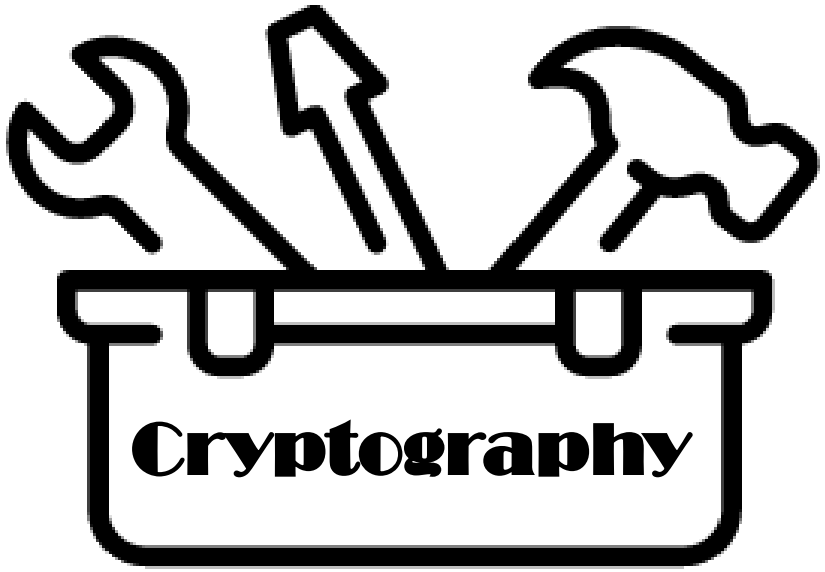
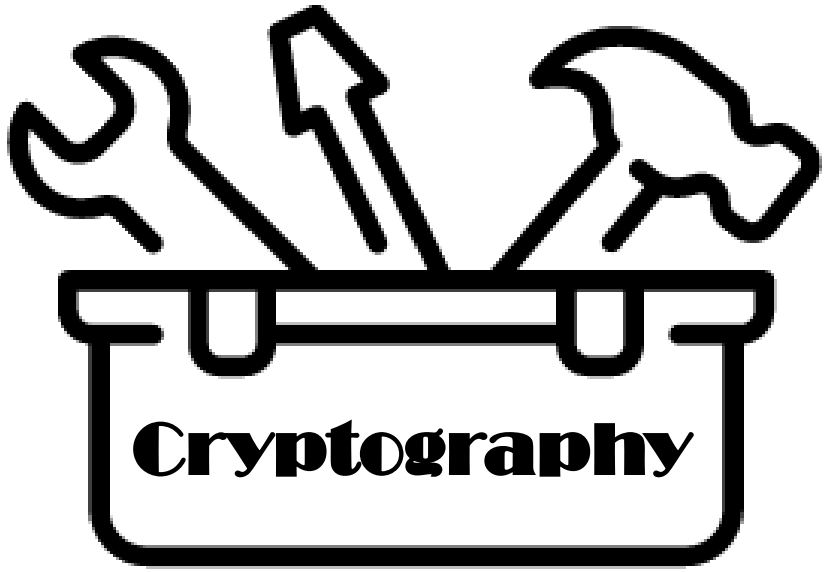
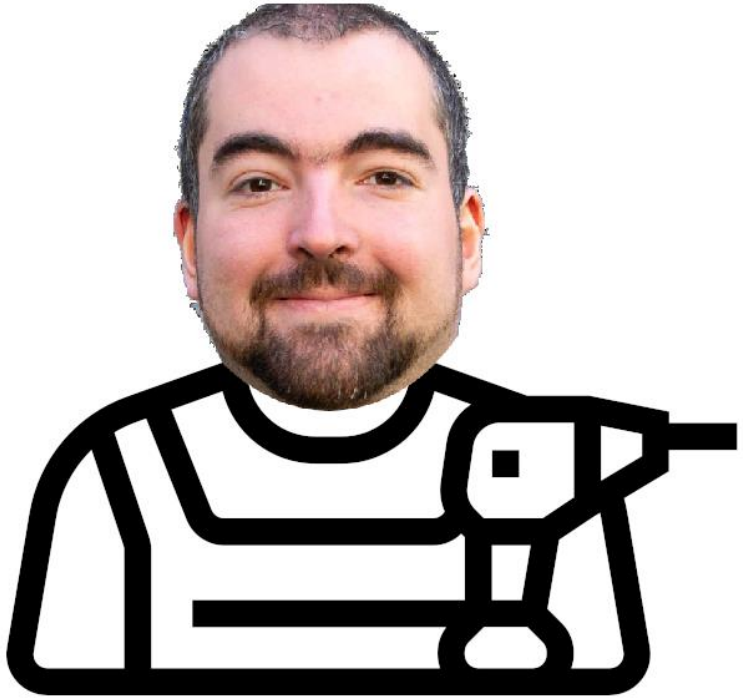
Secure Application Design

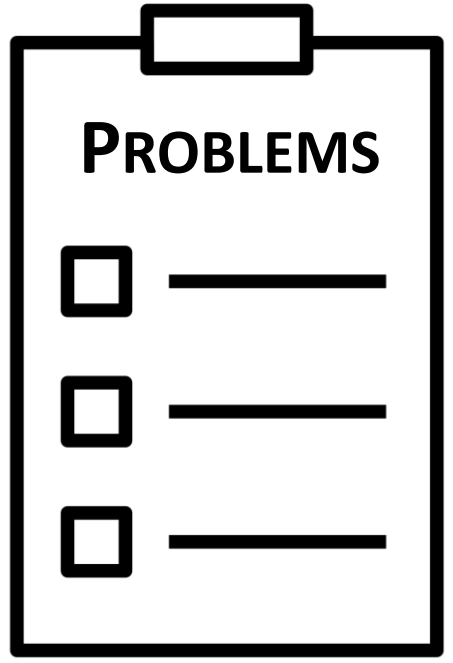
Summer 2025

Jakob Heher, www.isec.tugraz.at
he/his









Your SEAD VO team

Jakob Heher

he/his

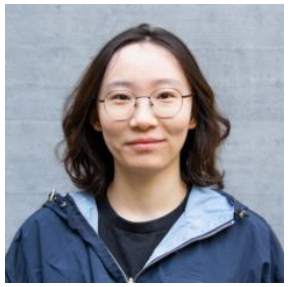
Various Guest Speakers

Your SEAD KU team

Hannes Weissteiner

he/his

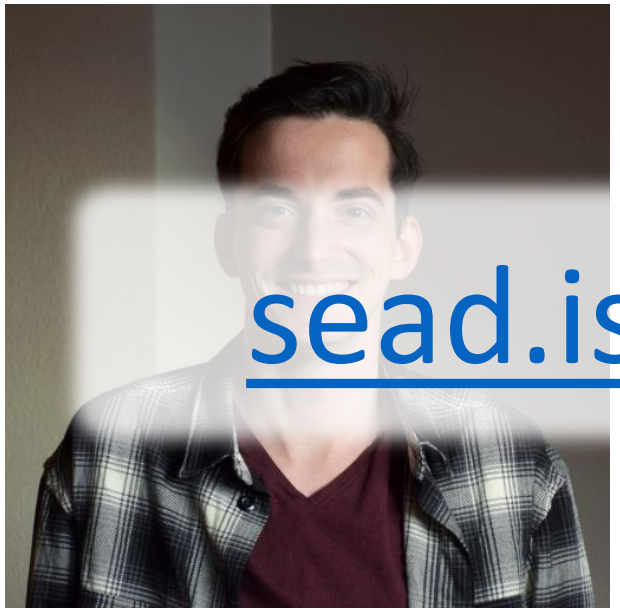
Jakob



Xufan

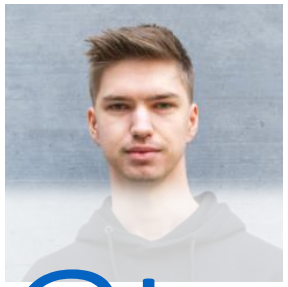
Jakob

Your SEAD KU team



Hannes Weissteiner

he/his



Jakob

Xinyan

Jakob

sead.isec@tugraz.at

What, When & Where?

