ROYAL INSTITUTE OF TECHNOLOGY

VPN Project Introduction

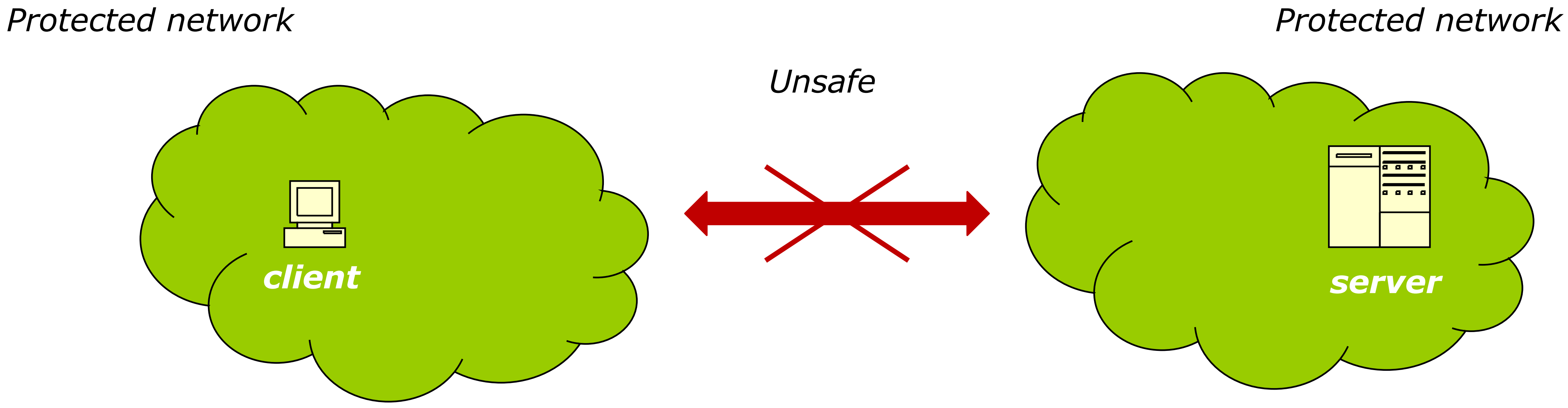
Peter Sjödin

psj@kth.se

Overview

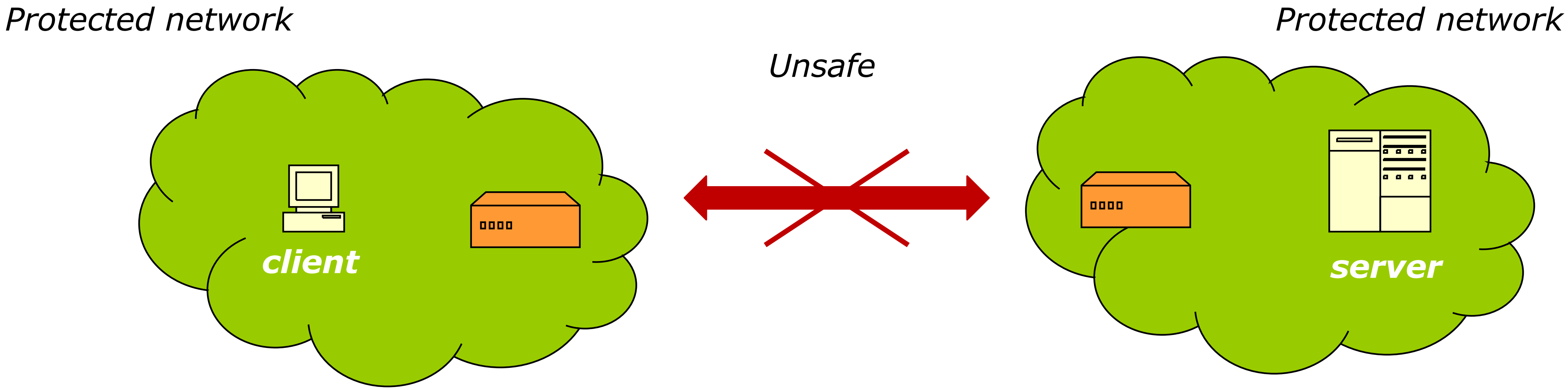
- Background to VPN
- Software support modules
- Assignment organization

