

Needham-Schroeder Based PDF Print Server

1. Introduction

This report provides an in-depth analysis of the secure, multi-threaded PDF print server implemented using the Needham-Schroeder protocol. The system consists of three primary components:

- **Key Distribution Center (KDC):** Handles authentication and session key distribution.
- **Print Server (PrnSrv):** Processes PDF conversion requests from clients.
- **Client:** Requests PDF conversion and retrieves the encrypted output.

The implementation is done in **C**, utilizing OpenSSL for cryptographic operations and POSIX threads for multi-threading.

2. System Architecture

2.1 Communication Flow

1. **Client authenticates with KDC:**
 - Sends username and derives a secret key.
 - KDC generates a session key and encrypts it.
2. **Client decrypts session key:**
 - Uses it to securely communicate with Print Server.
3. **Client requests file conversion:**
 - Sends a file to Print Server.
 - Print Server converts the file and encrypts the result.
4. **Client receives and decrypts PDF.**

3. File Descriptions and Code Breakdown

3.1 `config.h`

Purpose: Defines configuration constants for ports, key lengths, and security parameters.

```
#define KDC_PORT 5000
#define PRN_PORT 5001
#define MAX_CLIENTS 10
#define THREAD_POOL_SIZE 5
#define LOG_FILE "server.log"
#define KDC_SALT "fixed_salt_bhargav"
#define KEY_LEN 32
#define NONCE_LEN 12
#define TAG_LEN 16
#define SALT_LEN 16
```

3.2 `cryptoUtils.h` & `cryptoUtils.c`

Purpose: Handles cryptographic functions including key derivation, encryption, and decryption.

Functions:

- `deriveKey()`: Derives an AES-256 key using PBKDF2.
- `encryptData()`: Encrypts data using AES-GCM.
- `decryptData()`: Decrypts AES-GCM encrypted data.
- `generateNonce()`: Generates a random nonce.

Key Implementation (AES-256-GCM Encryption):

```
void encryptData(unsigned char *plaintext, int plaintextlen, unsigned char *key,
                unsigned char *nonce, unsigned char *ciphertext, unsigned char
                *tag) {
    EVP_CIPHER_CTX *ctx = EVP_CIPHER_CTX_new();
    EVP_EncryptInit_ex(ctx, EVP_aes_256_gcm(), NULL, NULL, NULL);
    EVP_CIPHER_CTX_ctrl(ctx, EVP_CTRL_GCM_SET_IVLEN, NONCE_LEN, NULL);
    EVP_EncryptInit_ex(ctx, NULL, NULL, key, nonce);
    EVP_EncryptUpdate(ctx, ciphertext, &len, plaintext, plaintextlen);
    EVP_EncryptFinal_ex(ctx, ciphertext + len, &len);
    EVP_CIPHER_CTX_ctrl(ctx, EVP_CTRL_GCM_GET_TAG, TAG_LEN, tag);
    EVP_CIPHER_CTX_free(ctx);
}
```

3.3 `logger.h` & `logger.c`

Purpose: Logs security events to `server.log`.

```
void logMessage(const char *message) {
    FILE *logFile = fopen(LOG_FILE, "a");
    fprintf(logFile, "[%s] %s\n", ctime(&now), message);
    fclose(logFile);
}
```

3.4 `client.c`

Purpose: Implements client-side communication with KDC and Print Server.

Execution Flow:

1. Connects to **KDC** for authentication.
2. Sends username and receives an encrypted session key.
3. Decrypts session key and connects to **Print Server**.
4. Sends a file for conversion.
5. Receives and decrypts the converted PDF.

3.5 `kdcServer.c`

Purpose: Implements the KDC authentication server.

Execution Flow:

1. Receives authentication request.
2. Derives the user's key using PBKDF2.
3. Generates a session key and encrypts it with the user's key.
4. Sends the encrypted session key and nonce to the client.

3.6 prnSrv.c

Purpose: Implements the Print Server.

Execution Flow:

1. Receives file from client.
2. Converts file to PDF using `enscript` or `img2pdf`.
3. Encrypts the PDF using the session key.
4. Sends the encrypted PDF back to the client.

