Techniques for Anonymity

Privacy-enchancing technoques

Level	What to protect	Method
3	Existence of message	Steganography
2	Metadata of message	Privacy-enhancing technologies
1	Content of message	Encryption
0	Nothing	None

Anonymity in communications

- One of the ways to protect privacy is to make a communication **anonymous,** so an adversary that
  - can monitor and/or compromise certain parts of the systems
  - would not be able to match a message (request) sender with the recipient (sender-recipient matching).
- Most widespread methods for anonymity in the Web communications based on the idea of **third trusted party** serving as **anonymizer** (special proxy server)

Anonymizer

- Early example: Anonymizer.com (many available now)
- Upon a request from an user anonymizer fetches the web page and displays is within your browser
- In doing so, anonymizer does not leave information about your request on the web-server: it re-directs your request, replacing all sensitive information (IP address, etc) with its own details
- Additionally it may provide
  - encryption of traffic between an user and itself
  - Blocking and removing potential active privacy and security threats: web bugs, spyware, viruses, etc

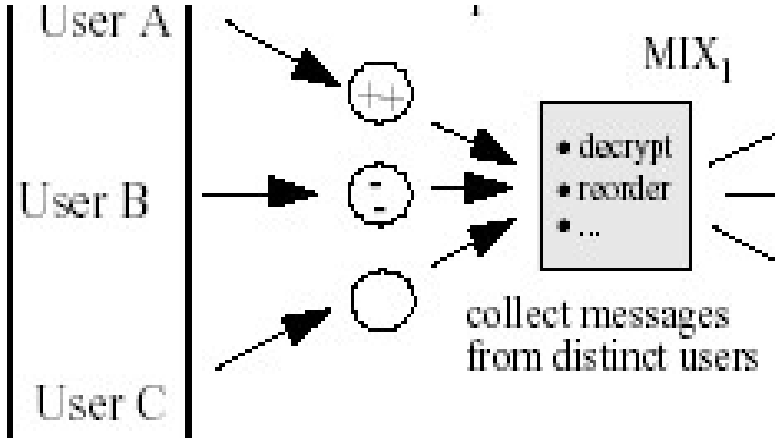
- More recently: VPN (Virtual Private Networks) implement the idea of anonymizer

Anonymizer

- Good protection and reasonable cost
- Privacy protection is based on the trusted central proxy
- Central proxy “knows everything” about communication – attacks by “insiders” are possible

Mix-Networks

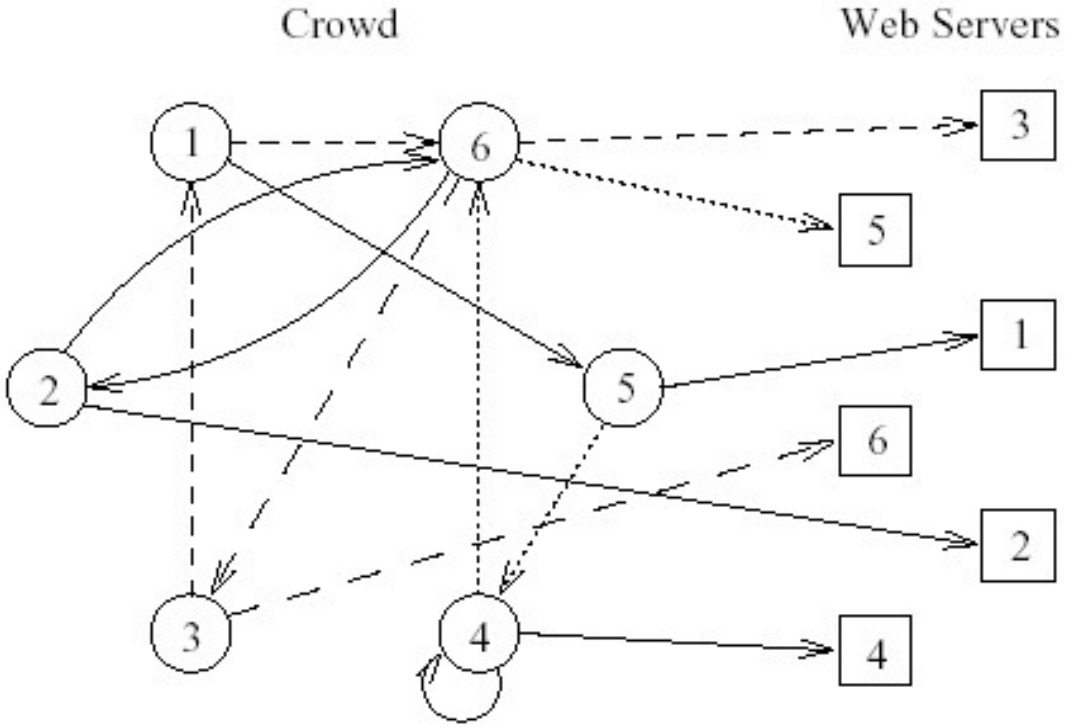
- D.Chaum (1981):
- A mix node is a node in the network that takes a number of incoming messages (packets), modifies them and output in a random order