- Lecture: every Friday 14:00-16:00
  - 14:00 sharp
  - in HS i11 *(here)*
  - Recordings available on request
- Practicals:
  - Introduction with  : after today's lecture *(here)*
  - P2 Intro with : 21st of March *(replaces the lecture)*
- Questions?
  - Email us: sead.isec@tugraz.at
  - Ask on ISEC Discord: https://discord.gg/9KKGfndsD5

How To Pass This Course

Lecture Exam

- Written exam
- End of semester *(probably July 4th)*
- Partial Open-book exam
  - One hand-written A4 sheet
  - Two-sided
  - Write whatever you want on it
- Didn't attend the main exam?
  - Ask for an oral exam date later!

OR

Seminar Talk

- Prepare a topic and:
  1. submit a ≥ 7 page report
  2. give a 45min presentation
- How this works:
  1. Choose a topic by March 23rd
  2. Email us at sead.isec@tugraz.at
  3. We approve/reject your topic
- Explicitly also invited:
  - CS Ethics, Usability, …
- Talk to us if you're unsure!

March 7th	Recap: Cryptography
March 14th	Common Vulnerabilities
March 28th	Trust & Privacy
April 4th	Identity
April 11th	Web Authentication Factors
	------ EASTER BREAK ------
May 2nd	OpenID Connect & FedCM
May 9th	Transparency
May 16th	Trust in Keys & Software
May 23rd	Case Study: TLS
June 6th	Case Study: ID Austria & eIDAS
June 13th	Case Study: towards the EU Digital Identity Wallet
June 20th	Current Topics Spotlight
June 27th	Seminar Presentations
July 4th	Lecture Exam

ENHANCE YOUR SKILLS
ISEC INTERNSHIP

Have you started planning your next summer? Are you interested in what we are doing at ISEC? Or do you want to broaden your knowledge in Security, Privacy, Cryptography or Verification? Looking for a great work and study environment where you can learn from the best while working on professional projects outside of your daily work?

At ISEC, we know how important our people are, and we value each and every one. Diversity is one of our highest goods, and we are happy to welcome people from different backgrounds to enrich our research.

Every year, we offer over a dozen summer internships where students will improve their knowledge and skills, get to know some of the best security research experts and work on professional research projects.
Our interns always become an important part of our project teams and contribute significantly to our work. During this time, our experienced team members help every intern to gain valuable working experience and develop and refine their skills.

Starting now until **March 31st 2025**, we are collecting applications for the

https://www.isec.tugraz.at/join/internships-and-student-staff/

Looking forward to seeing you next summer @ ISEC!

Secure Application Design

Recap: Cryptography

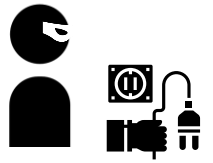
Summer 2025

Jakob Heher, www.isec.tugraz.at
he/his

What are our goals?

Confidentiality

Integrity



Availability

but also sometimes:

Privacy

Anonymity

Pseudonymity

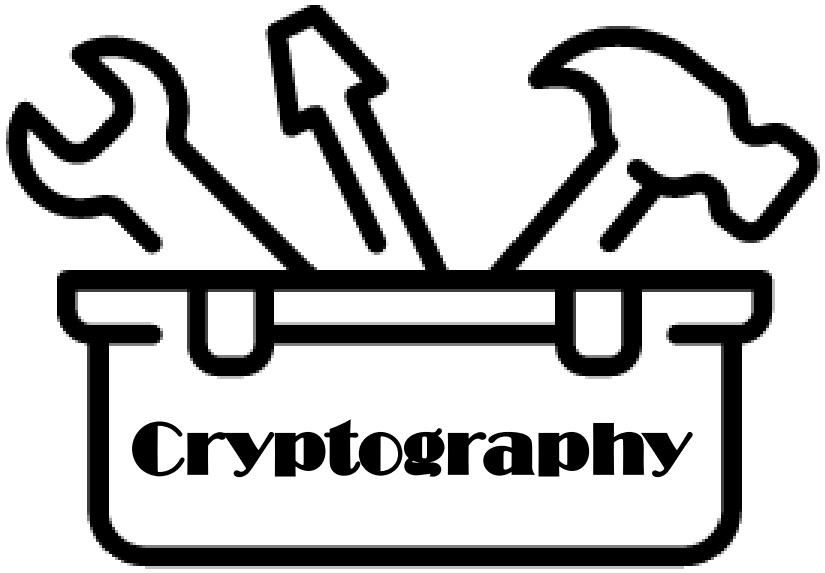
Commitment

Non-repudiation

Deniability

Time-stamping

What tools do we have so far?



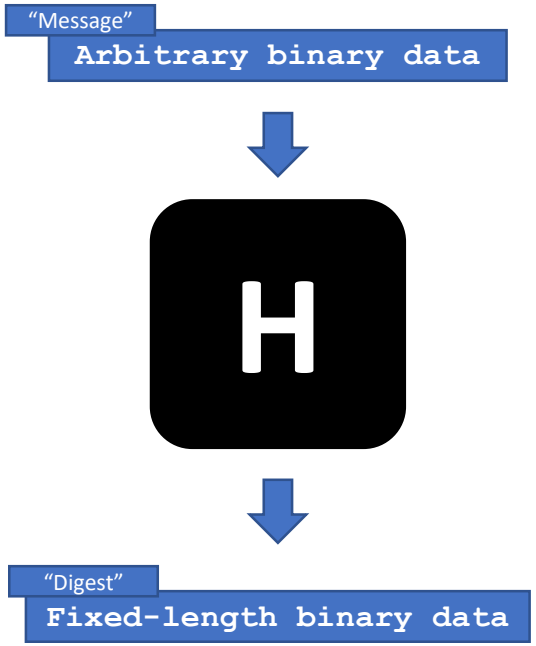
Hash Functions

- Finding a **pre-image**
  - Given $\mathbf{H(m)}$, find $\mathbf{m}$
- Finding a **second pre-image**
  - Given $\mathbf{m'}$, find $\mathbf{m}$ with $\mathbf{H(m) = H(m')}$
- Finding a **collision**
  - Find any $\mathbf{m}, \mathbf{m'}$ with $\mathbf{H(m) = H(m')}$

$\approx 2^n$ attempts

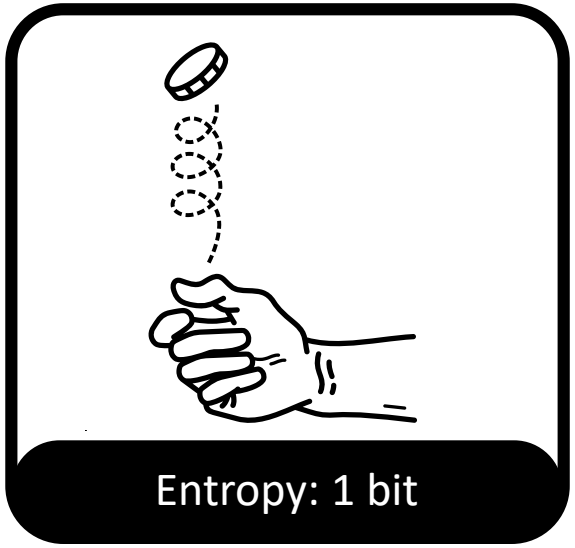
$\approx 2^{\left(\frac{n}{2}\right)}$ attempts