4. Execution Steps

4.1 Compilation

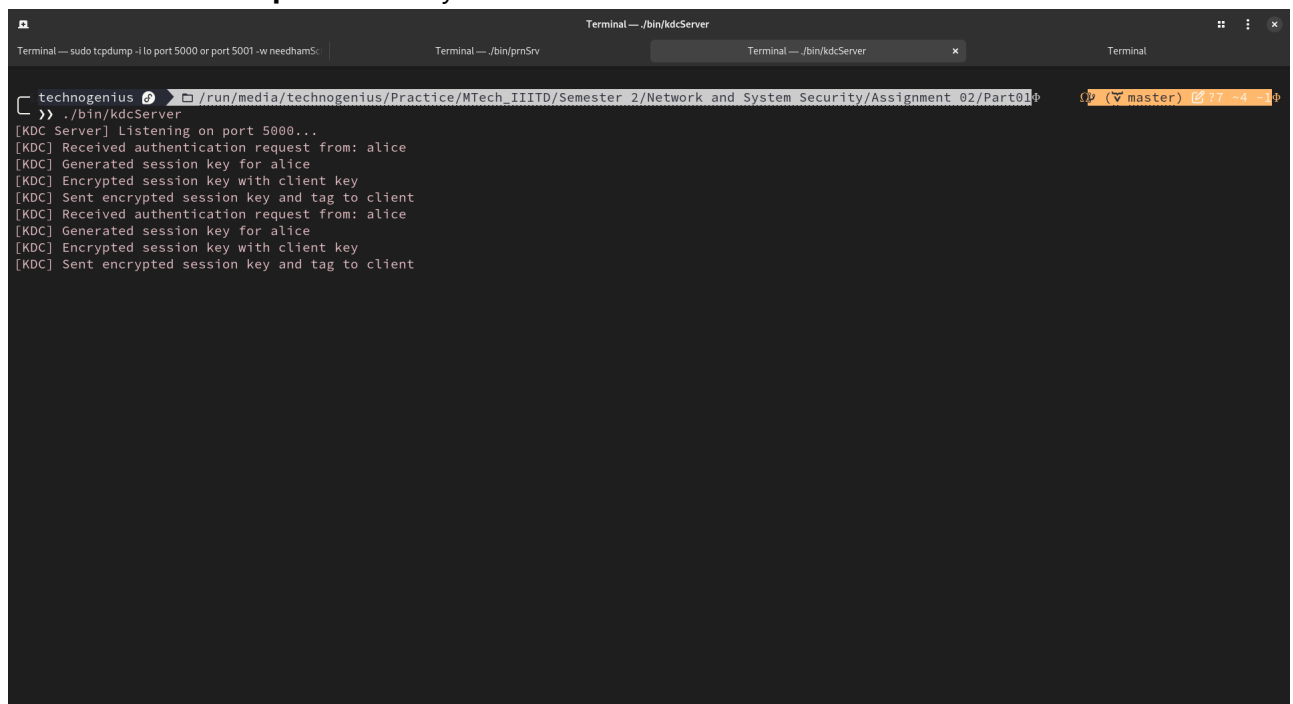
```
gcc -o client client.c cryptoUtils.c logger.c -lssl -lcrypto
gcc -o kdcServer kdcServer.c cryptoUtils.c logger.c -lssl -lcrypto -lpthread
gcc -o prnSrv prnSrv.c cryptoUtils.c logger.c -lssl -lcrypto -lpthread
```

