**CS311 Yoshii – Week 13B (Notes-13B) Encryption for Security**

**Questions from this lecture will be on the final exam.**

**Cyber security** has become a very important topic these days.

**Cybersecurity**

The field of cybersecurity is a new and rapidly developing field. Open job listings for cybersecurity positions throughout the nation **rose 73 percent** in the five years through 2012, 3.5 times faster than postings for computer jobs as a whole, according to Boston-based Burning Glass, a labor market analytics firm that collects data from more than 22,000 online jobs sites. Furthermore, the data show “employers literally just posting and reposting” their positions, suggesting that there are not enough qualified professionals to fill those positions. There were **64,383 jobs** related to cybersecurity listed for the twelve months through April, about 3 percent of all information technology positions, according to the company.

San Diego has been identified as a Cybersecurity hub with the development of Securing our eCity program, Stop Think Connect campaign and the Cyber Hive collaboration program. The San Diego Labor statistics reflect an above average increase, 12.3%, in cybersecurity related jobs projected for 2013-2020.

**Cybersecurity involves: 1) communication safety, 2) prevention of intrusion, 3) penetration testing (ethical hacking) , and 4) detection of intrusion and recovery.**

**\*\* Show the PSM web site \*\***

This lecture will introduce you to the world of **encryption** so that you will become interested in security issues.

Also, **decryption** is related to **“P == NP?”**

With hashing you converted one number (e.g. social security) into another number (the slot).

**Math plays an important role** in this conversion.

In encryption, you also use math to convert a number into another number. But it is more complex because you will **have to be able to get back the original from the encoded information.**

For example, if I convert every number by doing Num mod 2, the result will be 1 in many cases.

From 1, I cannot get back Num. Math is fascinating.

***Cryptography/Encoded Messages:***

For cryptography, you need **a lock (encryption tool)** and **a key (decryption tool).**

The more mathematically complex the scheme is, the more secure the system is. For the math behind cryptography, take MATH523 and CS538.

The lock is referred to as the **encryption key (EK)** while the key that opens a lock is referred to as the **decryption key (DK**)**. It is confusing because both are called keys. So, from here on, let me refer to them as simply EK and DK.**

***Basic Cryptography Algorihms: (We will go over 4 methods)***

1. **Symmetric-Key Encryption (Good between two close friends)**:

**EK and DK are shared by close friends.**

1. Computer A and Computer B both have EK ***where DK can be figured out easily from EK.***
2. Computer A encrypts a message using EK and sends it.
3. Computer B decrypts the message using DK.
4. Computer B encrypts a message using EK and sends it.
5. Computer A decrypts the message using DK.

**Example:** EK is to convert every letter by the letter that occurs next in the alphabet. (A -> B). DK is then to do the reverse (B -> A)

**\*Inter\* Think of another symmetric EK-DK example. If you had to use it from today to send documents between you and your friend, how will you make sure the method stays between just you two?**

**Analogy = “Let’s share the lock and the key”:**

**Mary C:\Users\rtuuser\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\A9392OR9\MC900389750[1].wmf C:\Users\rtuuser\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\A9392OR9\MC900389750[1].wmfJohn**

* 1. Mary first puts the secret message in a box, and locks the box using a padlock to which she has a key.
  2. Mary sends the box to John through regular mail.
  3. John uses a **copy of Mary's key** **to open the box**, and reads the message.
  4. John uses the **same padlock** **to send his secret reply**.

***Advantage:***  The mathematics involved in encoding and decoding is simple.

***Problem:***

* *This algorithm works well between two people who know each other well but if you are communicating with numerous people, how do you share EK and DK with everyone?*
* *Also, even between friends, how do you get EK/DK to your friend securely?*

**Note that since it is easy to figure out DK from EK, you should not give out EK publicly. It would be easier if you could give out EK publicly.**

**Data Encryption Standard** (**DES**) was the first major symmetric algorithm developed for computers in the United States:

**2. Public/Asymmetric-Key Encryption (All my friends have to use this lock):**

**Your friends know how to encode. Only you know how to decode.**

Each user has a pair of keys – a **public EK** and a **private DK**. The publicly EK is widely distributed to your friends, while the private DK is known only to you. Your friends use your public EK to send messages to you.

1. Computer A encrypts a message using publicly given EKB and sends it to B.
2. Computer B decrypts the message using its DKB.
3. Computer B encrypts a message using publicly given EKA and sends it to A.
4. Computer A decrypts the message using its DKA.

This is possible because calculating the private DK from the publicly available EK is **either impossible or prohibitively expensive.**

**Analogy = “Give me your lock for me to use”**

**May uses John’s lock John uses his secret key**

* Mary asks John to send his open padlock (EK) to her through regular mail. John keeps his key (DK) to himself.
* When Mary receives it she uses it to lock a box containing her message, and sends the locked box to John.
* John can then unlock the box with his key and read the message.
* To reply, John must similarly get Mary's open padlock (encoding key) to lock the box before sending it back to her.

