Privacy and Security of Data in Communication

MoshAhmed
Edmodo
2019-10-04

Consumer Privacy Issues

- Cookies and behavioural targeting
- Consumer privacy issues on social media
- Using private data in ML/AI
- Communication

Information in Protocols -1

Example

- Alan: Do you know Eve's phone?
- Bob: Yes.
- Alan: Prove it.
- Bob: 650-555-1234

Is this a good protocol?

Problems

- Now Bob knows the number
- Can be used only once

Protocol 2

Example

- Alan: Do you know Eve's phone?
- Bob: Yes.
- Alan: Prove it.
- Bob: 650-…

Is this a good protocol?

Problems

- Bob knows part of the number each time
- Can be used only a few times

Protocol 3

Example

- Alan: Do you know Eve's phone?
- Bob: Yes.
- Alan: Prove it, what is last digit?
- Bob: 4

Is this a good protocol?

Protocol 4

Example

- Alan: Do you know Eve's phone?
- Bob: Yes.
- Alan: Prove it, what is the sum of digits?
- Bob: 36

Is this a good protocol?

Problem

- Number can be computed after many rounds.

One Way Function

Hashing/fingerprinting, given the knowledge (account number) you can compute the sum (fingerprint) but you can't get the account number from the sum.

Protocol 5 – one way hashing

Example

- Alan: Do you know Eve's phone?
- Bob: Yes.
- Alan: Prove it, what is the sha2 hash digits?
- Bob: 2dd619305603f60f68bc…

Is this a good protocol?

Protocol 6 – hash + challenge response

Example

- Alan: Do you know Eve's phone?
- Bob: Yes.
- Alan: Prove it, what is the first 3 hex digits of the sha2 hash digits?
- Bob: 0x2dd

Is this a good protocol?

Pros/Cons

- Can be used only once
- Hash to number can be computed using massive “Rainbow tables” (map: strings -> hashes).

Zero knowledge proofs

Zero-knowledge proof or *Zero-knowledge protocol* is a method by which one party (the *prover*) can prove to another party (the *verifier*) that a given statement is true, without conveying any information apart from the fact that the statement is indeed true

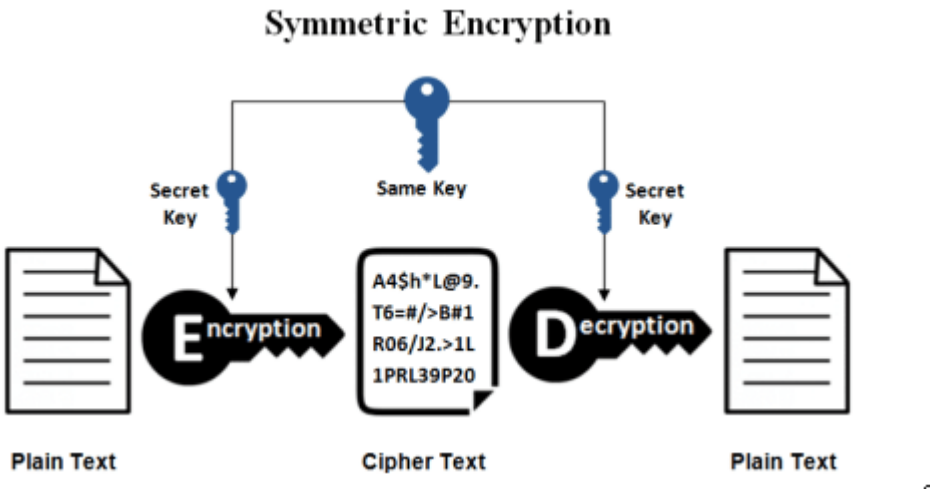
Pros/Cons

- Some math required
- Protocol can be used forever with no loss of private information.
- Bob can prove he knows the number without revealing anything (0 knowledge)
- Used by ssh login – server challenges the user to prove she has the private-password, by sending an encrypted random number (challenge response).

