Digital Signatures

Rohit Musti

CUNY - Hunter College

March 23, 2022

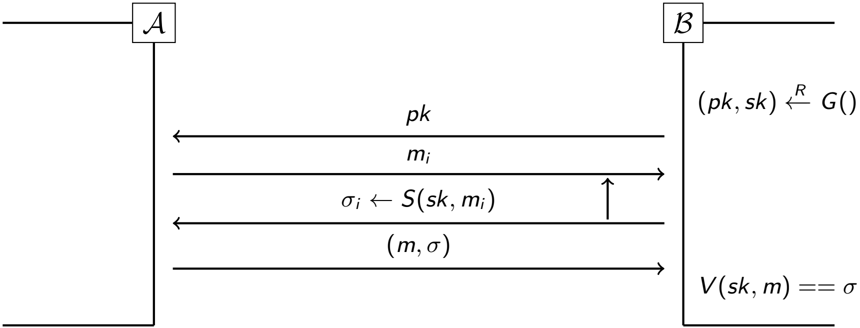
Table of Contents

1. Overview

Example Use Case

- Let's pretend you are a software company
- Occasionally, you will distribute updates to your software
- Customers will need a way to verify that the software is really from the company without sharing a secret key with every single customer
- Enter digital signatures!

Digital Signatures Attack Game



Non-Repudiation

1. Non-Repudiation: there is no way to deny that the signature was not produced by someone with access to the secret key
2. This sounds legally useful right? Unfortunately not, it is very easy for someone to claim the public key isn't theirs or their secret key was exposed/leaked/stolen by a hacker.
3. Even more maliciously, Alice could have leaked her key right after or before sending a signatunre

Limits

1. Binding Signatures: Easy for a signer to issue multiple yet contradicting signed messages
2. Duplicate Keys: Given a signature σ for a public key pk, could I generate a pk', sk', m that produces $\sigma' == \sigma$
3. Future HW Queston: Is there an easy way to get around duplicate keys attacks???

2. Trapdoor Construction of Digital Signatures

Construction

$$S(sk, m) = y \leftarrow H(m), \sigma \leftarrow I(sk, y)$$

$$V(pk, m\sigma) = y \leftarrow F(pk, sigma), y == H(m)$$

Security

- In order to prove security we most model the hash function as a random oracle
- This means that y is essentially a random point in the space, and there is no way for an attacker to forge its signature
- Without this hashing, an attacker could trivially generate secure signatures! Future HW question: How???