CS 6160 Cryptology Lecture 9: Formalizing notions of security

Maria Francis

September 30, 2020

Computational Security

- In a way, we are going to revisit concepts we learned and then formalize them.
- The concepts we look at in this lecture are:
  1. Computational Security
  2. Concrete Security Vs Asymptotic Security
  3. Semantic Security
  4. Proofs By Reduction
  5. Security for Multiple Encryption
  6. CPA-security
- and later: Modes of Operation for block ciphers, CCA-security, Padding Oracle Attacks
- Reading : Chap 3 of Katz & Lindell (3.1, 3.2, 3.4, 3.6, 3.7)

Computational Security

- Perfect Secrecy requires absolutely no information about the message to be leaked even for an Eve with unlimited computational power.
- Too strong, practically, we only need a scheme to be secure if it leaks only a tiny amount of information to Eves with bounded computational power.
- In practice that would mean for e.g: scheme that leaks with probability $< 2^{-60}$ to Eves that need to invest at least 200 years of computational effort on the fastest available supercomputer.
- Such a security definition is computational and NOT information-theoretic.

Computational Security

- The former allows for computational limits on attacks (Probabilistic Polynomial Time adversaries) and a small probability of failure (negligible chance to succeed)
- NOTE: We do not give up rigorous mathematical approach!We still need proofs and definitions but we rely on weaker notions of security.

Concrete Approach

- Quantified the security of a scheme by explicitly bounding the maximum success probability of any randomized adversary running for some specific time.

Definition

A scheme is (t, ϵ)-secure if any adversary running for time at most t succeeds in breaking the scheme with probability at most ϵ.

- We still have not formally defined what break is for the scheme.
- It could be measured in time like in the previous discussion or in terms of computational effort like CPU cycles: using at most 2^{80} cycles the probability of you breaking the scheme is not better than 2^{-60}.

Concrete Approach - Some Examples

- SKE schemes give optimal security in this sense: for a key length n (or key space 2^n), an adversary running for time t(/ computer cycles) succeeds in breaking it with probability $< ct/2^n$ for some fixed constant c.
- I.e. only a brute force search of the key-space!
- If $c = 1$, $n = 60$ provides adequate security against a desktop computer.
  - 4Ghz processor (4×10^9 cycles/sec), 2^{60} CPU cycles require $2^{60}/(4 \times 10^9)$ secs or 9 years.
  - Supercomputer that executes 2×10^{16} fp op/sec? Only 1 min!
  - But 2^{80} still takes 2 years!
- Recommended $n = 128$, i.e. 2^{48} times $> 2^{80}$. Physicists estimate 2^{58} secs have passed since the Big Bang!

Concrete Approach - Some Examples

- In terms of probability, an event that happens once in every 100 yrs is roughly estimated to occur with probability 2^{-30}
- An event that happens with probability 2^{-60} is even rarer, once in every 100 billion years
- And so if the chances of the attacker succeeding are in the same lines we are pretty safe!
- The concrete approach gives exact values and is important in practice.
- But for a scheme that is just being designed very hard to provide!
- We need to cover details like:
  - Types of computing power
  - Future advances in computing power (Moore's law estimates)
  - Do we assume generic algorithms or dedicated software?

Asymptotic Approach

- When concrete security is not an immediate concern then we use asymptotic approach.
- That is where the security parameter n comes into picture which parameterizes the scheme as well as the involved parties (attacker and honest parties).
- Efficient adversaries have probabilistic/randomized algorithms running in time polynomial in n.
- Honest parties also run in polynomial time but the adversary can run longer and maybe much more powerful.
- As discussed before $\mathrm{negligible}$ probability is $< 1/\mathrm{poly}(n)$.

Definition

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

Asymptotic Approach - Examples

- E.g: An adversary running for n^3 minutes can succeed in breaking the scheme with probability $2^{40} \cdot 2^{-n}$ a negligible function of n.
- For $n \leq 40$ this means an adversary running for 40^3 minutes (6 weeks) can break the scheme with probability 1.
- Not good!
- For $n = 500$, an adversary running for 200 years can break only with probability 2^{-50}. Great!
- Security parameter is a mechanism that allows honest parties to tune the security of a scheme to a level they like.
- Very large n means time to run the scheme is large and the length of the key is large but better security against attacks.

