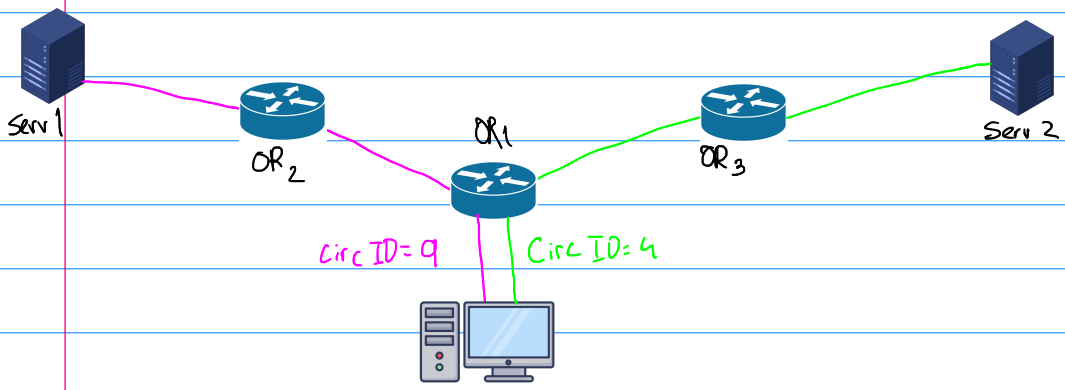
Cells

probably optional

- TOR uses two cells are (fixed size packets - 512 bytes)
  - ↳ Control: used for circuit creation and are interpreted by the OR who receives it
  - ↳ Relay: used for data-flow / responses from control cells

- The fundamental part of the cells are the Circuit ID (CircID). Every connection between two host has a unique CircID. For instance:



Since OP uses OR1 for both connections to serv 1 and serv 2, cells from OP to OR 1 need to tell what circuit they belong to so they are correctly forwarded

SO, OR1 will have a table with the Circ ID for its connection with OP:

Connection w/ OP

Incoming IP	Incoming	outgoing IP	outgoing Circ ID
OP - IP	4	OR3 - IP	6
OP - IP	9	OR 2 - IP	1

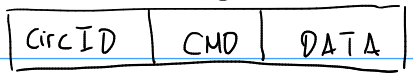
Likewise, OR 2 will also have able with the connections with OR1 to correctly forward them.

- Control cells commands:
  - ↳ create / created: to set up the circuit
  - ↳ destroy: to tear down the circuit
- Relay cells commands:
  - ↳ relay data: send data
  - ↳ relay begin / relay connected: start TCP connection from end OR to dest
  - ↳ relay close: close the TCP connection
  - ↳ Relay extend / relay ext

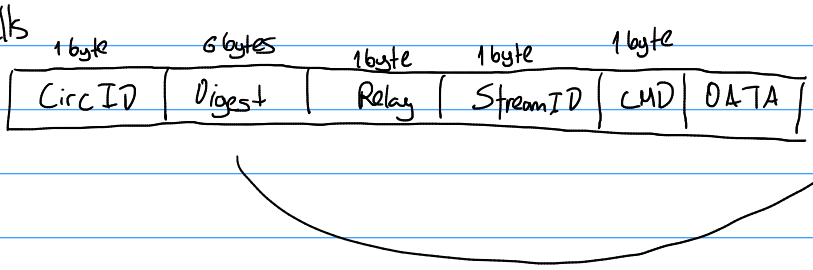
There are more, these seem to be the most relevant for the project

Since a gateway can have various TCP connections to the same server, relay cells also need a stream ID to identify what stream it belongs to.

