Complexity Theory
Lecture 9
Anuj Dawar

http://www.cl.cam.ac.uk/teaching/1920/Complexity
Alice wishes to communicate with Bob without Eve eavesdropping.
In a private key system, there are two secret keys

\( e \) – the encryption key

\( d \) – the decryption key

and two functions \( D \) and \( E \) such that:

for any \( x \),

\[
D(E(x, e), d) = x.
\]

For instance, taking \( d = e \) and both \( D \) and \( E \) as exclusive or, we have the one time pad:

\[
(x \oplus e) \oplus e = x
\]
One Time Pad

The one time pad is provably secure, in that the only way Eve can decode a message is by knowing the key.

If the original message $x$ and the encrypted message $y$ are known, then so is the key:

$$e = x \oplus y$$
In public key cryptography, the encryption key \( e \) is public, and the decryption key \( d \) is private. We still have,

\[
\text{for any } x, \quad D(E(x, e), d) = x
\]

If \( E \) is polynomial time computable (and it must be if communication is not to be painfully slow), then the following language is in \( \text{NP} \):

\[
\{(y, z) \mid y = E(x, e) \text{ for some } x \text{ with } x \leq_{\text{lex}} z\}
\]

Thus, public key cryptography is not \textit{provably secure} in the way that the one time pad is. It relies on the assumption that \( P \neq \text{NP} \).
A function \( f \) is called a \textit{one way function} if it satisfies the following conditions:

1. \( f \) is one-to-one.
2. for each \( x \), \(|x|^{1/k} \leq |f(x)| \leq |x|^k \) for some \( k \).
3. \( f \) is computable in polynomial time.
4. \( f^{-1} \) is \textit{not} computable in polynomial time.

We cannot hope to prove the existence of one-way functions without at the same time proving \( P \neq \text{NP} \).

It is strongly believed that the RSA function:

\[
f(x, e, p, q) = (x^e \mod pq, pq, e)
\]

is a one-way function.
Though one cannot hope to prove that the RSA function is one-way without separating $P$ and $NP$, we might hope to make it as secure as a proof of $NP$-completeness.

**Definition**
A nondeterministic machine is *unambiguous* if, for any input $x$, there is at most one accepting computation of the machine.

$UP$ is the class of languages accepted by unambiguous machines in polynomial time.
Equivalently, UP is the class of languages of the form

$$\{x \mid \exists y R(x, y)\}$$

Where $R$ is polynomial time computable, polynomially balanced, and for each $x$, there is at most one $y$ such that $R(x, y)$. 
We have

\[ P \subseteq \text{UP} \subseteq \text{NP} \]

It seems unlikely that there are any \( \text{NP} \)-complete problems in \( \text{UP} \).

One-way functions exist \textit{if, and only if}, \( P \neq \text{UP} \).
Suppose \( f \) is a \textit{one-way function}.

Define the language \( L_f \) by

\[
L_f = \{(x, y) \mid \exists z (z \leq x \text{ and } f(z) = y)\}.
\]

We can show that \( L_f \) is in \( \text{UP} \) but not in \( \text{P} \).
Suppose that $L$ is a language that is in UP but not in P. Let $U$ be an unambiguous machine that accepts $L$.

Define the function $f_U$ by

- if $x$ is a string that encodes an accepting computation of $U$, then $f_U(x) = 1y$ where $y$ is the input string accepted by this computation.
- $f_U(x) = 0x$ otherwise.

We can prove that $f_U$ is a one-way function.