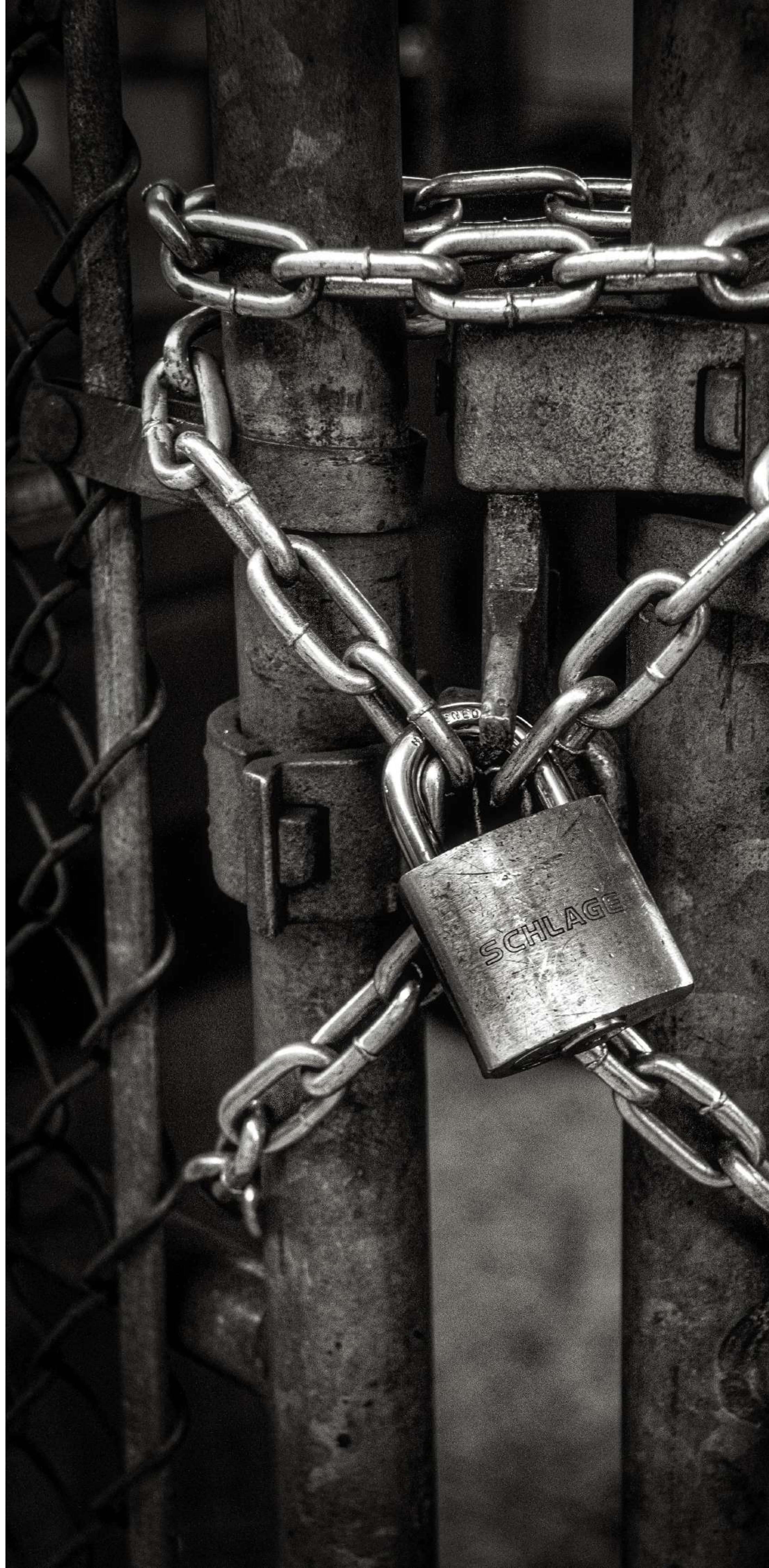
Symmetric Encryption

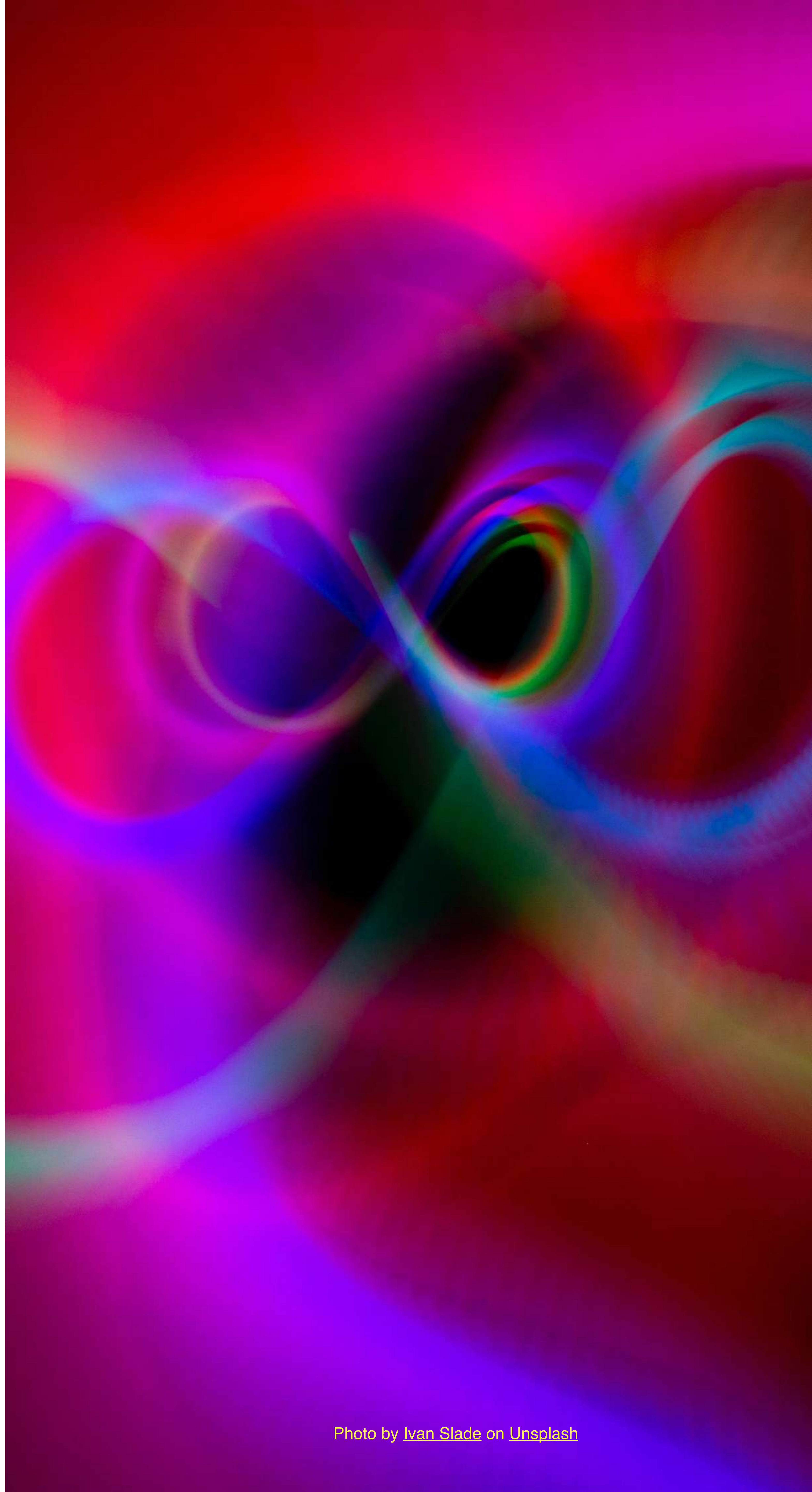
Computational Security

Vanesa Daza



Where do we stand?

