SPCS Cryptography Class Lecture 10

July 3, 2015

Attacks

We have seen many ways of sending cryptic messages by now. Here is some of them: Known ciphertext attack/Ciphertext-only attack (COA)

- This attack is based on knowing only ciphertexts.
- Examples:
  - Frequency analysis (for substitution ciphers), Kaisiski method (for Vigenere cipher)
  - If you know how to solve Diffie-Hellman problem, you can use a cipher-text only attack on Diffie-Hellman. Same for RSA.
  - Attack on WEP (protocol for protecting wireless connections).
  - Modern cipher are designed to be safe under ciphertext-only attacks - that's why when a new cipher is being adopted, they would be tested really thoroughly so that it is at least considered COA-safe.

Attacks

Known-plaintext attack (KPA)

- This attack is based on knowing the corresponding ciphertexts for a few known plaintexts.
- Examples:
  - Substitution/Vigenere/Columnar are prone to KPA: you can figure out the key!
  - Autokey cipher: guessing what English words show up.
  - Cryptanalysis in World War: either by stealing/intercepting messages, often they would be able to know/guess what certain ciphertexts mean. They keep analyzing from there.
  - Linear Cryptanalysis against block ciphers.
- Is RSA safe under KPA? (Randomness is important!)

Attacks

Chosen-plaintext attack (CPA)

- This attack is based on being able to know what the encryption for a number of plaintexts of your choice.
  - Substitution/Vigenere/Autokey/Columnar..
  - Differential Cryptanalysis against block ciphers.
  - Is RSA safe under CPA?

Chosen-ciphertext attack (CCA)

- This attack is based on being able to know what the decryption for a number of ciphertexts of your choice.
  - Substitution/Vigenere/Autokey/Columnar..
  - Is RSA safe under CCA?

Attacks

Chosen-ciphertext attack (CCA)

- Bob uses RSA. Bob publishes his public key (n, e).
- Suppose that Alice sends Bob m with ciphertext m^e mod n.
- Eve intercepts the ciphertext m^e. Suppose that she can trick Bob to tell her what certain ciphertext corresponds to.
- Eve doesn't want to be obvious, so she sends Bob $2^e m^e = (2m)^e$. Bob tells her what the decryption is - do you know the decryption is?
- Can Eve figure the original message m now?
- Would Bob be able to guess that Eve is being malicious, by denying her request? Realizing that $2^e m^e$ is related to $(2m)^e$?

Attacks

- This property of RSA is called malleability. It means that a ciphertext can be transformed into another ciphertext, such that after decryption this is related to the original plaintext.
- Usually undesirable, but also has its advantage.
- Example: Your document is encrypted. You want to change a little things but you will need to decrypt, change, then encrypt again. Would be much easier if you can modify the ciphertext directly.Such encryption methods are called *homomorphic* encryption.

We will fix this by padding.

Padding

Idea: add some randomness to our system.

