Disclaimer. Draft note. No guarantee on completeness nor soundness. Read with caution, and shoot me an email at fsong@pdx.edu for corrections/comments (they are always welcome!)

Logistics. DRC: note taker request 2nd round. New course in Spring'17: Intro to quantum computing. Clarification on HW 5 a), keyed functions, and oracle.
Last time. Block ciphers, PRFs, PRPs.
Today. Chosen-Plaintext Attacks, CPA-secure encryption.

Quick notes. HW 5a). F does not need to be length-preserving. The notion of PRF applies more generally. A keyed function $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ is a collection of functions indexed by the keys. Namely $F = \{F_k : \{0,1\}^* \to \{0,1\}^*\} \subseteq \mathscr{F} := \{f : \{0,1\}^* \to \{0,1\}^*\}$, where each F_k is a fixed function. Therefore picking a random key k is equivalent to sampling a function uniformly at random from this collection. Although it's only a small subset of all possible functions, we require that no efficient distinguisher can tell a difference from the setting where we sample a function uniformly from all possible functions.

Oracle: i.e., black-box; no one sees the inside, can only access input-output interface.

1 Stronger security for encryption: CPA

Limits of computational secrecy (against eavesdropping). So far we've been dealing with a rather weak security definition of encryption - computational secrecy. Namely an adversary only passively eavesdrops on a single ciphertext. This means that to encrypt multiple messages, we would generate a fresh key independently for each message. This will be very inefficient and difficult to manage. It would be nice to encrypt multiple messages under one key. However, this is not guaranteed by computational secrecy. In fact, many of the schemes we've seen will be broken completely if we reuse the key. For example

- Stream cipher: $E_k(m) := G(k) \oplus m$. This suffers from the same issue of OTP. $c \oplus c' = m \oplus m'$.
- ECB for one block: $E_k(m) := F_k(m)$. Not secure for two or more blocks.

If we think about it, the fundamental problem here is that the encryption algorithms are *deterministic*, therefore if you see two identical ciphertexts, you will know for sure the underlying plaintexts are *identical*. This contradicts (at least) our intuitive security goal that seeing a ciphertext should not reveal any additional information about the plaintext. All this can be made formal and it is worth stating in a Theorem. But I will just call it *multi-message secrecy* without giving a formal definition for computational secrecy for multiple messages here (Cf. [KL: Def. 3.19]). It's enough to let the intuitive idea guide you for now, and we will be more precise in a second when talking about CPA-security.

Proposition 1 (KL-Prop. 3.20, Thm. 3.21. Necessity of random encryption for multi-message secrecy)**.** *There is a private-key encryption scheme that is computationally secret (in the presence of an eavesdropper) but is* ***NOT*** *multi-message secret. In fact, any scheme that has a* deterministic *encryption algorithm cannot be multi-message secret.*

1.1 Chosen-Plaintext Attacks

It is time to strengthen our security definition. Recall two components of a security definition: security goal and attack model. We consider a stronger type of attacks: *chosen-plaintext attacks* (CPA).

Here we assume an adversary has control, at least partially, over what messages the users encrypt. Consider two honest parties, sharing a key and applies some encryption scheme to protect their communication. We assume that an adversary could somehow issue a bunch of messages m_i of his/her choice and then get the corresponding ciphertexts c_i by listening into the channel. We would hope that such information should not help the adversary to decipher any other ciphertext.

FS NOTE: draw communication channel and CPAttacks

Is CPA feasible?. You'd wonder, is this giving the attacker too much power? Well, it is actually realistic or even common. The most popular example of CPA you will often hear about is probably the stories during WW II. For instance, The British placed mines at certain locations and (intentionally) managed to let the Germans discover them. Sounds dumb? Well, they knew that the Germans would encrypt the locations and send back to the headquarters. These ciphertexts were exploited at Bletchley Park to break the Enigma machine. A classic CPA indeed!

Let's look at a modern example. We use Google everyday, and your search traffic is all encrypted. This includes in particular one important object on a search page! The one that Google makes great revenue from – *ads*.Then if an attacker has some fund at hand, he/she could place any ads on Google and obtain the ciphertexts from Google.

Without further delay, we will discuss formally secure encryption against CPA.

1.2 Defining CPA-security

How do we model the ability of getting encryptions of messages chosen by the adversary? We've seen a similar scenario: hand the encryption algorithm with a secret key $E_k(\cdot)$ as an oracle to the adversary. Let $\Pi = (G, E, D)$ be an encryption scheme.

Definition 2 (KL-Def. 3.22)**.** Π is *CPA-secure* or has *indistinguishable encryptions under a chosen-plaintext attack*, if for any PPT adversary $\mathcal{A}$ it holds that

$$\Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n) = 1] \leq 1/2 + \mathrm{negl}(n).$$

Remark 3. $\mathcal{A}$ can make queries adaptively, i.e., ask new queries based on what s/he sees so far. After receiving the challenge cipher, s/he still has access on $E_k(\cdot)$. So the queries can depend on c and any prior knowledge. Now you should see there is no way that a deterministic encryption can achieve CPA-security!