4.2 Running the Servers

Start KDC Server

```
./kdcServer
```

1. KDC authentication process: Verify successful authentication.

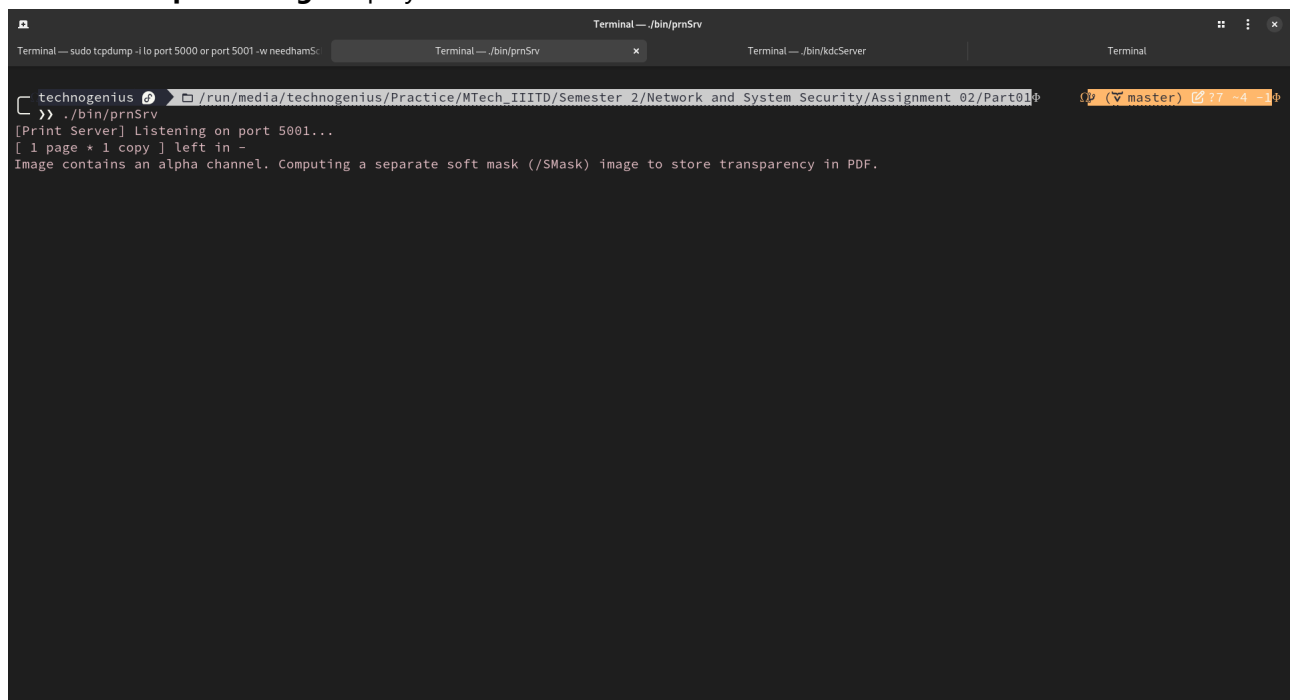
A terminal window titled 'Terminal — /bin/kdcServer' showing the execution of the kdcServer program. The output shows the server listening on port 5000, receiving authentication requests from 'alice', generating session keys, and sending encrypted session keys and tags back to the client. The process is repeated twice.

```
technogenius@ /run/media/technogenius/Practice/MTech_IIITD/Semester 2/Network and System Security/Assignment 02/Part01
>> ./bin/kdcServer
[KDC Server] Listening on port 5000...
[KDC] Received authentication request from: alice
[KDC] Generated session key for alice
[KDC] Encrypted session key with client key
[KDC] Sent encrypted session key and tag to client
[KDC] Received authentication request from: alice
[KDC] Generated session key for alice
[KDC] Encrypted session key with client key
[KDC] Sent encrypted session key and tag to client
```

Start Print Server

```
./prnSrv
```

2. Print Server processing: Display PDF conversion.

A terminal window titled 'Terminal — /bin/prnSrv' showing the execution of the prnSrv program. The output shows the server listening on port 5001, receiving a request, and processing it by computing a separate soft mask for transparency in PDF.

```
technogenius@ /run/media/technogenius/Practice/MTech_IIITD/Semester 2/Network and System Security/Assignment 02/Part01
>> ./bin/prnSrv
[Print Server] Listening on port 5001...
[ 1 page + 1 copy ] left in -
Image contains an alpha channel. Computing a separate soft mask (/SMask) image to store transparency in PDF.
```