- Perfect secrecy: absolutely no information about an encrypted message is leaked.
  - Worthwhile goal, but unnecessarily strong.
  - OTP achieves it.



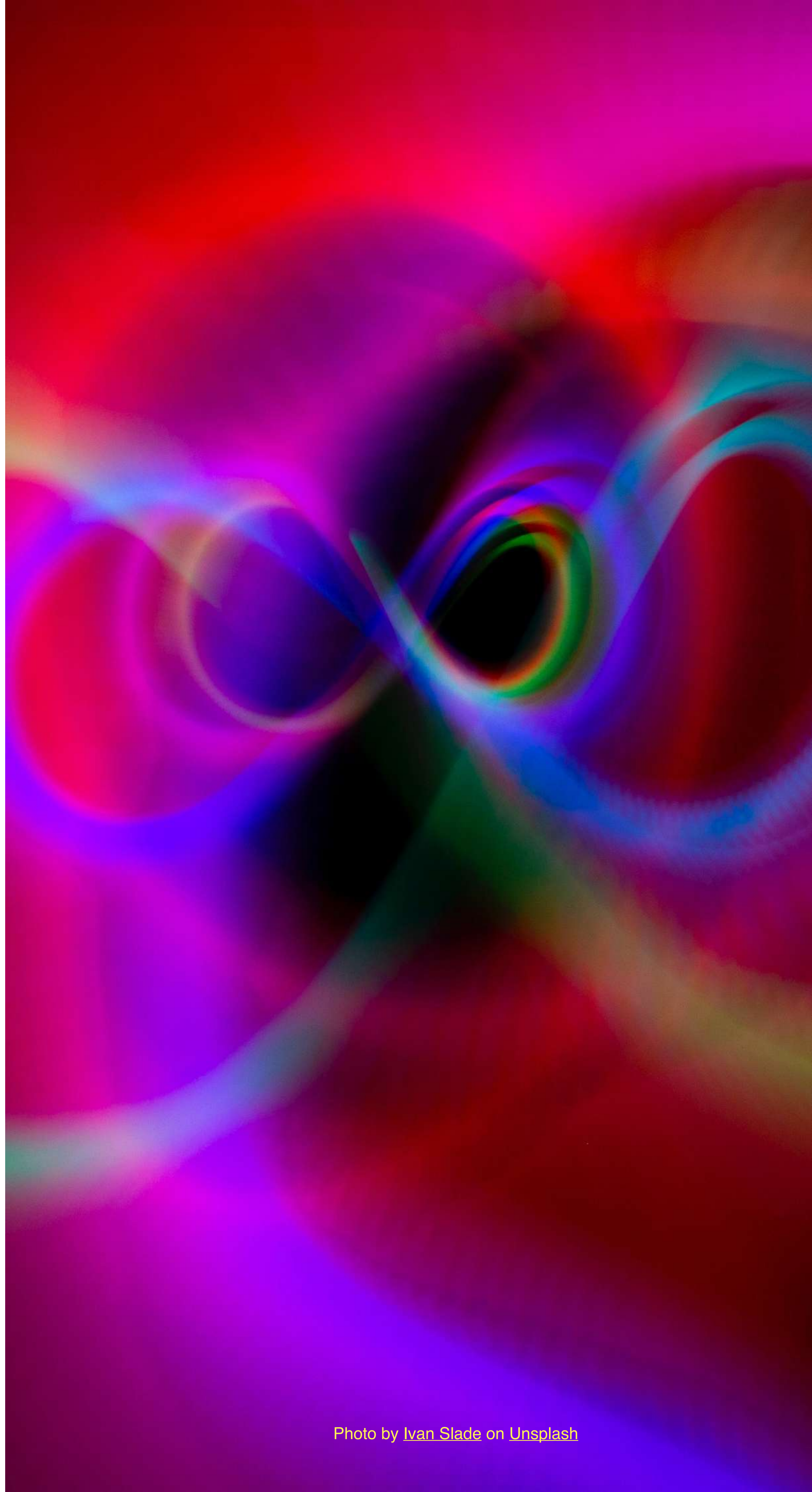
Where do we stand?

