

Questions

Question 2.1 Think about why it is better to use UDP in the tunnel, instead of TCP, write your answer in your lab report.

Answer: Because VPN operates on the network (IP) layer, which is stateless and does not require the other features in TCP, e.g., reliable transmission, congestion control. These features should be implemented in upper layer applications that use the VPN. Thus using TCP connection for VPN is an over design.

Question 2.2 Think why it is not recommended to implement your own algorithm; and write that in your lab report.

Answer: Encryption and HMAC algorithms are a critical part of the VPN, which ensure confidentiality and integrity. Self-implemented algorithms can be error-prone. It's better to use stable and widely used implementations, such as OpenSSL, which are actively maintained by its community.

Question 2.3: Why is it important for the server to release resources when a connection is broken?

Answer: To save resources on the server side. Also, to reduce the attack surface.

Implementation

task 1

Modify `simpletun.c` to use UDP protocol, i.e., `SOCK_STREAM` -> `SOCK_DGRAM`, `read()` -> `recvfrom()`, `write()` -> `sendto()`. See `src/simpletun_udp_task1.c`.

task 2

Use OpenSSL implementation of AES256 and HMAC-SHA256 because they are among the strongest block ciphers and HMAC algorithms without effective attack today. see `src/minivpn_task2.c`, `src/openssl_utils.h`.

task 3-5

Use `fork()` to fork 2 processes after ssl certificate verification. Parent process handles control channel messages (e.g., close request from client and related handling at server) through ssl connection. Child process handles vpn traffic using UDP. The parent uses unnamed pipe to inform the child process to close connection and release resources. See `src/minivpn_task3_server.c`, `src/minivpn_task3_client.c`, `src/openssl_utils.h`.

task 6 and bonus

At the client side, use fork() to accept multiple connection requests. Upon receiving a new request, assign a new UDP port and send it back to the client using the ssl control channel. Bonus feature of session key and iv dynamic reconfiguration is implemented similar to the close request handling, i.e., handling the request through ssl control channel and use unnamed pipe to inform the child process. See src/minivpn_task6_server.c, src/minivpn_task6_client.c, src/openssl_utils.h.

Screenshots

1. Multiple connections

Servers can accept multiple connection requests using ssl certificate for authentication; assign different UDP ports for different connections; exchange keys and ivs with clients separately; encrypt, decrypt and verify traffic through the VPN tunnels.

