Cosc 412: Cryptography and security
Lecture 3 (22/7/2020)
One time pads,
Stream ciphers,
Semantic security.

Michael Albert
michael.albert@cs.otago.ac.nz

This week

- One time pads
- Pseudo-random generators (PRGs) and stream ciphers
- Statistical tests against PRGs and semantic security
- The problem of key agreement

The one time pad

The system:

$$
\begin{aligned}
\mathbf{2} &= \{0, 1\} && \text{the set of } \textit{bits} \\
\mathcal{M} = \mathcal{K} &= \mathbf{2}^n && \text{the set of } n\text{-bit strings} \\
n &\sim 2^{30} && \text{message size, e.g. 1 gigabit} \\
E(k, m) &= k \oplus m && \text{encoding} \\
D(k, m) &= k \oplus m && \text{decoding}
\end{aligned}
$$

- The key is chosen uniformly at random from the key space.
- Very fast.
- Not generally practical (key exchange).

Shannon's observation

If,

- for any two messages m_0 and m_1,
- and any ciphertext c,
- the number of keys that encode with m_0 to get c is the same as the number that encode with m_1 to get c,

then an attacker learns nothing about the message from c (assuming keys are randomly chosen). So, no ciphertext only attack is possible and we say the cipher has *perfect secrecy*.

Stream ciphers

- Use a *pseudo-random* key.
- A pseudo-random generator, G, is a function that takes a *seed* and produces a much longer sequence efficiently:

$$G : \mathbf{2}^s \to \mathbf{2}^n \quad \text{where } s << n.$$

- If we agree about the seed (key) then we have access to a "long" sequence of agreed bits, which we can use as if it were a one time pad:

$$E(k, m) = G(k) \oplus m \quad D(k, c) = G(k) \oplus c.$$

- In what sense can the corresponding cipher be secure?

Predictability

- Obviously predictability of a PRG is a bad thing.
- If we choose a very weak condition for predictability (i.e., lots of generators satisfy it) then the corresponding notion of unpredictability will be strong.
- So - "for some i, given i bits of the generator, we can predict the next bit with probability significantly different from 1/2 (assuming the key is chosen uniformly at random)".

Attacks on the one time pad/stream ciphers

- Two time pads are completely insecure
- Systematic key modification can also be insecure (WEP example)
- Integrity fails, an attacker can potentially modify the ciphertext in predictable ways e.g., by taking the exclusive or of c with some string p, then

$$D(k, m \oplus k \oplus p) = m \oplus p.$$

Indistinguishable from random

- The set of outputs of a PRG is tiny in the output space – how could it possibly look random?
- The adversary has to be testing the output and is somewhat limited
- So we ask: "Is there an effective algorithm which correctly identifies the output of our PRG as non-random?"

Advantage

- A *statistical test* is a map $A : \mathbf{2}^n \to \mathbf{2}$ (think of 0 as meaning "I don't recognise this as random", and 1 as "I recognise this as random").
- The *advantage* of a statistical test A over a PRG, G is:

$$\mathrm{Adv}(A, G) = \left|\mathrm{Pr}_{k \leftarrow K}[A(G(k)) = 1] - \mathrm{Pr}_{r \leftarrow \{0,1\}^n}[A(r) = 1]\right|$$

- That is - how different does A find G from true randomness?
- Examples

Security

A PRG is secure if there is no efficient statistical test which has a non-negligible advantage over G.

- Why is efficiency important here?
- Can we build a provably secure PRG? (Probably not!)
- Security and unpredictability are equivalent!

Semantic security

- An adversary gives you two explicit messages m_0 and m_1
- You (choose a random key and) encode one of them and return the encoded version
- The adversary tries to guess which one was encoded

Their advantage is the difference between their probability of guessing 1 when you chose 0 and their probability of guessing 1 when you chose 1 (probability over your random choice of key).

Security of stream ciphers

- If only we had a secure PRG, the corresponding stream cipher would be semantically secure.
- "Proof": Our advantage in the semantic security game for the stream cipher built on a PRG can be at most twice our advantage in the PRG-security game (trying to distinguish the PRG from a truly random stream).
- See notes (and video).

The problem of key agreement

- Alice and Bob need to efficiently carry out an encrypted conversation of some length (upwards of tens of kilobytes)
- They have access to a fast and secure shared-key cryptosystem requiring a key of not more than a few tens of bytes
- Unfortunately they have no shared key
- They need to “agree a secret” across an open channel