VPN Problem



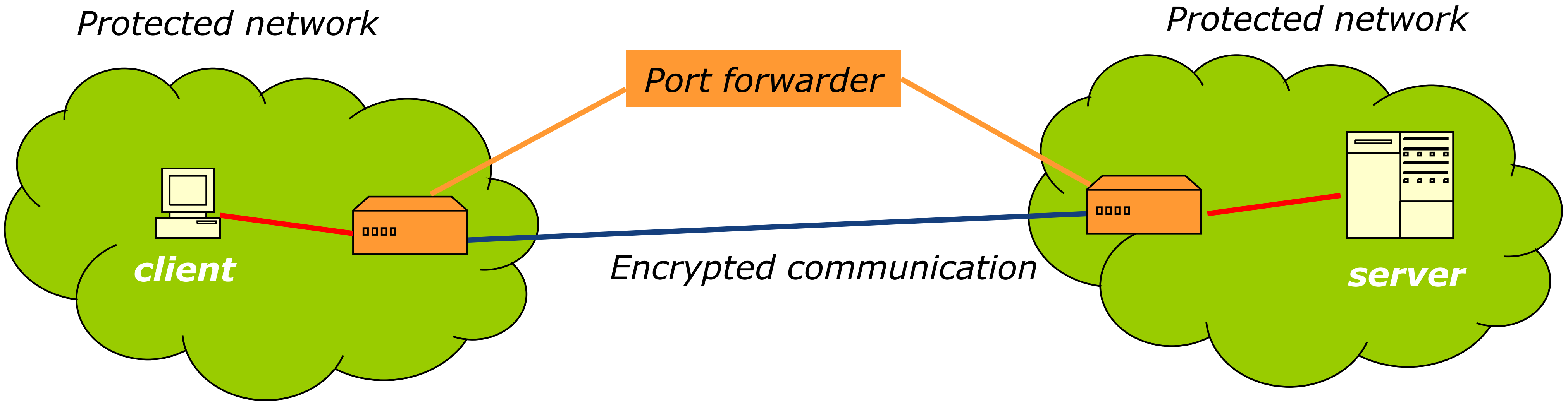
- Client wants to connect to server
  - Client and server are both on protected networks
  - But communication between them is unsafe

Intermediate Devices



- Could be:
  - NAT with port forwarding
    - Gives connectivity, but does not add protection
  - Layer 3 VPN device (OpenVPN, IPsec VPN)
    - Puts client on same network as server (logically)
    - May not always be what we want

Port Forwarding



- Access on a per-service basis
- Port forwarder forwards data between layer 4 ports
  - For instance, between TCP connections
- Connection over unsafe network protected with encryption
  - Like, for example, port forwarding in SSH

Port Forwarding Step by Step

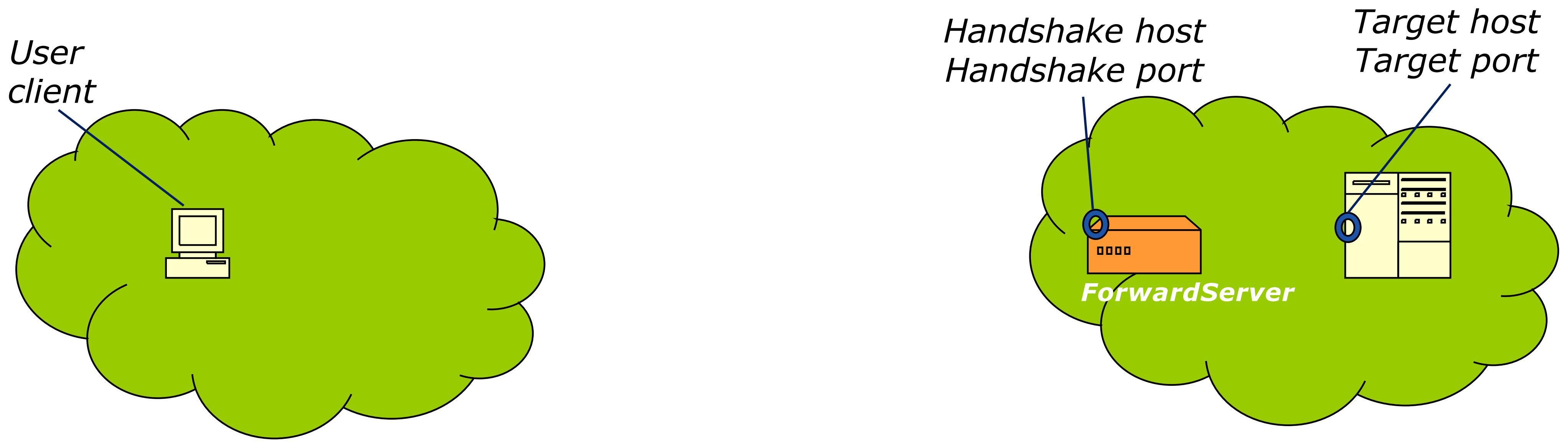
- There are several steps to setting up port forwarding
- Many TCP connections involved
- And many port numbers and host names
- Let’s take it step by step
- And introduce terminology on the way
    - Names for ports and hosts