• server:

```
NET2TAP 7: Read 128 bytes from the network
HMMA verification succeed: 0x5c2aa9327ba0f502d0ad61d1963acf37e1d42c9bed55f4a360f6f9648ca9c43e
NET2TAP 7: Written 84 bytes to the tap interface
TAP2NET 7: Read 84 bytes from the tap interface
TAP2NET 7: Written 130 bytes to the network
NET2TAP 1: Read 128 bytes from the network
HMMA verification succeed: 0x96fed62eb7354f7af232bf2272816db9a93539938c0513451810675826d8e19b
NET2TAP 1: Written 84 bytes to the tap interface
TAP2NET 1: Read 84 bytes from the tap interface
TAP2NET 1: Written 130 bytes to the network
NET2TAP 8: Read 128 bytes from the network
HMMA verification succeed: 0x728f52d2408b1671966e7679f3eaa9120b8fa3b8f399abdc33572d484a9ebc5
NET2TAP 8: Written 84 bytes to the tap interface
TAP2NET 8: Read 84 bytes from the tap interface
TAP2NET 8: Written 130 bytes to the network
NET2TAP 2: Read 128 bytes from the network
HMMA verification succeed: 0x64f9e7578c5924ca8a5b50d1513370b7be08c58c0958a2a1fe7fdd1497cfd9d
NET2TAP 2: Written 84 bytes to the tap interface
TAP2NET 2: Read 84 bytes from the tap interface
TAP2NET 2: Written 130 bytes to the network
NET2TAP 9: Read 128 bytes from the network
HMMA verification succeed: 0xf9b0c9df0afba9d9c28a1502e813f2d9a3cf5997763adfd6991263ecf0b6ea8
NET2TAP 9: Written 84 bytes to the tap interface
TAP2NET 9: Read 84 bytes from the tap interface
TAP2NET 9: Written 130 bytes to the network
NET2TAP 3: Read 128 bytes from the network
```

```
[03/02/2024 08:37] seed@ubuntu:~/zzhong/hw/cs528/lab2/src$ ./setup_tun_server_multiple_conne
[03/02/2024 08:37] seed@ubuntu:~/zzhong/hw/cs528/lab2/src$ ping 10.0.4.1
PING 10.0.4.1 (10.0.4.1) 56(84) bytes of data:
64 bytes from 10.0.4.1: icmp_req=1 ttl=64 time=0.684 ms
64 bytes from 10.0.4.1: icmp_req=2 ttl=64 time=0.491 ms
64 bytes from 10.0.4.1: icmp_req=3 ttl=64 time=0.526 ms
64 bytes from 10.0.4.1: icmp_req=4 ttl=64 time=0.551 ms
64 bytes from 10.0.4.1: icmp_req=5 ttl=64 time=0.605 ms
64 bytes from 10.0.4.1: icmp_req=6 ttl=64 time=0.568 ms
64 bytes from 10.0.4.1: icmp_req=7 ttl=64 time=0.465 ms
64 bytes from 10.0.4.1: icmp_req=8 ttl=64 time=0.545 ms
64 bytes from 10.0.4.1: icmp_req=9 ttl=64 time=0.531 ms
^C
--- 10.0.4.1 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 7999ms
rtt min/avg/max/mdev = 0.465/0.542/0.685/0.043 ms
[03/02/2024 08:37] seed@ubuntu:~/zzhong/hw/cs528/lab2/src$ ping 10.0.2.1
PING 10.0.2.1 (10.0.2.1) 56(84) bytes of data:
64 bytes from 10.0.2.1: icmp_req=1 ttl=64 time=0.680 ms
64 bytes from 10.0.2.1: icmp_req=2 ttl=64 time=0.584 ms
64 bytes from 10.0.2.1: icmp_req=3 ttl=64 time=0.595 ms
64 bytes from 10.0.2.1: icmp_req=4 ttl=64 time=0.598 ms
64 bytes from 10.0.2.1: icmp_req=5 ttl=64 time=0.548 ms
64 bytes from 10.0.2.1: icmp_req=6 ttl=64 time=0.552 ms
64 bytes from 10.0.2.1: icmp_req=7 ttl=64 time=0.583 ms
64 bytes from 10.0.2.1: icmp_req=8 ttl=64 time=0.536 ms
```

• client1:

```
TAP2NET 7: Read 84 bytes from the tap interface
TAP2NET 7: Written 130 bytes to the network
NET2TAP 7: Read 128 bytes from the network
HMMA verification succeed: 0x8ae036216870f15a530f859ccdaefde9f4b2fd53a89072e2da54a6838ac8af23
NET2TAP 7: Written 84 bytes to the tap interface
TAP2NET 8: Read 84 bytes from the tap interface
TAP2NET 8: Written 130 bytes to the network
NET2TAP 8: Read 128 bytes from the network
HMMA verification succeed: 0xb8cc7e9c7500ad63a530b4f763b459b2d949ce645b85ff373e5b37bd2e36c8bf
NET2TAP 8: Written 84 bytes to the tap interface
TAP2NET 9: Read 84 bytes from the tap interface
TAP2NET 9: Written 130 bytes to the network
NET2TAP 9: Read 128 bytes from the network
HMMA verification succeed: 0x635f6531d1e2723f20ef8e81552cf905c03f1edcc02d344926bcf4d3b0ae3af6
```

```
[03/02/2024 08:41] seed@ubuntu:~/zzhong/hw/cs528/lab2/src$ ./setup_tun_client.sh
[03/02/2024 08:42] seed@ubuntu:~/zzhong/hw/cs528/lab2/src$ ping 10.0.1.1
PING 10.0.1.1 (10.0.1.1) 56(84) bytes of data:
64 bytes from 10.0.1.1: icmp_req=1 ttl=64 time=0.748 ms
64 bytes from 10.0.1.1: icmp_req=2 ttl=64 time=0.616 ms
64 bytes from 10.0.1.1: icmp_req=3 ttl=64 time=0.588 ms
64 bytes from 10.0.1.1: icmp_req=4 ttl=64 time=0.648 ms
64 bytes from 10.0.1.1: icmp_req=5 ttl=64 time=0.569 ms
64 bytes from 10.0.1.1: icmp_req=6 ttl=64 time=0.623 ms
64 bytes from 10.0.1.1: icmp_req=7 ttl=64 time=0.591 ms
64 bytes from 10.0.1.1: icmp_req=8 ttl=64 time=0.672 ms
64 bytes from 10.0.1.1: icmp_req=9 ttl=64 time=0.621 ms
64 bytes from 10.0.1.1: icmp_req=10 ttl=64 time=0.696 ms
64 bytes from 10.0.1.1: icmp_req=11 ttl=64 time=0.643 ms
```