Asymptotic Approach - Examples

- What about faster computers?
- Consider a scheme that can run for $10^6 n^2$ cycles for honest parties and an adversary running for $10^8 n^4$ cycles can succeed in breaking the scheme with probability at most $2^{-n/2}$.
- Say all parties have 2Ghz computers and $n = 80$.
- Honest parties run for $10^6 6400$ cycles (3.2 sec) and an adversary running for $10^8 (80)^4$ cycles (3 weeks) can break with probability 2^{-40}.
- For 8 Ghz computers we can make $n = 160$ and still honest parties can maintain 3.2 sec running time but adversary has to run over 13 weeks to achieve success probability of 2^{-80}.
- The effect of faster computers made the adversary job harder. But then you assumed honest parties also got faster computers!

Asymptotic Approach - details

- Asymptotic approach cannot be used when you are actually deploying the scheme, you need concrete security then.
- But asymptotic approach can be translated to concrete security for any desired value of the security parameter.
- Recall, security parameter is given a unary representation, i.e. n is represented as 1^n.
- Probabilistic algorithms that may consider the outcome of tossing a coin in each step is what we assume all algorithms to be.
- Why? Randomness is inherent everywhere, e.g: when we choose a key.
- And two because we believe that this additional power is something we can assume for realistic attacks.

Asymptotic Approach - details

- Negligible function to indicate the chance of succeeding.

Definition

A function $f : \mathbb{N} \to \mathbb{R}^+$ is negligible or negl if for every positive polynomial p there is an N s.t. for all integers $n > N$ it holds that $f(n) < 1/p(n)$.

- I.e, for every polynomial p and all sufficiently large values of n $f(n) < 1/p(n)$.
- Examples: $2^{-n}, 2^{-\sqrt{n}}$, $n^{-log\ n}$.
- Results:
  1. $\mathrm{negl}_1(n) + \mathrm{negl}_2(n)$ is negligible,
  2. For any positive poly p, $p(n) \cdot \mathrm{negl}_1(n)$ is negligible.
- Last one implies the negligible chance of succeeding does not get better even if the adversary repeats the attack polynomial number of times.

Asymptotic Approach - details

- The previous result also gives rise to this observation: if g is not negligible then neither is $f(n) = g(n)/p(n)$ for any positive polynomial p.
- The advantage of using PPT algorithms:
  1. All reasonable models of computation are polynomially equivalent. So we need not specify whether we have to use TMs, boolean circuits or random-access machines.
  2. Closure properties: polynomial calls to a poly-time subroutine will itself run in poly time.

Definition of Security

- We first look at security against single message encryption, i.e. security against a ciphertext-only attack where the adversary can observe only a single ciphertext.
- Threat model: What are the powers of the adversary?
  - Eavesdropping computationally bounded adversary, only listens in
- What about adversary's strategy?
  - Typically, adversary should be unable to learn any partial information about the plaintext from the ciphertext.
- Semantic Security formalizes this idea in computationally secure encryption.
- An equiv. definition indistinguishability is simpler to look at.
- Remember the assigment question which gave an indistinguishability equiv. definition of perfect secrecy!

Indistinguishability with an eavesdropper

- We look at an experiment in which an PPT adversary $\mathcal{A}$ outputs two messages m_0, m_1.
- $\mathcal{A}$ is given an encryption of one of those messages using a uniform key.
- The security of a scheme Π is defined as :if no $\mathcal{A}$ can determine which is the message that was encrypted with probability negligibly greater than 1/2, equiv. to a random guess.
- $PrivK^{eav}_{\mathcal{A},\Pi}(1^n)$: experiment with security parameter n and output $= 1$ indicates $\mathcal{A}$ succeeds in identifying which message was encrypted.
- Adversary should first output two messages m_0, m_1 of equal length. So we do not require our scheme to hide the length of the plaintext.

