Lecture 10. Data Integrity: Message Authentication Schemes

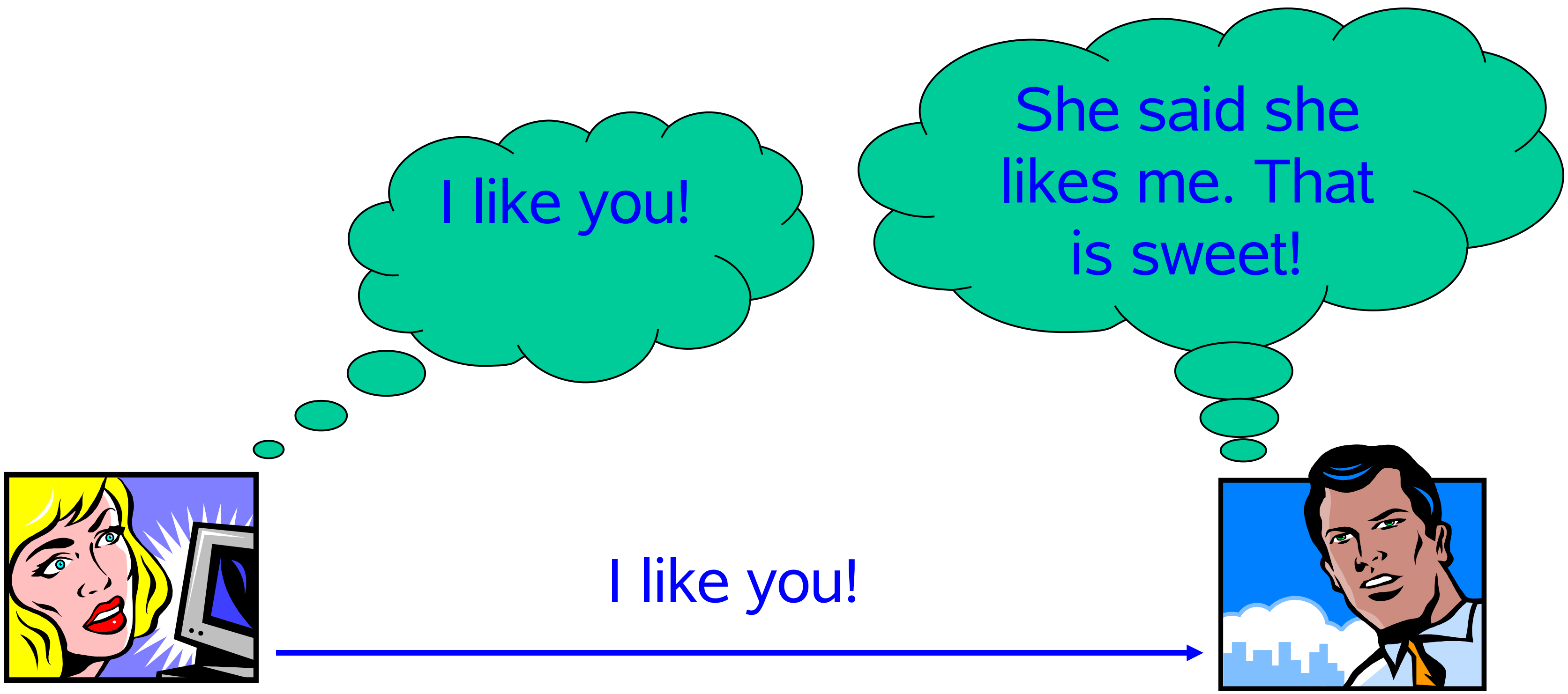
Roadmap

- Problem Statement
- Definition
- Constructions
- Remarks

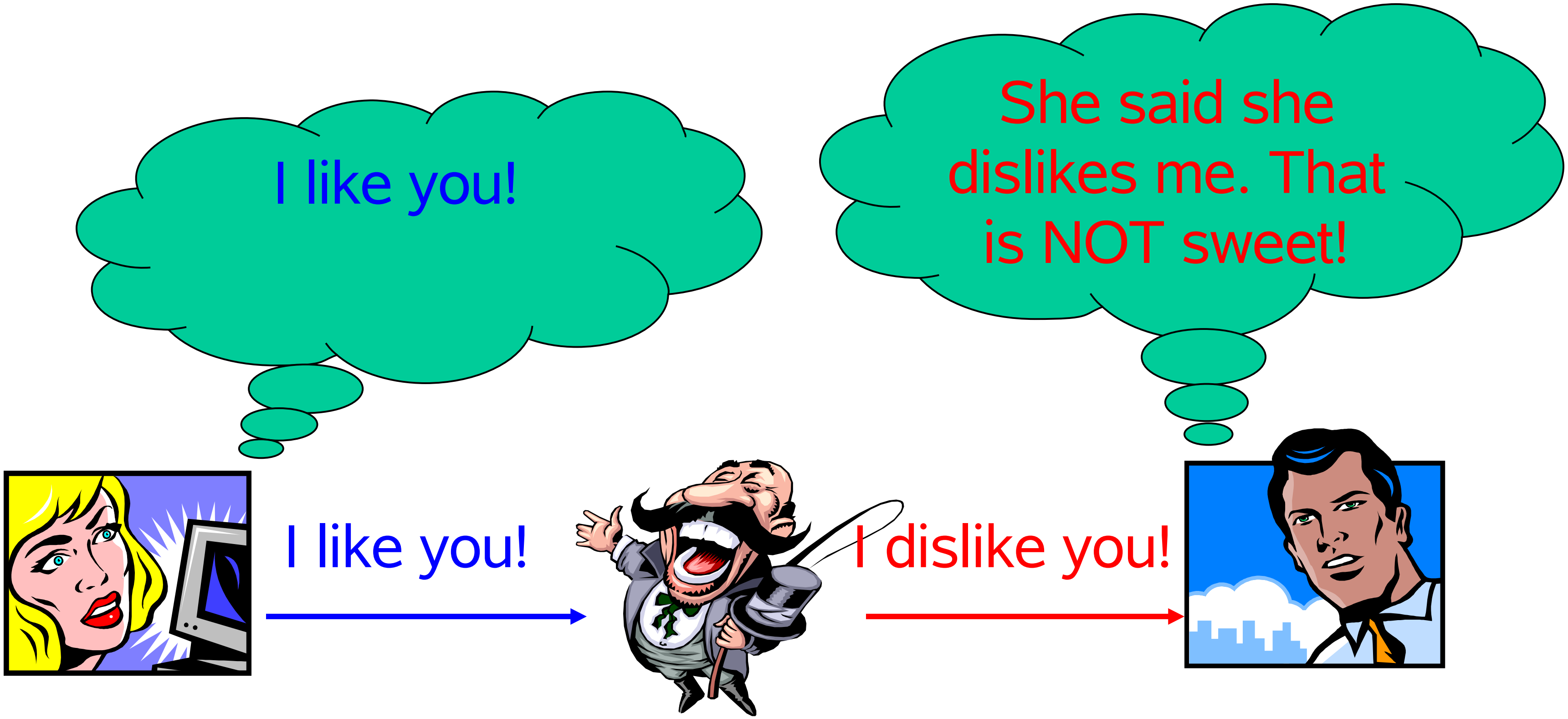
Problem Statement

- Suppose you are communicating (via any channel: the air, the wire, whispering, …) with your friend
- Sure you want what your friend "received" is exactly what you "sent"
- Yeah, this is trivially ensured in the face-to-face case
- Now we ask the question: what if the "channel" is under the control of some bad guy?

Sweet, If Channel Is Secure



What If There Is a Bad Guy…



The bad guy controls the channel (man-in-the-middle attack) …

How We Get There?

- So, we need a technical mechanism to ensure that the output of the channel is the same as the input to it (and detectable otherwise)
- How we achieve this in the physical world?
- For 10^3 years, people knew how to use "sealed envelope"

Not It's Clear What We Want …

- Message authentication scheme can be viewed as the analogy of "sealed envelope" in the physical world
- That is, emulating an envelope so that nobody can tamper with the content!
  - Of course, we ignore confidentiality for simplicity
  - You may think the envelope is transparent. But Indeed …

For Your Curiosity …

- A classic part of cryptography is **to emulate an envelope**
  - nobody can **tamper with** the content – message authentication
  - Nobody can **peep** the content – encryption (not that simple, it took many years to understand what it is!)