Port Forwarding Step 1

User client
Target host
Target port

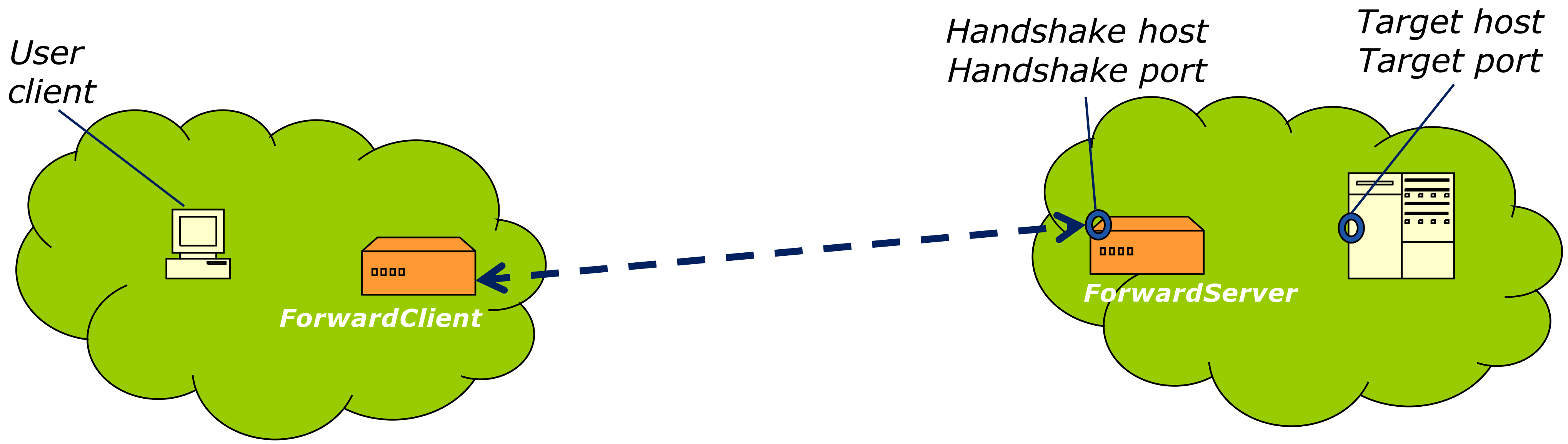
- In the beginning, there is a *user client* that wants to access a service at a *target port* on a *target host*
  - (We don’t need to know the host name and port number of the user client)

Port Forwarding Step 2



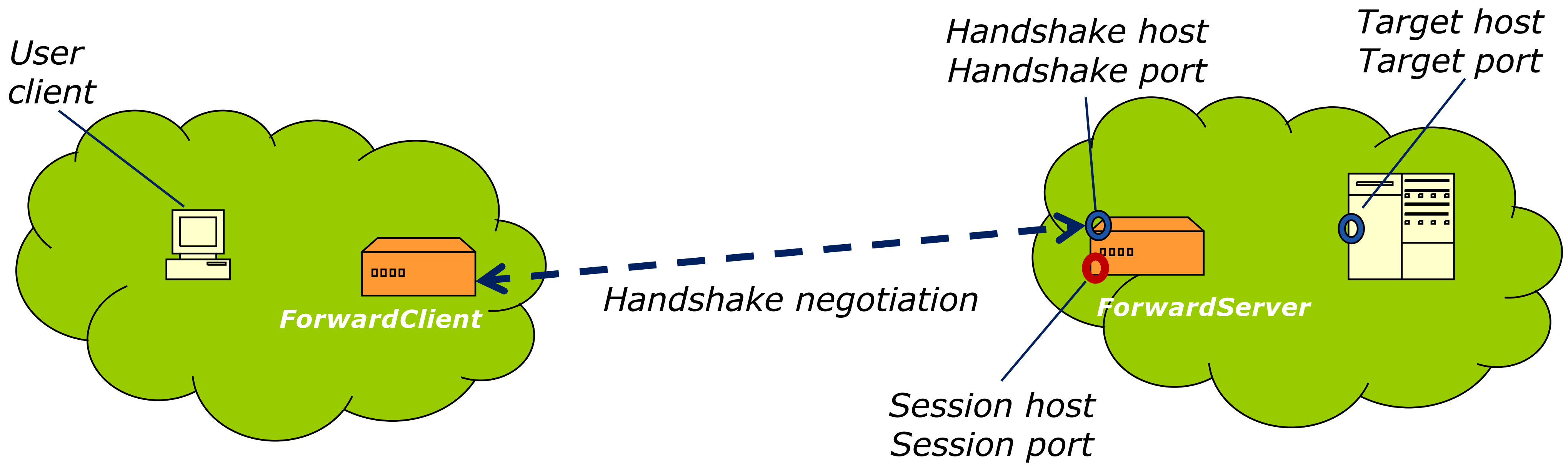
- Launch a ForwardServer application as a gateway to target’s network
- The host where ForwardServer runs is the *handshake host*
- ForwardServer listens for incoming TCP connections at *handshake port*

