Crypto VII: Digital Signatures

Cryptographic Hardware for Embedded Systems
ECE 3170

Fall 2025

Assoc. Prof. Vincent John Mooney III

Georgia Institute of Technology

Reading Assignment

- Please Chapter 2 of the course textbook by Schneier
- NOTE that you are responsible for everything that is explained in lecture!!!

Notation from Katz and Lindell

- $\{X\}$ is a set of elements of type X
- m is a message in plaintext
  - m is composed of smaller blocks m_i suitable for individual encryption steps
  - $m = \{m_i\}$
- c_i is ciphertext corresponding to message block m_i
- c is ciphertext corresponding to message m
- Enc_k is encryption with key k
  - $c \leftarrow Enc_k(m)$
- Dec_k is decryption with key k
  - $m \leftarrow Dec_k(c)$
- MAC_k is generation of a message authentication code t with key k
  - $t \leftarrow Mac_k(m)$ or, alternatively, $t \leftarrow Mac_k(c)$
- $\langle a,b \rangle$ is a concatenation of a followed by b

Protocols

- A protocol accomplishes a task through a series of steps involving two or more parties
- Everyone involved must know all of the steps in advance
- Everyone must agree to the protocol
- The exact actions required for each step must be unambiguous
- Completeness: every possible situation must have a specified action

Cryptographic Protocols

- Use cryptography
- Specify trusted and untrusted parties
    - Trusted third party (if no TTP is used, the protocol is said to be “self-enforcing”)
    - Cheaters
- Specify an attack surface
- Specify limits
    - Learning more or doing more than what is provided for in the protocol should not be allowed

Attacks Against Protocols

- Different goals
  - Recover useful information
  - Alter outcomes
  - Spoof
- Passive
  - Typically information gathering via eavesdropping
  - Follow the protocol
- Active
  - Substitute messages
  - Degrade systems performance
  - Remove & delete information
  - Disrupt the protocol
- Motives: money, revenge, terrorism, …

One-way Functions

- Central to public-key cryptography
- Given x, it is easy to compute f(x)
- Given f(x), it is hard to compute x
- “Hard”: given hundreds or millions of years of massive computational power, it is statistically very unlikely to be able to compute x given f(x)
- Often the word “infeasible” is used in place of “hard”
- To this day there is no mathematical proof that one-way functions exist, but there is a lot of evidence

Trapdoor One-way Functions

- Given f(x), it is hard to compute x
- Unless you have a secret trapdoor
- In RSA, the secret trapdoor is the private key
- Another related concept is a “backdoor” not revealed to anyone but surreptitiously placed / available in the one-way function
  - Example: if RSA keys are selected which are known to not be based on prime numbers (i.e., the large numbers in fact have factors) but yet pass a particular primality test known to be in use
  - Another example could be a pseudorandom number generator whose number sequences pass the NIST tests yet can be predicted by means known only to the backdoor generating party

NOTE: Be Careful in Phrasing Questions

- True story: in the 1990s a relative of mine went to the airport and was asked (back then a ticket agent would ask passengers questions prior to issuance of a boarding pass) the following
  - “Is it possible that someone put something in your bags without you knowing about it?”
- The following question is similar
  - “Is it possible that there is a backdoor we do not know about?”
- Better questions
  - “Did someone you do not know very well ask you to put anything in your bags?”
  - “Is there any evidence that a backdoor may exist for this function?”

Digital Signatures: Qualities

- Authenticity: the signature convinces the recipient that the signer deliberately signed the document
- Unforgeable: the signature is proof that only the signer, and not anyone else, deliberately signed the document
- Not reusable: the signature is both part of the signed document and cannot be moved to another document
- Unalterable: the signed document cannot be modified in an undetected manner
- Nonrepudiation: the signer cannot later claim that he or she did not sign the document

Inferred Signature

- If the symmetric cryptographic key used by Alice, K_A, can be relied upon, one can infer that messages encrypted with the symmetric key of Alice were messages generated by Alice
- The fact that the message was encrypted by K_A results in an inferred signature by Alice
- However, questions remain
  - When did Alice generate the message?
  - Am I the intended recipient?
  - Is the message still valid?
  - It is possible that the above questions may be answered appropriately in context

Inferred Signing with Public Key Cryptography

- Modern networking protocols involve sessions where authentication is carried out first using public keys
- During this authentication, a symmetric key is exchanged
- The symmetric key is only valid for a *session* usually of short duration
  - Timestamps may be added by the networking protocol or separately
- Each session provides appropriate context to remove the doubts raised earlier about intended recipient, etc.

Explicit (Non-Inferred) Digital Signatures

- Most efficient approach: generate a hash from the document to be signed, encrypt the hash with one's private key
- RSA has the property that both $Dec_{private}(Enc_{public}(h)) = h$ and $Enc_{public}(Dec_{private}(h)) = h$ (where h is a hash value)
- Thus, the only person who could have used Alice's private key to sign the hash value is Alice
- Additional signatures can be added, e.g., Bob can (i) first verify Alice's signature and then can (ii) regenerate the hash value and sign it himself, adding his signature to the document after the signature of Alice