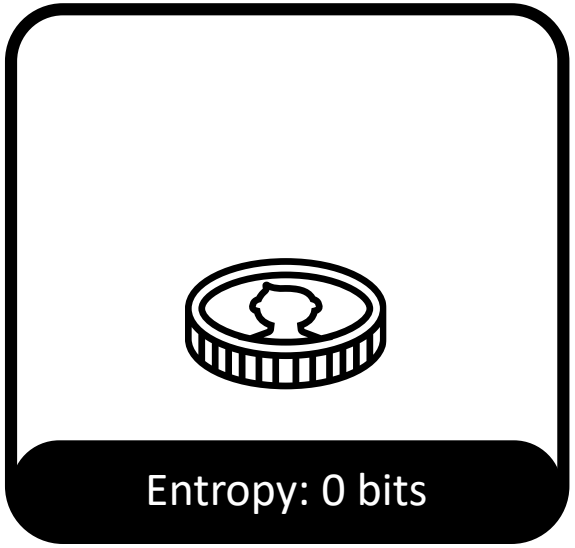
- Secure hash functions are **no worse than** the generic bounds

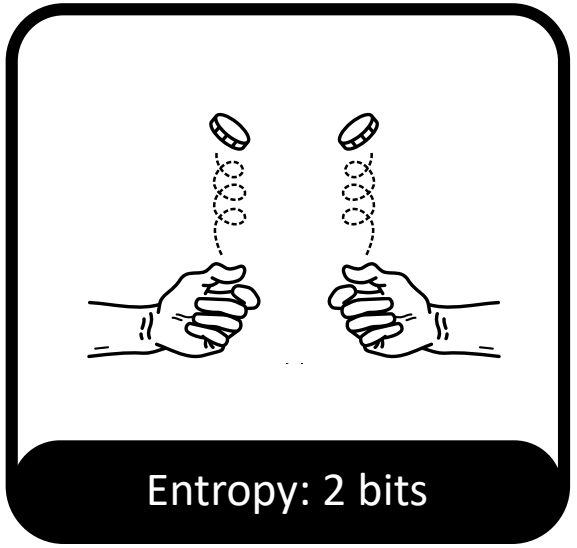


Entropy

- Measure of *unpredictability*





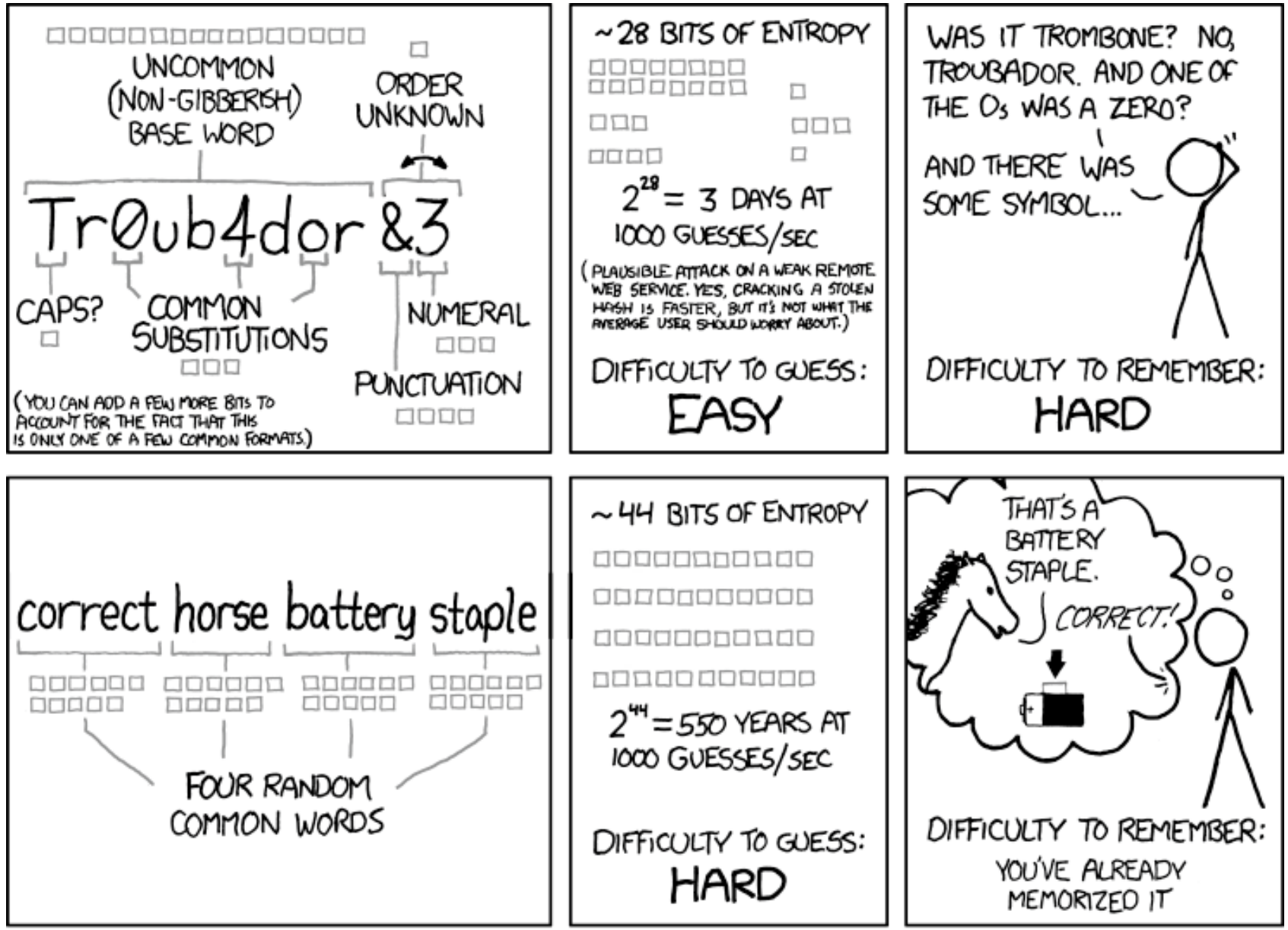


Entropy

- How long does it take to brute force a hash pre-image?

$$(time\ to\ brute\ force) = \frac{(number\ of\ attempts) \times (time\ per\ attempt)}{(number\ of\ parallel\ attempts)}$$

$2^{entropy}$



Password Hash Functions

- Designed to provide security for low-entropy input

$$(time\ to\ brute\ force) = \frac{(number\ of\ attempts) \times (time\ per\ attempt)}{(number\ of\ parallel\ attempts)}$$

