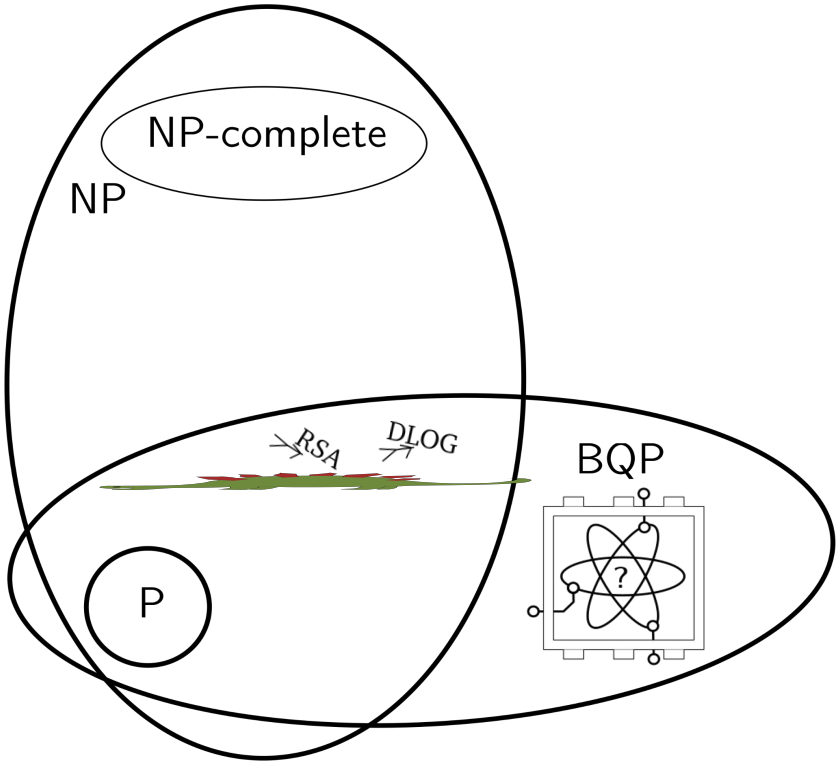
# The Why and the How



# The Why and the How



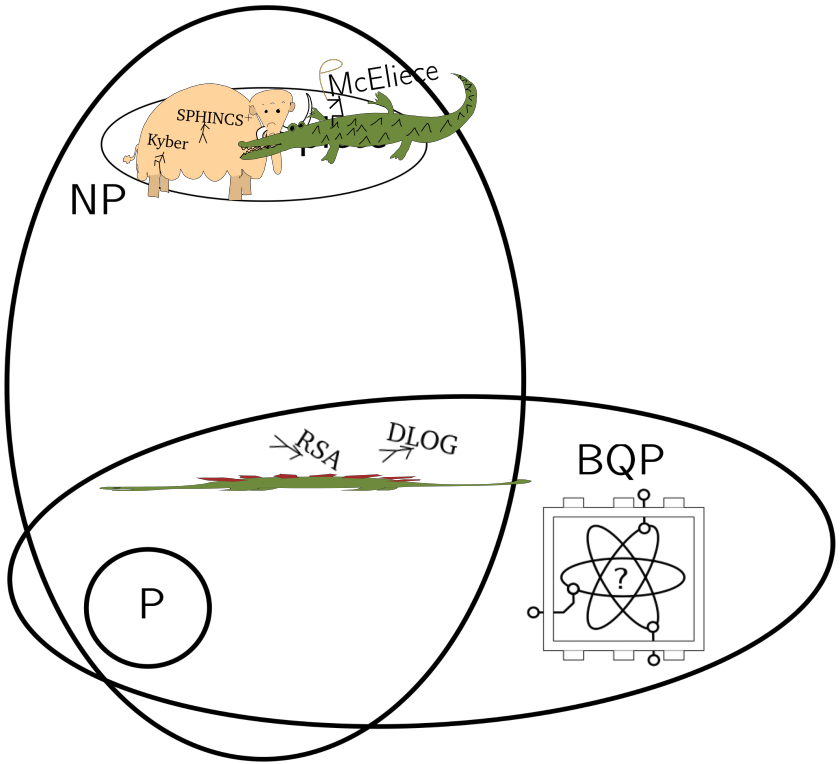
# The Why and the How




First **bold, unproven** claim

**Quantum computers cannot solve NP-complete problems in polynomial time**

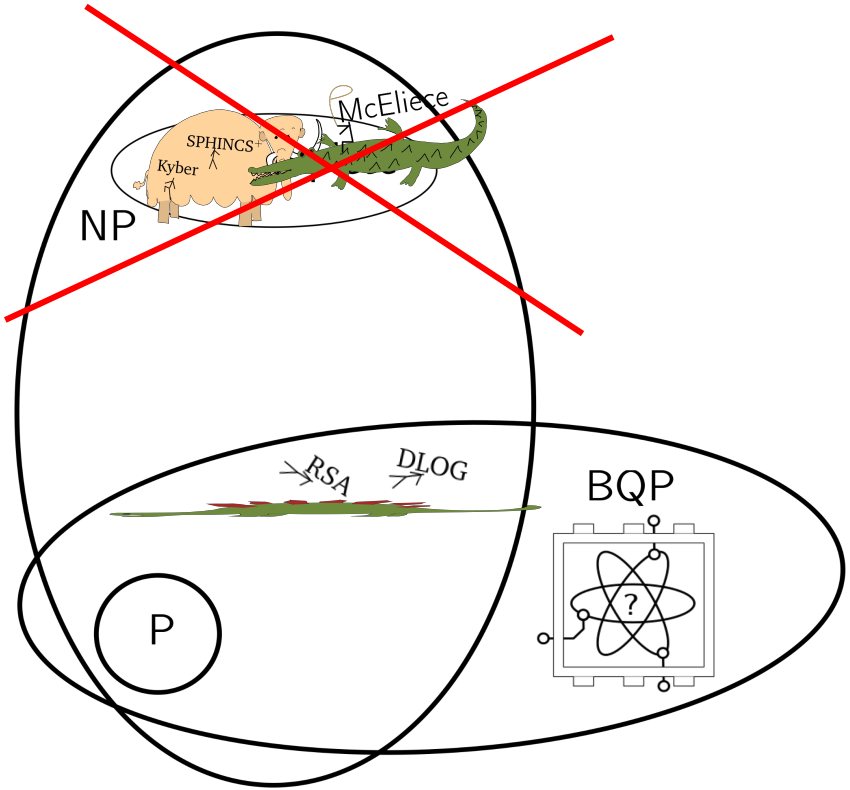
# The Why and the How




First **bold, unproven** claim