Warning

Encryption does not provide data integrity

- Pro argument: If “I like you” is encrypted, how can the adversary change the ciphertext to a new one corresponding to the plaintext “I dislike you”?
- This has been incorrectly understood by many people for many years! So do not commit the same mistake!

Roadmap

- Problem Statement
- Definition
- Constructions
- Remarks

How Do We Specify a MAS?

- Let's recall what happens in the physical world
  - Before you write the first letter to your friend,
    - You two agree on a handwriting style (only you know)
  - Before you mail you letter (in a transparent envelope), you knew how you friend will verify it
    - You already had a "tagging" algorithm in your mind!
  - When your friend receives your mail, he/she knows how to verify if the letter is from you, because
    - He/she had a "verification" algorithm in mind

How Do We Specify a MAS?

- Let's recall what happens in the physical world
  - Before you write the first letter to your friend,
    - ✓ Key Generation (and Distribution)!
  - Before you mail you letter (in a transparent envelope), you knew how you friend will verify it
    - ✓ Tagging
  - When your friend receives your mail, he/she knows how to verify if the letter is from you, because
    - ✓ Verification

Definition

A message authentication scheme MAS = (KeyGen, Tag, Ver)

- KeyGen is a randomized algorithm that returns a key k. That is, $k \leftarrow$ KeyGen(SecurityParameter).
- Tag is an algorithm that takes input k and a message m, and return a tag δ. This algorithm may be probabilistic. That is, $\delta \leftarrow$ Tag(k,m). We also denote it by $\delta \leftarrow Tag_k(m)$.
- Ver is a deterministic algorithm that takes as input the key k, a message m, and a tag δ, and returns a bit valid/invalid. That is, Ver(k,m,δ) returns valid or invalid. We also denote it by $Ver_k(m,\delta)$.

Disclaimer

- Message Authentication Scheme (MAS) and Message Authentication Code (MAC) are interchangeable.
- In the context of MAS, tag generation algorithm is typically the same as the tag verification algorithm. Moreover, the keys are the same.

Security

- We have specified the syntax of message authentication schemes
- What is the semantics? In particular
  - What that means when we say a message authentication scheme is secure?

Security

- Intuition 0: If the key is leaked, then an adversary can arbitrarily forge message authentication tags.
  - So we have to ensure that seeing polynomial many tags does not enable the adversary to recover the key, because
  - we cannot afford to not allow the adversary to see the genuine authentication tags, because the network is open.

Security

- Definition 0: we say a message authentication scheme is secure if seeing polynomial many authentication tags does not enable the adversary to recover the message authentication key.
- Is this definition good enough?

Security

- What if the adversary can forge, say, a single genuine authentication tag (e.g., even though the adversary is still unable to get the key)?

- So we need to refine the definition …

Security

- Intuition 1: Seeing polynomial many genuine authentication tags still does not enable the adversary to generate a single genuine authentication tag on a different message.

- Is this definition good enough?

Security

- What if the genuine authentication tags are for messages chosen by the adversary?
- Lunch time attack: the operator is out for lunch, the adversary has physical access to the message authentication machine to generate message authentication tags for the messages he prepared in the morning!
- So we need to refine the definition!

Security

- Intuition 2: Security means that seeing polynomial many genuine authentication tags on messages chosen by the adversary does not enable the adversary to generate even a single genuine authentication tag on a new message!
- Is this definition good enough?

Security

- What if the adversary can choose the next message based on the state information (or history of the output of the message authentication machine)?
- This is indeed called adaptive chosen message attack!

Security

- Intuition 3: Security means that seeing polynomial many message authentication tags on messages adaptively chosen by an adversary does not enable the adversary to generate even a single genuine authentication tag on a new message!

Security

- Intuition 3.5: Is the above definition enough?
- To the best of our knowledge, there is currently no more sophisticated definition.
- But if you can have a more sophisticated, yet meaningful one, then this lecture is paid off ☺

Now We Are Ready to …

- Combine the above discussions together to get a full-fledged version of the definition of message authentication schemes

Definition (syntax)

A message authentication scheme MAS = (KeyGen, Tag, Ver)

- KeyGen is a randomized algorithm that returns a key k. That is, $K \leftarrow$ KeyGen(SecurityParameter).
- Tag is an algorithm that takes input k and a message m, and return a tag δ. This algorithm may be probabilistic. That is, $\delta \leftarrow$ Tag(K,m). We also denote it by $\delta \leftarrow Tag_K(m)$.
- Ver is a deterministic algorithm that takes as input the key K, a message m, and a tag δ, and returns a bit valid/invalid. That is, Ver(K,m,δ) returns valid or invalid. We also denote it by $Ver_K(m,\delta)$.