- Perfect secrecy: absolutely no information about

an encrypted message is leaked.

  - Worthwhile goal, but unnecessarily strong.
  - OTP achieves it.

Practical purposes: an encryption scheme would be considered **secure** if it leaks information with some tiny probability to eavesdroppers with bounded computational power. **Computational Security.**



Two Relaxations

Relative to Perfect Secrecy

1. Security is only guaranteed against **efficient attacker** that run for some **feasible amount of time**.
    - Schemes is unbreakable if the **resources required** to break the scheme larger than those available to any realistic attacker.
2. Attacker can **potentially succeed** with some tiny probability.

Photo by Chris Thompson on Unsplash

Tiny probabilities

- Say security fails with probability 2^{-60}.

  - Should we be concerned about this?

Tiny probabilities

- Say security fails with probability 2^{-60}.
  - Should we be concerned about this?

probability	*equivalent*
2^{-10}	*full house in 5-card poker*
2^{-20}	*royal flush in 5-card poker*
2^{-28}	*you win this week's Powerball jackpot*
2^{-40}	*royal flush in 2 consecutive poker games*
2^{-60}	*the next meteorite that hits Earth lands in this square* $\rightarrow$

M. Rosulek, The Joy of Cryptography

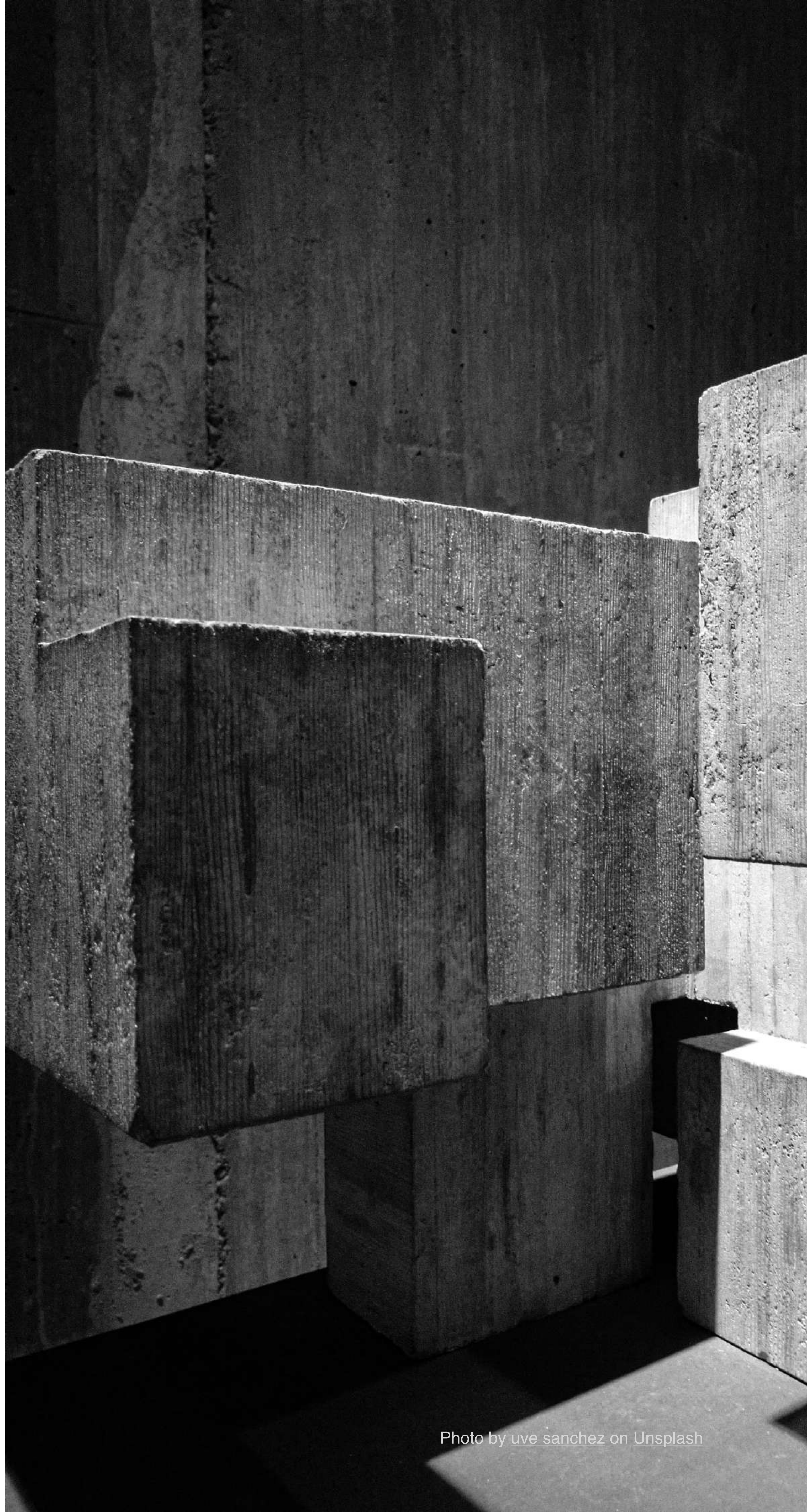
Resources
Monetary Value

clock cycles	*approx cost*	*reference*
2^{50}	*\$3.50*	*cup of coffee*
2^{55}	*\$100*	*decent tickets to a Portland Trailblazers game*
2^{65}	*\$130,000*	*median home price in Oshkosh, WI*
2^{75}	*\$130 million*	*budget of one of the Harry Potter movies*
2^{85}	*\$140 billion*	*GDP of Hungary*
2^{92}	*\$20 trillion*	*GDP of the United States*
2^{99}	*\$2 quadrillion*	*all of human economic activity since 300,000 BC*
2^{128}	*really a lot*	*a billion human civilizations' worth of effort*

