CMSC 28400 Introduction to Cryptography **Autumn 2021**

Notes #10: Chosen-Ciphertext Security for Encryption

Instructor: David Cash

These notes present and briefly study a definition for *chosen-ciphertext security* of an encryption scheme. To begin understanding the motivation for this definition, consider a setting where a server S_1 holding a key k is connected to the internet, and expects to talk to another computer S_2 that also has k. But by its very design the internet infrastructure is *unauthenticated*: Bits are delivered to hosts by routers, and those bits could claim to be from anywhere. So when our server S_1 receives a packet containing a ciphertext c with the IP address indicating it came from S_2, it can't depend on the IP address alone to ensure the packet is actually from S_2.

Now suppose an adversary controls a router near S_1, and injects a packet that looks like it came from S_2. What can S_1 do? In reality it performs some sort of *cryptographic authentication*, which we discuss in the next notes, but for now let us approach the question naively. Since S_1 has no idea if the packet is "real" or not, *it must run the decryption algorithm on the received ciphertext c*. Perhaps the decryption algorithm returns garbage, or perhaps c has been crafted to return something more controlled. In any case, S_1 *must react to c in some way that is observable to the adversary.* Perhaps S_1 closes the connection, or maybe it attempts to send an error code, or maybe it even sends the next message in a protocol.

In summary, the ability to inject one or more ciphertexts c gives an adversary the power to carry out attacks fitting the following template:

1. Cook up some ciphertext $\hat{c}$.
2. Have a receiver run $\hat{m} \leftarrow \mathsf{Dec}(k, \hat{c})$.
3. Observe how the receiver reacts to the message $\hat{m}$.

These are called *chosen-ciphertext attacks*, or CCAs. It's hard to intuit if CCAs like this could ever be damaging. Let's look at an example that highlights how things might go catastrophically wrong.

Example 10.1. *Let $\Pi = (\mathsf{Enc}, \mathsf{Dec})$ be the (deterministic) OTP. Suppose a server uses Π to decrypt messages it receives. Suppose further that if the server decrypts a ciphertext c and obtains a message m that does not have the correct format (e.g. is not proper HTTP), then it replies with m in the clear to the sender and requests a retransmission.*

Here is a CCA that recovers the secret key from the server:

1. *Set $\hat{c}$ to a random string, and send it to the server.*
2. *The server computes $\hat{m} \leftarrow \mathsf{Dec}(k, \hat{c})$. It's likely that $\hat{m} = k \oplus \hat{c}$ will not be a properly-formatted message, so the server replies with $\hat{m}$.*
3. *The attacker computes $k = \hat{c} \oplus \hat{m}$.*

The above example is extreme, but bugs like this have been known to leak keys; One problem in Project 2 gives a more practical version.

So we want a definition that ensures CCAs won't be damaging. We certainly want that keys won't be leaked, but we'd like to ensure even more, akin to the guarantees we got from CPA security. Thus we will aim for a left/right indistinguishability definition like CPA, but we need to give the adversary some power similar to the template above. The types of reactions a server might have to a ciphertext $\hat{c}$ are quite wide, and we can't hope to enumerate them. Instead, we'll go completely overkill, and *give the adversary the ability to decrypt ciphertexts as it likes.* We do this formally with another oracle. That way, since the adversary can learn how any $\hat{c}$ to decrypts to some $\hat{m}$, it can also learn about any possible reaction a server might have. And if a scheme resists these attacks, then it should resist more realistic attacks fitting the template above.

There is one wrinkle, however: We must stop the adversary from using its decryption ability to trivially win the experiment. After all, if the adversary is trying to learn what message was encrypted (m_0 or m_1), decryption of arbitrary ciphertexts will allow it easily figure that out. Thus we need an extra rule saying "free win" ciphertexts are not allowed to be decrypted.

The formal definition follows. It fits the template we've had so far, but with an extra oracle.

Definition 10.1. *Let* $\Pi = (\mathsf{Enc}, \mathsf{Dec})$ *be a randomized encryption scheme with key-space* $\mathcal{K}$*, message-space* $\mathcal{M}$*, randomness-space* $\mathcal{R}$*, and ciphertext-space* $\mathcal{C}$*. Assume that* $\mathcal{M} \subseteq \{0,1\}^*$*. Let* $\mathcal{A}$ *be an algorithm. Define algorithm* $\mathbf{Expt}^{\mathrm{cca}}_{\Pi}(\mathcal{A})$ *as*

```
Alg $\mathbf{Expt}^{\mathrm{cca}}_{\Pi}(\mathcal{A})$
01 Pick $k \xleftarrow{\$} \mathcal{K}, b \xleftarrow{\$} \{0,1\}$
02 Run $\mathcal{A}^{\mathrm{LR}_{k,b}(\cdot,\cdot),\mathsf{Dec}(k,\cdot)}$, where the first oracle is given below. Eventually $\mathcal{A}$ halts with output $\hat{b}$
03 If $\mathcal{A}$ ever queried $\mathsf{Dec}(k,\cdot)$ at a ciphertext $c$ previously output by $\mathrm{LR}_{k,b}(\cdot,\cdot)$: Output $\perp$.
04 If $\hat{b} = b$: Output 1
05 Else: Output 0

Oracle $\mathrm{LR}_{k,b}(m_0, m_1)$
    If $m_0, m_1$ are not the same length: Return $\perp$
    Pick $r \xleftarrow{\$} \mathcal{R}$
    Compute $c \leftarrow \mathsf{Enc}(k, m_b, r)$
    Return $c$
```

Define the CCA advantage of $\mathcal{A}$ against Π *as*

$$\mathbf{Adv}^{\mathrm{cca}}_{\Pi}(\mathcal{A}) = \left| \Pr[\mathbf{Expt}^{\mathrm{cca}}_{\Pi}(\mathcal{A}) = 1] - \frac{1}{2} \right|$$

This definition is exactly the CPA definition, except for the $\mathsf{Dec}(k,\cdot)$ oracle and for line 03, which enforces the "free win" rule (see the next paragraph). Recall that with an oracle, an algorithm can submit whatever it likes, and submit as many queries as it likes (bounded only by its runtime). Thus an adversary above can ask for decryptions of many ciphertexts (millions), each of which it gets to form. They could be malformed, all-zeros, or whatever else might help a break. The idea is that all of the decryptions should not help the adversary break any of the "challenge" ciphertexts from the LR oracle.