***Advantage:*** You don’t have to worry about keeping the EK a secret. You can share it widely.

***Problems:***

* Computationally expensive.
* How do you know that the public EK you have received belongs to the person or entity claimed, and has not been tampered with or replaced by a malicious third party? Example: Does EKA really belong to Computer A? What if Computer Z says “This is EKA. Send messages to A using this.”

**Any publicly made available EK should be certified. Usual solution is PKI.**

**Public Key Infrastructure** (PKI) has a **certificate authority (CA)** that both issues and verifies the **digital certificates**. The digital certificate **certifies the ownership of a public EK**.

**RSA** is a widely used public-key algorithm. It uses **prime numbers** to generate keys. There are infinitely many prime numbers to choose from.

**How RSA Works:**

The public EK and private DK for the RSA algorithm are generated the following way.

**First you need to get n and e. Example:**

1. Choose two distinct randomly picked large prime numbers ***p*** and ***q***.
2. Compute ***n*** = *pq*.
3. Compute φ(*n*) = φ(*p)*φ(*q*) = (p-1)(q-1), where **φ is Euler's totient function**.
4. Choose an integer ***e*** such that 1 < *e* < φ(*n*) and [gcd](http://en.wikipedia.org/wiki/Greatest_common_divisor)(*e*, φ(*n*)) = 1

🡺The **public EK consists of *n* and *e*.** We will show you later how to use them.

1. Given ***d****\*e* ≡ 1 (mod φ(*n*)), solve for **d**.

🡺The **private DK consists of *n* and *d*,** where d must be kept secret.

***P* and *q* must also be kept secret** because they can be used to calculate *d*.

**But from just n and e, it is impossible to figure out d.**

### How to Encrypt using *n* and *e*:

### The sender first turns message *M* into an integer *m*, such that 0 ≤ *m* < *n and then computes c:*

 c \equiv m^e \pmod{n} .

### How to Decrypt using n and d:

### The receiver converts c back to m:

 m \equiv c^d \pmod{n} .

**3. Using a Combination to Speed Up Things (Sending EK/DK securely):**

**May uses John’s lock John uses his secret key**

Our shared EK/DK

**Public-key encryption described above takes a lot of computing**, so most systems use a **combination of public-key and symmetric key encryption**. The following allows you to send the symmetric key to your friends securely:

1. Computer A creates symmetric (shared) EK/DK and sends it to Computer B using public-key encryption (i.e. using EKB).
2. Computers A and B can then communicate privately using symmetric-key encryption.
3. Once the session is finished, each computer discards the symmetric EK/DK used for that session.

**Analogy = “Send me the secret key using this lock”:**

* Mary asks John to send his public open padlock (EK) to her through regular mail. John keeps his DK to himself.
* When Mary receives it she uses it to lock a box containing the shared key, and sends the locked box to John.
* John can then unlock the box with his key and get the shared key for the future communications between them.

**4. Alternative - Doubling the Encryption to Avoid Sharing:**

Neither party needs to even touch the other party's padlock (or key) if we use a **double locking** system:

1. Computer A encrypts a message using his EKA and sends it to B. (one lock)
2. Computer B encrypts the message using his EKB and sends it back to A. (two locks)
3. Computer A decrypts the message using his DKA and sends it to B again. (one lock)
4. Computer B decrypts the message using his DKB to read it. (no lock)

**Analogy = “I will lock it until you can lock it yourself”**

**Mary uses her private EK John uses his private EK**

**Mary uses her DK John uses his DK**

1. Mary puts the secret message in a box, and locks the box using a padlock to which only she has a key. She then sends the box to John through regular mail.
2. When John receives the box, he adds his own padlock to the box, and sends it back to Mary.
3. When Mary receives the box with the two padlocks, she removes her padlock and sends it back to John again.
4. When John receives the box with only his padlock on it, John can then unlock the box with his key and read the message Mary.

***Problem:*** This is only possible if **commutative ciphers are used**. A commutative cipher is one in which the **order of encryption and decryption is interchangeable**. It is more mathematically complex.

**In-Class Exercise (Week13B) - Cryptography - (MAX 2 people per team) 5 pts**

**Print your name first.**

1. **Describe the differences between symmetric and asymmetric methods.**
2. **Why is it difficult to figure out p and q from n (which is p\*q)?**

**-------------------------------- break -----------------------------------------------------**

**\*\*\* IoT Handout : Preview of the rest of the topics on where CS is headed. \*\*\***

**The End of the Note Section:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

***®Summarize here what you have learned in your own words and also write down your own thoughts/reactions/questions.***