How to transmit knowledge when others are listening

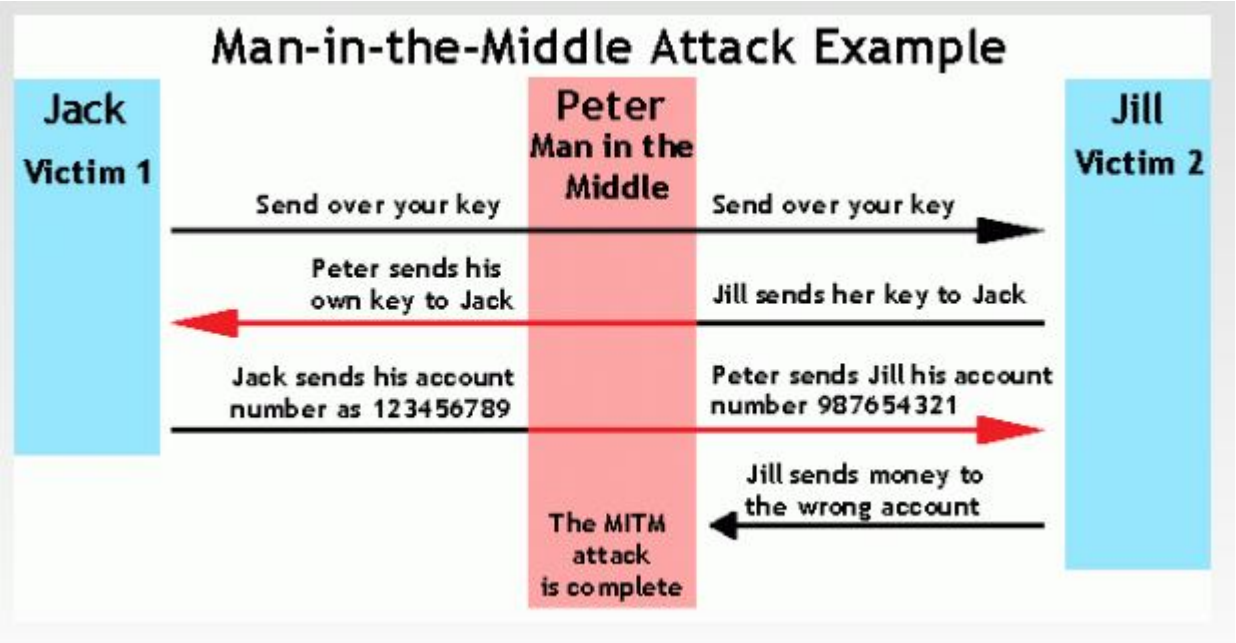
MITM attack, postman can eat the chocolate?

Solution 1 – communication over insecure channel



Problem – how to exchange keys?

MITM attack



Solution 2 – communication over insecure channel

Send a box of chocolate via rogue courier?

A. Sends locked box with chocolate

B. Puts his own lock and send back to A.

A. Removes his own lock and sends it to B with B's lock.

Is this solution good?

