Massachusetts Institute of Technology
6.857: Network and Computer Security
Professors Ronald L. Rivest and Yael Tauman Kalai

Handout R6
March 13, 2020
Adrian Sy

Recitation 6: Digital Signatures and Blockchain

1 Digital Signatures

1.1 Digital Signatures Definitions

Cryptographic primitive in a public key setting for verifying authenticity / integrity. Analogous to MAC schemes in the symmetric key setting.

Defined by the following three algorithms:

- $\text{Gen}(1^k) = (\text{sk}, \text{pk})$ - generates a secret key and public key pair
- $\text{Sign}(m, \text{sk}) = t$ - generates a signature (also referred to as tag t) for message m
- $\text{Ver}(m, t, \text{pk})$ - verifies if the tag t is a valid signature for message m

The strongest notion of security for signature schemes is existential unforgability under chosen message attack. Similar to the MAC definition for security (R4), this means any PPT adversary who has access to a signing oracle cannot generate a signature for any new message.

1.2 Applications of Digital Signatures

The applications of Digital Signatures fall under three major categories:

1. Authentication - the tag t for message m proves that message m was created by the party with public key pk and not anyone else (because the signature is unforgeable).
2. Integrity - since the tag t only verifies for message m, then modifying m is impossible without creating a corresponding t' to the modified m'
3. Non-repudiation - the party who generated pk cannot deny having made the message m with tag t since t could not have been generated by anyone else.

1.3 Digital Certificates

In cryptography, digital certificates are used as a proof of identity tying an entity to a public key that it publishes for digital signature schemes.

Typically requires a trusted party, known as a Certificate Authority (CA), to verify the identity of the public key owner. These CAs are fixed entities and their public keys are installed in most browsers (or other relevant network interface) and trusted by default. The way this is used in protocols like HTTPS is described below:

1. A website owner who has just registered their domain creates a certificate containing their website domain, public key/s (for encryption/ digital signatures) and other meta-information.
2. Assuming the website is valid, a CA can approve the message by creating a digital signature for the certificate which includes the CA's identity. This signed certificate is stored in the website servers.
3. When a new user enters the website, the website provides the signed certificate to the user. If the CA listed in the certificate is trusted by the user's browser, the user's browser can verify if the signature for the certificate is valid and if the listed domain matches the website.

4. The browser then uses the public key to perform key exchange with the server to generate a short-term symmetric key for symmetric key encryption for all future communications in that session.

This avoids man-in-the-middle (MITM) attacks where an adversary with control over the user's network would pretend to be one of the websites the user visits and switch the actual website's public key with their own key. They are unable to pose as a different entity because they cannot forge the CA's digital signature.

2 Blockchain

2.1 Blocks

The underlying structure for Bitcoin and several other crypto currencies is the blockchain which is made up of blocks. Each block contains the following:

- Meta-data (e.g. timestamp)
- List of signed transactions
- Hash of previous block
- Proof of work (e.g. hash of previous items plus a nonce value satisfies some requirement)

2.2 Protocol

From a high-level perspective, the protocol proceeds as follows:

1. The blockchain starts with a genesis block that is broadcasted to everyone on the network.
2. Whenever a transaction occurs, the one giving bitcoin creates a transaction record, signs it and broadcasts it to the network.
3. Miners can generate blocks by accumulating several valid transactions that are not yet on the chain and showing proof of work. The block is then broadcasted.
4. Each user on the network adds the first block that gets broadcasted and assuming enough parties are honest, consensus will be reached.

2.3 Security

2.3.1 Digital Signatures

Each transaction has to be signed so that malicious miners cannot forge transactions to give themselves more cryptocurrency. Since the transaction records are unforgeable, the worst a malicious miner can do is to not include the transaction into a block. However, the miners are incentivized to do so because transactions typically include a small portion of currency that the miner can give themselves without breaking the signature validity.

2.3.2 Sybil Attacks

Sybil attacks refer to attacks which can be exploited by making multiple accounts (since the only identifier is a public key). For example, if mining a block is easy, a miner can simply create multiple accounts to mine simultaneously. The blockchain avoids this by using proof of work to limit the each user to the amount of computational power available to them.

2.3.3 Hash Functions

Hash functions are used in two places: to identify the previous block and to provide proof of work.

- To connect to the existing blockchain, each new block must contain the hash of the block preceding it. This prevents a malicious entity from making a block in advance and just appending it to the current chain, this entity must also be faster than all the other miners in the system. This requires the hash function to be collision resistant.
- To provide proof of work, the miner must find a nonce such that when the nonce value is combined with the other data and hashed, the resulting hash value is part of a limited subset of the possible output values, e.g. has k leading 0 bits. This requires the hash function to be pseudorandom.