Port Forwarding Step 3



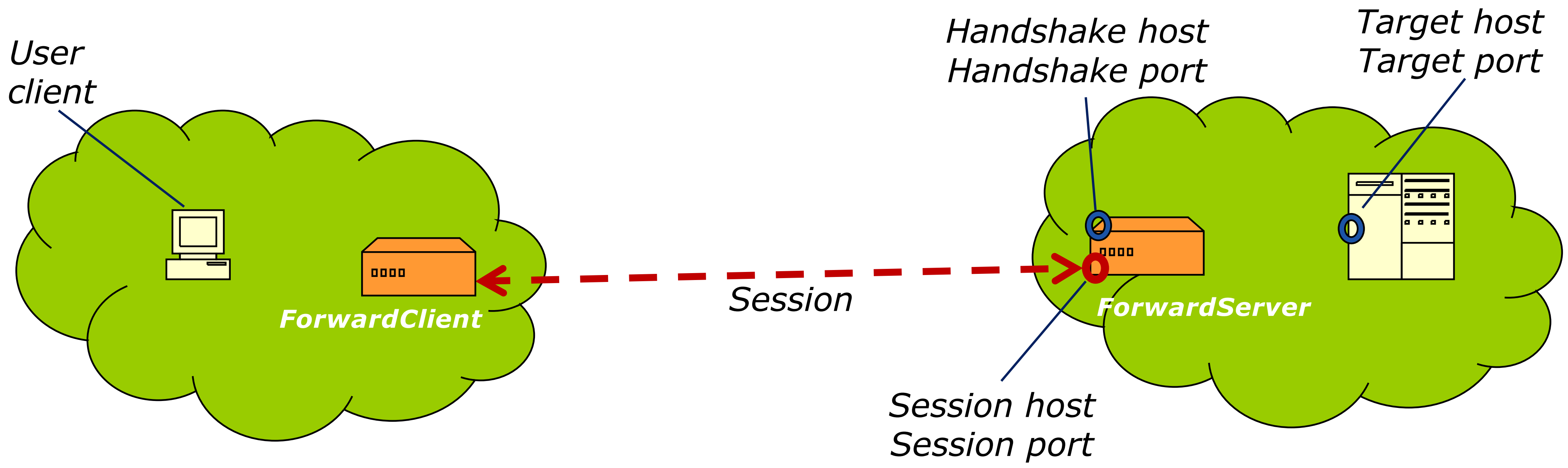
- Launch ForwardClient application on the User client’s network
- ForwardClient contacts ForwardServer by setting up TCP connection to *handshake host*/*port*

Port Forwarding Step 4



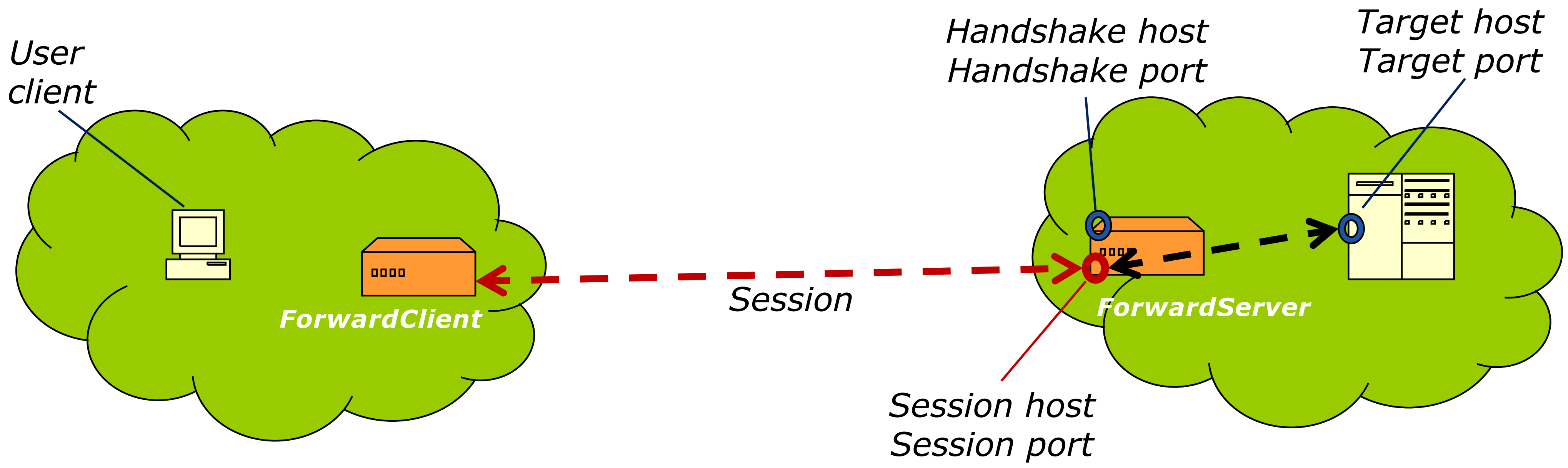
- ForwardClient and ForwardServer performs handshake negotiation to establish:
  - Session key/IV and session host/port
  - Handshake is protected through encryption (RSA) with ForwardClient's public key
- After successful handshake, ForwardServer listens for incoming connection at session host/port