Mix-networks

- The mix nodes can be used for anonymous communication as follows:
- The message will be sent through a sequence of mix nodes (a route) p_1, p_2, p_3, …p_d. The user encrypts the message with node p_d key, the result encrypt with the node p_{d-1} key, etc
- Every mix node receives several messages, decrypt them, re-order and send to the next nodes in the route
- Every nodes "knows" only previous and the next node in the route => compromising a single, or even several (not all) mix nodes does not give an attacker an information about sender-receiver matching
- It is more expensive than anonymizer solution but gives more privacy protection.

Crowds



Paths in a crowd. Picture by M.Reiter and A. Rubin

Anonymity, further approaches

- M.Reiter, A. Rubin, 1998, Crowds: anonymity for Web Transactions
- Based on the idea “blending into a crowd”, that is hiding one’s actions within the actions of many others
- To execute a web transaction a user first joins a “crowd” of other users;
- Then the user’s request to a web server is passed to a random member of the crowd;
- That member can either submit the request to the server, or forward it to another randomly chosen member of the crowd and so on.

Privacy protection by the crowd

- When the request submitted to the end server, it is submitted by a **random** member of a crowd, so identity of an initiator is hidden (“in the crowd”) from an external observer
- Members of the crowd cannot identify initiator as well, they “just passing requests”

Crowds vs anonymizers and mixes

- Unlike an anonymizer crowds provide no single point, where an attacker can compromise anonymity of all users
- Crowds does not provide anonymity against a global adversary able to oversee all communications. In contrast, mix-networks protect anonymity in that case.
- Crowds admit very efficient implementations in comparison with mixes: no encryption/decryption operations, no inflation of message lengths.

Tor: anonymity online

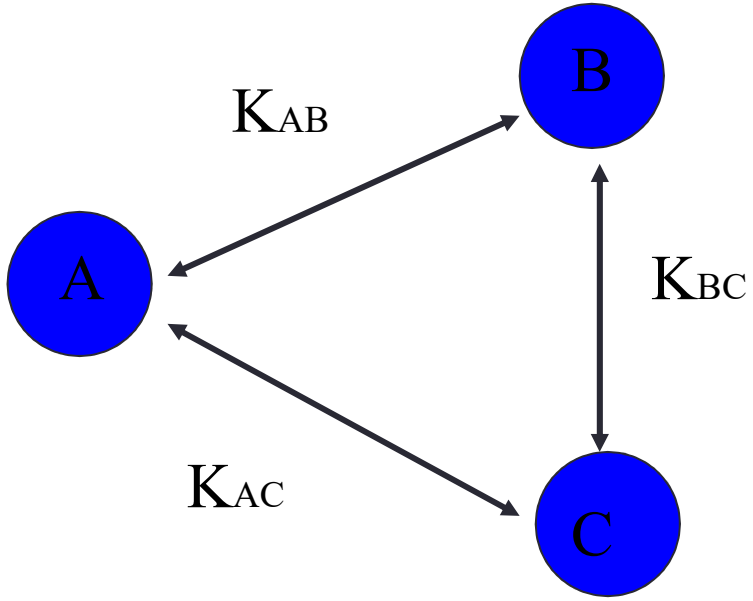
- Tor (from The onion routing project, originated in the US Naval Research Lab) – practical solution for anonymity protection : www.torproject.org
- It is free and open source and available for major platforms: Windows, Mac, Linux/Unix, Android
- Can be used for web browsing and instant messaging, prevents people from learning your location or browsing habits