**Quantum computers cannot solve NP-complete problems in polynomial time**

# The Why and the How

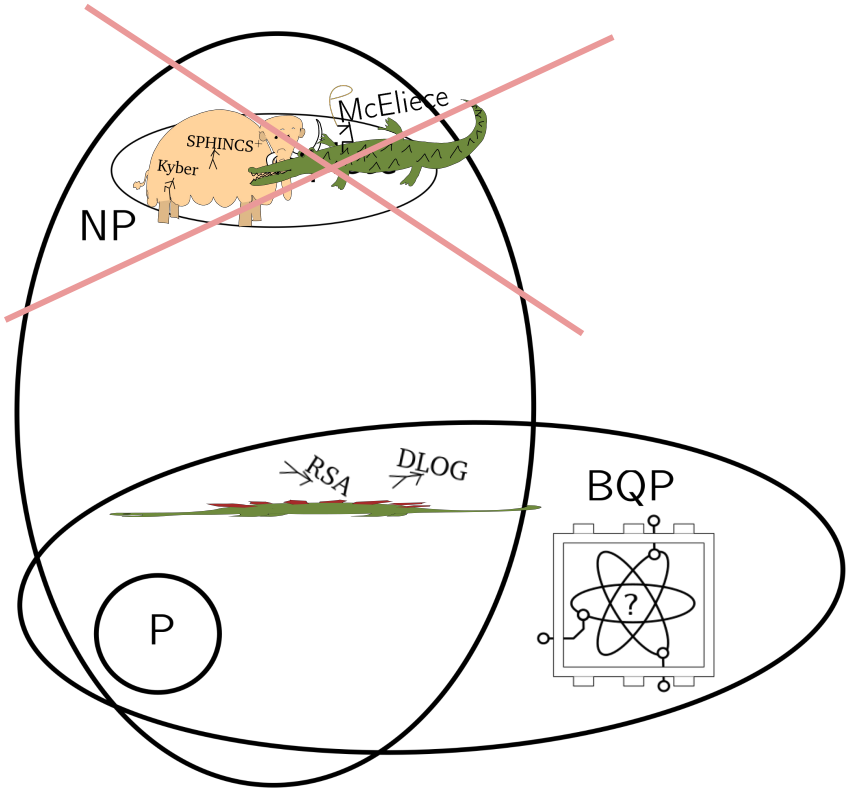



First **bold, unproven** claim

**Quantum computers cannot solve NP-complete problems in polynomial time**

Cryptography as difficult to break as NP-complete problem **not known** to be possible!