• client2:

```
HMMA verification succeed: 0x2fb502d657d7be08279d3b642e3f777c22cb9442eae2bea0d04d0638e0aa737
NET2TAP 64: Written 84 bytes to the tap interface
TAP2NET 64: Read 84 bytes from the tap interface
TAP2NET 64: Written 130 bytes to the network
NET2TAP 65: Read 128 bytes from the network
HMMA verification succeed: 0xcd45bb0c040cdde344ad45da5d966de6e3e799bf5824d6e25478ced0be2ccb
NET2TAP 65: Written 84 bytes to the tap interface
TAP2NET 65: Read 84 bytes from the tap interface
TAP2NET 65: Written 130 bytes to the network
NET2TAP 66: Read 128 bytes from the network
HMMA verification succeed: 0x381b0e9eb1cdf1a93248b7013e5a9b9fa269a93db0614bb07410c7c81195a7f
NET2TAP 66: Written 84 bytes to the tap interface
TAP2NET 66: Read 84 bytes from the tap interface
TAP2NET 66: Written 130 bytes to the network
NET2TAP 67: Read 128 bytes from the network
```

```
[03/02/2024 08:38] seed@ubuntu:~/zzhong/hw/cs528/lab2/src$ ./setup_tun_client2.sh
[03/02/2024 08:42] seed@ubuntu:~/zzhong/hw/cs528/lab2/src$ ping 10.0.3.1
PING 10.0.3.1 (10.0.3.1) 56(84) bytes of data:
64 bytes from 10.0.3.1: icmp_req=1 ttl=64 time=0.771 ms
64 bytes from 10.0.3.1: icmp_req=2 ttl=64 time=0.617 ms
64 bytes from 10.0.3.1: icmp_req=3 ttl=64 time=0.528 ms
64 bytes from 10.0.3.1: icmp_req=4 ttl=64 time=0.583 ms
64 bytes from 10.0.3.1: icmp_req=5 ttl=64 time=0.762 ms
64 bytes from 10.0.3.1: icmp_req=6 ttl=64 time=0.517 ms
64 bytes from 10.0.3.1: icmp_req=7 ttl=64 time=0.586 ms
64 bytes from 10.0.3.1: icmp_req=8 ttl=64 time=0.468 ms
64 bytes from 10.0.3.1: icmp_req=9 ttl=64 time=0.470 ms
64 bytes from 10.0.3.1: icmp_req=10 ttl=64 time=0.557 ms
64 bytes from 10.0.3.1: icmp_req=11 ttl=64 time=0.573 ms
64 bytes from 10.0.3.1: icmp_req=12 ttl=64 time=0.541 ms
```

2. Dynamic Reconfiguration (bonus feature)

Clients can send "keyiv" requests to the server and update the key and iv used.