Port Forwarding Step 5



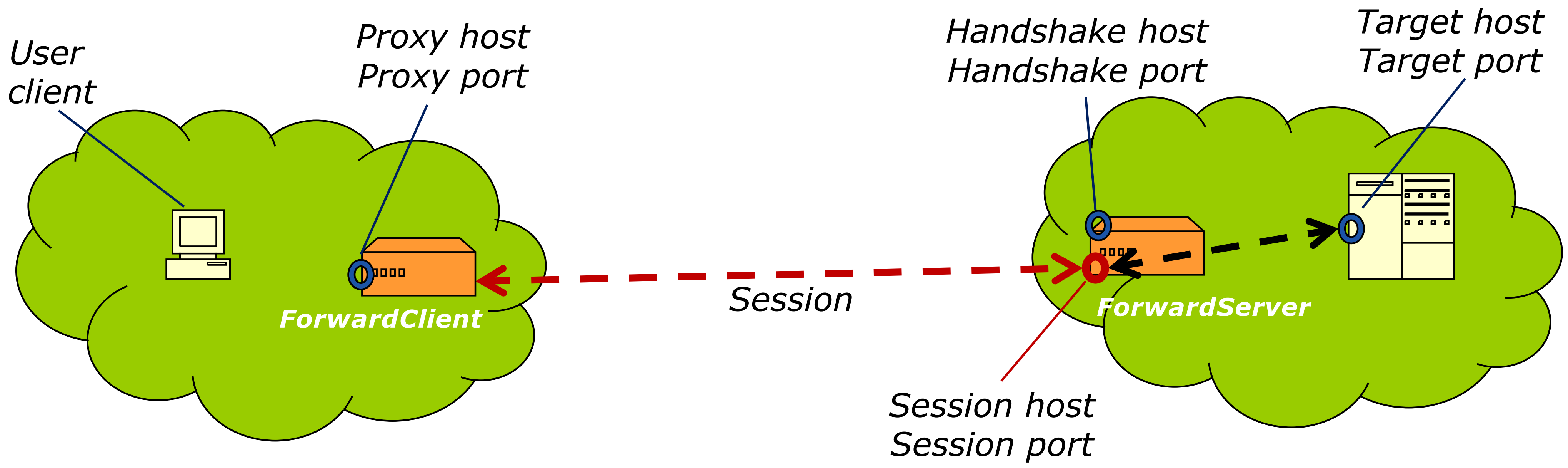
- ForwardClient sets up session – a TCP connection to session host/port
  - This connection is encrypted with session key/IV (AES in CTR mode)

Port Forwarding Step 6



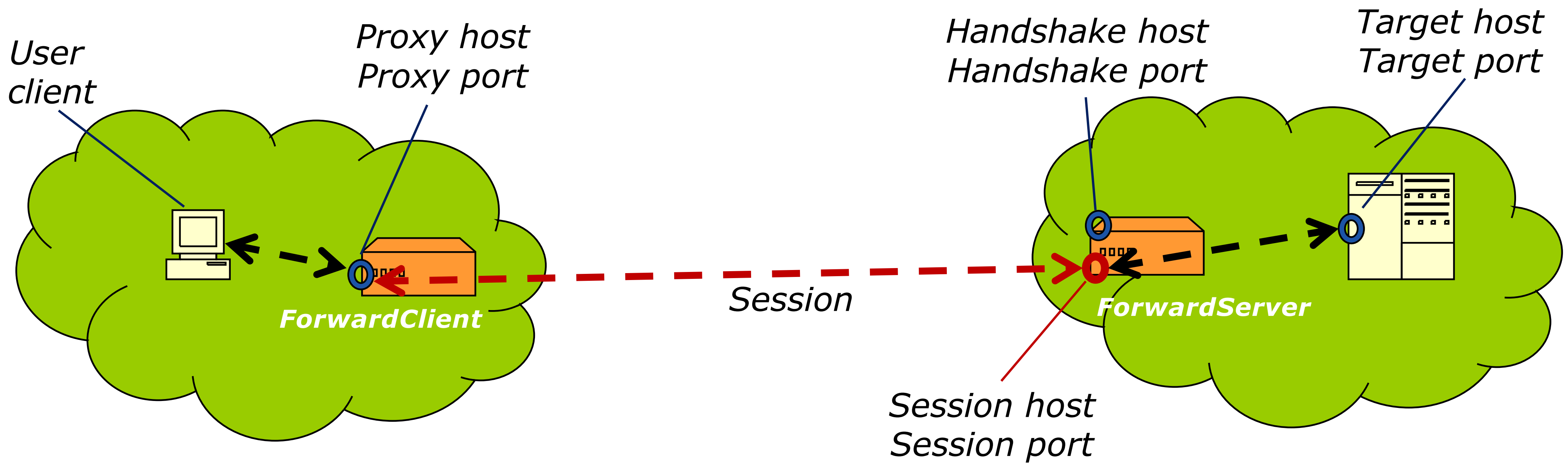
- ForwardServer sets up port forwarding (or TCP relaying) between session host/port and target host/port