We introduce CPA-security for the purpose of encrypting multiple messages. Let's be more precise about what we want. We change the CPA-indistinguishability game and define multi-message CPA indist. game $\mathsf{PrivK}^{\mathsf{mult\text{-}cpa}}_{\mathcal{A},\Pi}(n)$:

FS NOTE: Draw cpa diagram

1. CH generates key $k \leftarrow G(1^n)$.
2. Adversary $\mathcal{A}$ is given 1^n and oracle access to $E_k(\cdot)$ (i.e., $\mathcal{A}$ can make queries m_i and obtain $E_k(m_i)$).
3. $\mathcal{A}$ outputs a pair of messages $m_0, m_1 \in \mathcal{M}$ with $|m_0| = |m_1|$.
4. CH picks uniform $b \leftarrow \{0,1\}$, and computes *challenge ciphertext* $c \leftarrow E_k(m_b)$. c is given to $\mathcal{A}$.
5. $\mathcal{A}$ continues to access $E_k(\cdot)$ and outputs a bit b' in the end.
6. Define the output of the experiment $\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n) = 1$ if $b' = b$, and we say $\mathcal{A}$ succeeds in this case. Otherwise define $\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n) = 0$.

Figure 1: CPA indistinguishability experiment $\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n)$

- $\mathcal{A}$ generates two tuples of messages $\vec{m}_0 = (m_0^1, \ldots, m_0^t)$ and $\vec{m}_1 = (m_1^1, \ldots, m_1^t)$ for some $t \geq 1$. We require that $|m_0^j| = |m_1^j|$ for all j.
- CH picks random $b \leftarrow \{0,1\}$, and computes $\vec{c} := (E_k(m_b^1), \ldots, E_k(m_b^t))$.

Similarly, we define the experiment output $\mathsf{PrivK}^{\mathsf{mult\text{-}cpa}}_{\mathcal{A},\Pi}(n) = 1$ if $b' = b$, and $\mathsf{PrivK}^{\mathsf{mult\text{-}cpa}}_{\mathcal{A},\Pi}(n) = 0$ otherwise.

Definition 4 (KL-Hybrid of Def. 3.19 & 3.23)**.** Π is *multi-message CPA-secure* or has *indistinguishable multiple encryptions under a chosen-plaintext attack*, if for any PPT adversary $\mathcal{A}$

$$\Pr[\mathsf{PrivK}^{\mathsf{mult\text{-}cpa}}_{\mathcal{A},\Pi}(n) = 1] \leq 1/2 + \mathrm{negl}(n).$$

Note that our definition of multi-message CPA security subsumes CPA-security as a special case. It turns out that our effort so far has not been in vain.

Theorem 5 (KL-Thm. 3.24)**.** *CPA-security* $\equiv$ *multi-message CPA-security.*

Its proof will be deferred (if at all). Intuitively, if one can tell apart $E_k(\vec{m}_0)$ and $E_k(\vec{m}_1)$, then there must be a pair (m_0^j, m_1^j) such that $\mathcal{A}$ can tell apart $E_k(m_0^j)$ and $E_k(m_1^j)$, contradicting CPA-security of the scheme.

This immediately gives some advantages of CPA-security: we don't have to explicitly worry about multi-message security, since it follows from single-message CPA-security. We can also get an encrytion for messages of arbitrary length from an encryption that can only encrypt messages of a fixed length, say just one bit. $E'_k(m) := E_k(m_1) \| \ldots \| E_k(m_\ell)$ is CPA-secure. 1-BIT ENCRYPTION IS COMPLETE FOR CPA.

1.3 Constructing CPA-secure ciphers

How to get CPA-secure schemes? By the simple rule, we know that none of stream ciphers, and two modes of block ciphers, ECB and Deterministic Counter mode, can be CPA-secure. The other two modes of operations of block ciphers we discussed, RCTR and CBC, they are randomized (do you recall where the randomness is introduced?), and they are indeed CPA-secure. Read the proofs if interested.

Here we give a simple construction from any PRF with a proof of CPA-security. Again, think about our good old friend OTP and its generalization to stream cipher where we use a pseudorandom string as our pad. We know the output from a PRF would look random, can we just use the output as our pad? But what input we give to PRF? Remember we have to introduce some randomness. How about we evaluate PRF on a random input? Sounds good. Let's write down what we have:

FS NOTE: Draw diagram of encryption alg.

Let $F_k(\cdot): \{0,1\}^n \to \{0,1\}^n$ be a PRF. Construct $\Sigma := (G, E, D)$:

- $G(1^n)$: pick uniform $k \leftarrow \{0,1\}^n$.
- $E_k(m)$: pick uniform $r \leftarrow \{0,1\}^n$ and output $c := (r, F_k(r) \oplus m)$
- $D_k(c)$: parse $c = (r, s)$ and output $m := F_k(r) \oplus s$.