Indistinguishability experiment $PrivK^{eav}_{\mathcal{A},\Pi}(1^n)$

1. $\mathcal{A}$ is given input 1^n, it outputs m_0, m_1 s.t. $|m_0| = |m_1|$.
2. Running key-gen algorithm we get a key k, and $b \in \{0, 1\}$ is chosen. Ciphertext $c \leftarrow Enc_k(m_b))$ is given to $\mathcal{A}$. It is called *challenge ciphertext*.
3. $\mathcal{A}$ outputs a bit $b^{'}$.
4. If $b = b^{'}$ output 1, else 0. If $PrivK^{eav}_{\mathcal{A},\Pi}(1^n) = 1$, then $\mathcal{A}$ succeeds.

EAV secure

- $\mathcal{A}$ can only eavesdrop is implicit from the fact that its input is limited to a single ciphertext and there is no further interaction.
- How do this experiment come in the picture of security definitions?

Definition

A SKE $\Pi = (Gen, Enc, Dec)$ has indistinguishable encryptions in the presence of an eavesdropper or is EAV-secure if for all probabilistic polynomial-time adversaries $\mathcal{A}$,

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

Equiv. def: every PPT adversary behaves the same whether it is encryption of m_0 or m_1. (Def 3.9 in textbook).

Semantic Security

- In layman terms, it is the computational complexity equivalent of perfect secrecy.
- I.e. given the ciphertext no PPT algorithm can determine any partial information about the corresponding message with non-negligible probability.
- Perfect secrecy means that the ciphertext reveals no information about the plaintext message and semantic security says you cannot obtain any information about the plaintext in a computationally feasible manner.
- Easier to work with indistinguishable encryptions.

Theorem

A SKE has indistinguishable enryptions in the presence of an eavesdropper iff it is semantically secure in the presence of an eavesdropper.

Proofs by Reduction

- We need to show something is computationally secure.We have to rely on unproven assumptions.
- We assume some mathematical problem is hard, or a low-level cryptographic primitive is secure.
- Then prove that a given construction based on this problem/primitive.
- The proof has a reduction : transforms any efficient adversary $\mathcal{A}$ that succeeds in breaking the scheme into an efficient algorithm $\mathcal{A}'$ that solves the hard problem.
- Let X be a problem that cannot be solved by any prol-time algorithm.
- We need to show some scheme Π is secure.
- Consider a PPT adversary $\mathcal{A}$ and $\epsilon(n)$ its chances of succeeding.

Proofs by Reduction

- Construct an efficient algo $\mathcal{A}'$ called the reduction that attempts to solve X using $\mathcal{A}$.
- For $\mathcal{A}'$, $\mathcal{A}$ is a blackbox that attacks Π.
- On input instance x of X, $\mathcal{A}'$ will simulate for $\mathcal{A}$ an instance of Π s.t.:
  - For $\mathcal{A}$ it is the same view as interacting with Π even if it is running as a subroutine in $\mathcal{A}'$.
  - If $\mathcal{A}$ breaks the instance of Π that is being simulated by $\mathcal{A}'$, it should allow for $\mathcal{A}'$ to solve X it was given with at least inverse polynomial probability, $1/p(n)$.
- This implies $\mathcal{A}'$ solves X with prob. $\epsilon(n)/p(n)$. If $\epsilon(n)$ is not negligible neither is $\epsilon(n)/p(n)$.
- But our assumption of X shows that no efficient PPT $\mathcal{A}$ can break Π with non-negligible probability and Π is computationally secure.

Proofs by Reduction

- When we build stream ciphers with pseudorandom pads, we did not unconditionally prove that it is secure.
- We show that if we have a pseudorandom generator then it is secure.
- We are reducing the security of a higher-level construction to a lower-level primitive.
- It is easier to design a lower-level primitive that is secure than a higher level one.
- It is easier to analyze too, than analyze a complicated scheme.
- But this does not mean constructing a PRG is easy!