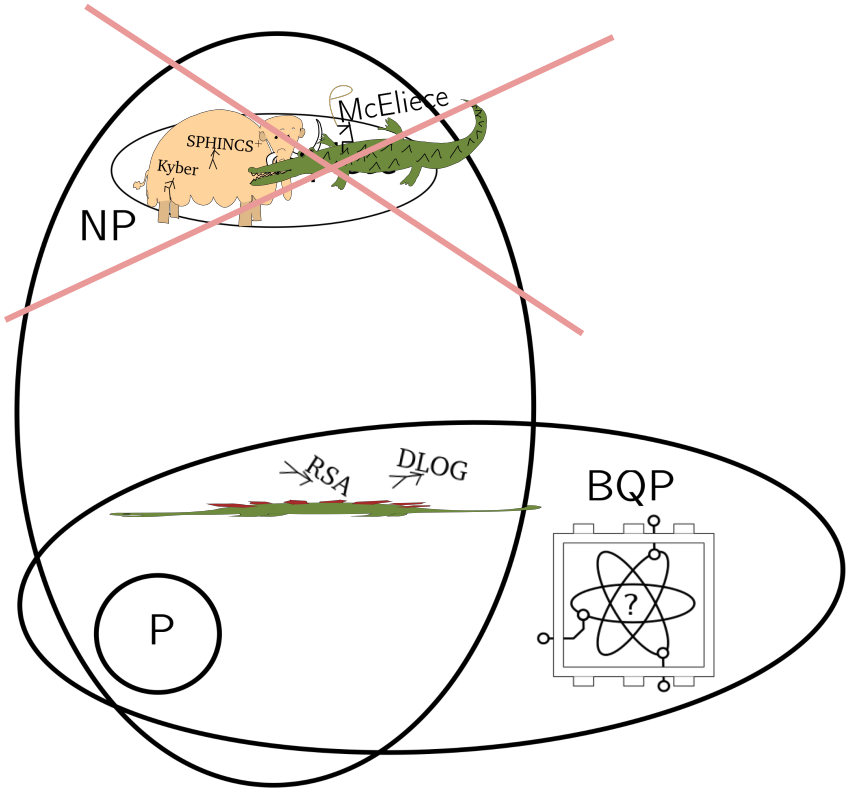
# The Why and the How




**NP** Solution can be guessed and verified in poly-time

**co-NP** Non-existence of solution can be guessed and verified in poly-time

# The Why and the How



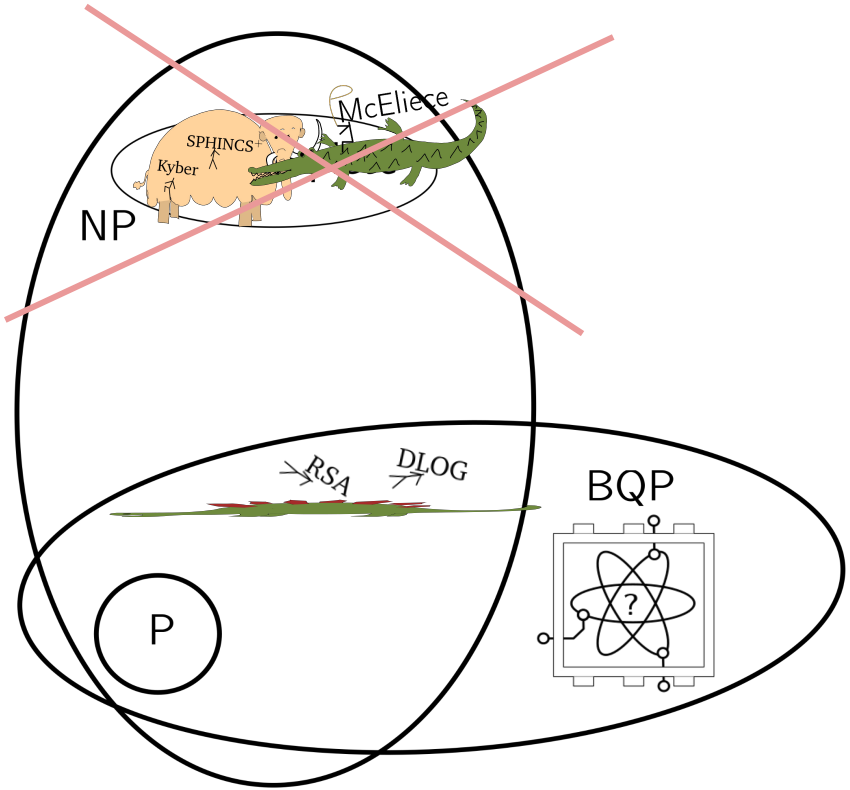

**NP** Solution can be guessed and verified in poly-time

**co-NP** Non-existence of solution can be guessed and