- Control cells: 1 byte 1 byte

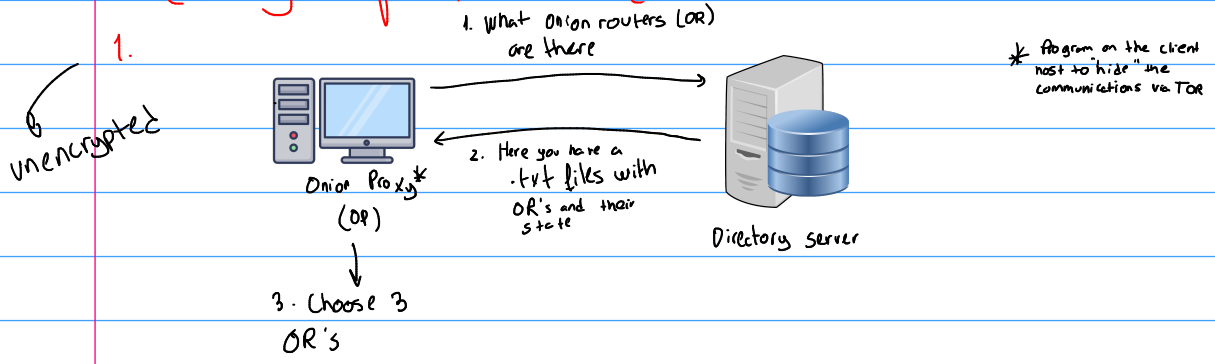


- Relay cells

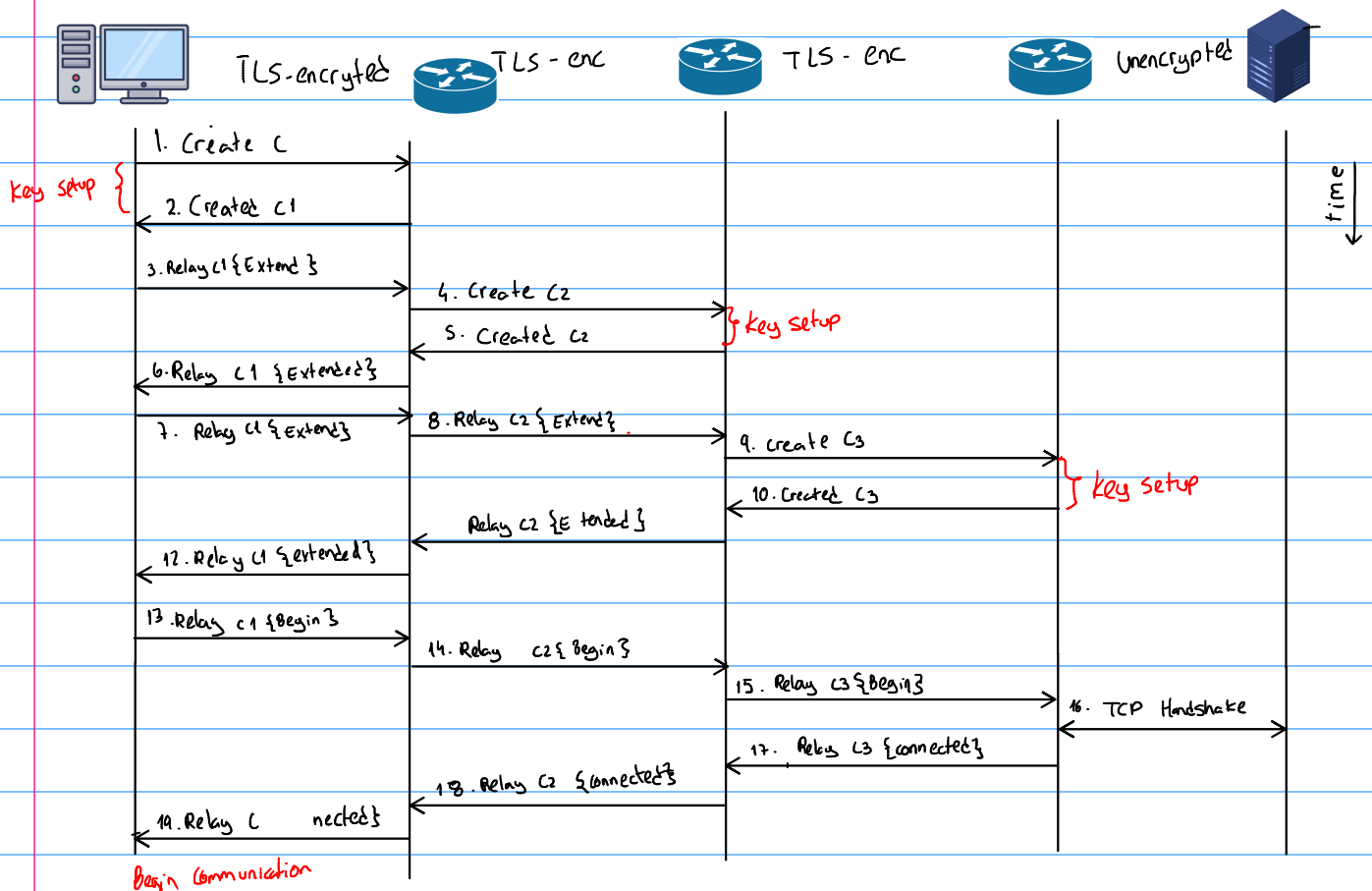


Setting up the circuit

1.