- server:

```
^C[03/02/2024 08:40] seed@ubuntu:~/zzhong/hw/cs528/lab2/src$ ./run_server_multiple_connectio 599/1845
rm -f simpletun simpletun_udp_task1 minivpn_task2 minivpn_task3_server minivpn_task3_client minivpn_t
sk6_server minivpn_task6_client
gcc minivpn_task6_server.c openssl_utils.h -o minivpn_task6_server -lssl -lcrypto
Enter PEM pass phrase:
[Server] waiting for ssl handshake ... done.
[SSL REQ] Received: hello
[Server] assign port: 55555
[SSL REQ] Received: keyiv
[Server] update key: 0x51357a28f6c716157ebbe330c7d495f6b99ea3765ae4e245c02878c25e8a9d54
[Server] update iv: 0x187d607590c6a984198af1f2d6dd4c15
Successfully connected to interface tun0
[Server] waiting for ssl handshake ... done.
[SSL REQ] Received: hello
[Server] assign port: 55556
[SSL REQ] Received: keyiv
[Server] update key: 0x38ef5c8cd7a7c3eddafbee11fc89cdf0592a7b678031c6dac7d10dc4bc19d4c2
[Server] update iv: 0x6544695e0dd7daba00c7dc93b0d20634
Successfully connected to interface tun1
NET2TAP 1: Read 128 bytes from the network
HMAC verification succeed: 0xd5bb4d15c29688a30e4666dcf8a351f054346a9ed0d91a60254ce9d11452031c
NET2TAP 1: Written 84 bytes to the tap interface
TAP2NET 1: Read 84 bytes from the tap interface
TAP2NET 1: Written 130 bytes to the network
NET2TAP 2: Read 128 bytes from the network
HMAC verification succeed: 0x59d0b5e4c57105416ef0a052b578bf623e3ee565073aa67d73e549914cd608af
NET2TAP 2: Written 84 bytes to the tap interface
TAP2NET 2: Read 84 bytes from the tap interface
TAP2NET 2: Written 130 bytes to the network
NET2TAP 3: Read 128 bytes from the network
HMAC verification succeed: 0x76e02aa88e6803bd962d0a42ce0d93a578efd9ed2d7f6bad2b24d902ac7cfaf2
NET2TAP 3: Written 84 bytes to the tap interface
TAP2NET 3: Read 84 bytes from the tap interface
TAP2NET 3: Written 130 bytes to the network
NET2TAP 4: Read 128 bytes from the network
HMAC verification succeed: 0x925e2c2d5d7cc10bf8a1b57f7676643d3d315f0ac70803c59f2f832e248e4440
NET2TAP 4: Written 84 bytes to the tap interface
TAP2NET 4: Read 84 bytes from the tap interface
TAP2NET 4: Written 130 bytes to the network
NET2TAP 5: Read 128 bytes from the network
HMAC verification succeed: 0x7a5e2dcc6b7bc45b712b33b1871121497d0eb8a37542ca6b898d8ef30303a6b5
NET2TAP 5: Written 84 bytes to the tap interface
TAP2NET 5: Read 84 bytes from the tap interface
TAP2NET 5: Written 130 bytes to the network
NET2TAP 6: Read 128 bytes from the network
HMAC verification succeed: 0x5b0529bcc1edcf3705b1f0d85d2f38459b4fd182485f820beb5f747d7c5a590c
NET2TAP 6: Written 84 bytes to the tap interface
TAP2NET 6: Read 84 bytes from the tap interface
TAP2NET 6: Written 130 bytes to the network
[SSL REQ] Received: keyiv
[Server] update key: 0xd384ec2ee2681935bd145542a618adbf61934b612d1654a90769d688ab6467d4
[Server] update iv: 0xcd1b17ca9e61c72a9db9c8953d938e0b
NET2TAP 7: Read 128 bytes from the network
HMAC verification succeed: 0x5c2aa9327ba6f502d6ad61d1963acf37e1d42c9bed55f4a360f6f9648ca9c43e
```