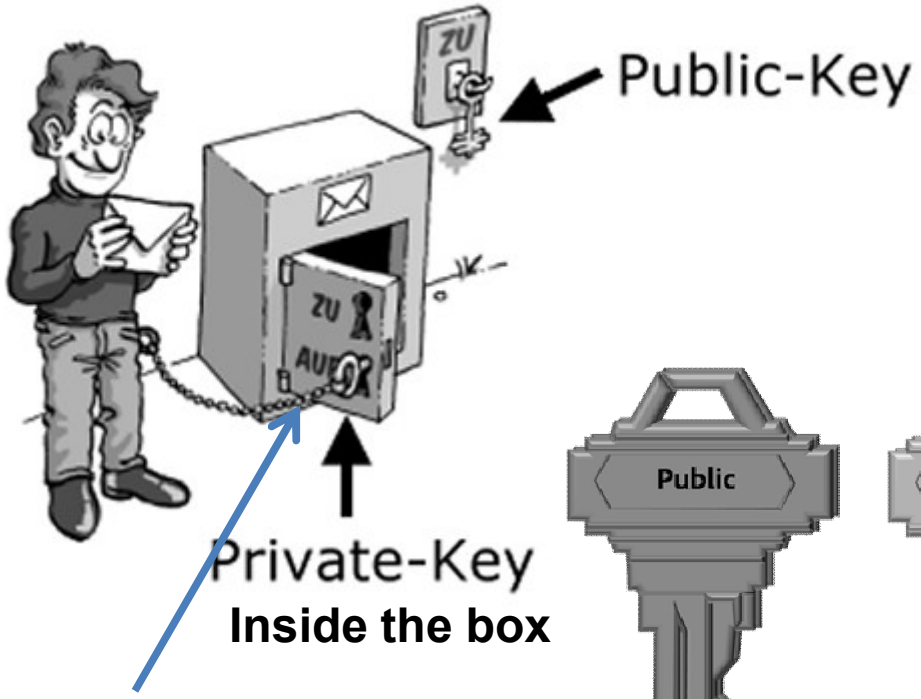
Or-Locks [Series]
(opens with any one key)

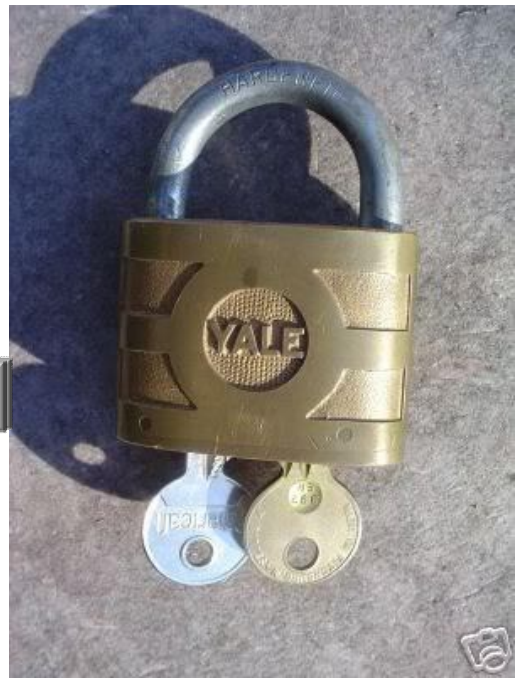
And-Locks [Parallel]
(open needs all keys)

Shamir Secret Sharing

- (k of N) keys needed to unlock.
- Instead of one key, you require minimum k keys to unlock the secret.
- Having less than k keys is no use.

Lock analogy for PK





Application - SSH

- You never send your private key to login server
- Server sends a challenge - a random number encrypted with your public key
- Only you can open it and send the answer back to the server and prove you have the private key.
- Server lets you login.

Ssh protecting private key

- Your private key is kept safe with a passphrase.
- Private key never leaves your machine
- On local machine private key is kept with ssh-agent (private process), you just ask ssh-agent to compute the response to the challenge.
- In Credit-card has private key on a chip, private-key never leaves the chip on the card, only response are available from the card.
- Can be used million times without loss of info.

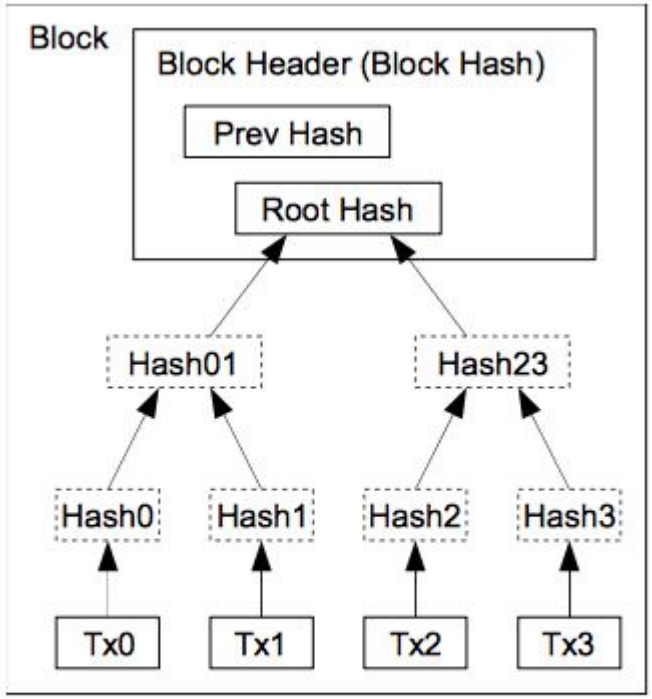
Problem: How do you find out the Average salary with no one revealing their own salary?

