CSE 5852, Modern Cryptography: Foundations Fall 2016

Lecture 26

Prof. Benjamin Fuller *Scribe: Tham Hoang*

1 Last Class

Last class we introduce the Schnorr identification scheme [Sch91]. The scheme is to provide a protocol that allows Alice (A - Sender) to validate herself with Bob (B - Receiver) in secret channel. It required the interaction between Alice and Bob to work on the tuple values $(y, r, rx + y)$ in $\mathbb{Z}_p^*$. To begin with, Alice need to broadcast her public key that has been generated (p, g, g^x) to everyone. Bob at the end check the correctness: $g^{xr+y} = (g^x)^r.g^y$. The details are on the following:

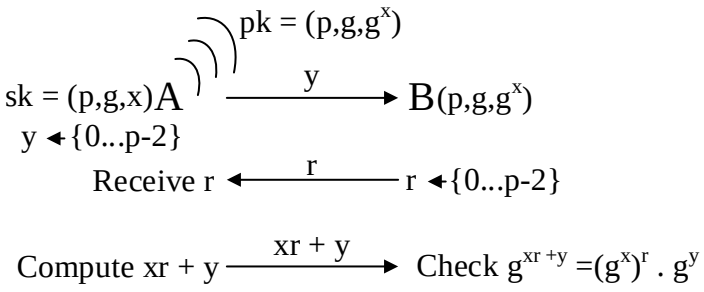

Figure 1: Schnorr identification scheme.

We want to use this identification scheme for digital signatures. Here, A want to create transcript, no one else can impersonate the protocol. Let us review the digital signature scheme.

Signature scheme $\pi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vfy})$

$\mathsf{Gen}(1^n) \to (pk, sk)|pk|, |sk| \geq n$

$\mathsf{Sign}_{sk}(m) \to \sigma$ (Sign may only be defined for certain messages).

$\mathsf{Vfy}_{pk}(m, \sigma) \in \{0, 1\}$ (This function is deterministic).

Correctness: $\mathsf{Vfy}_{pk}(m, \mathsf{Sign}_{sk}(m)) = 1$

The digital signature detail is as following:

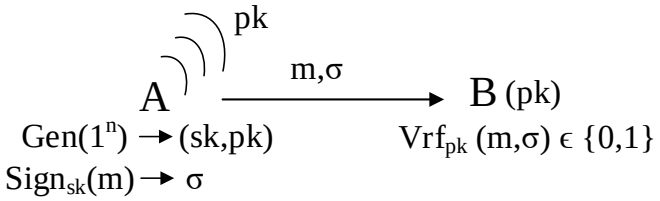

Figure 2: Digital Signature.

In digital signature scheme, there is no sharing key ahead of time. Symmetric key does not require interactions between $\mathcal{A}$ and $\mathcal{B}$. Moreover, conversation is required for identification scheme which digital signature does not. Today class has two parts: 1) to study the transformation process to turn identification scheme to digital signature and 2) what is bitcoin problem and how to solve it.

2 Fiat-Shamir transformation [FS86]

Idea: "Use m as the challenge r in the protocol".

Objections:

- Alice may not generate g^y before seeing m.
- Alice can "forge" arbitrary distribution on r.

F-S strategy:

- Pick a hash function: $h : \{0,1\}^* \to \{0 \dots p-2\}$ with * arbitrary length.
  treate h as completely random function (RO: Random Oracle).

F-S signature:

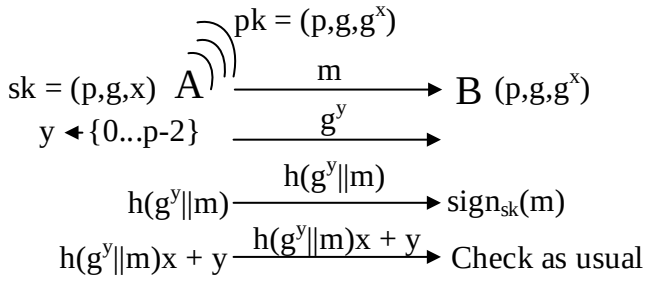

Figure 3: F-S signature scheme.