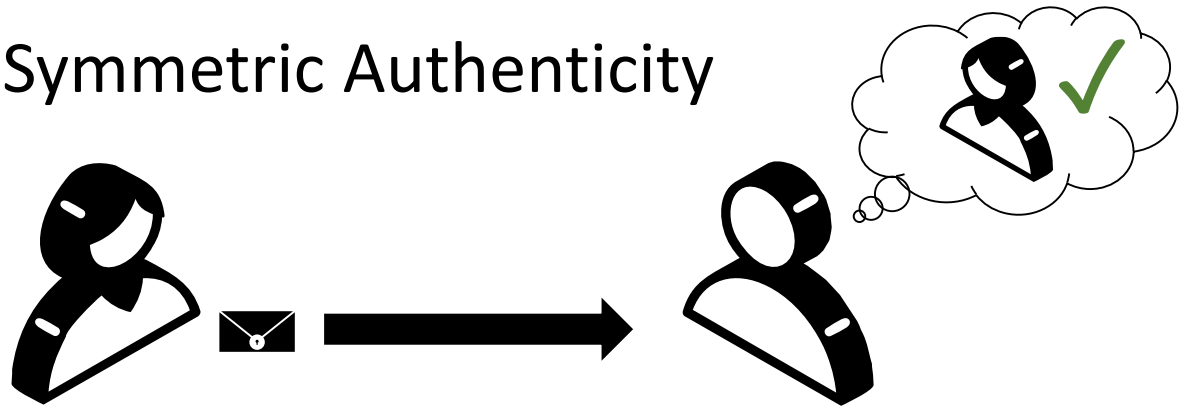
Slow to calculate

Hard to parallelize

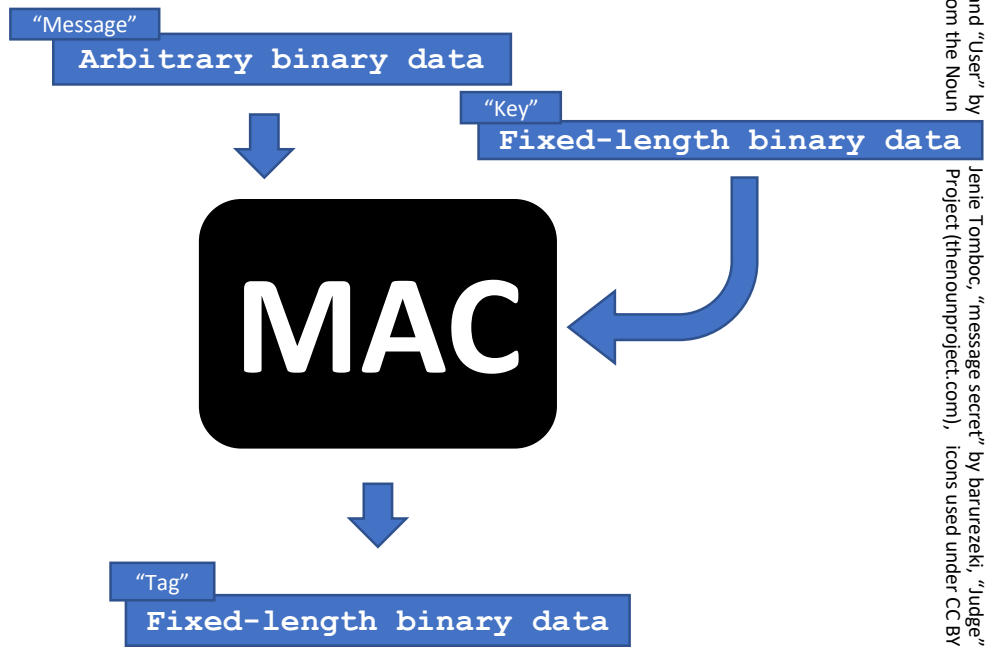
- **Examples:** Argon2, PBKDF2

Message Authentication Codes

- Symmetric Authenticity

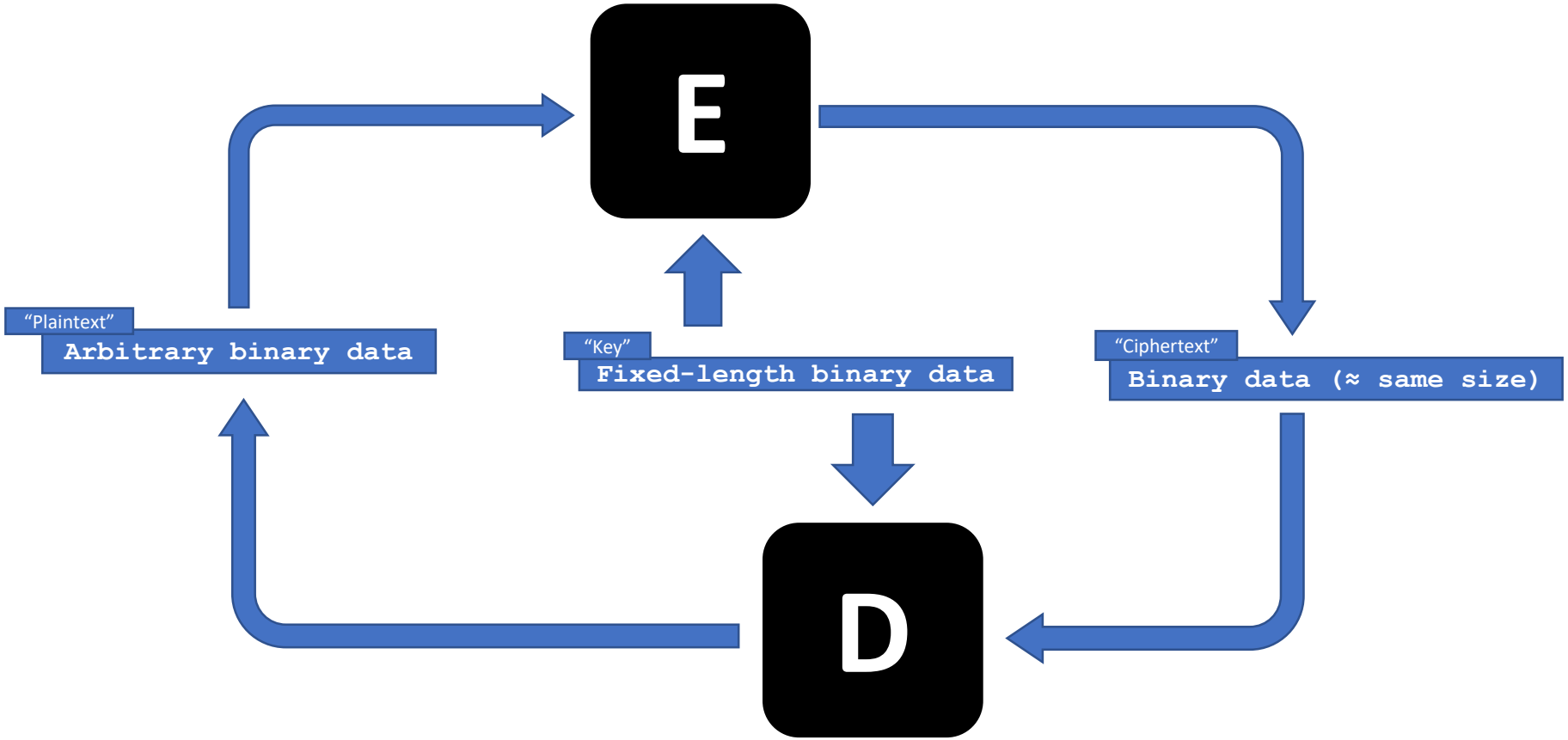


- Cannot prove origin to third party



Symmetric Encryption

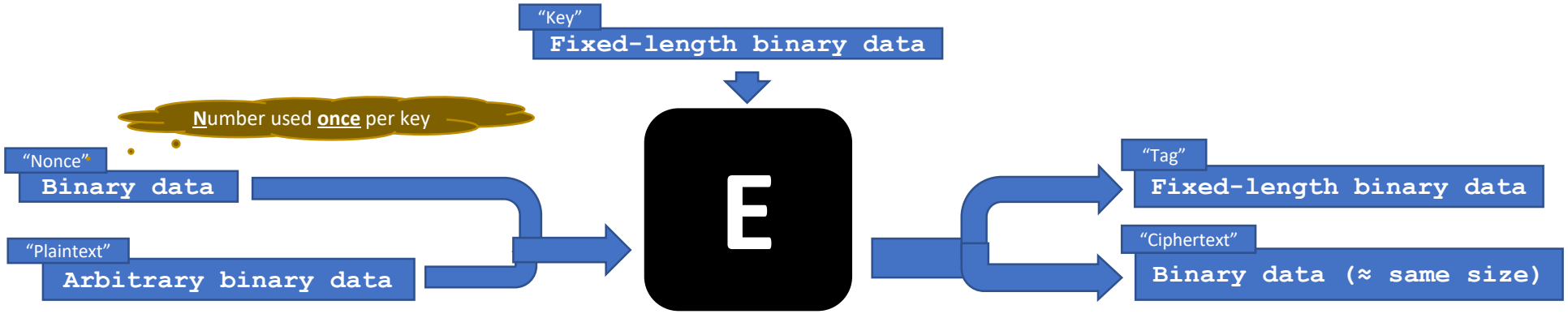
Don't use this directly!
(unless you have to)



Authenticated Encryption (with Additional Data)

This is the one to use!
