Length Extension Attacks

Lecture notes of Alexander Wood
awood@jjay.cuny.edu

John Jay College of Criminal Justice

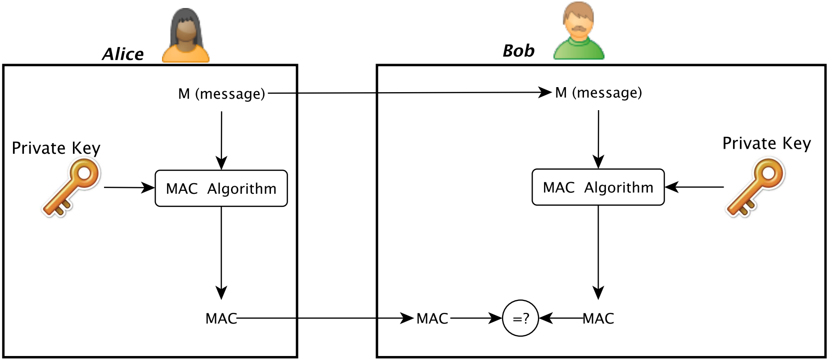
MACs: Review

A **message authentication code (MAC)** is a key-dependent one-way hash function.

They satisfy the same properties as one-way hash functions. In addition they have a **key**.

MACs are used to **authenticate** files between users. It checks its **authenticity** (confirms the sender) as well as its **integrity** (it has not been tampered with).

MAC Visualization

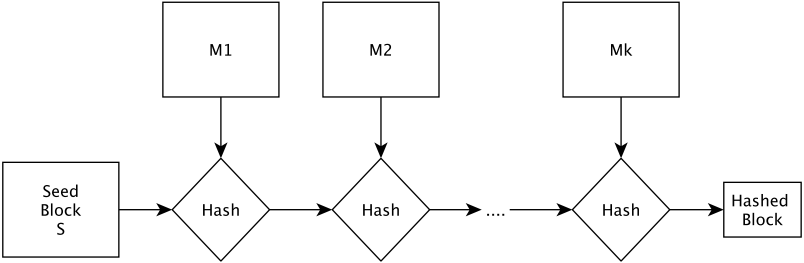


MAC Algorithms

- `KeyGen` - generates a key 1^n uniformly at random.
- `Sign` - Alice inputs her key k and message M, receives output t (tag).
- `Verify` - Bob verifies the authenticity of Alice's message.

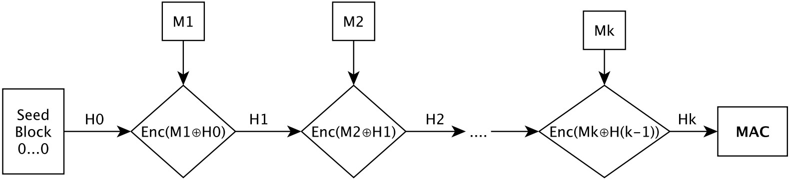
Merkle-Damgard Construction

We should now be familiar with the **Merkle-Damgard Construction** of a hash function.



CBC-MAC

MACs can be constructed similarly by including a **key.** Recall that this is called **CBC-MAC.**



CBC-MAC

CBC-MAC is secure for fixed-length messages *if the underlying block cipher used is secure*.

Length-Extension Attacks

Today we look at a different attack, the **length extension attack**. This attack works specifically against **Merkle-Damgard based hashes** which are inappropriately used as MACs.

Thus, algorithms like MD5, SHA-1, and SHA-2 are susceptible. SHA-3 and HMAC are not susceptible to this form of attack.

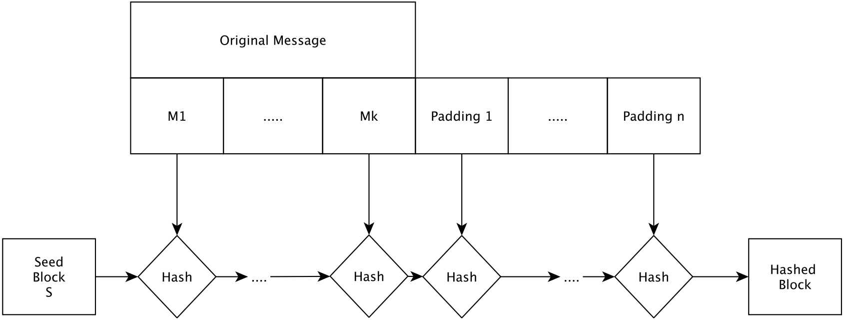
Length-Extension Attacks

A length extension attack is carried out as follows. Let H be a hash function and M_1 a message.