M. Rosulek, The Joy of Cryptography

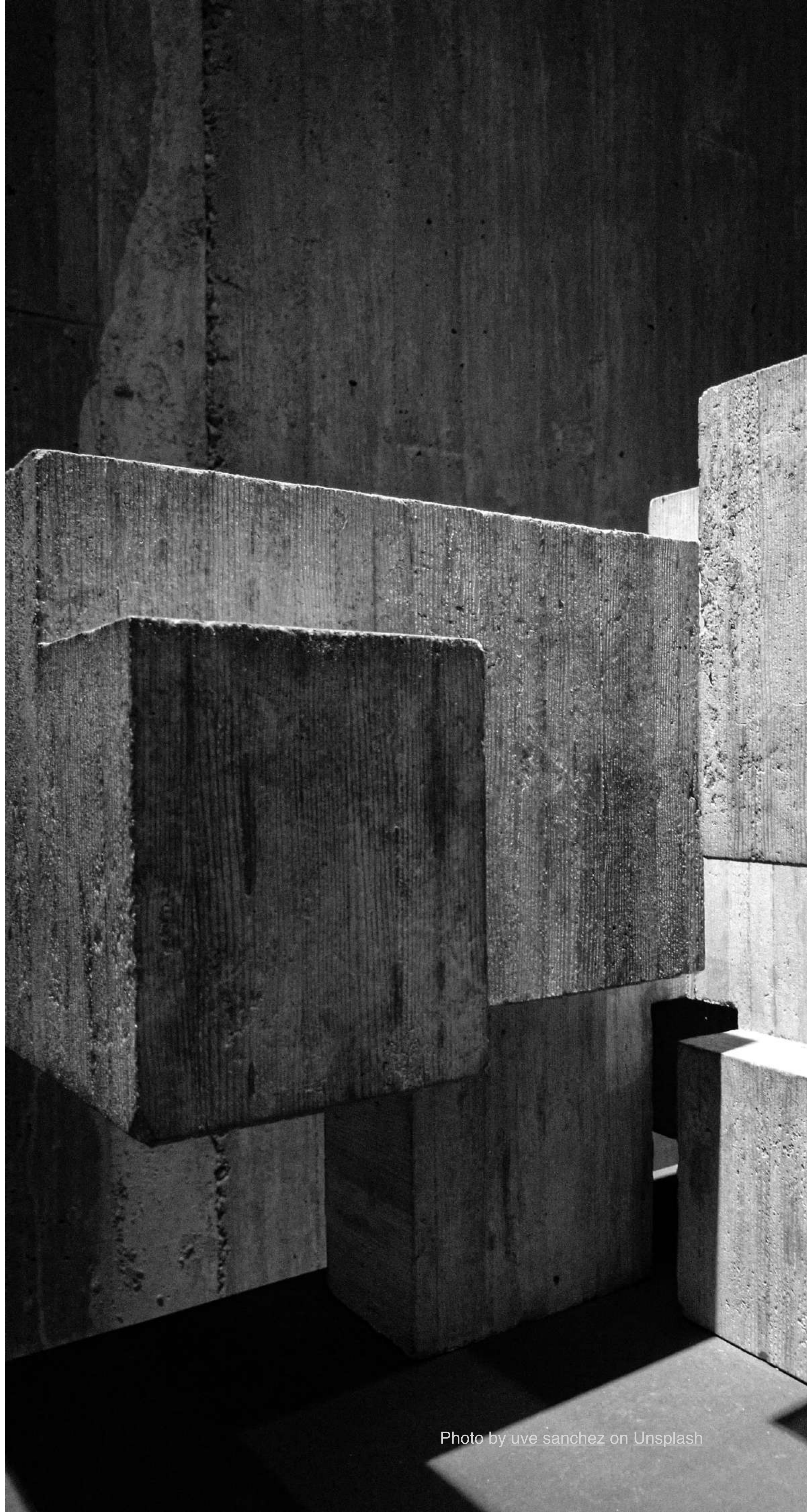
Concrete Approach

- Bounds the maximum success probability of a (randomized) adversary running
  - specified amount of time
  - investing some specified amount of computational effort.



Concrete Approach

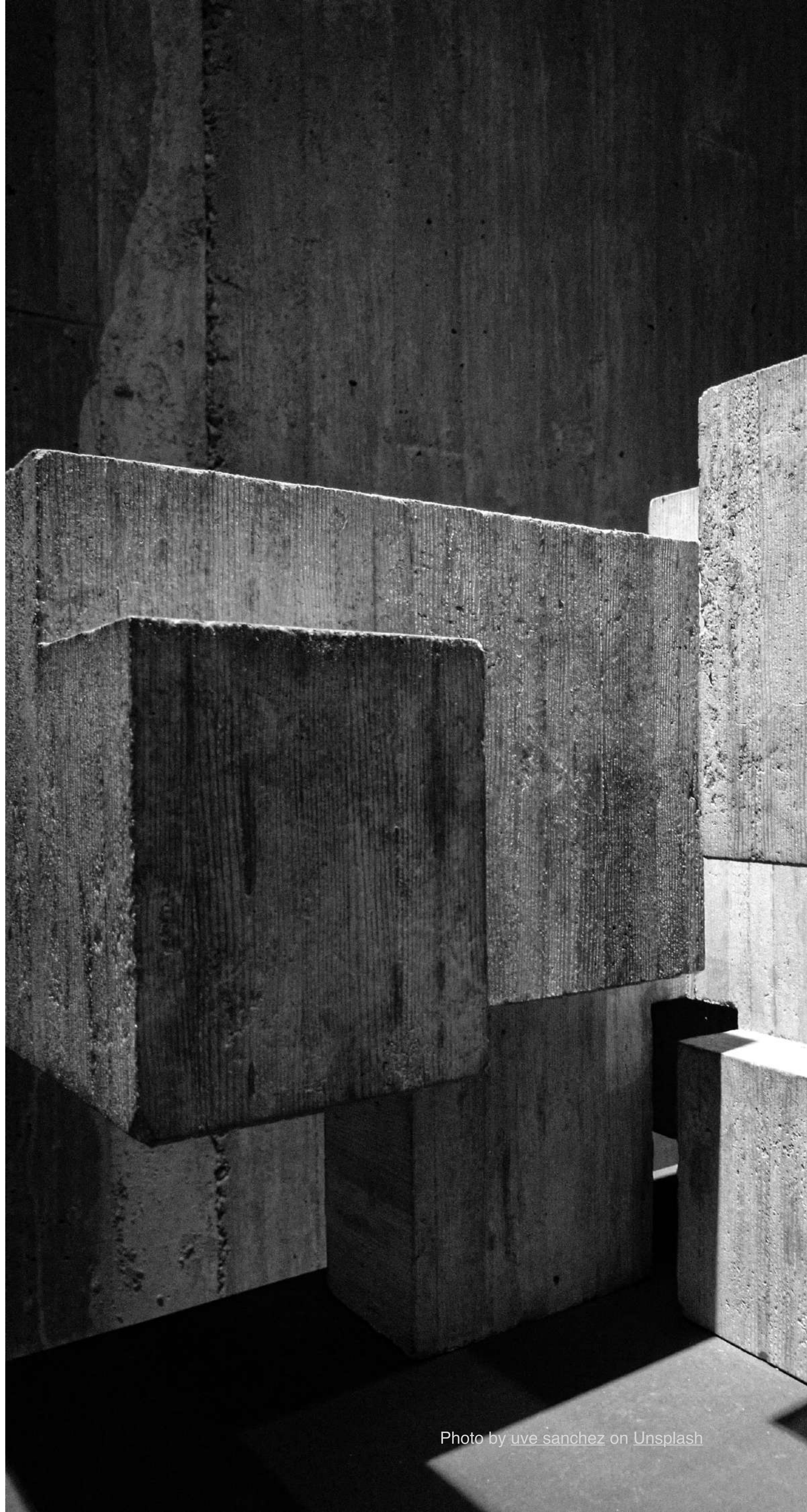
- Bounds the maximum success probability of a (randomized) adversary running
  - specified amount of time
  - investing some specified amount of computational effort.
- Need to define *break the scheme* in question.
- Example: no adversary running for at most 200 years can succeed in breaking the scheme with probability better than 2^{-60}.