(unless you have reasons not to)

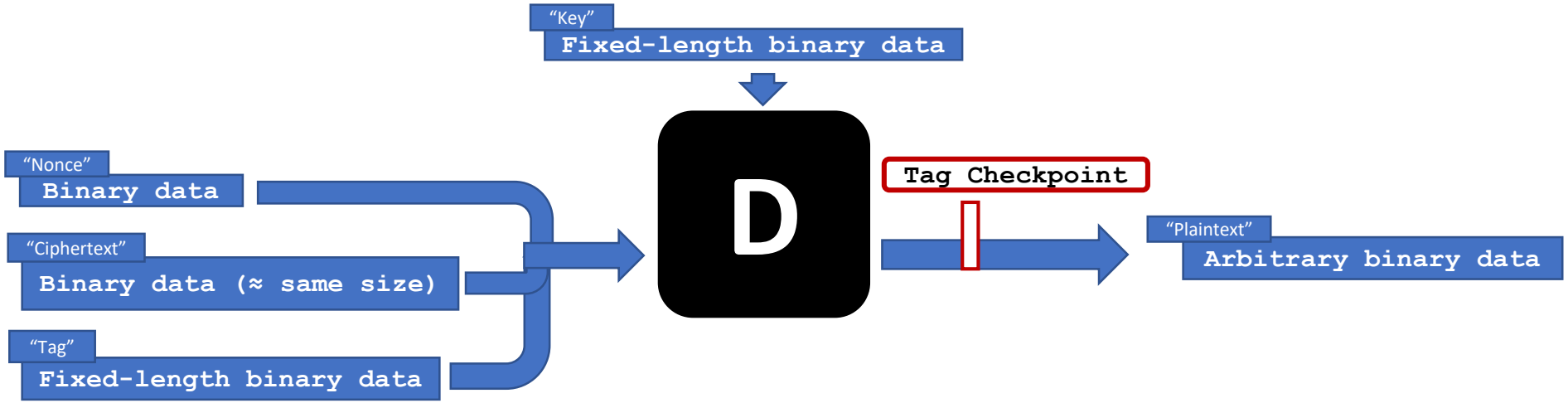
- Ciphertext comes with authenticity “tag”



Authenticated Encryption (with Additional Data)

This is the one to use!
(unless you have reasons not to)

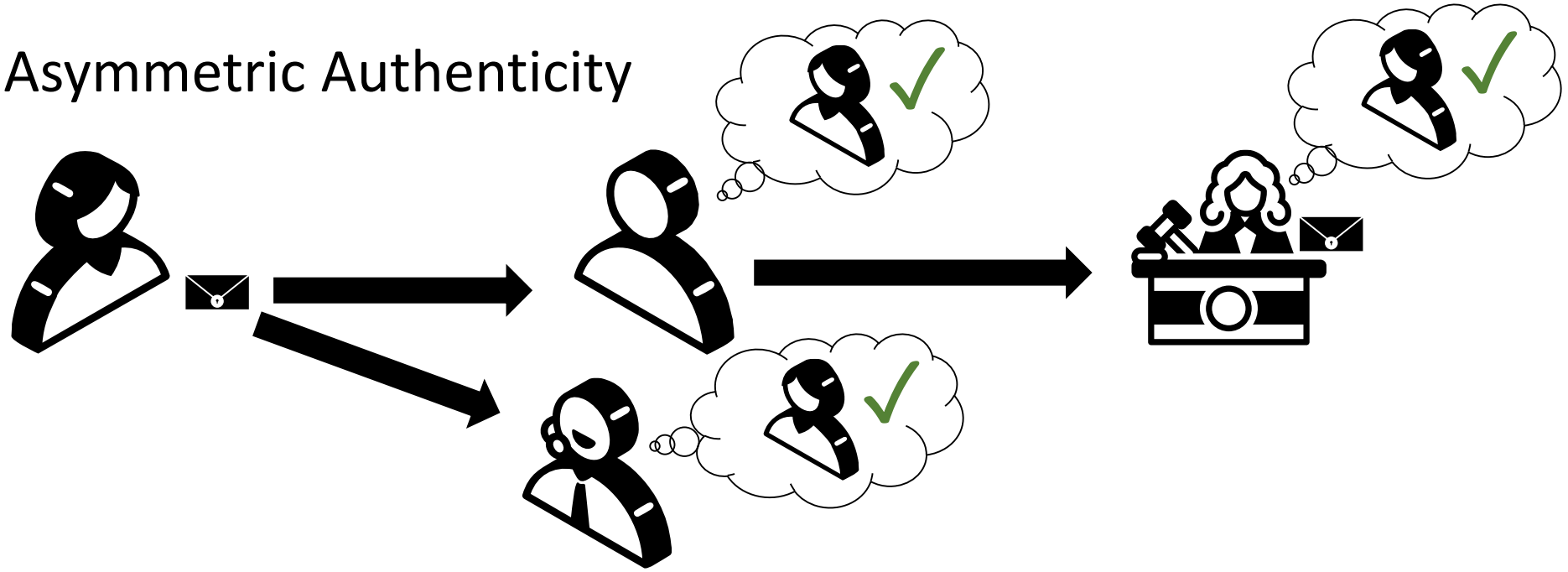
- Ciphertext comes with authenticity "tag"
- Decryption will fail unless tag is correct



- **Examples:** AES-GCM-SIV, AES-SIV, XChaCha20-Poly1305, AES-CCM
- **Be careful with:** AES-GCM