Security for Multiple Encryptions

- We looked at a weak model of passive eavesdropping and one ciphertext.
- Next we consider communicating parties sending multiple ciphertexts to each other using same key and an eavesdropper observing all of them.
- Description of $PrivK^{mult}_{\mathcal{A},\Pi}(1^n)$:
  1. $\mathcal{A}$ outputs a pairs of equal length lists of messages $M_0 = (m_{0,1}, \ldots, m_{0,t})$ and $M_1 = (m_{1,1}, \ldots, m_{1,t})$ with $|m_{0,i}| = |m_{1,i}|\ \forall i$.
  2. k is generated and a uniform bit $b \in \{0, 1\}$ is chosen. For all i, $c_i \leftarrow Enc_k(m_{b,i})$ and the list $C = (c_1, \ldots, c_t)$ is given to $\mathcal{A}$.
  3. $\mathcal{A}$ outputs a bit b'.
  4. $PrivK^{mult}_{\mathcal{A},\Pi}(1^n) = 1$ if $b' = b$ and 0 otherwise.

Security for Multiple Encryptions

- How do this experiment come in the picture of security definitions?

Definition

A SKE $\Pi = (Gen, Enc, Dec)$ has indistinguishable multiple encryptions in the presence of an eavesdropper if for all probabilistic polynomial-time adversaries $\mathcal{A}$,

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

Security for Multiple Encryptions – is it stronger?

- Any scheme that is secure w.r.t. $PrivK^{mult}$ is also secure w.r.t. $PrivK^{eav}$. The list has only one message.
- But is our new definition strictly stronger?

Theorem

There is a SKE that has indistinguishable encryptions in the presence of an eavesdropper but not indistinguishable multiple encryptions in the presence of an eavesdropper.

- OTP! It is secure w.r.t. $PrivK^{eav}$. But consider $\mathcal{A}$ outputting $M_0 = (0^\ell, 0^\ell)$ and $M_1 = (0^\ell, 1^\ell)$.
- Let $C = (c_1, c_2)$ be the ciphertexts $\mathcal{A}$ receives.
- If $c_1 = c_2$, then $\mathcal{A}$ says $b' = 0$ else 1.

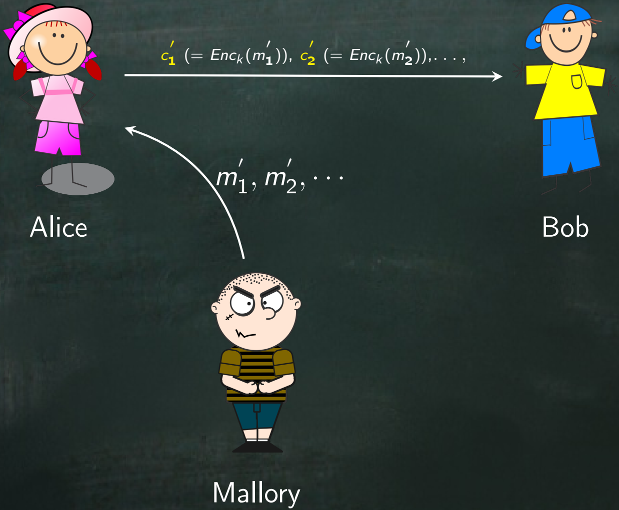
OTPs and $PrivK^{mult}$

- What is the probability that $b^{'} = b$?
- The same message encrypted twice will yield the same ciphertext. That is OTP encryption is deterministic.
- Thus if $b = 0$ then $c_1 = c_2$ and so $\mathcal{A}$ outputs 0 in this case.
- If $b = 1$ then a different message is encrypted each time and so $c_1 \neq c_2$ and $\mathcal{A}$ outputs 1.
- So probability is 1 that the adversary will succeed.
- Thus OTPs are not secure w.r.t. $PrivK^{mult}$. We need probabilistic encryption.

Theorem

If Π is a (stateless) encryption scheme in which Enc is a deterministic function of the key and message then Π cannot have indistinguishable multiple encryptions in the presence of an eavesdropper.

Chosen-Plaintext Attacks



Mallory gets Alice to encrypt $m_1^{'}, m_2^{'}, \ldots$ and eavesdrops for the corresponding ciphertexts.