Concrete Approach

- Bounds the maximum success probability of a (randomized) adversary running
  - specified amount of time
  - investing some specified amount of computational effort.
- Need to define *break the scheme* in question.
- Example: no adversary running for at most 200 years can succeed in breaking the scheme with probability better than 2^{-60}.

Note: Large times, small probabilities



The Asymptotic Approach

- **Security Parameter:** value that parameterizes both cryptographic schemes as well as all involved parties (honest parties and attacker).
- When using a scheme, a security parameter is chosen.
- Functions of the security parameter
  - Running time of the adversary,
  - Success probability

Photo by Sergey Shmidt on Unsplash

Two important concepts

Polynomial time and negligible probability

- "Efficient adversaries" = randomized algorithms running in **polynomial time** in the security parameter.
  - PPT: there is some polynomial p such that the attacker runs for time at most $p(n)$ when the security parameter is n.
  - "Small probabilities of success" = **negligible probability.**
  - A function is negligible if for every polynomial p there is an N such that for all $n > N$ it holds that $f(n) < \frac{1}{p(n)}$.
  - Or if for all p, $\lim_{\lambda \to \infty} p(\lambda) f(\lambda) = 0$.

Photo by Jens Jakob on Unsplash

Asymptotic Security

Definition Security

Practical purposes: an encryption scheme would be considered **secure** if it leaks information with some tiny probability to eavesdroppers with bounded computational power. **Computational Security.**