verified in poly-time

**Crypto** Enc(pk, M) → C
Dec(sk, C) → M
∈ NP ∩ co-CPs

**Secure** If finding M from (pk,C) is NP-complete.

# The Why and the How




**NP** Solution can be guessed and verified in poly-time

**co-NP** Non-existence of solution can be guessed and verified in poly-time

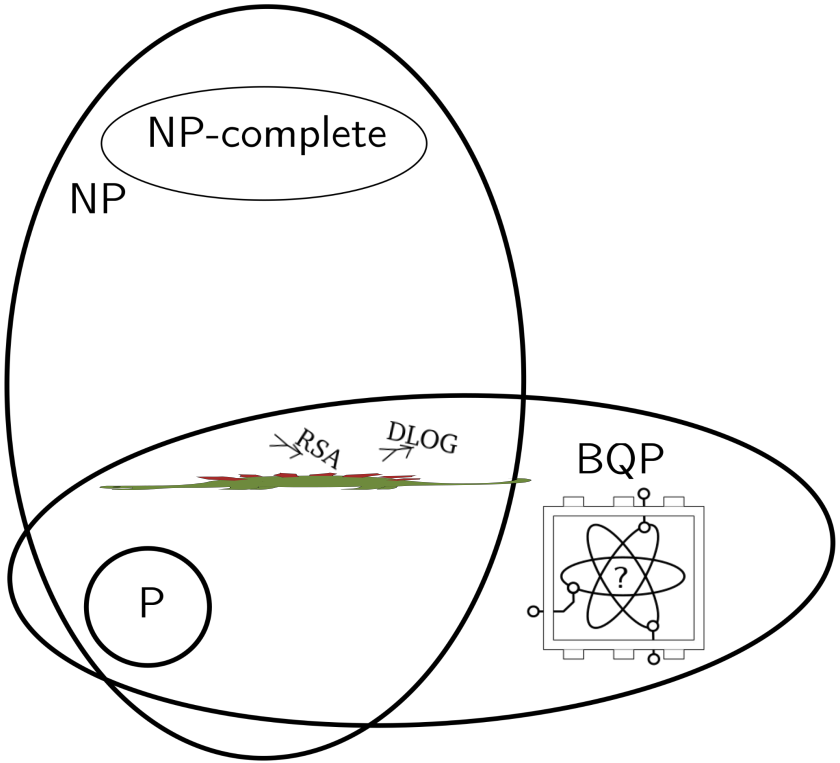
**Crypto** Enc(pk, M) → C
Dec(sk, C) → M
∈ NP ∩ co-CPs

