How to Keep a Secret

Eleanor McMurtry

May 2020

1 How to keep a secret

What does it mean for a cryptosystem to be *secure*?

Definition 1. A *cryptosystem* is three algorithms:

- Gen which generates keys (symmetric or asymmetric)
- Enc which encrypts a message m to produce a *ciphertext* $\{m\}$
- Dec which decrypts a ciphertext $\{m\}$ to recover the *plaintext* m.

In an ideal world, a cryptosystem would be *impossible* to break. We will imagine an *adversary* $\mathcal{A}$, which is an algorithm that attempts to recover m from $\{m\}$ without knowing any required secret keys.

Here's what we would *like*:

Definition 2. A cryptosystem is *information-theoretic secure* if for all adversaries $\mathcal{A}$, $\mathcal{A}(\{m\}) \neq m$

Intuitively, "information-theoretic secure" means that any adversary, even with an infinite amount of time and computational power, cannot recover the plaintext — because it *does not have enough information* about m to do so. Unfortunately, it turns out that distributing a public key breaks this requirement:

Proposition 1.1. *A public key cryptosystem cannot be information-theoretic secure.*

Back to the drawing board. We imagine the adversary as playing a *game* with a challenger $\mathcal{C}$: $\mathcal{C}$ gives $\mathcal{A}$ the public key with a ciphertext $\{m\}$, and $\mathcal{A}$ *guesses* what m is.

Definition 3. An *adversary* $\mathcal{A}$ is a probabilistic polynomial time algorithm that aims to break a cryptosystem.

We require that $\mathcal{A}$ is polynomial time because we have conceded that if $\mathcal{A}$ had enough time to e.g. brute force the private key (an exponentially-hard problem), it could break our encryption.

It should however be *very unlikely* that $\mathcal{A}$ can guess the plaintext. Here's what that means:

Definition 4. A function $f : \mathbb{N} \to \mathbb{R}$ is *negligible* if for all polynomials $P(x)$, there exists $N > 0$ such that for all $n > N$,

$$|f(n)| < \frac{1}{P(n)}$$

The idea is that f is much smaller than any polynomial, so it goes to 0 very quickly. Here's an example of a *game* based on the not-very-good Caesar cipher:

Definition 5. Consider the Caesar cryptosystem described below for an alphabet of N letters:

- Gen: choose a random integer $c \in \mathbb{Z}$
- Enc: take each letter in m and shift it c letters along. That is, $\{m\} = m + c \mod N$.
- Dec: take each letter in $\{m\}$ and shift it $-c$ letters along. That is, $m = \{m\} - c \mod N$.

The adversary $\mathcal{A}$ plays a game with a challenger $\mathcal{C}$ as follows:

1. $\mathcal{C}$ runs Gen to obtain c.
2. $\mathcal{C}$ chooses a message m and encrypts it to obtain $\{m\}$.
3. $\mathcal{C}$ sends $\{m\}$ to $\mathcal{A}$.
4. $\mathcal{A}$ outputs a message m^*.

$\mathcal{A}$ wins if $m = m^*$. The Caesar cipher is *secure* if for all PPT adversaries $\mathcal{A}$,

$$\Pr[\mathcal{A} \text{ wins}] < \mathsf{negl}(N)$$

where $\mathsf{negl}(N)$ is some negligible function.

Proposition 1.2. *The Caesar cipher is not secure.*

Proof. Let $\mathcal{A}$ choose a random $c^* \in \mathbb{Z}$, and output $\{m\} - c^* \mod N$. With non-negligible probability $1/N$, $c = c^*$, and A outputs m. □

This is a fairly basic idea of "secure", and it turns out we can do a little better. We will require *indistinguishability*: it should be hard for $\mathcal{A}$ to guess the difference between two ciphertexts. If $\mathcal{A}$ can break the encryption outright, the game is easy — it can just decrypt the ciphertexts. So, if $\mathcal{A}$ *can't* tell the difference, it also can't decrypt the ciphertexts.

Definition 6. The *indistinguishability game* for a cryptosystem Π, written G^{Π}_{ind}, runs as follows:

1. $\mathcal{C}$ runs Gen to obtain a key.
2. $\mathcal{A}$ sends $\mathcal{C}$ two messages: m_0 and m_1.
3. $\mathcal{C}$ flips a coin $b \in \{0, 1\}$ and sends $\{m_b\}$ to $\mathcal{A}$.
4. $\mathcal{A}$ outputs a bit b^*.

$\mathcal{A}$ wins if $b = b^*$.