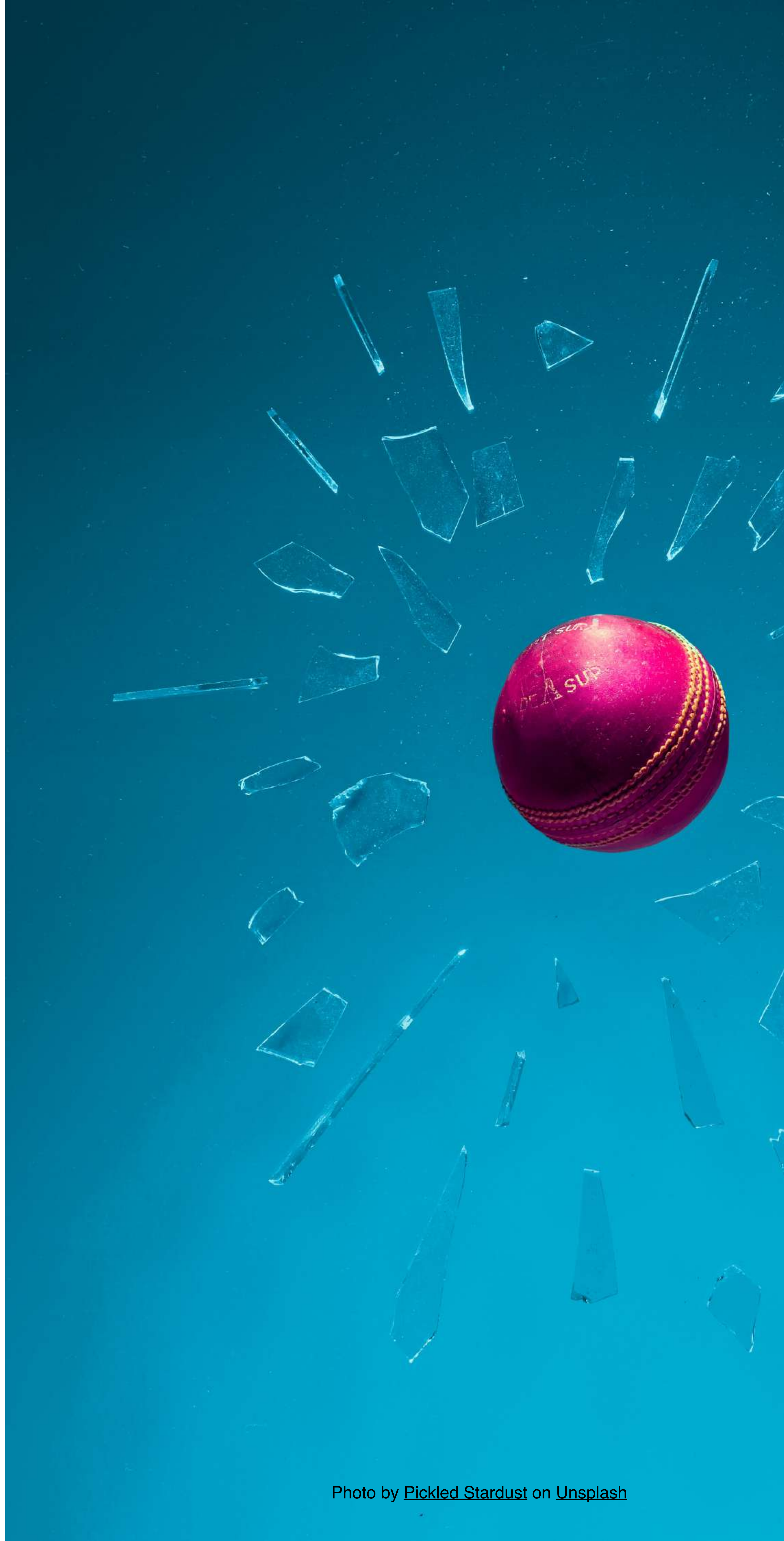


Photo by Pickled Stardust on Unsplash

Asymptotic Security

Definition Security

Practical purposes: an encryption scheme would be considered **secure** if it leaks information with some tiny probability to eavesdroppers with bounded computational power. **Computational Security.**