**Secure** If finding M from (pk,C) is NP-complete.

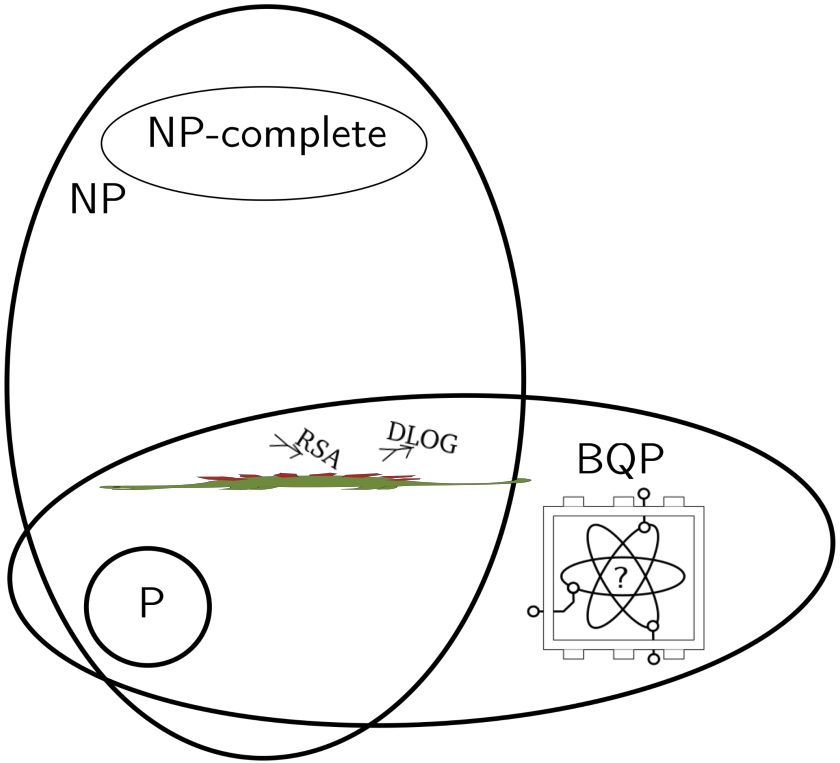
Showing **Crypto** as **Secure** as **NP-complete** problem difficult

means **proving** **NP = co-NP**
(and also P ≠ NP)