Tor: main principles

- A combination of (variants of) mix-network and crowd mechanisms
  - Using a set of relay nodes (~crowd); currently >6000
  - Routing using random choice (similar to crowds)
  - Encrypted connections between neighbouring nodes (similar to mix-networks)
  - It uses public key cryptography to share the secret keys between neighbouring nodes => uses the shared secret to perform symmetric encryption in further communications between neighbours
  - temporarily available virtual channels

DC-networks

- D.Chaum, 1988: **D**ining **C**ryptographer networks

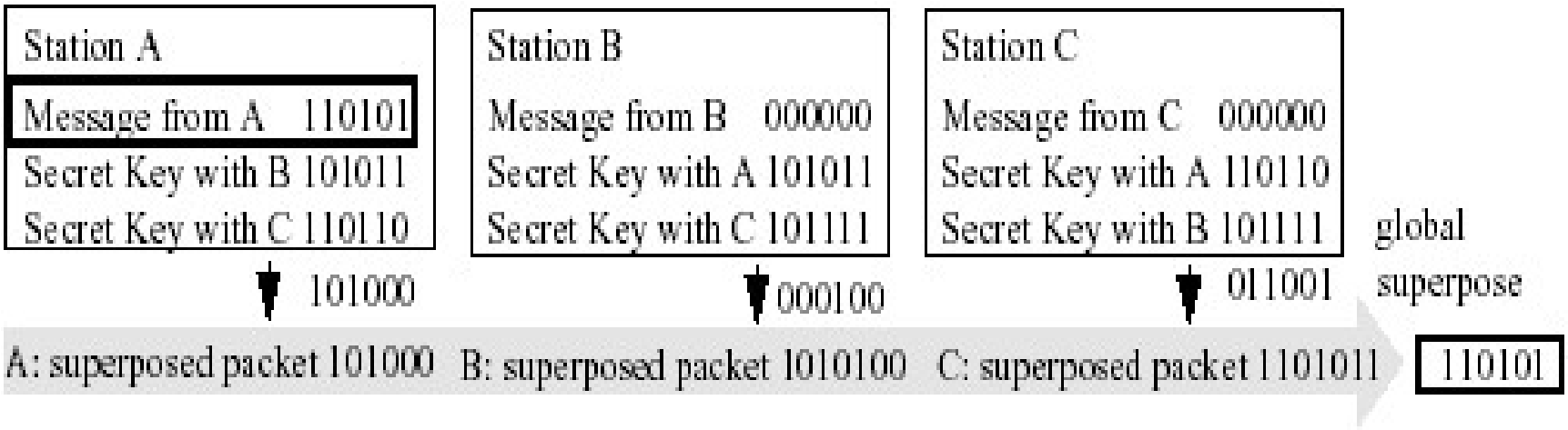


• At the preliminary stage between some pairs of nodes (at the picture between all) secret keys (sequences of bits) are exchanged

DC-networks

- To send a message M (sequence of bits), a node, say A, *broadcasts* the value **$(M +_2 K_{AB} +_2 K_{AC})$**, i.e. superposition of the message and all keys of A, here **$+_2$** stands for bitwise addition modulo 2 (or XOR operation)
- All other nodes broadcast superpositions of all their keys. So, B broadcasts **$(K_{AB}+K_{BC})$** and C broadcasts **$(K_{AC}+K_{BC})$**
- All nodes then superpose all received messages and get **$(M+_2K_{AB}+_2K_{AC}+_2K_{AB}+_2K_{BC}+_2K_{AC}+_2K_{BC}) = M$**
- (the initial message !!!)

DC-network



A message sent by A in the DC-network.

Picture by A.Pfitzmann

Anonymity by DC-networks

- DC-networks provide for *sender* anonymity because an adversary is unable to decide whether the packets he may observe contain a message or not;
- DC-networks can be used in combination with other mechanisms, such as mix-networks to enhance anonymity
- A major drawback is that DC-Networks require the preliminary stage exchanging the secret keys between participants
- Every round of communication requires a new set of keys
- Every node needs to participate every time a message is broadcasted => high load on the nodes => impractical in large networks

Recent developments in DC-networks

- Dissent system (~2012):
  - Scalable to thousands nodes
  - Client-server architecture with several servers and small groups served by a server
  - Retro-active blame mechanism to deal with *jamming*
  - XOR together with more complicated *group multiplication* operations are used
  - See further details at *dedis.cs.yale.edu/dissent*