4.3 Running the Client

```
./client sample.txt
```

3. Client request handling: Show file being sent and encrypted.

```

Terminal — sudo tcpdump -i lo port 5000 or port 5001 -w needhamSchroeder.pcap
Terminal — ./bin/prnSrv
Terminal — ./bin/kdcServer
Terminal

technogenius @ /run/media/technogenius/Practice/MTech_IIITD/Semester 2/Network and System Security/Assignment 02/Part01
>> ./bin/client hello.txt
[Client] Requesting conversion of file: hello.txt
[Client] Connected to KDC
[Client] Sent username 'alice' to KDC
[Client] Received nonce from KDC
[Client] Received encrypted session key from KDC
[Client] Received authentication tag from KDC
[Client] Decrypted session key successfully
[Client] Connected to print server
[Client] Sent nonce to print server
[Client] Sent filename 'hello.txt' to print server
[Client] Receiving PDF of size: 3801352667 bytes
Read PDF data: Success

technogenius @ /run/media/technogenius/Practice/MTech_IIITD/Semester 2/Network and System Security/Assignment 02/Part01
>> ./bin/client inputs.png
[Client] Requesting conversion of file: inputs.png
[Client] Connected to KDC
[Client] Sent username 'alice' to KDC
[Client] Received nonce from KDC
[Client] Received encrypted session key from KDC
[Client] Received authentication tag from KDC
[Client] Decrypted session key successfully
[Client] Connected to print server
[Client] Sent nonce to print server
[Client] Sent filename 'inputs.png' to print server
[Client] Receiving PDF of size: 261344229 bytes
Read PDF data: Success

technogenius @ /run/media/technogenius/Practice/MTech_IIITD/Semester 2/Network and System Security/Assignment 02/Part01
>>

```

5. Security Verification

5.1 Capturing Encrypted Traffic

Run the following command to capture and analyze network traffic:

```
sudo tcpdump -i lo -w needhamSchroeder.pcap port 5000 or port 5001
```

4. Encrypted network traffic: Wireshark capture demonstrating AES-GCM encryption.

```

Terminal — sudo tcpdump -i lo port 5000 or port 5001 -w needhamSchroeder.pcap
Terminal
Terminal
Terminal

technogenius @ /run/media/technogenius/Practice/MTech_IIITD/Semester 2/Network and System Security/Assignment 02/Part01
>> ls
hello.txt  include  inputs.png  Makefile  Networks_and_Systems_Security_2___w2025_A2.pdf  output.pdf  run.sh  Screenshot  server.log  src

technogenius @ /run/media/technogenius/Practice/MTech_IIITD/Semester 2/Network and System Security/Assignment 02/Part01
>> make
gcc src/kdcServer.c src/cryptoUtils.c src/logger.c -o bin/kdcServer -Wall -Wextra -I./include -lssl -lcrypto -pthread
gcc src/prnSrv.c src/cryptoUtils.c src/logger.c -o bin/prnSrv -Wall -Wextra -I./include -lssl -lcrypto -pthread
gcc src/client.c src/cryptoUtils.c src/logger.c -o bin/client -Wall -Wextra -I./include -lssl -lcrypto -pthread
src/client.c: In function 'connectToPrintServer':
src/client.c:103:55: warning: passing argument 4 of 'decryptData' discards 'const' qualifier from pointer target type [-Wdiscarded-qualifiers]
   103 |         int ret = decryptData(encryptedPdf, pdfSize, kap, nonce, decryptedPdf, tag);
       |                                     ^~~~~~
In file included from src/client.c:9:
./include/cryptoUtils.h:15:52: note: expected 'unsigned char *' but argument is of type 'const unsigned char *'
   15 |         unsigned char *key, unsigned char *nonce,
       |                                     ~~~~~~^~~~~~

technogenius @ /run/media/technogenius/Practice/MTech_IIITD/Semester 2/Network and System Security/Assignment 02/Part01
>> ls
bin  include  Makefile  Networks_and_Systems_Security_2___w2025_A2.pdf  output.pdf  Screenshot  src
hello.txt  inputs.png  run.sh  server.log

technogenius @ /run/media/technogenius/Practice/MTech_IIITD/Semester 2/Network and System Security/Assignment 02/Part01
>> sudo tcpdump -i lo port 5000 or port 5001 -w needhamSchroeder.pcap
[sudo] password for technogenius:
dropped privs to tcpdump
tcpdump: listening on lo, link-type EN10MB (Ethernet), snapshot length 262144 bytes

