What is cryptography?

Goal: to construct systems and prove their security

What is a system and what does security mean?

Simplest answer: a system is a set of (efficiently computable) algorithms. *This is not 100% rigorous!

Cryptosystem: a set (Gen, Enc, Dec) where

$Gen: \mathbb{N}^+ \to PK \times SK$

- probabilistic map from λ to key pairs (pk, sk) of length λ

$Enc: PK \times M \to C$

- probabilistic map where M is the message space and C is the ciphertext space

$Dec: SK \times C \to M$

- probabilistic and possibly partial map

such that if $(pk, sk) \leftarrow Gen(\lambda)$, then $\forall m \in M$, $Dec(sk, Enc(pk, m)) = m$.

What about security? For many years research used games.
Let the adversary A and challenger C be interactive probabilistic algorithms. A game takes an interaction transcript and outputs a bit indicating who won. We will see one way to understand public key security as a game. The IND-CPA game $G^{\lambda,S}_{IND\text{-}CPA}(A)$ works as follows: $(S = (Gen, Enc, Dec), \lambda \in \mathbb{N}^+)$

1) C runs $(pk, sk) \leftarrow Gen(\lambda)$ and sends pk to A
2) A chooses $m_0, m_1 \in M$ (however it likes) and sends them to C
3) C chooses $b \leftarrow \{0,1\}$ unif. at random and sends $Enc(pk, m_b)$ to A
4) A outputs $b^* \in \{0,1\}$.

The game outputs 1 (i.e. A wins) if $b = b^*$.

The idea is that an adversary can always encrypt messages, but should still not be able to tell the difference between encryptions of two possible messages. (Textbook RSA does not satisfy this.)

Def$^{\underline{n}}$ 1

A function $f: \mathbb{N} \to \mathbb{R}$ is negligible if for all $c \in \mathbb{R}_{>0}$,

$$\lim_{n \to \infty} f(n) \cdot n^c = 0$$

Def$^{\underline{n}}$ 2 * Note: asymmetric cryptosystems are "nicer", mathematically

A cryptosystem $S = (Gen, Enc, Dec)$ is IND-CPA secure if for all probabilistic polynomial-time (PPT) algorithms A, $\left| \Pr\left[G^{\lambda, S}_{IND\text{-}CPA}(A) = 1\right] - \frac{1}{2} \right| \in negl(\lambda)$

(probability w/ A as the random variable) (because the adversary could always just guess)

Some details I glossed over: what exactly is a probabilistic interactive algorithm? We'll return to this in future weeks; for now, think of it as a Turing machine with a "randomness tape" and some way to "interact" with other TMs.

Definitions like the above work but have some limitations.

- Composition: If I use an IND-CPA cryptosystem to share a symmetric key that is also IND-CPA, is the overall system secure?

- Strength: the security guarantee depends entirely upon the (somewhat artificial) game we defined. Is it too weak — can I actually trust an IND-CPA system? Or perhaps it's too strong, and a weaker guarantee suffices in real life.

In this seminar series we will explore these two questions through a framework called universal composability, so-named because its theorems guarantee that security guarantees compose universally, i.e. in any environment. We introduce UC next week. For the rest of today I will motivate the above points in more detail.

The one-time pad works as follows:

- $Gen(\lambda)$: sample $k \leftarrow \{0,1\}^\lambda$ uniformly at random, output k * symmetric
- $Enc(k, m) = k \oplus m$ * bitwise XOR (+ mod 2)
- $Dec(k, c) = k \oplus c$

Correctness is easy to verify. We say it achieves perfect security because given $k \sim Gen(\lambda)$, for all $m_0, m_1 \in \{0,1\}^\lambda$ and all $c \in \{0,1\}^\lambda$,

$$\Pr[Enc(k, m_0) = c] = \Pr[Enc(k, m_1) = c]$$

In the symmetric version of IND-CPA, the adversary wins exactly half of the time.

Quantum key exchange is more complicated, but allows two parties to agree on $k \in \{0,1\}^\lambda$ such that when all parties measure their states, the adversary has zero information about k.

If I combine these, I should be able to communicate with total secrecy and no pre-shared key, right? Wrong! The QKE security game does not consider what happens if the adversary waits to measure their state _after_ messages are sent using OTP. The security guarantees do not compose!

In real-world applications, an adversary can often obtain the decryption of a ciphertext it controls. For example, they may send an encrypted message to a server, which either gives an encrypted reply or an error code (if the message was "invalid", e.g. as a HTTP request). This is the source of the infamous Bleichenbacher attack against RSA.

The IND-CPA game does not guarantee security in this case. We need a stronger game.

The IND-CCA game $G^{\lambda,S}_{IND\text{-}CCA}(A)$ works as follows:

1) C runs $(pk, sk) \leftarrow Gen(\lambda)$ and sends pk to A.
2) A sends $c \in C$ to C, which responds with $m = Dec(sk, c)$. This repeats as many times as A likes.*
3) A chooses $m_0, m_1 \in M$ and sends them to C.
4) C chooses $b \in \{0,1\}$ uniformly at random and sends $\hat{c} = Enc(pk, m_b)$ to A.
5) Step 2 repeats, except C only responds to queries where $c \neq \hat{c}$.
6) A outputs b^*.

$G^{\lambda,S}_{IND\text{-}CCA}$ outputs 1 if $b = b^*$ and 0 otherwise.

Def^n 3

A cryptosystem $S = (Gen, Enc, Dec)$ is IND-CCA secure if for all PPT adversaries A,

$$\left| Pr\left[G^{\lambda,S}_{IND\text{-}CCA}(A) = 1 \right] - \frac{1}{2} \right| \in negl(\lambda)$$

Problem: This is too strong! Take any CCA system and add a random bit to the ciphertext that is ignored in decryption; now the system is spuriously insecure.

There are intermediate notions of security that try to address this, but none are standard yet. We will try a different approach next week to handle both this issue and also composability.

---

* Technically, this requires some sort of polynomial bound.

Summary:

System: set of efficiently-computable algorithms w/ correctness property

Security: for now, defined by the prob. an adversary algorithm can win a game

(Asymmetric) cryptosystem: satisfies either IND-CPA or IND-CCA