EE 382V: Social Computing **Fall 2018**

Lecture 8: November 10

Lecturer: Vijay Garg *Scribe: Shiyang Cheng*

8.1 Ceaser Cipher

In cryptography, *Caeser Cipher* is one of the simplest encryption techniques. It exploit substitution cipher in which each letter in the plaintext is replaced by a letter some fixed number of postions down the alphabet.

For example:

Cipher: $HPPE\ NPSOJOH$

Plain: *good morning*

In which, H maps to g, and so on.

8.2 RSA - Asymmetric cryptography

RSA stands for Rivest-Shamir-Adleman.

8.2.1 History

Cryptanalysis of the Enigma ciphering system enabled the western Allies in World War II to read substantial amounts of Morse-coded radio communications of the Axis powers that had been enciphered using Enigma machines. This yielded military intelligence which, along with that from other decrypted Axis radio and teleprinter transmissions, was given the codename Ultra. This was considered by western Supreme Allied Commander Dwight D. Eisenhower to have been "decisive" to the Allied victory.

The Enigma machines were a family of portable cipher machines with rotor scramblers. Good operating procedures, properly enforced, would have made the plugboard Enigma machine unbreakable. However, most of the German military forces, secret services and civilian agencies that used Enigma employed poor operating procedures, and it was these poor procedures that allowed the Enigma machines to be reverse-engineered and the ciphers to be read.

The German plugboard-equipped Enigma became Nazi Germany's principal crypto-system. It was broken by the Polish General Staff's Cipher Bureau in December 1932, with the aid of French-supplied intelligence material obtained from a German spy. A month before the outbreak of World War II, at a conference held near Warsaw, the Polish Cipher Bureau shared its Enigma-breaking techniques and technology with the French and British. During the German invasion of Poland, core Polish Cipher Bureau personnel were evacuated, via Romania, to France where they established the PC Bruno signals intelligence station with French facilities support. Successful cooperation among the Poles, the French, and the British at Bletchley Park continued until June 1940, when France surrendered to the Germans.

From this beginning, the British Government Code and Cypher School (GC&CS) at Bletchley Park built up

an extensive cryptanalytic capability. Initially, the decryption was mainly of Luftwaffe (German air force) and a few Heer (German army) messages, as the Kriegsmarine (German navy) employed much more secure procedures for using Enigma. Alan Turing, a Cambridge University mathematician and logician, provided much of the original thinking that led to the design of the cryptanalytical bombe machines that were instrumental in eventually breaking the naval Enigma. However, the Kriegsmarine introduced an Enigma version with a fourth rotor for its U-boats, resulting in a prolonged period when these messages could not be decrypted. With the capture of relevant cipher keys and the use of much faster US Navy bombes, regular, rapid reading of U-boat messages resumed[WikiEnigma].

In 1970, *James H. Ellis*, a British cryptographer at the UK Government Communications Headquarters (GCHQ), conceived of the possibility of "non-secret encryption", (now called public key cryptography), but could see no way to implement it. In 1973, his colleague *Clifford Cocks* implemented what has become known as the RSA encryption algorithm, giving a practical method of "non-secret encryption", and in 1974, another GCHQ mathematician and cryptographer, *Malcolm J. Williamson*, developed what is now known as *Diffie-Hellman* key exchange[WikiRSA].

8.2.2 RSA Use Case

There are three individuals: *Alice*, *Bob*, *Eve*. *Alice* want to communicate with *Bob* as figure-8.1. *Eve* is a passive attacker, who intercepts all the message between *Alice* and *Bob*. *Eve* also could send message to each other pretend to be one of them. *Alice* and *Bob* want to keep the message secrecy and authentication at the same time.

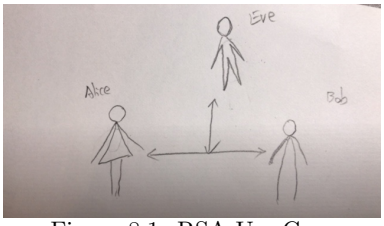

Figure 8.1: RSA Use Case

We use the properties:

1. Private and public keys can invert each other.

If we use private key to encrypt a message, we can use public key to decrypt the message. Otherwise, if we use public key to encrypt a message, we can use private key to decrypt the message.

2. If we have public key, we cannot find private key (because of the fact that a one way function)

Solution:

Let's define:

e_A: *Alice*'s public key

d_A: *Alice*'s private key

e_B: *Bob*'s public key

d_B: *Bob*'s private key

E: encryption function

D: decryption function

Assume that *Alice* gets *Bob*'s public key e_A, and *Bob* gets *Alice*'s public key. Nobody shares private key to anyone.

Alice want to send message M. She applies the encryption function to M, using encryption function E with *Bob*'s public key e_B as $E(M, e_B)$. Then encrypt the encrypted message again using her own private key d_A, which is $E(E(M, e_B), d_A)$.

Bob could use *Alice*'s public key e_A to decrypt the message into $E(M, e_B)$. If and only if the message comes from *Alice*, *Bob* could get the message $E(M, e_B)$, and decrypt the message again using his own private key d_B into message M.

In this way, *Bob* use *Alice*'s public key e_A to authenticate *Alice*. And use his own private d_B to decrypt the message which provide the secrecy at the same time.

8.2.3 RSA Algorithms

8.2.3.1 Generate Key Pairs

Algorithm 1 generateKeyPair()

1. Take two big prime numbers p and q, let $n = p^q$
2. Compute *Euler*'s totient function $\Phi(n) = (p-1) * (q-1)$
3. Choose any number e that relatively prime to $\Phi(n)$, which means $gcd(e, \Phi(n)) = 1$
4. Find d such that $d * e \equiv 1 \ mod \ \Phi(n)$ (take *mod* both side)
5. throw away p, q, $\Phi(n)$
6. return (n, e) as public key, (n, d) as private key

This section gives the algorithms is meant to generate key pair which is (e, d) as algorithm-1. The function to encrypt message M as C:

$C = M^e \ mod \ n$

The function to decrypt encrypted message C back to M:

$M = C^d \ mod \ n$

8.2.3.2 Modulo Properties

Modulo operation finds the remainder after division of one number by another, which represent as *mod*.

Modulo operation has properties:

1. $(a + b) \ mod \ n = ((a \ mod \ n) + (b \ mod \ n))$
2. $(a * b) \ mod \ n = ((a \ mod \ n) * (b \ mod \ n))$

References

[WikiRSA] WIKIPEDIA, Public-key cryptography *wikipedia.org* (2018),

[WikiEnigma] WIKIPEDIA, Cryptanalysis of the Enigma *wikipedia.org* (2018),