We say Π is *indistinguishable-secure* if for all PPT adversaries $\mathcal{A}$,

$$\Pr\left[\mathcal{A} \text{ wins } G^{\Pi}_{\text{ind}}\right] < \frac{1}{2} + \mathsf{negl}(\lambda)$$

for a *security parameter* λ (e.g. key length).

In a public key setting, $\mathcal{A}$ can encrypt m_0 and m_1 itself, and see whether either matches $\{m_b\}$. This is called a *chosen plaintext attack*, and breaks "textbook RSA" — RSA used without random padding.

In this setting, $\mathcal{C}$ gives $\mathcal{A}$ the algorithm Enc (including the public key) in step 1. A cryptosystem that is indistinguishable-secure under this condition is called *IND-CPA secure*. To prove this property, we will use *proof by reduction*: we will show that *if* $\mathcal{A}$ can win $G^{\Pi}_{\text{IND-CPA}}$, *then* it can win another game that we assume is hard.

Below is a common such assumption — it forms the basis of the Diffie-Hellman and ElGamal schemes.

Definition 7 (The Decisional Diffie-Hellman game)**.** Let $\mathbb{G}$ be a cyclic group, and g be an element of (prime) order q. The game $G^{\mathbb{G},g,q}_{\text{DDH}}$ is defined as follows:

1. $\mathcal{C}$ chooses $a, b, c \leftarrow \mathbb{Z}_q$ uniformly at random and calculates
$$x_0 = g^c \qquad\qquad x_1 = g^{ab}$$
2. $\mathcal{C}$ flips a coin $i \in \{0, 1\}$ and sends g^a, g^b, and x_i to $\mathcal{A}$.
3. $\mathcal{A}$ outputs a bit i^*.

$\mathcal{A}$ wins if $i = i^*$. We say that *the DDH assumption holds in* $\mathbb{G}$ if there exists g, q such that for all PPT adversaries $\mathcal{A}$,

$$\Pr\left[\mathcal{A} \text{ wins } G^{\mathbb{G},g,q}_{\text{DDH}}\right] < \frac{1}{2} + \mathsf{negl}(q)$$

The idea: $\mathcal{A}$ shouldn't be able to easily tell the difference between g^{ab} and g^c. If it could compute discrete logarithms, then it could easily win — and that's supposed to be hard.

We are, at last, ready to prove that ElGamal is IND-CPA secure. The proof below mirrors that of [1].

Theorem 1.1. *If the DDH assumption holds in $\mathbb{G}$, the ElGamal cryptosystem over $\mathbb{G}$ is IND-CPA secure.*

Proof. Let $\mathcal{A}$ be a PPT adversary that wins $G^{\text{ElGamal}}_{\text{IND-CPA}}$ with probability $\frac{1}{2} + \varepsilon(q)$.

Consider an adversary $\mathcal{D}$ that attacks DDH. It receives input g^a, g^b, x and acts as a challenger to $\mathcal{A}$, but instead of Enc it gives $\mathcal{A}$ access to $\mathsf{Enc}_{\mathcal{D}}(m) = \left(g^b, m \cdot x\right)$. If $\mathcal{A}$ wins, $\mathcal{D}$ outputs 1; otherwise, $\mathcal{D}$ outputs 0.

- **Case 1:** If $x = g^c$, $\mathcal{D}$ needs to output 0 to win. $\mathcal{A}$ receives the "ciphertext"

  $$\left(g^b, m \cdot g^c\right)$$

  This is not a valid encryption so $\mathcal{A}$ can do no better than guessing, and wins with probability $\frac{1}{2}$. Then $\mathcal{D}$ wins with probability $\frac{1}{2}$.

- **Case 2:** If $x = g^{ab}$, $\mathcal{D}$ needs to output 1 to win. $\mathcal{A}$ receives the ciphertext

  $$\left(g^b, m \cdot g^{ab}\right)$$

  This is a valid encryption of m with private key a and random factor b, so $\mathcal{A}$ wins with probability $\frac{1}{2} + \varepsilon(q)$, meaning that $\mathcal{D}$ outputs 1 and also wins with probability $\frac{1}{2} + \varepsilon(q)$. But since the DDH assumption holds in $\mathbb{G}$,

  $$\Pr\left[\mathcal{D} \text{ wins } G^{\mathbb{G},g,q}_{\text{DDH}}\right] = \frac{1}{2} + \varepsilon(q) \leq \frac{1}{2} + \mathsf{negl}(q)$$

  so that $\varepsilon(q) \leq \mathsf{negl}(q)$.

$\square$

References

[1] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography*. Chapman and Hall/CRC, 2014.