Digital Signatures

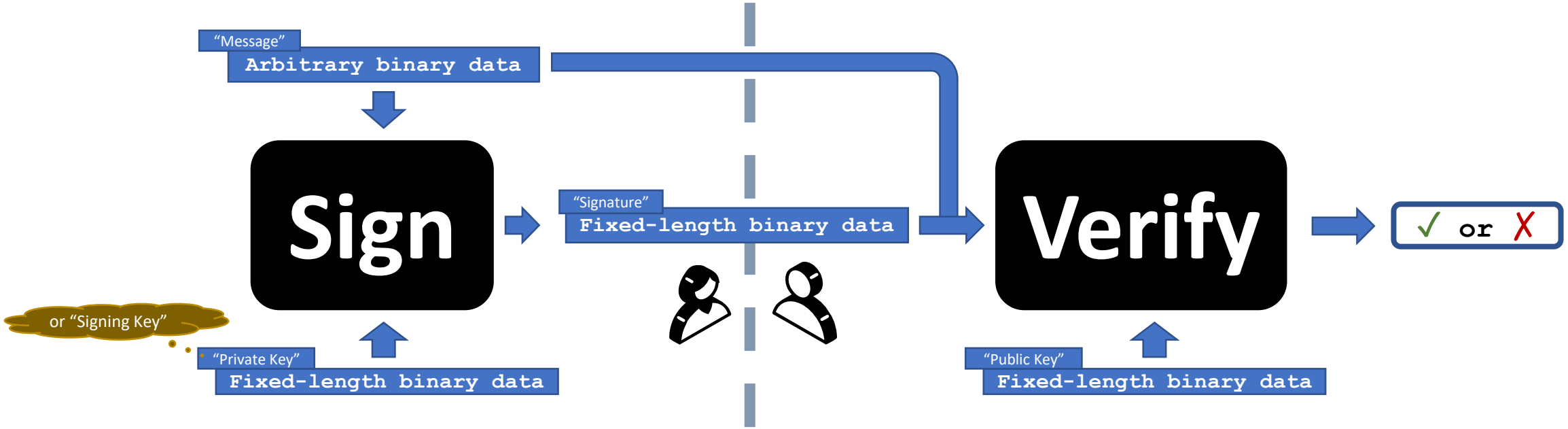
- Asymmetric Authenticity



- No repudiation possible!

Digital Signatures

- Asymmetric Authenticity
  - No repudiation possible!

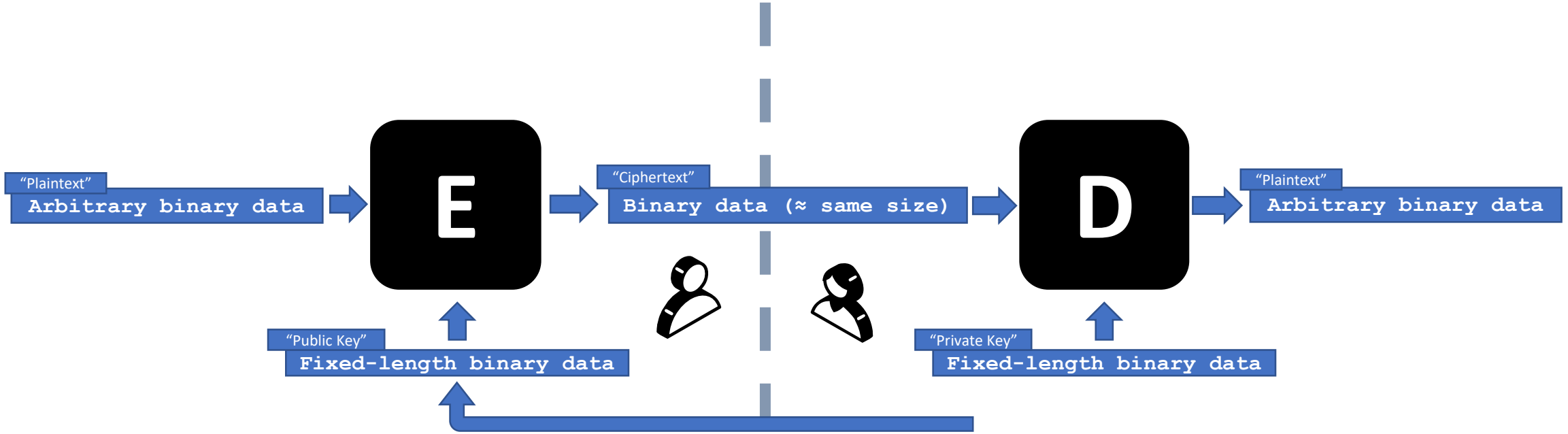


- **Examples:** Ed25519, (deterministic) ECDSA

Asymmetric Encryption

Don't use this directly!
(unless you have to)

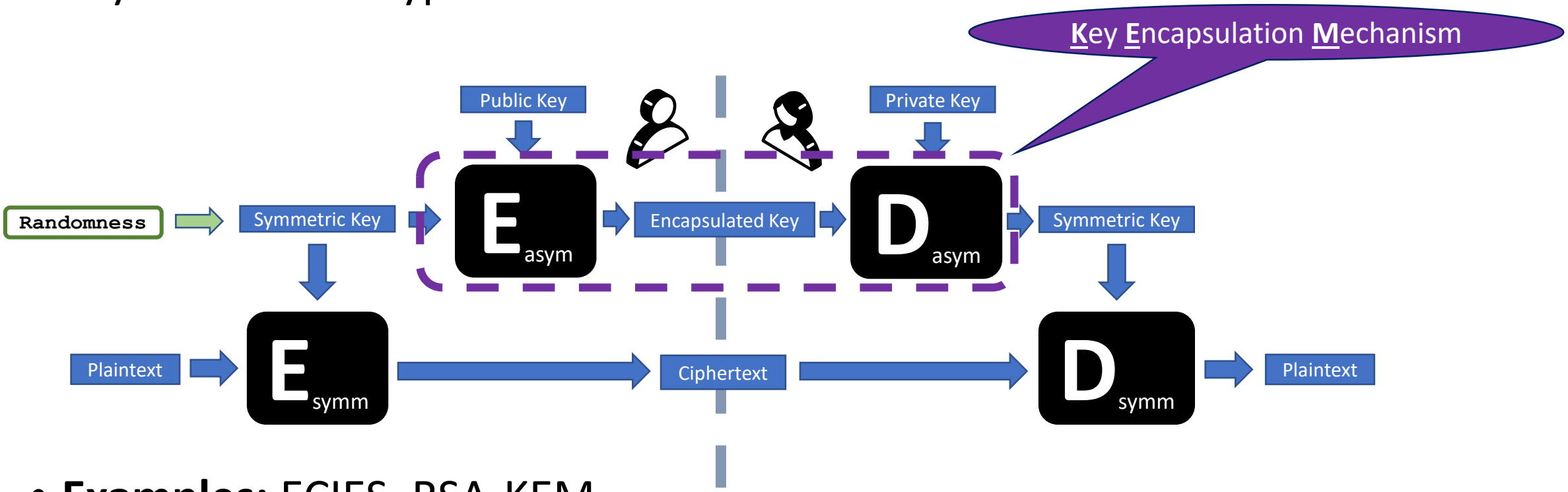
- Very slow processing speed
- Private Key leaks are devastating



This is the one to use!
(unless you have reasons not to)

Hybrid Encryption

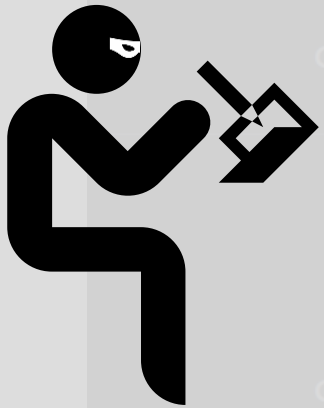
- Asymmetric encryption with a twist!