Chosen-Plaintext Attacks

Alice

$c = Enc_k(m)$, m is m_0 or m_1

m_0 and m_1 are unknown

Bob

Mallory

Can Mallory tell
which message was encrypted
with probability better than
random guessing?

CPA in the real world

- CPA encompasses known-plaintext attacks and that is easy to see in the real world.
- How can adversary have significant influence over what messages got encrypted?
- $\mathcal{A}$ types on a terminal which in turns encrypts what $\mathcal{A}$ typed using the shared key of the server.
- In WWII, British placed mines in certain locations so that their locations will get encrypted by Germans and they can use that to break the scheme.
- More examples from WWII and real world!

CPA security

- $\mathcal{A}$ has access to an encryption oracle $Enc_k()$, it is viewed as a blackbox that encrypts messages of $\mathcal{A}$'s choice using a key k but won't show how it is done to $\mathcal{A}$.
- $\mathcal{A}$ queries this oracle with m and $Enc_k()$ returns $c \leftarrow Enc_k(m)$.
- For a randomized encryption, the oracle also uses fresh randomness each time.
- $\mathcal{A}$ can interact with this oracle as many times as it likes.
- We do not worry about the efficiency of the oracle.

CPA indistinguishability experiment

$PrivK^{cpa}_{\mathcal{A},\Pi}(1^n)$

1. A key k is generated considering the security parameter 1^n.
2. $\mathcal{A}$ has oracle access $Enc_k()$ and outputs a pair of messages m_0, m_1 of the same length.
3. A uniform bit $b \in \{0, 1\}$ is chosen and then a ciphertext $c \leftarrow Enc_k(m_b)$ given to $\mathcal{A}$.
4. $\mathcal{A}$ continues to have oracle access to $Enc_k()$ and outputs a bit b'.
5. $PrivK^{cpa}_{\mathcal{A},\Pi}(1^n) = 1$ if $b' = b$ ($\mathcal{A}$ succeeds) and 0 otherwise.

A private-key encryption scheme Π has indistinguishable encryptions under a CPA or is CPA secure if for all PPT $\mathcal{A}$

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

CPA for Multiple Encryptions

- Slightly different approach to take into consideration modeling attackers that can adaptively choose plaintexts to be encrypted even after observing previous ciphertexts.
- There is a left-to-right oracle, $LR_{k,b}$ that on input (m_0, m_1) returns $c \leftarrow Enc_k(m_b)$ s.t. if $b = 0$, $\mathcal{A}$ receives an encryption of left plaintext else it received encryption of right plaintext.
- The attacker has to guess b.
- This generalizes multiple message lists, instead of deciding which list the encrypted messages belong to we sequentially query

$$LR_{k,b}(m_{0,1}, m_{1,1}), \ldots, LR_{k,b}(m_{0,t}, m_{1,t})$$

LR-oracle experiment

$PrivK^{LR-cpa}_{\mathcal{A},\Pi}(1^n)$

1. A key k is generated considering the security parameter 1^n. A uniform bit $b \in \{0,1\}$ is chosen.
2. $\mathcal{A}$ has oracle access $LR_{k,b}(\cdot,\cdot)$ as defined previously.
3. $\mathcal{A}$ outputs a bit b'.
4. $PrivK^{LR-cpa}_{\mathcal{A},\Pi}(1^n) = 1$ if $b' = b$ ($\mathcal{A}$ succeeds) and 0 otherwise.

A private-key encryption scheme Π has indistinguishable multiple encryptions under a CPA or is CPA secure for multiple encryptions if for all PPT $\mathcal{A}$

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

LR-oracle experiment

- CPA-security for multiple encryptions implies it is CPA-secure for single encryption too.
- But unlike eavesdropping adversaries, the converse also holds : CPA-security (for single encryptions) implies CPA-security for multiple encryptions.

Theorem
Any SKE that is CPA-secure is also CPA-secure for multiple encryptions.

- We skip the proof.
- Big advantage for CPA-security – enough to show only for single encryption.
- Security against CPA is a minimal requirement for most schemes!

Summary

Summary

Summary