Solution

- Everyone adds their own secret random number to their information.
- All the numbers are added up in some order.
- Everyone subtracts their own random number from the total in different order.
- Divide the total by number of participants.
- Assume no one is giving wrong information.

Application – Block Chain

- Merkle Trees, distributed hashes of data

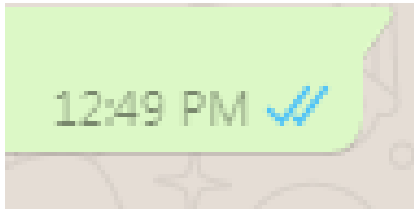


Transactions Hashed in a Merkle Tree

Merkle Trees in Bitcoin

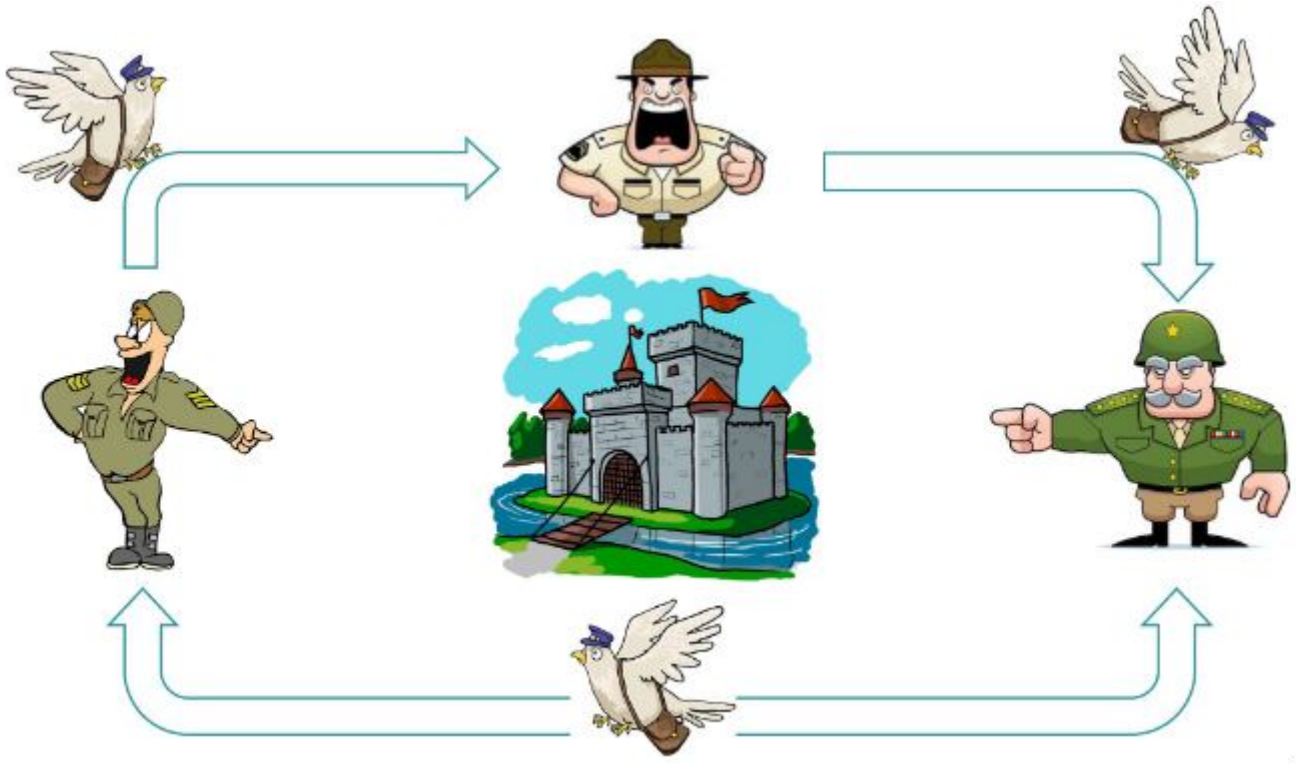
Common Knowledge

- Common knowledge is: I know that you know that I know that you know that …
- Achieving common-knowledge overcoming uncertainties.
- Reaching consensus over unreliable networks.
- Products: Whatsapp blue-ticks, last-seen.



Byzantine General's Problem