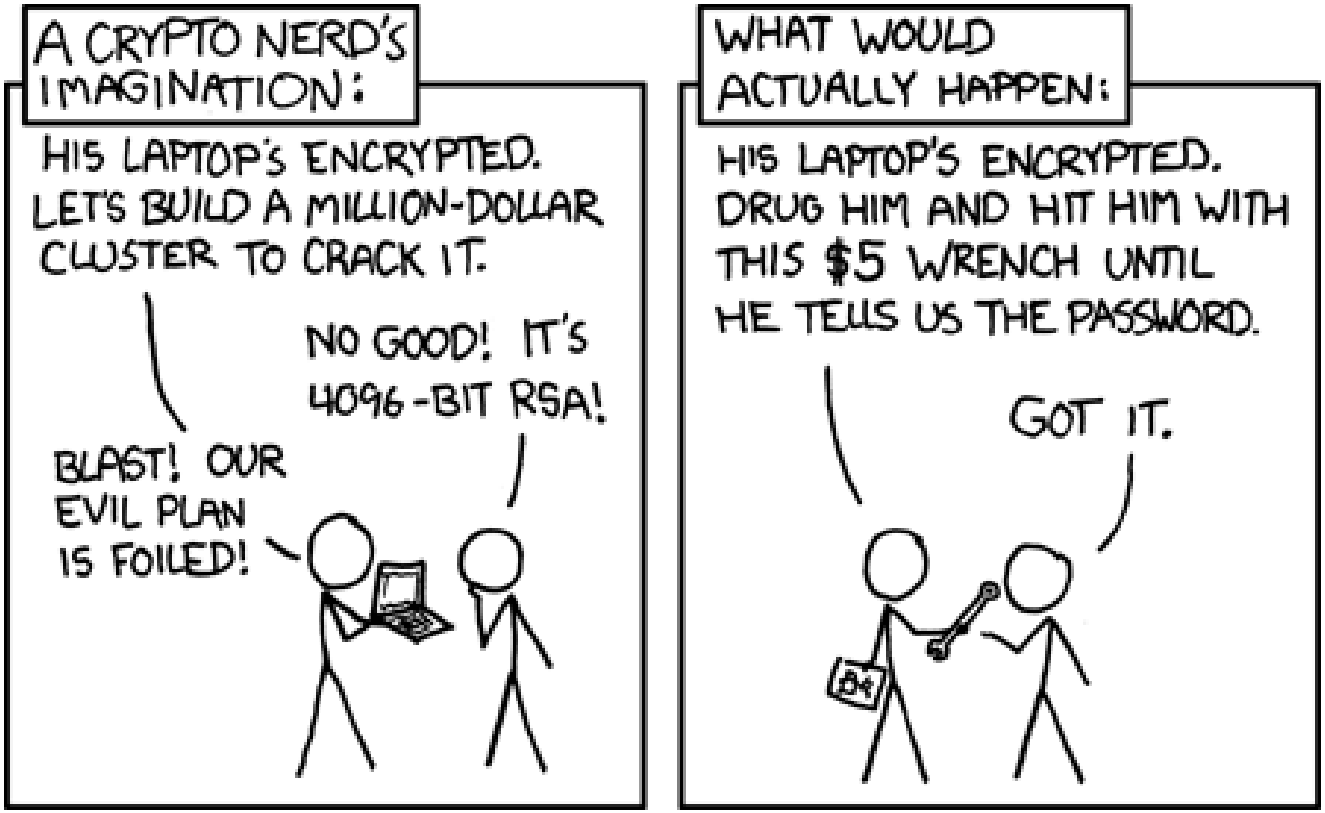
- For example, Alice can append 30 random digits to the end of her plaintext message before encrypting it. There are 10^{30} ways of appending random digits, so it's not likely for Eve to guess the particular sequence used, thus impossible to launch a chosen-plaintext attack.
- The said chosen-ciphertext attack is not possible either, because the randomness at the end makes this version of RSA not malleable anymore.
- This idea of filling in a plaintext message is known as *padding*.
- Padding may also be used elsewhere. For example in cryptographic hash functions or symmetric key cryptography - whenever you need space fillers.
- Many padding schemes available, e.g. OAEP.

Attacks

Some other attacks:

- Man-in-the-middle attack. Is RSA prone to Man-in-the-middle attack?
- Side-channel attacks. Some examples:
  - Timing attack.
  - Power attack. (Example: Differential power analysis) `http://www.cryptography.com/technology/dpa.html`
  - Fault attack. (Example: Differential fault attack) `https://en.wikipedia.org/wiki/Differential_fault_analysis`
  - Electromagnetic attack. (Example: `https://en.wikipedia.org/wiki/Tempest_(codename)`)

Attacks



https://xkcd.com/538/

Attacks

Security of a cryptosystem needs to address

- Authentication: Can Bob be certain that the message is from Alice?
- Confidentiality: Can Alice and Bob be certain that the plaintext cannot be read?
- Integrity: Can Alice and Bob be certain that the message sent was not tampered?
- Non-repudiation: Can a third party be certain that the message is from Alice?

Hashing

The first thing we will address is Integrity: how we can make sure that data is not tampered. Before we do that though, we will first briefly talk about the idea of hash table. This is a kind of data structure that helps to speed up searching.

Example

Suppose there are 10000 students with 8 digit student ID's. We want to store the data of these students so that we can look things up quickly.