Note that the game is easy to win if we omit the rule on line 03; We leave this as an exercise to check.

Exercise 10.1. *Suppose the definition above omitted line* 03*. Assume* Π *has more than one allowed message, i.e.* $|\mathcal{M}| > 1$*. Give an efficient adversary* $\mathcal{A}$ *such that*

$$\mathbf{Adv}^{\mathrm{cca}}_{\Pi}(\mathcal{A}) = 1/2.$$

Building encryption from a block cipher that resists chosen-ciphertext attacks is considerably harder than chosen-plaintext attacks. In fact, we won't even try in these notes and will defer it to the next set. The following example is meant to highlight the challenge.

Example 10.2. *Let* $E : \{0,1\}^n \times \{0,1\}^\ell \to \{0,1\}^\ell$ *be a block cipher. Define a randomized encryption encryption scheme* $\Pi = (\mathsf{Enc}, \mathsf{Dec})$ *with key-space* $\{0,1\}^n$*, message- and randomness-spaces* $\mathcal{M} = \mathcal{R} = \{0,1\}^\ell$*, and ciphertext-space* $\mathcal{C} = \{0,1\}^\ell \times \{0,1\}^\ell$ *by*

$$\mathsf{Enc}(k, m, r) = (r, E(k, r) \oplus m)$$

and $\mathsf{Dec}(k, (r, c)) = E(k, r) \oplus c$*.*

Previously we showed that breaking the CPA-security of Π *required breaking the PRF-security of* E*. We now give an efficient adversary* $\mathcal{A}$ *that strongly breaks the CCA-security of* Π*, satisfying*

$$\mathbf{Adv}^{\mathrm{cca}}_{\Pi}(\mathcal{A}) = 1/2.$$

This shows the construction is not at all secure against chosen-ciphertext attack. The adversary works follows:

```
Adversary $\mathcal{A}^{\mathrm{LR}_{k,b}(m_0,m_1),\mathsf{Dec}(k,\cdot)}$
    Let $m_0 \neq m_1 \in \mathcal{M}$, with $m_0 \neq 0^\ell$. Query $(r, c) \leftarrow \mathrm{LR}_{k,b}(m_0, m_1)$.
    Let $c' = c \oplus m_0$. Query $m' \leftarrow \mathsf{Dec}(k, (r, c'))$
    If $m' = 0^\ell$: Output 0
    Else: Output 1.
```

Let us check that the adversary always win $\mathbf{Expt}^{\mathrm{cca}}_{\Pi}(\mathcal{A})$*. First,* $\mathcal{A}$ *will never auto-lose the game, because* (r, c) *is never queried to the decryption oracle (only* (r, c') *is, but* $(r, c') \neq (r, c)$ *because* $c' \neq c$*, which is where we use that* $m_0 \neq 0^\ell$*). Now suppose* $b = 0$*, so* $(r, c) = (r, E(k, r) \oplus m_0)$*. Then the ciphertext* (r, c') *will decrypt as*

$$c' \oplus E(k, r) = c \oplus m_0 \oplus E(k, r) = E(k, r) \oplus m_0 \oplus m_0 \oplus E(k, r) = 0^\ell,$$

and $\mathcal{A}$ *will output* 0*. Otherwise, if* $b = 1$*,* (r, c') *will decrypt as*

$$c' \oplus E(k, r) = c \oplus m_0 \oplus E(k, r) = E(k, r) \oplus m_1 \oplus m_0 \oplus E(k, r) = m_1 \oplus m_0 \neq 0^\ell,$$

because $m_0 \neq m_1$*. Thus* $\mathcal{A}$ *will output* 1*. This shows that* $\mathcal{A}$ *always wins and thus has the desired advantage.*

Intuitively, the problem is *malleability of ciphertexts*: The adversary $\mathcal{A}$, without knowing the key, is able to modify a ciphertext c into a new ciphertext c' containing a message related to the original one. This new ciphertext can be submitted for decryption (because it is different), and the response can be used to win the experiment.

Don't overthink the next exercise: It's trivial.

Exercise 10.2. *Let* Π *be an encryption scheme. Show that for all* $\mathcal{A}$ *there exists a* $\mathcal{B}$ *running in the same time such that*

$$\mathbf{Adv}_{\Pi}^{\mathrm{cpa}}(\mathcal{A}) \leq \mathbf{Adv}_{\Pi}^{\mathrm{cca}}(\mathcal{B}).$$

How are CCA and CPA security related?

Unfortunately, CTR and CBC modes are not CCA-secure. We'll need new tools instead.

Exercise 10.3. *Let* Π *be either AES-CTR or AES-CBC, defined in the previous notes. For each one, find an efficient* $\mathcal{A}$ *such that*

$$\mathbf{Adv}_{\Pi}^{\mathrm{cca}}(\mathcal{A}) = 1/2.$$