Topics lecture will cover:

- Review
- Computational Adversary
- One Way Functions (OWF)
- OWF vs. Standard complexity theory assumption
- Weak vs. Strong OWF

Recall 1

1. Perfect Secrecy - \( \forall m_1, m_2 \forall c \Pr[E(K, m_1) = c] = \Pr[E(K, m_2) = c] \)
2. Shannon Secrecy - \( \forall m \forall c \Pr[m] = \Pr[m|E(K, m) = c] \)

Recall 2

- **Theorem 1** Perfect Secrecy \( \equiv \) Shannon secrecy

**Claim 2** One time pad satisfies perfect secrecy

**Theorem 3** If \((G, E, D)\) satisfies perfect secrecy over \(M\), then \(|k| \geq |M|\) provided that the number of keys \( \leq \) size of message sent and the size of key \( \leq \) size of cleartext sent using this key.

**Proof** (by contradiction) Suppose that \( \exists (G, E, D) \) such that \(|k| < |M|\). Let \( c \) be a ciphertext being transmitted. What \( EVE \) does is try every possible key space: \( \forall k \in K \) run \( m = D(k, c) \). When done, \( EVE \) has \( m_1, m_2, ..., m_{|k|} \). This implies that there is some message that exists that is not in this list. We can write this as \( \exists m \in M - m_i, \Pr[m|c] = 0 \). Since two messages have different probabilities, we have a contradiction and the theorem is true.

Modern Cryptography

1976 Diffie-Hellman described a new definition for an adversary. Namely, \( EVE \) is now computationally bounded. Note that from now on Professor Goldwasser defines “possible” to mean “efficient” and “impossible” to mean “infeasible”.

What is meant by efficient? As proposed by Diffie-Hellman, a computer algorithm that can flip coins and run in polynomial time. There are many reasons for this definition:

1. polynomials are composed (i.e. a polynomial can be made up of polynomials, and also that a linear combination of polynomials leads only to another polynomial).
2. we run into formal problems such as exponential functions (which blow up quickly) if we do not use polynomials.

**Definition 4** \( EVE \) is any algorithm such that:

1. Always runs in polynomial time \( \exists P, A(x) \) terminates within \( P(|x|) \) steps.
2. It may be randomized (denoted by PPT for probabilistic polynomial time).
Basically, we want to make $EVE$ as powerful as possible, but still prove security against her.

Now we need to create new definitions for secrecy. Since $EVE$ is PPT, there is no such thing as Shannon’s Perfect secrecy (by Theorem 3).

Let’s come up with a new definition.

**Polynomially Secure Encryption:** $(G,E,D)$ is polynomially secure (this is what we will mean by secure from now on) if $\forall$ PPT algorithms $EVE$, $Pr[EVE(c) = m_1 | c = E(k,m_1)] = Pr[EVE(c) = m_1 | E(k,m_2)]$. The equal sign means that the two probabilities are essentially equivalent. We want the difference between the two probabilities to be negligible in the security parameter. We also want to make a simplification to our previous definition; let’s add a security parameter “$n$”. We now choose a security parameter first, then the key. Basically, $n = |k|$. The value of $n$ depends on technology.

**Negligible Function:** We say that $f(n)$ is negligible if $\forall$ polynomial $P$, $\forall$ $n$ sufficiently large, $f(n) < 1/p(n)$. Since the probability is less than polynomial time, $EVE$ does not have enough time to witness this event. This is different from Shannon’s definition.

Our goal is that we want $(G,E,D)$ to be secure for many messages. We come to our next definition, One Way Function.

**One Way Function:** Polynomially secure encryption (for many messages). A cryptographic building block.

- $f : \{0,1\}^* \rightarrow \{0,1\}^*$
- $\exists A$ where $A$ is a PPT where $A(x) = f(x)$ is computable in polynomial time $\forall x$.
- $\forall$ PPT $B$, $Pr[B(f(x)) = x' | f(x) = f(x')] < neg(n)$; note that this probability is taken over all $x$’s such that $|x| = n$. This means that an attempt to invert will fail with negligible probability. This holds for large enough $n$. Also note that this is a strong requirement for a function.

There is also the notion of “exact” security: Any algorithm $B$ that runs in $t$ steps and succeeds with probability $\epsilon$. Define a $(t, \epsilon)$OWF.

OWF imply the existence of secure encryption by great pseudorandom number generators.

A big question is “do OWF’s exist?” We don’t know. Instead, we assume they exist. We can prove that secure encryption schemes exist under this assumption.

**Complexity Theory**

- $P$: set of problems that can be solved in polynomial time
- $NP$: set of problems that can be verified in polynomial time
- $BPP$: Problems that can be solved by a PPT algorithm
- $P/poly$: size specific algorithm. Problems solved by non-uniform polynomial time algorithm.

Side note: if OWF exist, then it implies that $NP \neq BPP$.

We want a candidate OWF. Here’s one:

- $f(x,y) = x \cdot y$ such that $|x| + |y| = n$.
- $f : \{0,1\}^* \rightarrow \{0,1\}^*$

RSA says that $x$ and $y$ should be choosen such that $|x| \cong |y| = n/2$ & $x,y$ are primes. The best inverting algorithm is subexponential: $e^{n^{1/3}logn^{2/3}}$

Even though we don’t know whether Strong OWF’s exist, Maybe we have a weak OWF:

- $\exists A$ PPT such that $A(x) = f(x)$
- $\exists$ polynomial $Q$ such that $\forall$ PPT $B \forall n$ sufficiently large, $Pr[B(f(x)) = x$ s.t. $f(x) = f(x')] > 1/Q(n)$ where $|x| = n$