- **Examples:** ECIES, RSA-KEM
- Private Key leaks are still devastating!

Forward Secrecy

1. Record Google Searches



2. Steal Private Key

3. Profit?

Forward Secrecy

- Attackers might compromise long-lived keys

- **Forward Secrecy** means:
  - Key compromise *after the fact* does not compromise any data

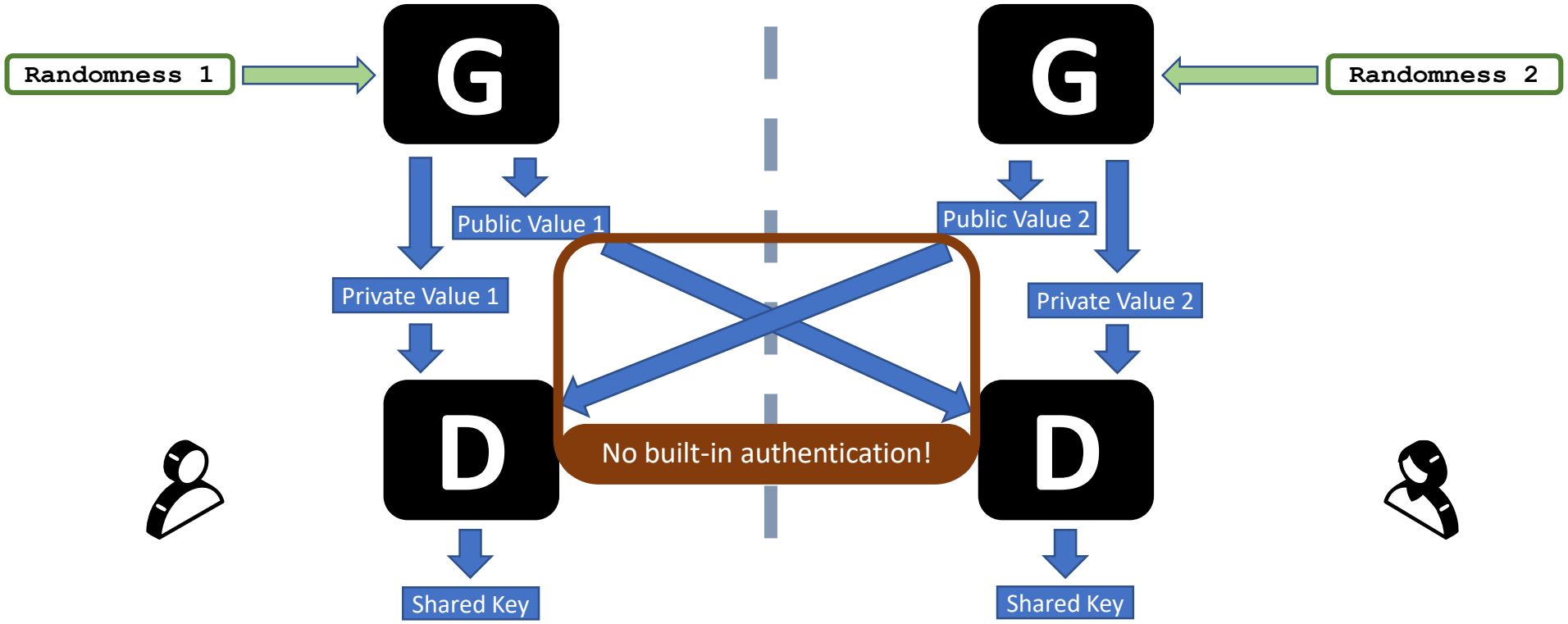
(Ephemeral)

Key Agreement

Generate value pair

Derive shared key

- Shared Secret over Insecure Channel

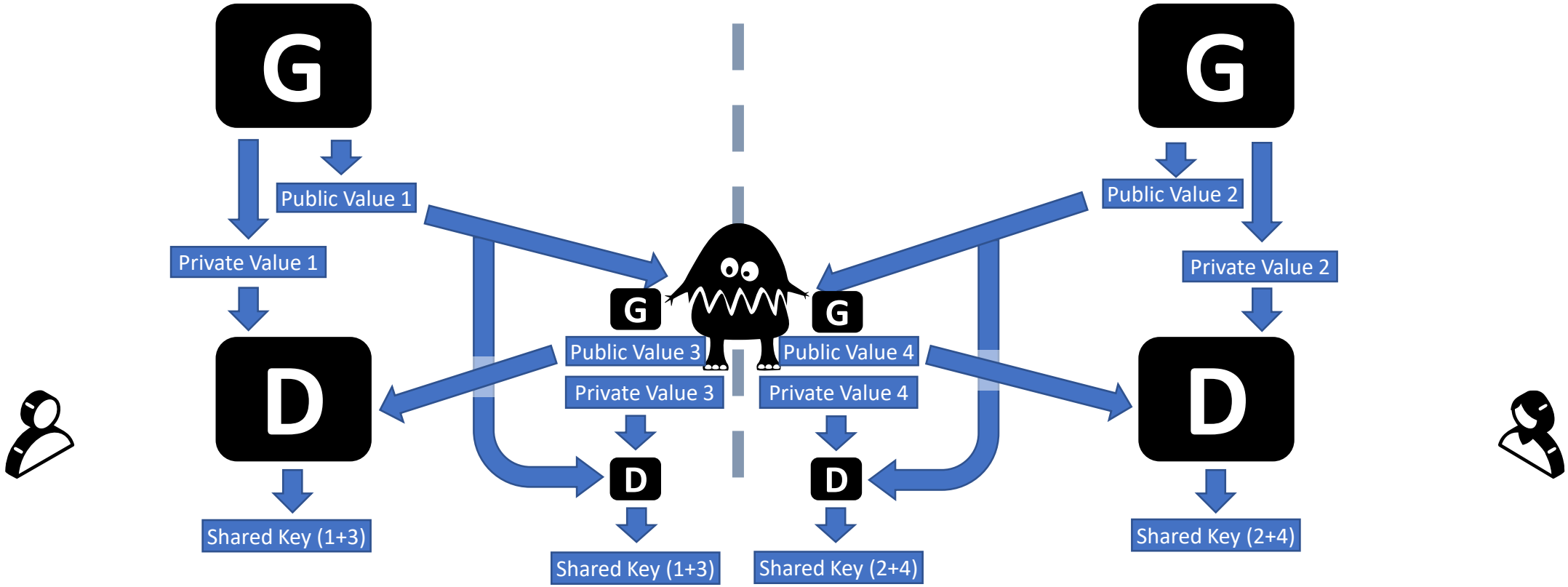


(Ephemeral)