- client:

```
^C[03/02/2024 08:40] seed@ubuntu:~/zzhong/hw/cs528/lab2/src$ ./run_client_multiple_connectio 228/1935
rm -f simpletun simpletun_udp_task1 minivpn_task2 minivpn_task3_server minivpn_task3_client minivpn_ta
sk6_server minivpn_task6_client
gcc minivpn_task6_client.c openssl_utils.h -o minivpn_task6_client -lssl -lcrypto
Enter PEM pass phrase:
[Client] establishing tcp connection to server... done.
[Client] establishing ssl connection to server... done.
[SSL RES] Received: Hello from server!
[Client] obtain server assigned port: 55555
[SSL RES] Received: keyiv succeed
[Client] update key: 0x51357a28f6c716157ebbe330c7d495f6b99ea3765ae4e245c02878c25e8a9d54
[Client] update iv: 0x187d607590c6a984198af1f2d6dd4c15
Successfully connected to interface tun0
TAP2NET 1: Read 84 bytes from the tap interface
TAP2NET 1: Written 130 bytes to the network
NET2TAP 1: Read 128 bytes from the network
HMAC verification succeed: 0x8875c53053d6f83ce7ff24e7ed7ffef5f29640bdf92e2f2b587bdcafef3db5f1
NET2TAP 1: Written 84 bytes to the tap interface
TAP2NET 2: Read 84 bytes from the tap interface
TAP2NET 2: Written 130 bytes to the network
NET2TAP 2: Read 128 bytes from the network
HMAC verification succeed: 0xcd3ee497988990977e870179a20c12cc1288bbae3a15bc1b9c7ddae704217c8b
NET2TAP 2: Written 84 bytes to the tap interface
TAP2NET 3: Read 84 bytes from the tap interface
TAP2NET 3: Written 130 bytes to the network
NET2TAP 3: Read 128 bytes from the network
HMAC verification succeed: 0x3acf637da0c9f0648670d9aad6753c0ebd46279caf2ecc4a4a11745480421e07
NET2TAP 3: Written 84 bytes to the tap interface
TAP2NET 4: Read 84 bytes from the tap interface
TAP2NET 4: Written 130 bytes to the network
NET2TAP 4: Read 128 bytes from the network
HMAC verification succeed: 0x16726aae33d478930ceca7f93e775c1d711ac979acfeff7960a16261476cadbe
NET2TAP 4: Written 84 bytes to the tap interface
TAP2NET 5: Read 84 bytes from the tap interface
TAP2NET 5: Written 130 bytes to the network
NET2TAP 5: Read 128 bytes from the network
HMAC verification succeed: 0xa1d06e7843d0025932b2a20ff1b1266366626c5c3123ac8049933717ea19a36f
NET2TAP 5: Written 84 bytes to the tap interface
TAP2NET 6: Read 84 bytes from the tap interface
TAP2NET 6: Written 130 bytes to the network
NET2TAP 6: Read 128 bytes from the network
HMAC verification succeed: 0x5a0d2a39613b15d8c978f805fdd89774af86be516efae5c5ae61022609e34ab5
NET2TAP 6: Written 84 bytes to the tap interface
[SSL RES] Received: keyiv succeed
[Client] update key: 0xd384ec2ee2681935bd145542a618adbf61934b612d1654a90769d688ab6467d4
[Client] update iv: 0xcd1b17ca9e61c72a9db9c8953d938e0b
TAP2NET 7: Read 84 bytes from the tap interface
TAP2NET 7: Written 130 bytes to the network
NET2TAP 7: Read 128 bytes from the network
HMAC verification succeed: 0x8ae036216870f15a530f859ccdaefde9f4b2fd53a89072e2da54a6838ac8af23
NET2TAP 7: Written 84 bytes to the tap interface
TAP2NET 8: Read 84 bytes from the tap interface
TAP2NET 8: Written 130 bytes to the network
NET2TAP 8: Read 128 bytes from the network
```