3 Bitcoin

We consider a scenario for classical banking.

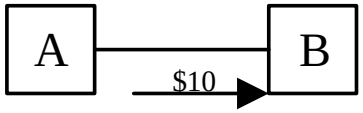

Figure 4: Classical banking.

What $\mathcal{B}$ need to do:

- Authenticate: $\mathcal{B}$ need to verify the identification of $\mathcal{A}$
- Balance check: $\mathcal{B}$ need to check the balance of $\mathcal{A}$

When $\mathcal{B}$ finished at least two above processes, the transaction requested from $\mathcal{A}$ may get further processing.

Issues:

- Customers requires trust in $\mathcal{B}$
- Centralized: The transaction through the bank $\mathcal{B}$ is much more different from Cash. It has to go through the bank in a certain place.
- Not "anonymous": Customers requires privacy (e.g, how much money they have). This one is not covering in this lecture.

Ideal solution: Bitcoin is a decentralized virtual crytocurrency. For simplicity, we assume that it is possible to access entire transactions recoded in a "ledger" in real time. There are some requirements for Bitcoin:

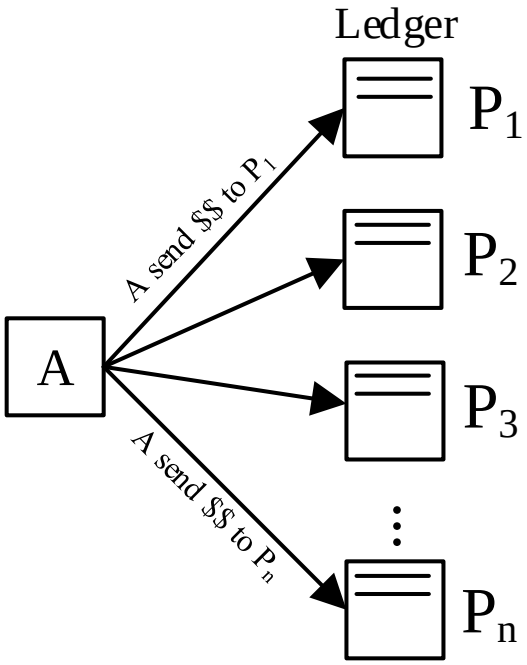

Figure 5: Bitcoin ledger.

- All participants have to maintain the ledger.
- Transactions are known to all parties.

Since all ledgers have to be updated at the same time so we may have issue with consistency. There are several attacks that could harm the Bitcoin system. See more on weakness of Bitcoin [Wik14]. One solution is to apply payment prior to a party can play or involve in any ledger or require the party to demonstrate the ability of computation.

Double Spending: A double spend is an attack where the given set of coins is spent in more than one transaction. Bitcoin protects against double spending by verifying each transaction added to the block chain to ensure that the inputs for the transaction had not previously already been spent.

Sybil Attack [Dou02]: A Sybil attack is a type of security threat when a node in a peer-to-peer network claims multiple identities. When an attack can control clients, it is very likely to connect only to attacker nodes.

Bitcoin solution [Nak08]: We consider following hash function:

$$60\ leading\ zeros \leftarrow h \begin{cases} \frac{t_1 \ldots t_n}{N} \\ TOP \end{cases}$$

This scheme is mostly trial and error: Adversary is successful if he can find N for t_i. It usually follows the longest path of ledgers.

References

[Dou02] John R Douceur. The sybil attack. In *International Workshop on Peer-to-Peer Systems*, pages 251–260. Springer, 2002.

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Conference on the Theory and Application of Cryptographic Techniques*, pages 186–194. Springer, 1986.

[Nak08] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

[Sch91] Claus-Peter Schnorr. Efficient signature generation by smart cards. *Journal of cryptology*, 4(3):161–174, 1991.

[Wik14] Bitcoin Wiki. Denial of service (dos) attacks. en. bitcoin. it/wiki. *Weaknesses, Accessed on*, 26, 2014.