Key Agreement – Monster in the Middle

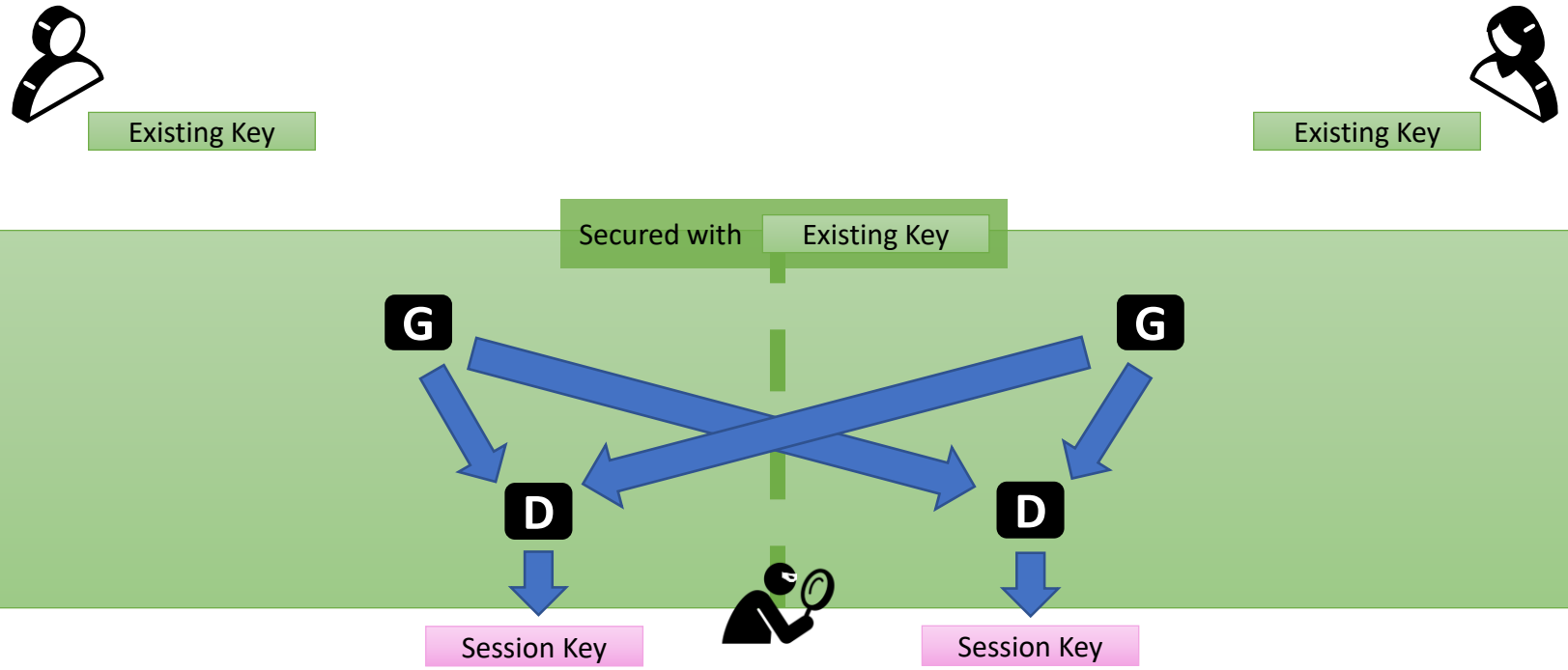
G enerate value pair

D erive shared key



(Ephemeral)

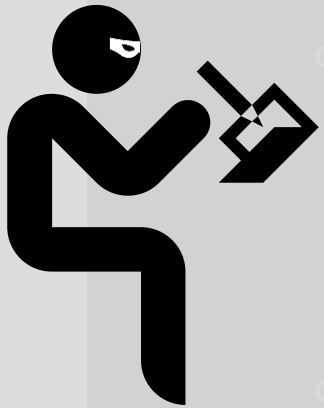
Session Key Agreement



- Compromise of existing key after the handshake?
  - Attacker only sees the public messages of the handshake!

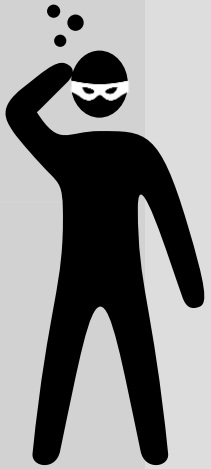
Forward Secrecy!

1. Record Google Searches



2. Steal Private Key

3. See a secure key exchange

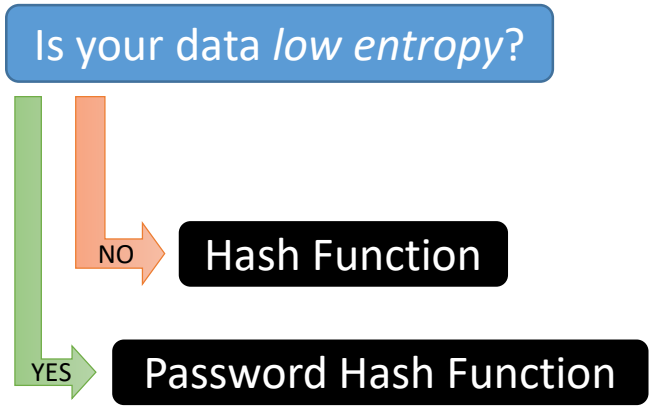


Recap: What Primitive Should I Use?

I want to …

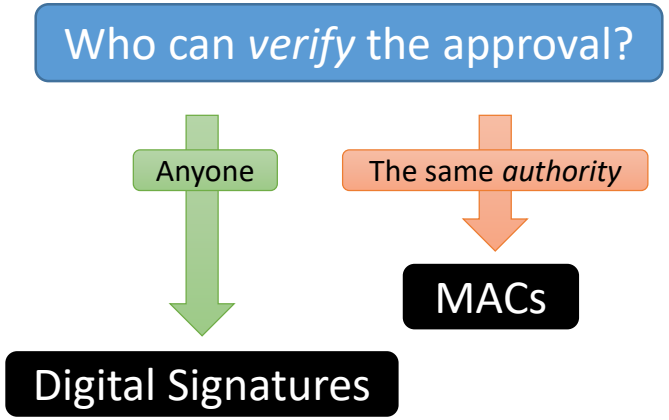
… store a *fingerprint* of the data, but *nobody* can retrieve the original.

(Cryptographic) Hash Functions



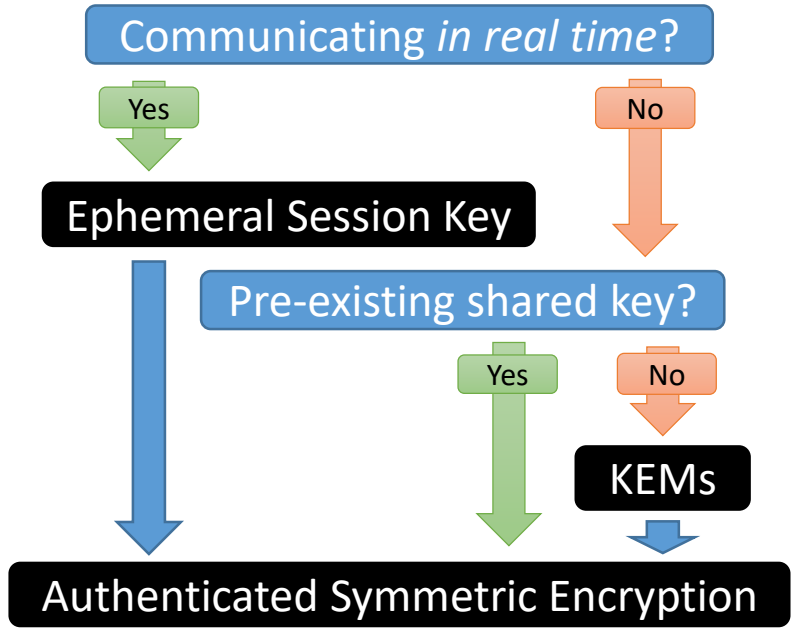
… guarantee that some *authority* has approved the data in question.

Authenticity Primitives



… prevent anyone *except the recipient* from seeing the data.

Some Kind Of Encryption



Even Better: Use Existing Protocols

- TLS
- OAuth
- SAML
- Double Ratchet
- WebAuthn
- and many more…