- An attacker, Eve, intercepts $H(M_1)$, the hash of message 1. Let L be the length of M_1.
- Eve calculates $H(M_1 \| M_2)$ for a message M_2 of her choosing.
- The value $H(M_1 \| M_2)$ now verifies as signed by the original sender.

SHA-1: Length Extension Attack

Recall that SHA-1 uses 512-bit blocks. In order to send a message, it is first **padded** in order to be a multiple of 512 bits.



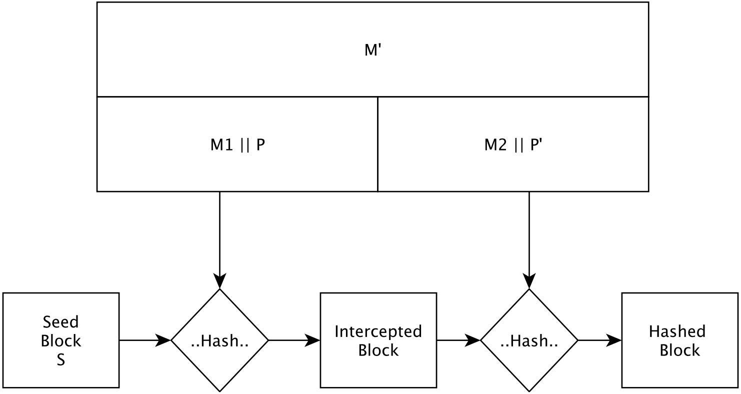
SHA-1: Length Extension Attack

Eve intercepts the hashed block, H. She knows:

- The hashed block H, which is the hash on the message $M_1 \| P$ for some padding P.
- The message $M_1 \| P$.
- The length of the key K.

SHA-1: Length Extension Attack

Let $M' = M_1 \| P \| M_2$. Pad this further to make it a multiple of 512 bits. Eve can now compute the hash of $M' \| H'$.



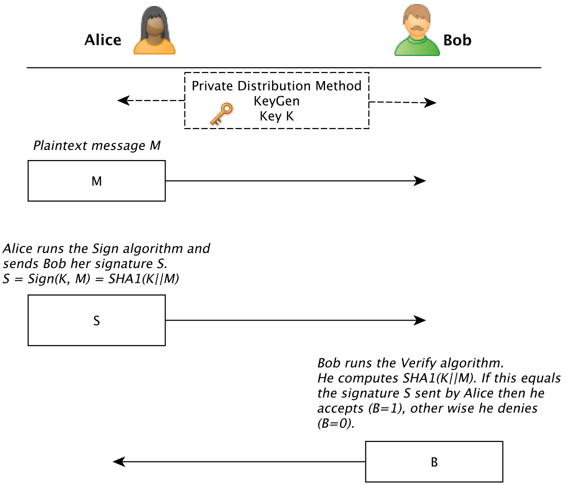
SHA-1: Length Extension Attack

How can Eve use this to her advantage? Suppose a MAC is built using SHA-1.

- `Sign`: Alice signs a message M with a key K by computing the value $S = \texttt{SHA1}(K \| M)$.
- `Verify`: Bob verifies the message M by computing $\texttt{SHA1}(K \| M)$ and verifying that this is is equal to the signature S sent by Alice.

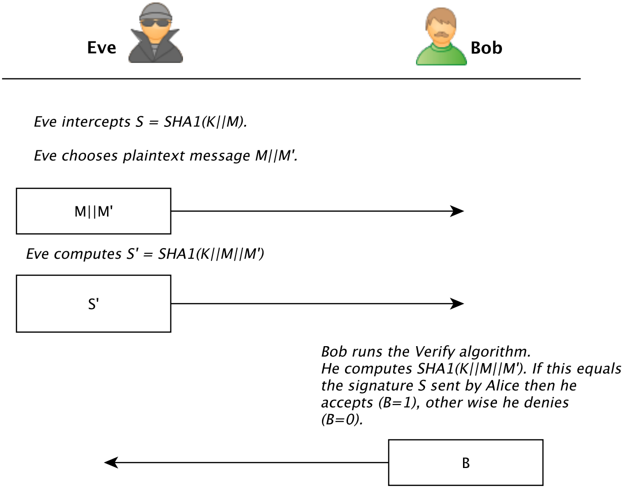
SHA-1: Length Extension Attack

This MAC protocol would work as follows.



SHA-1: Length Extension Attack

Eve could attack as follows.



How to protect against this attack

Avoid the Merkle-Damgard construction! Instead use something like HMAC (with nested hashing).

Alternatively append a message number or a timestamp to the beginning of your message so that extending that message is pointless.

References

- *Applied Cryptography* By Schneier, Chapter 18
- *Cryptography Engineering* by Schneier, Ferguson, Kohno, Chapter 6
- `https://lord.io/blog/2014/length-extension-attacks/`