Port Forwarding Step 7



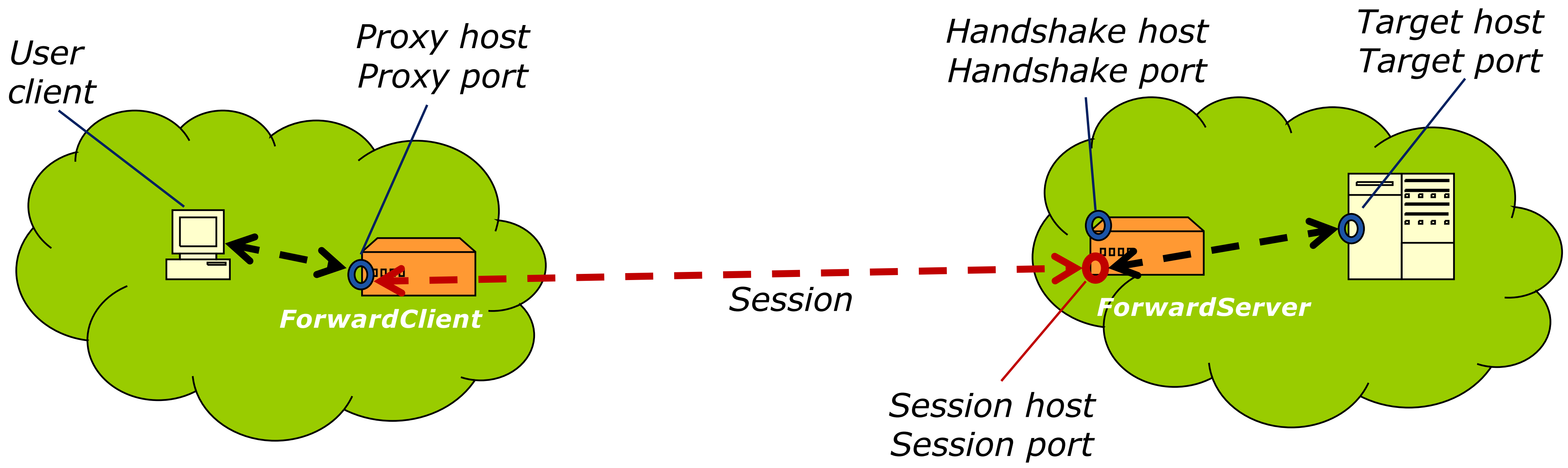
- After successful handshake, ForwardClient listens for incoming TCP connections at proxy host/port

Port Forwarding Step 8



- User client sets up TCP connection to proxy host/port
- ForwardClient sets up port forwarding between proxy host/port and session host/port

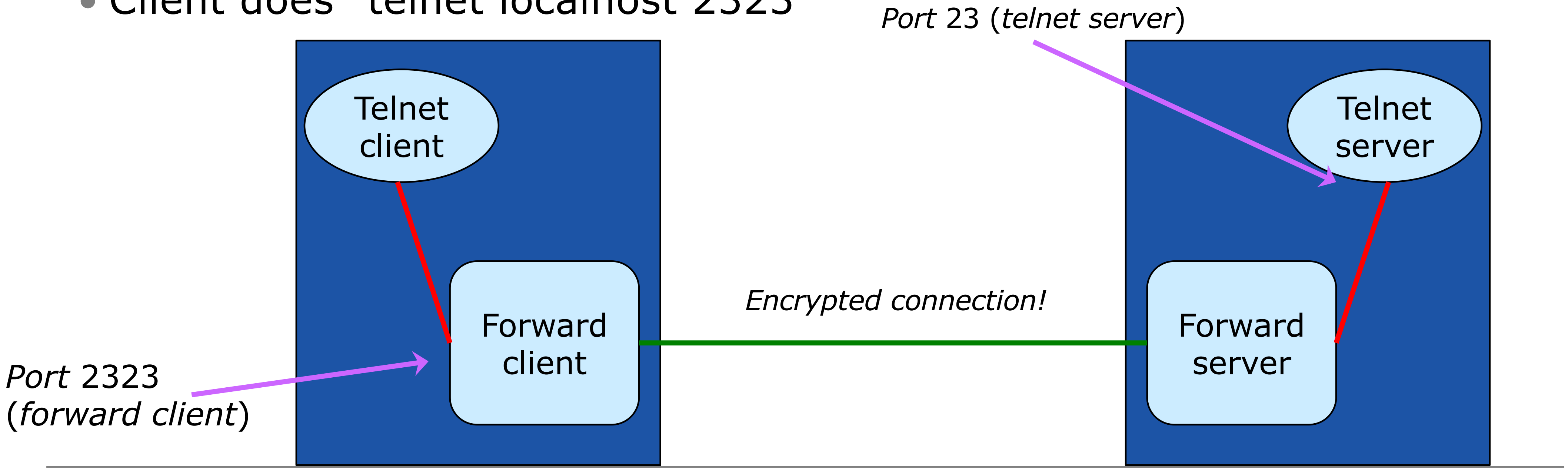
Port Forwarding Completed!