A scheme is **secure** if any PPT adversary succeeds in breaking the scheme with at most negligible probability.

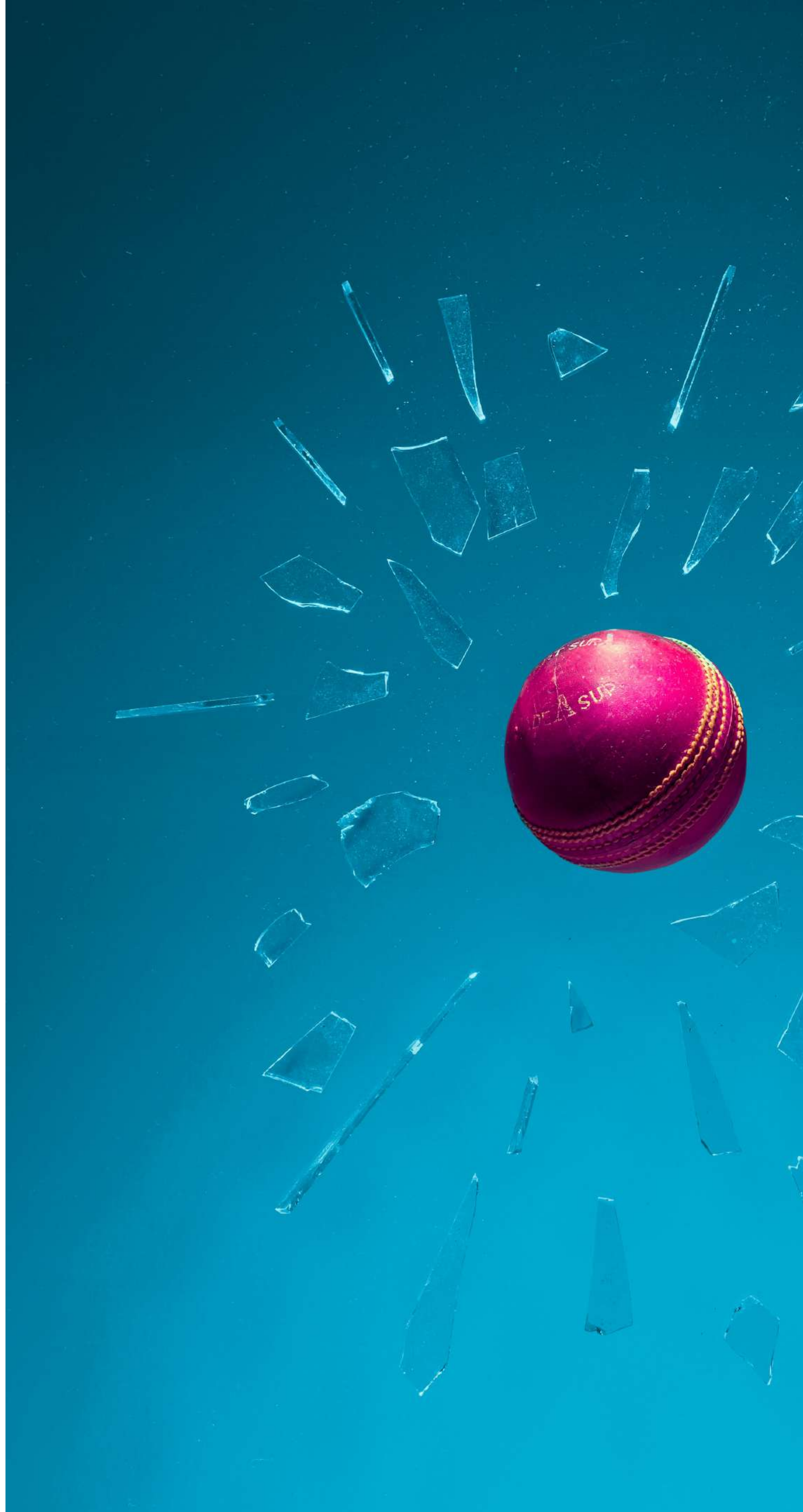


Photo by Pickled Stardust on Unsplash

Security Level

Definition 1.3 *A cryptographic scheme has* n-bit security *if the best known attack requires* 2^n *steps.*