Figure 2: CPA-secure encryption from PRF

Now if you stare at the diagram, I hope you can recognize it! It is nothing but the first iteration of Randomzed Counter Mode. Let's prove that it is CPA-secure, which will warm you up to read/prove RCTR as well as CBC modes.

Theorem 6 (KL-Thm. 3.31)**.** *Σ in Fig. 2 is CPA-secure (for message of length n).*

Idea: imagine F_k being a truly random function, then it will be one-time-pad with independently random key for each message, except when the random string r^* used in generating the challenge ciphertext coincides with some r used to answer $\mathcal{A}$'s queries to $E_k(\cdot)$ (assume $\mathcal{A}$ makes $q(n)$ queries for some polynomial $q(n)$). But since r^* is chosen at random, this occurs with tiny probability $q(n)/2^n$. Therefore the only possibility that an adversary breaks CPA would be by distinguishing F from truly random, but this contradicts F being a PRF.

Proof. Consider a variant of Σ, where we substitude a truly random function for the PRF. Call it $\tilde{\Sigma} = (\tilde{G}, \tilde{E}, \tilde{D})$ (imaginary scheme just for the sake of proof. We cannot implement it efficiently). Then for any adversary $\mathcal{A}$ who makes at most $q(n)$ encryption queries ($q(n)$ must be bounded by some polynomial, why?). We complete the proof in two lemmas:

- Lemma 7 says that $\mathcal{A}$ will succeed in the CPA-indist. game with the same probability whether it's Σ or $\tilde{\Sigma}$, except with negligible discrepency, assuming F is a PRF.

- Lemma 8 shows that when it is $\tilde{\Sigma}$, $\mathcal{A}$ succeeds with probability $1/2 + q/2^n$, only negligibly better than a random guess.

Combining them, we have that $\left|\Pr\left[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Sigma}(n)=1\right]\right| \le 1/2 + \mathrm{negl}(n)$. □

Lemma 7. $\left|\Pr\left[\mathit{PrivK}^{\mathit{cpa}}_{\mathcal{A},\Sigma}(n)=1\right] - \Pr\left[\mathit{PrivK}^{\mathit{cpa}}_{\mathcal{A},\tilde{\Sigma}}(n)=1\right]\right| \le \mathrm{negl}(n)$.

Proof. Notation:

$$p_{\mathcal{A},\Sigma} := \Pr\left[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Sigma}(n)=1\right], \qquad p_{\mathcal{A},\tilde{\Sigma}} \Pr\left[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\tilde{\Sigma}}(n)=1\right]$$
$$p_{D,F_k} := \Pr[D^{F_k}(1^n)=1 : k \leftarrow \{0,1\}^n], \qquad p_{D,f\leftarrow\mathcal{F}} := \Pr[D^f(1^n)=1].$$

This is done by a reduction. We construct a distinguisher D, and show that $|p_{\mathcal{A},\Sigma} - p_{\mathcal{A},\tilde{\Sigma}}| \le |p_{D,F_k} - p_{D,f\leftarrow\mathcal{F}}|$ which is negligible since we assume that F_k is a PRF.

FS NOTE: Draw reduction diagram.

Distinguisher D: given 1^n and oracle $\mathcal{O} : \{0,1\}^n \to \{0,1\}^n$

1. Run $\mathcal{A}(1^n)$. Whenever $\mathcal{A}$ makes encryption query m, answer as follows
   (a) Choose uniform $r \leftarrow \{0,1\}^n$
   (b) Query $\mathcal{O}(\cdot)$ and obtain response $y := \mathcal{O}(r)$.
   (c) Returen ciphertext $c = (r, y \oplus m)$.
2. When $\mathcal{A}$ outputs (m_0, m_1), pick random bit $b \leftarrow \{0,1\}$, and generate c on m_b as above.
3. Continue answering encryption queries as above till $\mathcal{A}$ outputs b'. Output 1 if $b' = b$ and 0 otherwise.

Observe that D runs in polynomial time as $\mathcal{A}$ does.

- If $\mathcal{O}$ is PRF F_k with a random key k. Then $\mathcal{A}$'s view is identical to its view in $\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Sigma}(n)$. Therefore $p_{\mathcal{A},\Sigma} = p_{D,F_k}$.
- If $\mathcal{O}$ is a random function. Then $\mathcal{A}$'s view is identical to its view in $\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\tilde{\Sigma}}(n)$. Therefore $p_{\mathcal{A},\tilde{\Sigma}} = p_{D,f\leftarrow\mathcal{F}}$.

Therefore $\left|\Pr\left[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Sigma}(n)=1\right] - \Pr\left[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\tilde{\Sigma}}(n)=1\right]\right| \le \mathrm{negl}(n)$. □

Lemma 8. $\left|\Pr\left[\mathit{PrivK}^{\mathit{cpa}}_{\mathcal{A},\tilde{\Sigma}}(n)=1\right]\right| \le 1/2 + q(n)/2^n$.

Proof. Skipped. See KL. □