- Two armies A1 & A2 want to launch a coordinated attack on camp B, if A1 and A2 are not coordinated, A1 and A2 will be defeated.

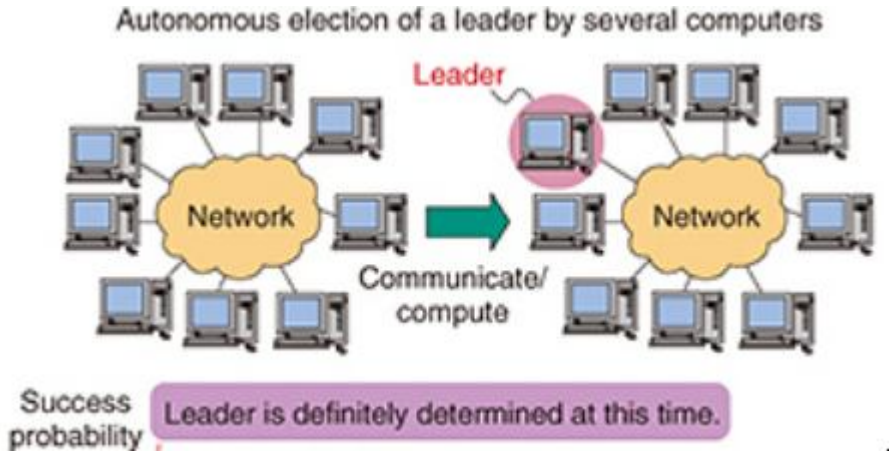


Byzantine General's Problem

- Messaging by running courier who may be lost.
- A1 -> A2 we will attack at 4am
- Now A1 waiting to confirm A2 got the message, so A2 replies "OK"
- Now A2 waiting to confirm from A1 that A1 got the ack and only then they can attack.

Leader Election in a network

- Generally Networks have masters and replicated slaves (which keep copies of the data) in case some machines fail.
- What happens when the master goes down?
- Rest of the machines must figure out that master is down (or not connected or slow) and elect a new master.
- What if the network gets disconnected at a hinge?