Definition (semantics)

Requirements on a message authentication scheme

MAS = (KeyGen, Tag, Ver), where

- Correctness: for any $m \in$ MessageSpace, we have $Ver_K(m, \delta = Tag_K(m)) = 1$
- Security: unforgeability under adaptive chosen-message attack

Definition (semantics)

$K \leftarrow$ **KeyGen(k)**

Pr

m_1 →

← $Tag_K(m_1)$

…

→

m_n →

← $Tag_K(m_n)$

m_{n+1}, $Tag_K(m_{n+1})$ →

← Ok, you win as $Ver(K, Tag_K(m_{n+1}))=1$

attacker

challenger

subject to: $m_{n+1} \notin \{m_1, m_2, \ldots, m_n\}$

= negligible(security-parameter k)

Roadmap

- Problem Statement
- Definition
- Constructions
- Remarks

Constructions

- PRF/CBC MACs [Bellare-Kilian-Rogaway Crypto'94]
- XOR MACs [Bellare-Guerin-Rogaway Crypto'95]
- MACing use cryptographic hash function (e.g., MD5, SHA-1)
- Universal Hash based MACs [Black-Halevi-Krawczyk-Krovetz-Rogaway Crypto'99]

Constructions (cont.)

- Can we use keyed hash functions such as MD5(k,m) and SHA1(k,m) directly as a message authentication code?
  - Currently, nobody knows, and no positive evidence
- But evidences do show that HMAC is good
  - Remains to be true, even if MD5/SHA1 is not collision resistant (because a weaker property suffices HMAC!)

HMAC

- HMAC [Bellare-Canetti-Krawczyk Crypto'96]: why hash?
  - faster than block ciphers in software implementation
  - software implementations are widely available
  - not subject to export restriction
- HMAC is now IETF mandatory MAS

HMAC

- Suppose H is a good(?) hash function (e.g., MD5 with 128-bit output, SHA-1 with 160-bit output).
- ipad = the byte 0x36 repeated 64 times
- opad = the byte 0x5C repeated 64 times
- k is the message authentication key
- $HMAC_k(m) = H(k \oplus opad, H(k \oplus ipad, m))$

HMAC

1. Append zeros to the end of k to create a 64 bytes string
2. XOR the 64 byte string computed in step (1) with ipad
3. Append m to the 64 bytes string resulting from step (2)
4. Apply H to the stream generated in step (3)
5. XOR the 64 byte string computed in step (1) with opad
6. Append the in step (4) to the 64 byte result in step (5)
7. Apply H to the output in step (6) and output the result

HMAC

1. The recommended length of the key is at least l bits, where l is the length of the output of the hash function (i.e., l=128 for MD5, and l=160 for SHA-1)
2. A longer key does not add significantly to the security of HMAC
3. HMAC allows truncation of the final output to, say, 80 bits

HMAC: Security

- Security of HMAC can be justified given some reasonable assumptions about the strength of the underlying H
- Assume only that H has a certain kind of weak collision-freeness and some limited unpredictability
- Details may be covered in “advanced cryptography course”

Roadmap

- Problem Statement
- Definition
- Constructions
- Remarks

Applications

- Friend-Or-Foe
- TESLA: multicast authentication
- As a building block in advanced protocols
- … find more … that is your contribution …☺

What MAC Isn't?

- When Alice and Bob are in honeymoon, Bob said "I will give all my property to Alice if we divorce someday".
- The statement is authenticated using a message authentication code with a key known to both Alice and Bob.
- Alice (knowing some crypto) is happy and keeps it in a safe.
- Life is really wonderful …
- many days passed … sweet …

What MAC Isn’t? (cont.)

- Unfortunately, bad days come up …
- Divorce is on agenda; Alice presents the safe to a judge …
- Can Alice convince the judge that Bob did make the promise?
- No way! Alice could have done it herself!

What MAC Isn't? (cont.)

- How we solve the Alice-Bob puzzle?
- digital signature …