

Claims

1. A system, that enables NAT traversing using UDP hole punching in a network comprising:
 - a. At least 1 client devices connected to the global network using NAT
 - b. At least 1 client devices connected to the global network and having publicly available IP addresses
 - c. Software for each client implementing a protocol to perform UDP hole punching using pre-defined addresses of clients with publicly available IP addresses, distribute information about clients having publicly available IP addresses, clients successfully passed the NAT traversing and clients using relay servers with relay servers' information
2. A system from claim 1, which uses Distributed Hash Tables to spread connection information across network
3. A system from claim 1, also comprising at least one server, which runs the same software as clients and has at least 1 "white" IP address which is known to the clients
4. A system from claim 1, also comprising a system for local blacklisting of clients, accidentally or maliciously spreading wrong connection information based on 2-layer signature checks of connection information and connection information synchronization response
5. A system from claim 1, also comprising a system for global blacklisting of clients, based on spreading information about their IDs, accidentally or maliciously spreading wrong connection information based on 2-layer signature checks of connection information and connection information synchronization response
6. A system to enable session key establishment when syncing parties are not able to be online (have network connection) at the same time, comprising:
 - a. Client software, that creates Diffie-Hellman half-keys and publishes them on a server
 - b. Client software that takes one of half-keys from a server, creates a session key and publishes half-key used to create a session key on a server
 - c. Server, that manages distribution of half-keys and distribution of encrypted messages

7. A system from claim 5, that uses another client to store half-keys and encrypted messages
8. A system from claim 6, that implements algorithm of shortest route search to find best candidates to store half-keys
9. A system from claim 7, that uses multiple clients to store same half-keys to improve their availability
10. A system from claim 8, where clients to store same half-keys are selected based on how close they are to receiver
11. A system from claim 8, where clients to store same half-keys are selected based on their availability
12. A system from claim 8, where clients to store same half-keys are selected based on their traffic throughput capabilities
13. A system from claim 5, that enables to sync incoming and outgoing messages between multiple devices keeping the perfect forward security using periodic database synchronization between devices using same account over a direct connection
14. A system from claim 5, that enables to sync incoming and outgoing messages between multiple devices keeping the perfect forward security using periodic database synchronization between devices using same account, exchanging their databases over a server and performing a merge operation
15. A system from claim 5, that enables to sync incoming and outgoing messages between multiple devices keeping the perfect forward security by establishing long-term keys between devices using the same account and using these long-term keys to encrypt random numbers, from which half-keys are generated, and publishing them along with half-keys
16. A system from claim 5, that enables to sync incoming and outgoing messages between multiple devices keeping the perfect forward security, where all devices publish their half-keys, all incoming messages are encrypted with session keys, generated from all devices half-keys and all outgoing messages are encrypted with session keys with session keys with receiver and session keys generated from half-keys of all other devices using same account