Paxos

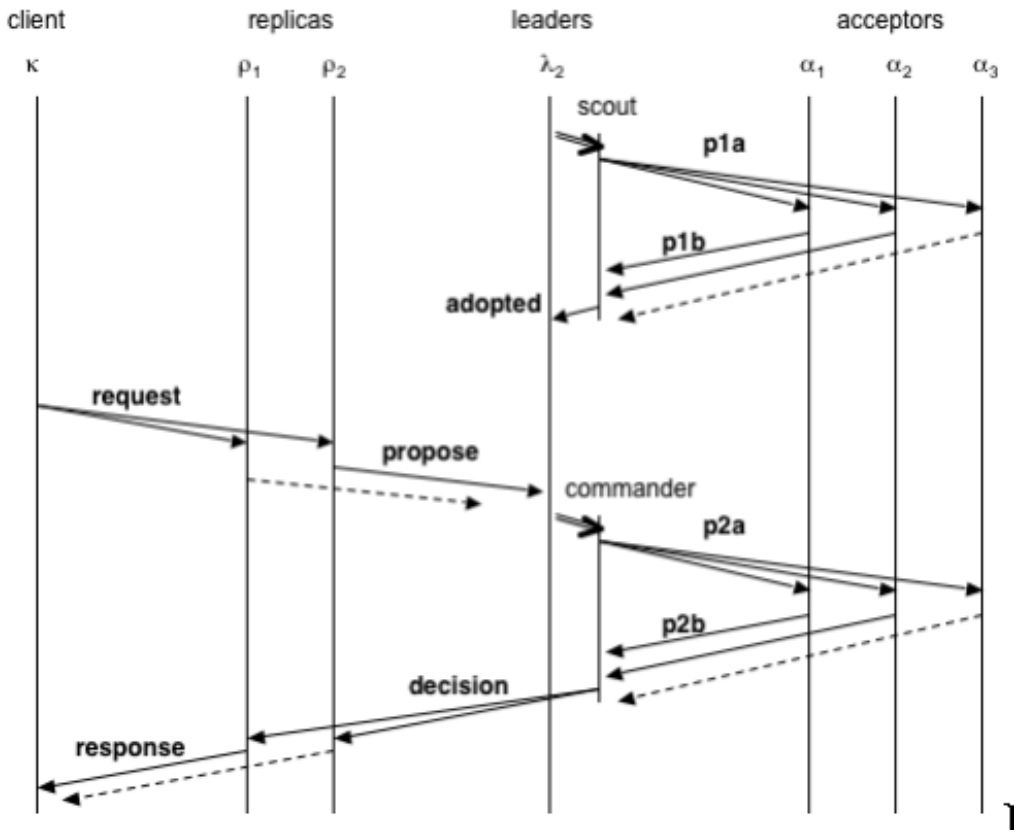
- **Paxos** is a family of protocols for solving consensus in a network of unreliable processors (that is, processors that may fail). Consensus is the process of agreeing on one result among a group of participants.
- This problem becomes difficult when the participants or their communication medium may experience failures.

Paxos – property guarantee

1 Validity (or *non-triviality*) Only proposed values can be chosen and learned.

2 Agreement (or *consistency*, or *safety*) No two distinct learners can learn different values (or there can't be more than one decided value)

3 Termination (or liveness) If value C has been proposed, then eventually learner L will learn some value (if sufficient processors remain non-faulty).



Paxos Protocol: The time diagram shows a client, two replicas, a leader (with a scout and a commander), and three acceptors, with time progressing downward. Arrows represent messages. Dashed arrows are messages that end up being ignored. The leader first runs a scout in order to become active. Later, when a replica proposes a command (in response to a client's request), the leader runs a commander, which notifies the replicas upon learning a decision. https://cs.nyu.edu/courses/fall18/CSCI-GA.3033-002/papers/paxos-renesse.pdf

Google’s Chubby

- Chubby based on Paxos, used by Borg cluster, Bigtable, GFS to synchronize multiple several thousand linux servers in a data center
- Dozens machine fail in a day, without impacting any user.

Applications

- Banking (identity and accounting)
- Block chain (distributed anon audit trails)
- Voting (audit trail, anonymity).
- Data analysis (dremel, sawzall), maintaining privacy.
- AI and ML (machine learning), without leaking personal info in output.

References

- **Wikipedia**: Zero knowledge proof, RSA, Public Key, Key Exchange, Diffie Hellman, Leader election, Paxos, Shamir Secret Sharing, Chubby
- *Applied Cryptography* by Bruce Schneier.
- *Forever Undecided, Puzzle Guide to Godel*, by Smullyan.
- Bitcoin lectures on youtube by Felton, Princeton.
- *Everybody Lies*, by Seth Stephens-Davidowitz (big data)
- *Freakanomics*, on small data analysis.
- **Basics CS/Math**: Algorithms by Cormen, Math for CS by Lehman, MIT, Introduction to Probability, by Bertsekas.

For details, search above terms in Google.

Questions?

Thank you – email: moshahmed