# The Why and the How



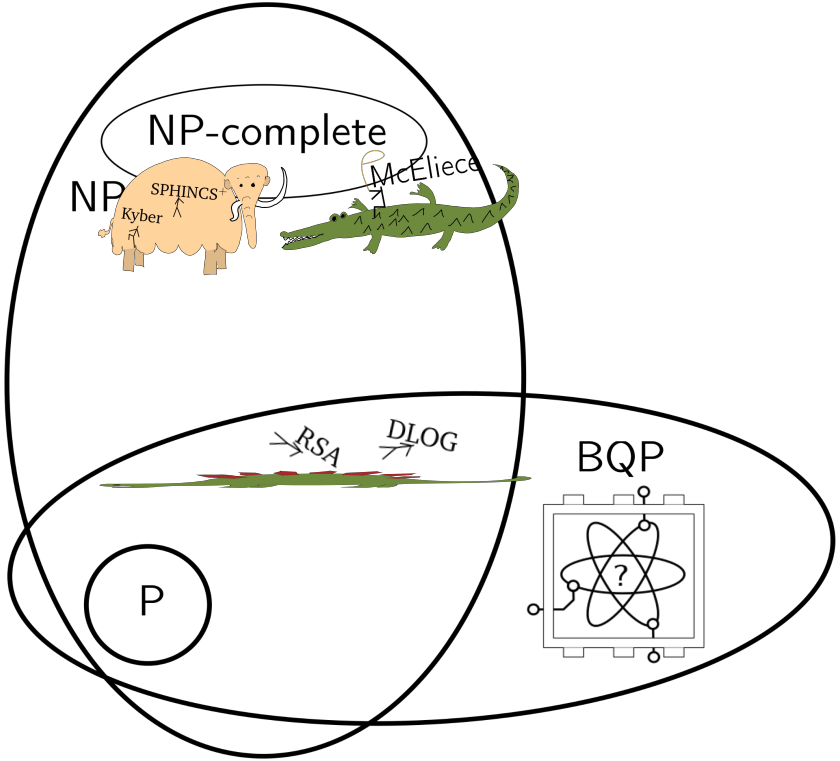
# The Why and the How




another **bold, unproven** claim

**Take NP-complete problem,**
**make it a bit easier**
**⇒**
**Crypto!**

# The Why and the How



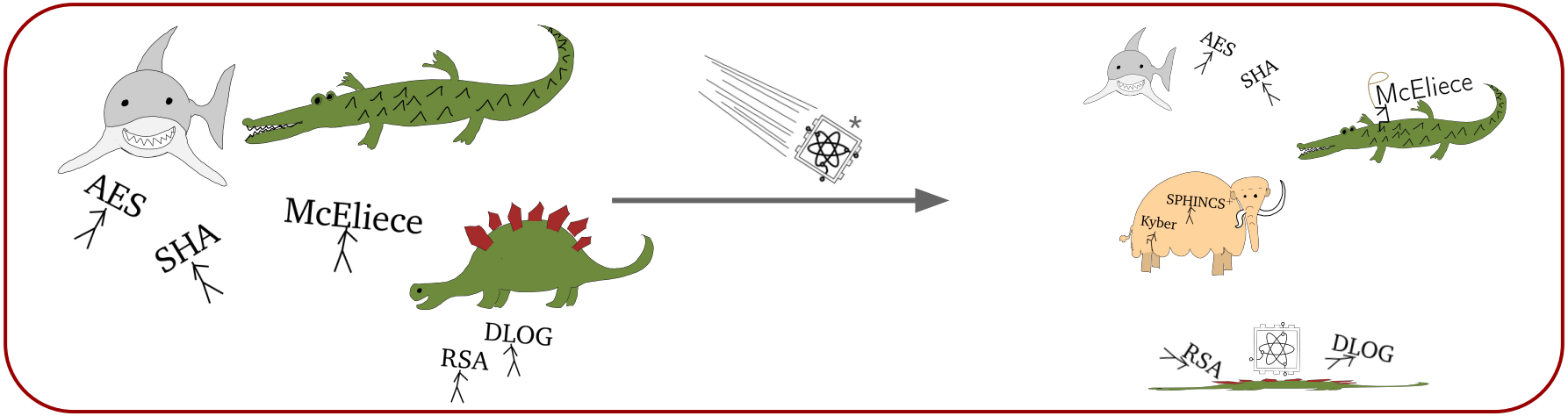

another **bold, unproven** claim

**Take NP-complete problem,**
**make it a bit easier**
**⇒**
**Crypto!**

# Goals of this talks

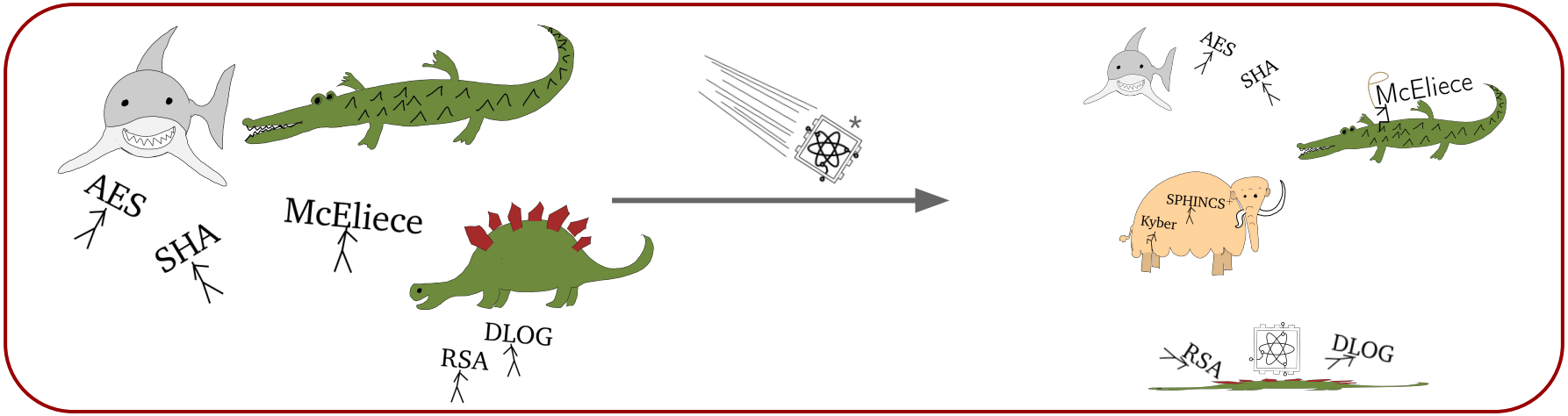
1. Oblivious learning about cryptography 
2. Explain the above with **bold** claims 

# Summary & Conclusion




*No dinosaurs were harmed during the making of these slides.

# Summary & Conclusion




**Quantum-secure cryptography can be build from nearly-NP-complete problems**

*No dinosaurs were harmed during the making of these slides.

Last **bold**, unproven claim