When the best known attack is a brute-force attack, then $n = \lambda$, but we will see many examples of the opposite, which makes n significantly smaller. In a few lessons, we will see the example of hash functions, for which, in the best case,

$$n = \frac{\lambda}{2}.$$

Security Level

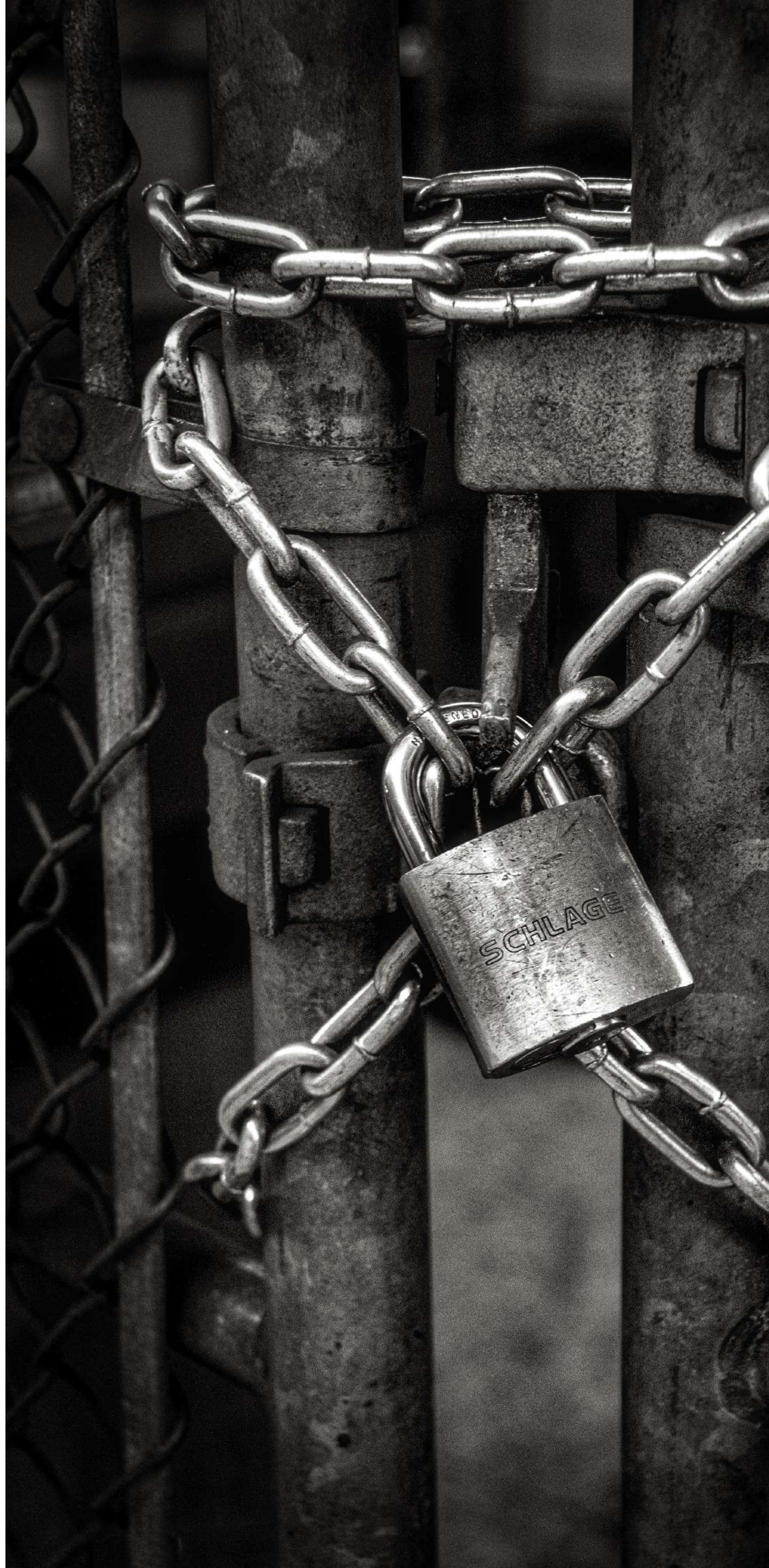
Definition 1.3 *A cryptographic scheme has* n-bit security *if the best known attack requires* 2^n *steps.* When the best known attack is a brute-force attack, then $n = \lambda$, but we will see many examples of the opposite, which makes n significantly smaller. In a few lessons, we will see the example of hash functions, for which, in the best case,

$$n = \frac{\lambda}{2}.$$

If we require a security level of 80 bits, this forces us to choose $\lambda = 160$, at the least. Another example is RSA, which is a famous encryption scheme that we will study later in the course. In that case, λ needs to be 1024 to achieve a security level of roughly 80 bits.

Symmetric Encryption

Computational Security



Vanesa Daza