- If the list is unsorted, linear search takes $O(n)$ time.
- If the list is sorted, binary search takes $O(\log n)$ time.
- Using an array of size 10^8, ordered by student ID's, take $O(1)$-access time but takes too much space.

Can we do it in $O(1)$ time with less space wasting?

Hashing

Example

- Suppose the student ID's are so special that they are all distinct modulo 15000.
- One method: store student with ID n in the array with index $n \bmod 15000$.
- This way, we only need an array of size 15000.
- With such a hash table, one can find the record with student ID n by accessing the $n \bmod 15000$-th entry.
- In this case, the *hash table* is the array of size 15000 loaded with students' data, and the *hash function* is $n \to n \bmod 15000$, where n is student ID.

Hashing

Example

- What if some students' ID are the same modulo 15000? (This is the problem of *collision*).
- If very few students' ID modulo 15000 collide, there are simple ways to get around it. (For example, separate chaining)
- What if the hash function is $n \to n \bmod 2$?
- So in this case we want the hash function to be *collision-resistant*.

Example

We can look things up in dictionary by

- Look for the first letter of the word.
- Then start searching from the first word starting with that letter. (Separate chaining!)

Hashing

Example

Recall in USPS money order,

- the 11-digit serial number consists of the 10-digit code plus a check digit.
- The check digit comes from sum of first 10 digits mod 9.
- The transformation from the first 10-digits to the check digit is again a hash function.

So what is hashing?

Hashing

- Hashing is any transformation of some data of variable size to data of fixed (and usually smaller) size.
- We may want different properties of the hash, depending on the purpose of hashing.
- Some applications:
  - Hash tables. (Searching)
  - Bloom filter. (Searching)
  - Cryptography.

Hashing used in cryptography is called cryptographic hashing.

Hashing

Example

When you log into an account online, how do they check your identity?

- Naive method: store your password when you register. Compare the password you typed in with the stored password when you login.
- Is the naive method good?
- Encrypt the password stored. Every time you log in, the system encrypts the typed password, and compare with the stored encrypted password.
- Is this method good?

Hashing

Example

- Third method: Hash the password stored.
  - Given a variable-length password, hashing produces a "fixed-length fingerprint".
  - (Collision resistance) Different password should have different fingerprints.
  - When you login, your typed password is hashed, and is compared to the stored hash.

However, we want at least some more properties for hashes to be useful:

- Pre-image resistance: Easy to compute the hash, but impossible to reverse the process. Why?
- Second pre-image resistance: Given a hash, you cannot come up with some string such that the hashed string is the given hash. Why?
- Avalanche effect: two similar inputs should have very different output. Why?

Hashing

Example (Student ID revisited)

Let's go back to the mod 15000 hashing. Let's see if it satisfies the security properties we mentioned.

- Collision-resistant: this was our starting assumption.
- Preimage-resistance: given student ID modulo 15000, can you recover the student ID? Yes.
- Second pre-image resistance: Given a hash H - can you come up with some string such that after hashing it becomes H? Yes - you can easily cook up 8-digit-number that's congruent to a given number mod 15000. So this is NOT second pre-image resistant.
- Avalanche effect: if I change the last few digits of student ID by a little, is the hash similar? Yes! They only differ 1 mod 15000. So this does not have Avalanche effect.

Hashing

Example

Verifying file integrity/Error detection - checksums

- Suppose you download OpenOffice installation program not from the developer, but from some random source.
- You may be worried that the file may have been tampered, what can you do?
- One thing you can do is to check the hash value of the downloaded file.
- If this hash value is the same as hash value provided by the developer, then it's very likely that the program was not tampered.
- `http://www.openoffice.org/download/checksums/3.4.1_checksums.html`

Digital signatures

Authentication:

- As stated, hashing is only useful for checking integrity. What about authentication?
- There are two methods we will discuss - signatures (public key) and message authentication code (symmetric key)

What are digital signatures?

- Similar to ordinary signatures - hopefully others cannot forge it.
- Would provide authentication, integrity, and non-repudiation.

RSA signatures

One example is the RSA signature scheme - very similar to RSA encryption! Suppose Alice sends Bob a message, encrypted as C, and she wants to sign this ciphertext C.