- Data from user client to proxy host/port is encrypted and forwarded by ForwardClient to session host/port
- Data received by ForwardServer at session host/port is decrypted and forwarded to target host/port
- And vice versa in the opposite direction…

Other Use: Legacy applications

- Existing application without encryption, such as telnet
- Forward client and server between telnet client/server
  - Only way to access telnet server
- Client does “telnet localhost 2323”



Project Assignment

- You are provided with a (skeleton) port forwarder
  - The Nakov Forward Server
  - Port forwarding client and server, without security
- Your job is to extend the forwarder
  - Handshake phase
  - Encrypted session

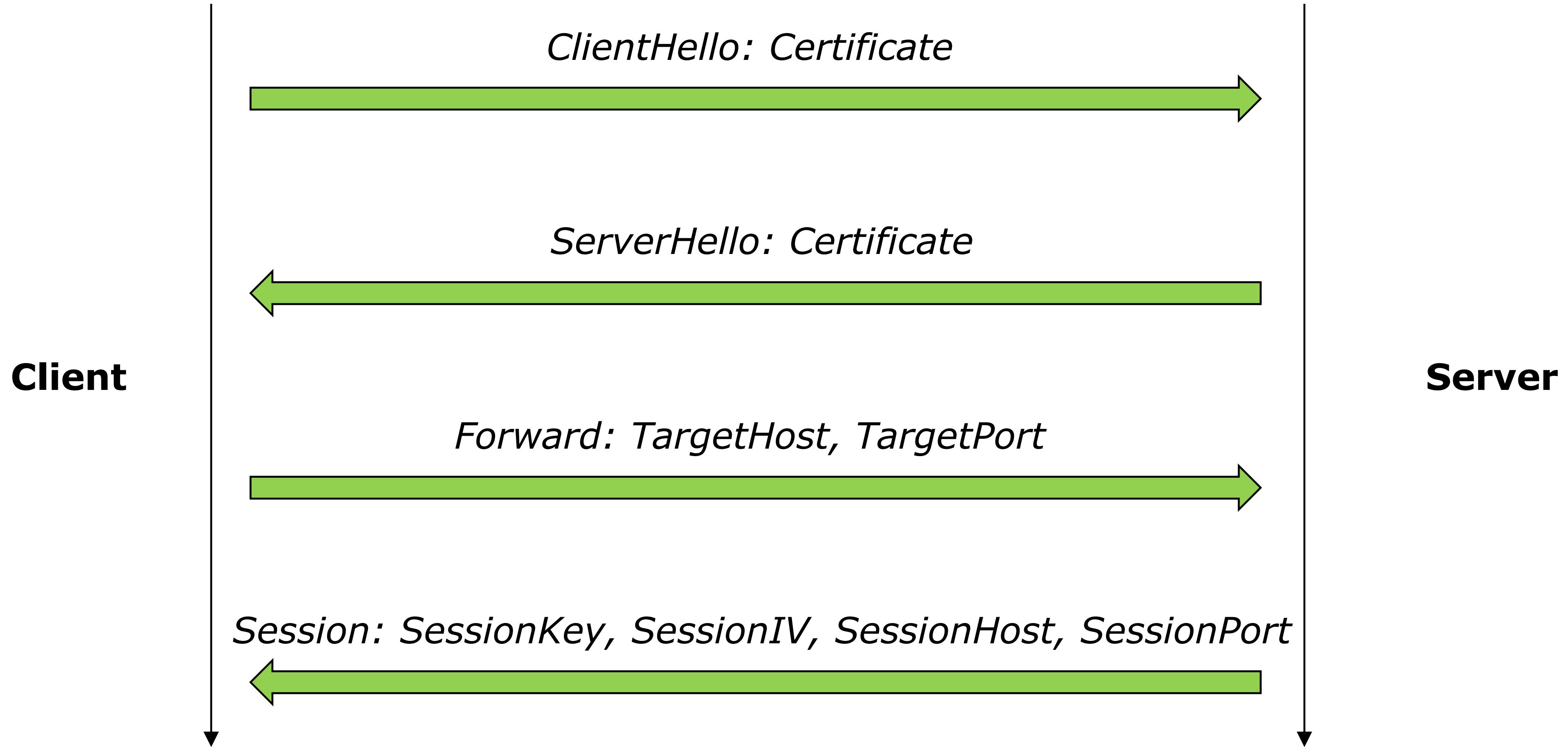
Handshake Phase

- Client and server authenticate each other
  - Certificate exchange
- Client requests forwarding to a target server
- Server creates *session key* and *IV* for session encryption
  - Used by both client and server
- Server creates *session port* and *host*
  - *A* new TCP endpoint (TCP port) to which client should connect
  - Communication over this connection encrypted using symmetric key encryption
  - (Here, session host will always be the same as handshake host, but we include it in anyway, for the sake of generality)