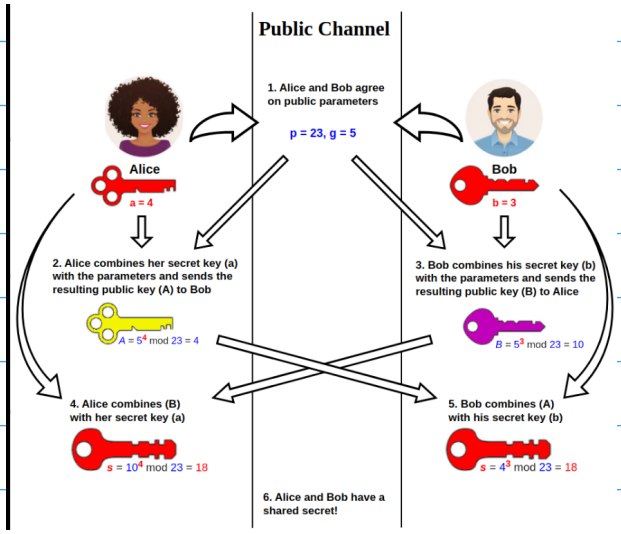


2.



Key setup

- Uses Diffie-Hellman key exchange



$OP \rightarrow OR_1 : E_{PK_{OR_1}}(g^x)$

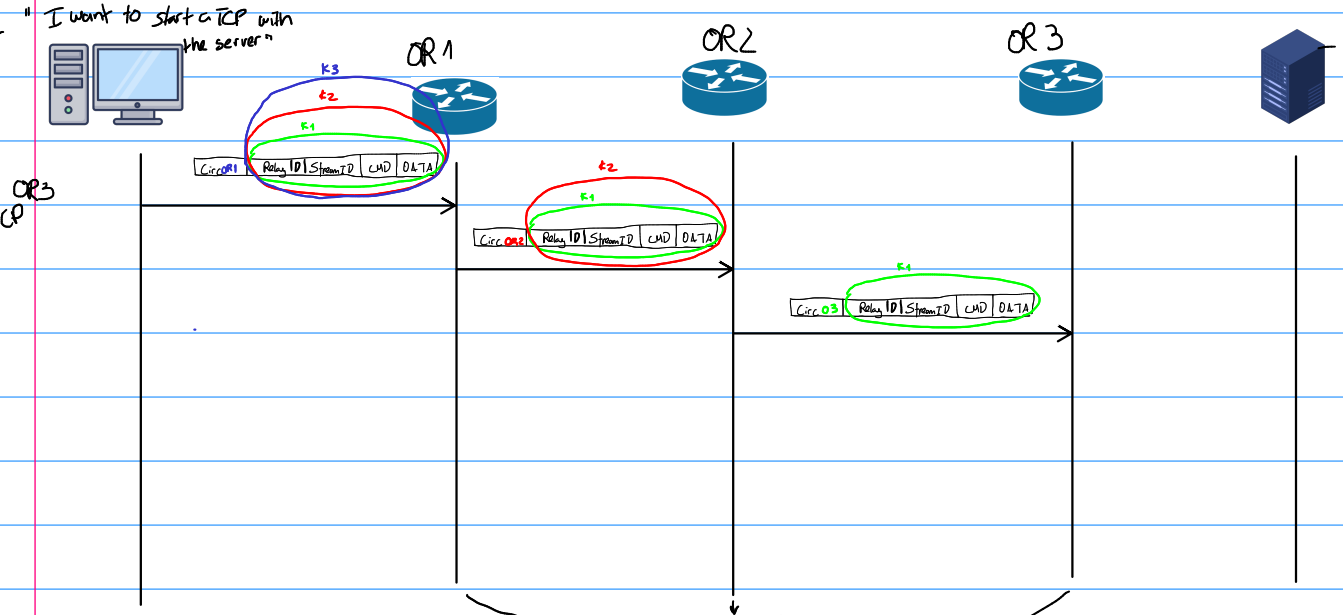
$OR_1 \rightarrow OP : g^y, H(k \mid \text{'handshake'})$

$k = g^{xy}$

secure because g^x is encrypted with OR_1's Public key

Encryption mechanism

CircID	Digest	Relay	StreamID	CMD	DATA



1. Create a relay packet to OR3 to start the TCP connection

Encryption / Decryption Protocol

when (receive relay packet) do:

decrypt (packet)
check (packet.digest)
if correct (digest):
execute (packet.cmd)
else:
change CircID (packet)
forward (Packet)