RSA signature scheme

- Alice chooses large primes p, q, compute $n = pq$.
- Alice also chooses e mod $\phi(n)$, and publishes her public key (n, e). Her private key is $d \equiv e^{-1}$ mod $\phi(n)$.
- To sign C, she sends the signature $S = C^d$ to Bob. (Note: this is where RSA signature is different from RSA crypto!)
- Bob can recover the ciphertext C by calculating S^e mod n, where e is the public key of Alice.

RSA signatures

What happens when Eve intercepts this message (C, S)?

- She can tamper with C, but she would also need to produce a signature that matches with the tampered message.
- Without knowing Alice's private key - this is hard; same difficulty as RSA problem!

Now that you know textbook RSA is not safe, you can probably imagine that this is not safe either..

- Existential forgery: Suppose Eve just wants to mess things up. Can she come up with a legitimate pair of (Ciphertext, Signature)?
- Yes!
- Thus in practice, people would first hash the message, then sign the hash. What's the difference?
- Can we use a random hash function?
- Should Alice's hash function be public?

Digital signatures

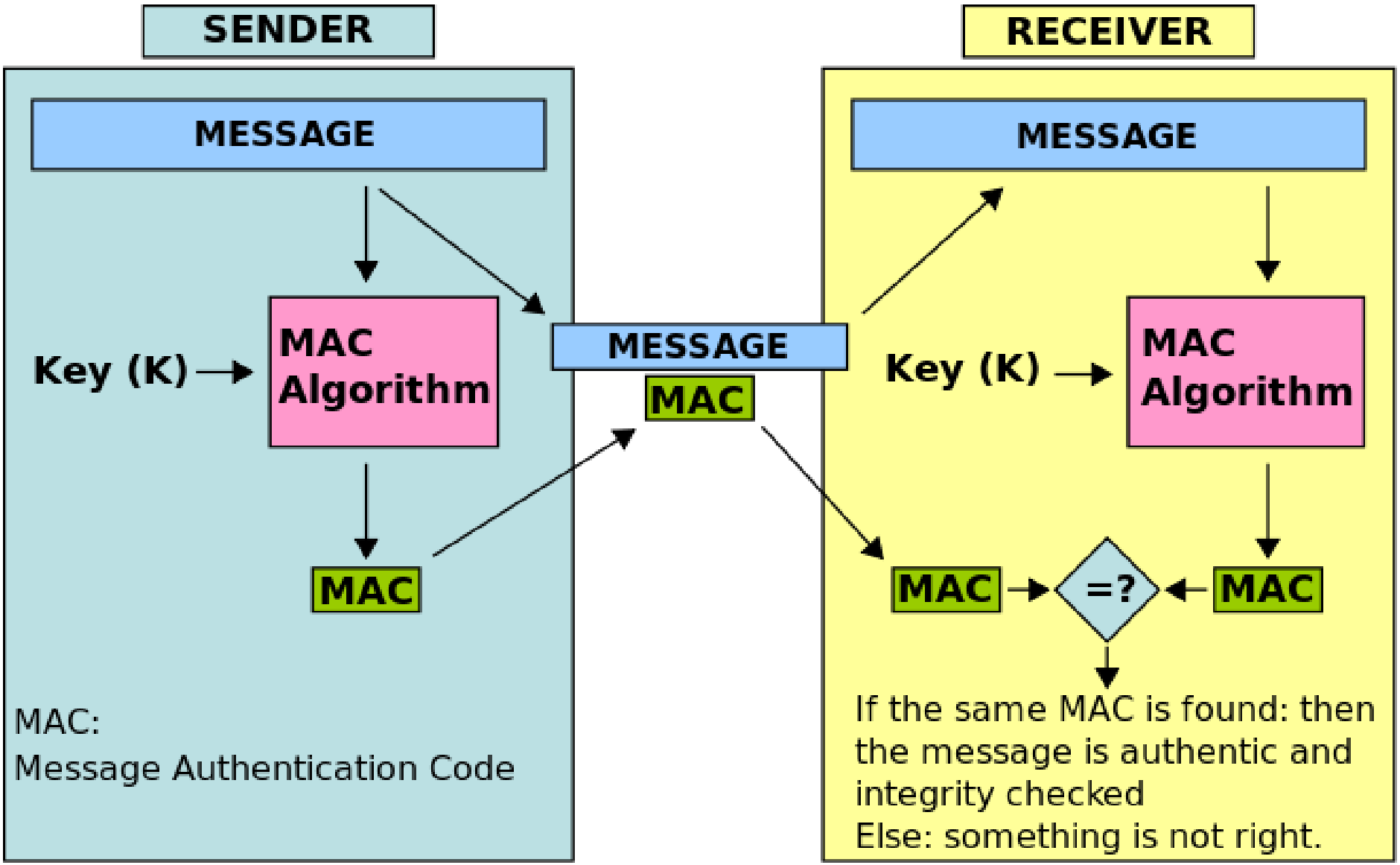
Summary:

- RSA is only one possible signature scheme. As with the encryption process, if you can find a hard problem, you may be able to turn it into a signature scheme.
- People often use RSA or DSA (Digital signature algorithm) for signing.

Message Authentication Code (MAC)

- This time we work in the symmetric key framework - which means Alice and Bob will do one key exchange first, so that they have a shared key.
- If Alice hashes her ciphertext and send it to Bob as a signature, would it work?
- What if Alice has many hash functions?
- She can use the secret key to agree on one particular hash function with Bob.
- Is this practical? (For example, UMAC)

Message Authentication Code (MAC)



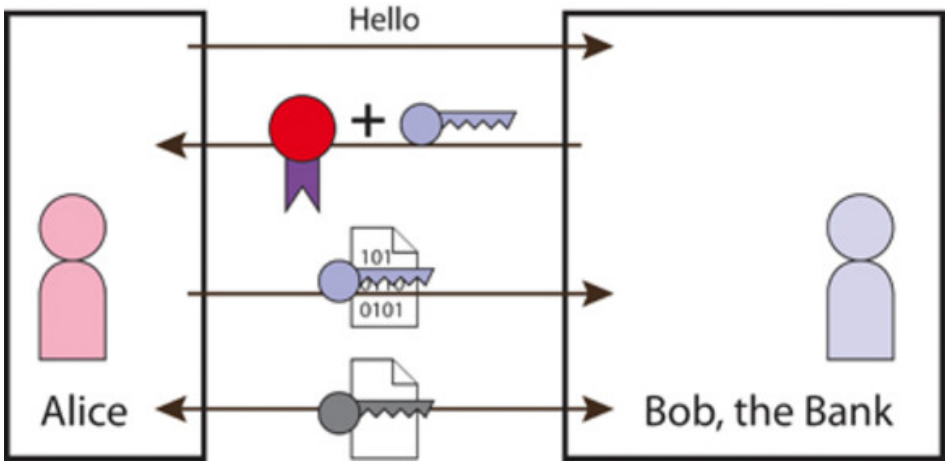
Public Key Infrastructure (PKI)

What happens if I publish a public key online, and claim that I am Bank of America, and you can do online banking with me?

- We really need to be able to match a public key with a real entity.
- Usually we leave this to a trusted third party - Certificate Authority (CA).
- Some examples of CA: Federal Bridge Certification Authority (FBCA) `http://www.idmanagement.gov/federal-public-key-infrastructure`
- Some examples of TLS CA: Comodo, Symantec, GoDaddy, etc.

Public Key Infrastructure (PKI)

Example:



Public Key Infrastructure (PKI)

Can we trust Certificate Authority(CA)?`http://venturebeat.com/2015/04/02/google-and-mozilla-decide-to-ban-chinese-certificate-authorit`

- On March 20, Google became aware of several authorized certificates for their domain.
- These certificates were (essentially) issued by CNNIC (China's CA)
- What does that mean?
- In early April, Google, Mozilla etc removed CNNIC from its root store, i.e. trusted CA's by default.

Various applications of Cryptography in daily(virtual) life

- TLS/SSL: https = http within SSL/TLS
- PGP: An encryption/decryption/signature system for email.
- SSH: Secure shell - use remote machines securely.
- VPN: Virtual private network
- RTP: Voice communication over Internet. For example, Google Hangouts uses SRTP. Skype? `http://www.theguardian.com/world/2013/jul/11/microsoft-nsa-collaboration-user-data`
- Bitcoin: Cryptocurrency

etc, etc.