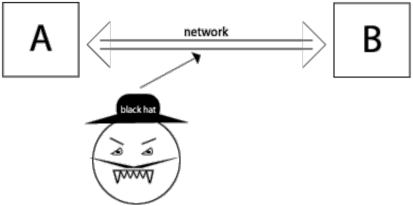
CSC3631 Cryptography

Message Authentication Code

Changyu Dong

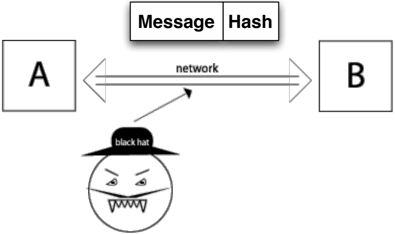
Newcastle University

Message Authnetication



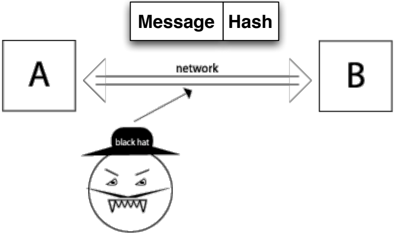
- If Alice wants to send a message to Bob, how can Bob be sure that
  - The message hasn't been modified
  - The message comes from Alice

First Try: Hash function



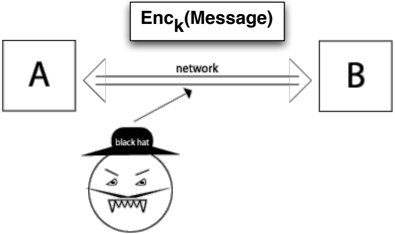
- Alice use a hash function, computes the hash value, appends it to the message and sends it to Bob
- Bob recomputes the hash value, and accepts if it is the same.
- Any problems?

First Try: Hash function



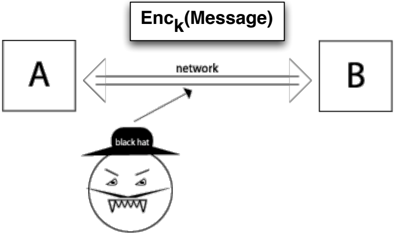
- Alice use a hash function, computes the hash value, appends it to the message and sends it to Bob
- Bob recomputes the hash value, and accepts if it is the same.
- Any problems?
  - An attacker can modify the message M', generate $H(M')$, and send $(M', H(M'))$ to Bob
  - Anyone can generate the hash value, no way to check whether the message is from Alice

Second Try: Encryption



- Alice shares a key with Bob, and encrypts the message before sending it to Bob
- Bob decrypts it, and accepts if it is decrypted correctly.
- Any problems?

Second Try: Encryption



- Alice shares a key with Bob, and encrypts the message before sending it to Bob
- Bob decrypts it, and accepts if it is decrypted correctly.
- Any problems?
  - Only Alice has the key, so the message comes from Alice
  - But encryption doesn't care about integrity: the message might have been modified. For example, if a stream cipher is used.
  - It might not easy to detect the modification

Message Authentication Code (MAC)

- A function with two inputs: a secret key K and an arbitrarily sized message M, output a fixed-length MAC value.
- The sender and the receiver share K
- The sender sends $(M, Mac_K(M))$
- The receiver receives (X, Y) and verifies that $Mac_K(X) = Y$. If so then accepts the message
  - The message hasn't been modified
  - The message comes from the real sender

Message Authentication Code (MAC)

A message authentication code consists of three PPT algorithms (**Gen, Mac, Vrfy**) such that

- The key generation algorihtm **Gen** takes as input the security parameter n and outputs a key k with $|k| \geq n$.
- The tag-generation algorithm **Mac** takes as input a key k and a message $m \in \{0,1\}^*$, and outputs a tag t, write as $Mac_k(m) \to t$.
- The deterministic verification algorithm **Vrfy** takes as input a key k. a message m, and a tag t. It outputs a bit b write as $b = Vrfy_k(m,t)$, with $b = 1$ meaning valid and $b = 0$ meaning invalid.

For every n, every k output by $Gen(n)$ and every $m \in \{0,1\}^*$, it is required that $vrfy_k(m, Mac_k(m)) = 1$ (correctness).

Security Model of MAC

- The adversary knows the algorithms, but not the key
- The adversary may have seen many messages along with their tags (the messages might even be chosen by the adversary)
- The adversary should not be able to forge a valid MAC for a message that the tag has not be seen by the adversary

MAC and replay attack

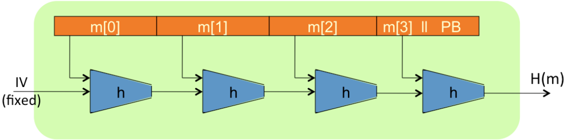
- Replay means the adversary capture a message and sends it again later.
- The security definition of MAC does not prevent replay attack
  - Alice sends (m, t) to Bob
  - Later Eve sends (m, t) to Bob again
  - Eve does not need to forge a tag
- However, application can add replay resistance by
  - include a timestamp with the message $T||m$
  - include a sequence number with the message $N||m$
- Eve captured $(T_i||m, Mac_k(T_i||m))$, but to replay, she needs to forge$(T_j||m, Mac_k(T_j||m))$ for the current time T_j

Hash-based Message Authentication Code (HMAC)

- Essentially a keyed hash function: $HMAC_K(M) = H(K \oplus a||H(K \oplus b||M))$ where H is a hash function and a, b are specified constants
- Keyless hash cannot be used as MAC.
  - Hash algorithms are public, anyone can generate the hash value for any message
- K should be at least n-bit, where n is the output size of the hash function
- H needs only to be weak collision resistant
- security level of $\frac{n}{2}$ bits if the hash function is secure (birthday attack)

Why not simply $H(k||m)$?

- This can be proven to be a secure MAC in the random oracle model
- BUT, if the hash function is based on Merkle-Damgård construction, then it is not a secure MAC.

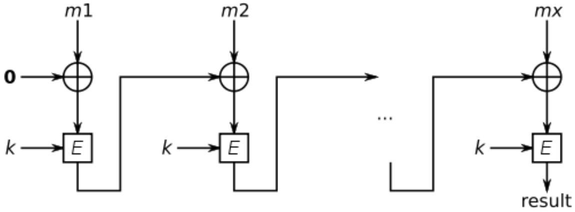


- given $(m_1, H(k||m_1))$, it is easy to forge the tag $H(k||m_1||PB||m_2)$ for $m_1||PB||m_2$
  - PB is the padding, used when computing $H(k||m_1)$.
  - m_2 can be any message.

Question: How about $H(m||k)$?

CBC-MAC

- Another MAC obtained from block ciphers
- Very much like the CBC encryption mode,
- IV is often defined as 0
- Only the last block of ciphertext is retained as MAC
- Security level of $\frac{n}{2}$-bit where n is the block size



Be Cautious With CBC-MAC

- The sender needs to tell the receiver the length of the message
  - either a pre-agreed fixed length
  - or this information has to be send and authenticated with the message itself
- Otherwise an adversary can forge a MAC easily
  - M_1, M_2 are all messages 1 block long
  - The adversary queries M_1 and receives its CBC-MAC
  - The adversary queries $M_3 = CBC\text{-}MAC_K(M_1) \oplus M_2$ and receives its CBC-MAC
  - The adversary can forge a message $M_1||M_2$ where
    - $CBC\text{-}MAC_K(M_1||M_2) = CBC\text{-}MAC_K(M_3)$

Reading

- Cryptography made simple §14.5,14.7
- Cryptography theory and practice §4.4