SYMMETRIC AND PUBLIC KEY ENCRYPTION LESSON

Mark Emry, McNeil High School, Round Rock ISD, Austin, TX

SYMMETRIC AND PUBLIC KEY ENCRYPTION

PUBLIC KEY ENCRYPTION REVIEW

This writeup is designed to string together all the main thoughts of the CCL.

This video is a good summary as well:

"RSA encryption: Step 4 | Journey into cryptography | Computer Science | Khan Academy" Apr 28, 2014. Khan Academy Labs. CC BY NC SA 3.0.

PUBLIC KEY

Alice wants ***to get*** a message from Bob.
Alice will take 2 prime numbers and multiply them together.
This is one component of Alice's PUBLIC key. Call this **n**.
Next you find what is called the (phi) ϕ of **n**.

Alice will subtract 1 from each prime and multiply them together to get ϕ**n**.

Find a number that is relatively prime to the ϕ**n**. There are potentially many, but let's keep it small.
Let's call this number **e**. '**e**' is used to encrypt our message and will be sent along with n to Bob that wants to send Alice a message.

Notice Alice sends the product of the two primes and a number that is relatively prime to ϕn. The two numbers are related and are built in a special way to let an inverse of the encrypting operation occur. It is impossible to find what the original n is unless we know the prime factors n was generated with.

Alice sends n and e to Bob. This is called the public key.
Alice wants Bob to send a private message to her. 👀👀👀

ENCRYPTION

Bob will take his message use Alice's **n** and **e** like this.

$$C = M^{e} \bmod n$$

M is the message as a number - this will be padded (encrypted in a pre-defined

way)
C is the resulting encrypted message.

DECRYPTION

Bob sends **C** to Alice and she can use a decrypting equation find the original **M**.

$$M = C^d \bmod \phi n$$

NOTE: Alice would have found her private key d by solving this equation:

$$d*e \bmod \phi n = 1$$

EXTENDED EUCLIDEAN ALGORITHM

Finding d can be done by trial and error, but this can be done by a computer using the Extended Euclidean Algorithm (video from GVSUmath Feb 12, 2014).

*Alice will not share **d** or ϕ**n**. These are Alice's private keys.*

The roundabout action of the two formulas (video by Kahn Academy) will give Alice the message using exponents and using the remainder. Look up Diffe-Hellman for more information.

NOTES:

This is a one-way encryption for Bob to send a message to Alice.

RSA encryption, when the keys are huge, is slow, so it's usually only done once per secure communication. The one time that it's done, a symmetric key is shared so that the two computers can communicate using traditional symmetric key encryption.

If the **n** is 2048bits (a little over 600 digits), it'll take a standard desktop computer in 2019 over 6 QUADRILLION years to factor it back to two prime numbers.

Here is an issue: Quantum computers are finally getting to be less error prone and more accessible for engineers to work with, so factoring a large number can theoretically be done much faster. Because of this, RSA may soon be obsolete.

RSA encryption has been broken before, but not because of the math. Many "side-channel" methods have been found, but most of them have been "patched" out.

There are other public key encryption algorithms such as elliptic curve cryptography that are also difficult to crack. ("*How does elliptic curve cryptography work?*" by Anastasios Arampatzis. Mar. 21, 2019. Dzone Security Zone.)

RSA encryption, even though it's known and studied many times, is still used today for most websites. It doesn't generate **security by obscurity**. Even though people know the exact math behind the algorithm, it's very difficult to break in a short period of time.

QUESTIONS TO PONDER

Listen:

🥚🥚🥚 "10 Hours of Wii Shop Channel (Remix)" (Nicky Flowers. Dec 1, 2015). 🥚🥚🥚

1. Can you find other public key encryption algorithms? What makes them secure? Is it more secure than RSA? Does it require less computing power?

2. Research how RSA encryption has been broken/hacked. What kind of ingenious side-channel hacks have been formulated? Are they still a threat?

3. How does quantum computing threaten RSA encryption? Will it break our security on the internet?

4. What are your thoughts about cybersecurity? Ever think about a career in it?