Handshake outcome

- ForwardClient learns session key+IV
- ForwardClient learns session port and host
- ForwardServer learns target port and host

Handshake Protocol


ClientHello: Certificate
ServerHello: Certificate
Client
Server
Forward: TargetHost, TargetPort
Session: SessionKey, SessionIV, SessionHost, SessionPort

Implementation

- Much code at your disposal – you job is to put it all together
- Results from preparatory tasks
  - SessionKey, SessionEncrypter, SessionDecrypter, …
- Stub port forwarder client and server programs
  - Without encryption
- TCP forwarding thread
  - bidirectional forwarding between TCP connections
- HandShakeMessage class
  - Encoding, decoding, and transmission of handshake messages

Examples of Running Port Forwarder Client and Server Programs

```
$ java ForwardClient
    --handshakehost=portfw.kth.se --handshakeport=2206
    --usercert=client.pem --cacert=ca.pem
    --key=client-private.der
    --targethost=server.kth.se --targetport=6789
```

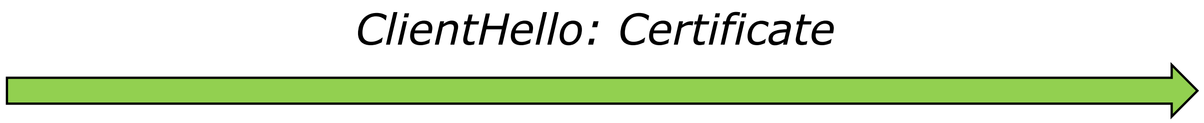
```
$ java ForwardServer --handshakehost=localhost --handshakeport=2206
    --usercert=server.pem --cacert=ca.pem
    --key=server-private.der
```

TCP Forwarding Thread

- Java thread to copy data between two TCP connections
- Does most of the Java socket plumbing work for you
- Based on Nakov Forward Server
  - See https://github.com/nakov/NakovForwardServer

HandShakeMessage class

- How to represent data in handshake messages
  - Encoding/decoding
- Many options to choose from
  - Binary, JSON, XML, CSV, URL, …



```
void putParameter(String parameter, String value)

String getParameter(String parameter)

void send(Socket socket)

void recv(Socket socket)
```

HandShakeMessage Class Example

```
HandShakeMessage clienthello = new HandshakeMessage();
clienthello.putParameter(”MessageType”, ”ClientHello”);
clienthello.putParameter(”Certificate”, encodeCertificate(clientcert));
clienthello.send(socket);
```

```
HandShakeMessage fromclient = new HandshakeMessage();
fromclient.recv(socket);
if (fromclient.getParameter(”MessageType”).equals(”ClientHello”)) {
    X509Certificate = decodeCertificate(fromclient.getParameter(”Certificate”));
    …
}
```

Handshake Messages Implementation

- Built-in support for encoding/decoding as XML
  - Format in which messages are transmitted ”on the wire”
- Extension of Java ”Properties” class
  - Use Properties methods for further inspection and debugging (if you need to)

Organisation of Project Assignment

- Implement secure forwarding client and server in Java
- Submit according to instructions
- Graded Pass/Fail

- Supervision sessions next two weeks if you need help
  - Sign up in Canvas
  - If no students have signed up before 18:00 the day before, we will cancel

- Make-up opportunity
  - If you made a serious attempt at submitting before the due date, you will get feedback and a chance to improve
  - Due date after exam

To Consider

- In the preparatory tasks, we have “micro-managed” you with given classes, methods, etc.
- Now it is up to you to organize your implementation
- Take all the pieces that you have and put it together into a running systems
  - Tasks
  - Nakov server
  - Handshake messages
  - Classes, methods,
- How you do this is up to you!
  - You are free to add classes, methods, etc

Evaluation

- First, your implementation will be checked for basic functionality
- However, you are really implementing a communication protocol here
- Therefore, your implementation must interoperate with other implementations
  - We will test your code against a reference implementation
  - For example, your ForwardClient against our reference ForwardServer
- So, make sure to follow the instructions in detail
  - Exact spelling in handshake messages, for instance

Evaluation

- Organize your submission according to instructions
  - If you use some other organization, for instance with extra packages, your code will not run and you fail
- If you fail, but have made a serious attempt to solve the assignment by the due date, you will get feedback and another chance
- Graded Pass/Fail