```

```

technogenius@ /run/media/technogenius/Practice/MTech_IIITD/Semester 2/Network and System Security/Assignment 02/Part01
>> make
gcc src/kdcServer.c src/cryptoUtils.c src/logger.c -o bin/kdcServer -Wall -Wextra -I./include -lssl -lcrypto -pthread
gcc src/prnSrv.c src/cryptoUtils.c src/logger.c -o bin/prnSrv -Wall -Wextra -I./include -lssl -lcrypto -pthread
gcc src/client.c src/cryptoUtils.c src/logger.c -o bin/client -Wall -Wextra -I./include -lssl -lcrypto -pthread
src/client.c: In function 'connectToPrintServer':
src/client.c:103:55: warning: passing argument 4 of 'decryptData' discards 'const' qualifier from pointer target type [-Wdiscarded-qualifiers]
103 |         int ret = decryptData(encryptedPdf, pdfSize, kap, nonce, decryptedPdf, tag);
    |                                     ^~~~~~
In file included from src/client.c:9:
./include/cryptoUtils.h:15:52: note: expected 'unsigned char *' but argument is of type 'const unsigned char *'
15 |         unsigned char *key, unsigned char *nonce,
    |         ~~~~~~^~~~~~

technogenius@ /run/media/technogenius/Practice/MTech_IIITD/Semester 2/Network and System Security/Assignment 02/Part01
>> ls
bin  hello.txt  include  inputs.png  Makefile  Networks_and_Systems_Security_2_2025_A2.pdf  output.pdf  run.sh  Screenshot  src

technogenius@ /run/media/technogenius/Practice/MTech_IIITD/Semester 2/Network and System Security/Assignment 02/Part01
>> sudo tcpdump -i lo port 5000 or port 5001 -w needhamSchroeder.pcap
dropped privs to tcpdump
tcpdump: listening on lo, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C77 packets captured
250 packets received by filter
96 packets dropped by kernel

technogenius@ /run/media/technogenius/Practice/MTech_IIITD/Semester 2/Network and System Security/Assignment 02/Part01
>>

```

5.2 Analyzing the Capture

Open the capture file in Wireshark:

```
tshark -r traffic.pcap -x
```

5. Wireshark

needhamSchroeder.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	127.0.0.1	127.0.0.1	TCP	74	48906 → 5000 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM TSval=1085906423 TSecr=0 WS=128
2	0.000027	127.0.0.1	127.0.0.1	TCP	74	5000 → 48906 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM TSval=1085906423 TSecr=1085906423 WS=128
3	0.000043	127.0.0.1	127.0.0.1	TCP	66	48906 → 5000 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=1085906423 TSecr=1085906423
4	0.000084	127.0.0.1	127.0.0.1	TCP	72	48906 → 5000 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=6 TSval=1085906423 TSecr=1085906423
5	0.000094	127.0.0.1	127.0.0.1	TCP	66	5000 → 48906 [ACK] Seq=1 Ack=7 Win=65536 Len=0 TSval=1085906423 TSecr=1085906423
6	0.045262	127.0.0.1	127.0.0.1	TCP	78	5000 → 48906 [PSH, ACK] Seq=1 Ack=7 Win=65536 Len=12 TSval=1085906468 TSecr=1085906423
7	0.045287	127.0.0.1	127.0.0.1	TCP	66	48906 → 5000 [ACK] Seq=7 Ack=13 Win=65536 Len=0 TSval=1085906468 TSecr=1085906468
8	0.045304	127.0.0.1	127.0.0.1	TCP	98	5000 → 48906 [PSH, ACK] Seq=13 Ack=7 Win=65536 Len=32 TSval=1085906468 TSecr=1085906468
9	0.045311	127.0.0.1	127.0.0.1	TCP	66	48906 → 5000 [ACK] Seq=7 Ack=45 Win=65536 Len=0 TSval=1085906468 TSecr=1085906468
10	0.045322	127.0.0.1	127.0.0.1	TCP	82	5000 → 48906 [PSH, ACK] Seq=45 Ack=7 Win=65536 Len=16 TSval=1085906468 TSecr=1085906468
11	0.045328	127.0.0.1	127.0.0.1	TCP	66	48906 → 5000 [ACK] Seq=7 Ack=61 Win=65536 Len=0 TSval=1085906468 TSecr=1085906468
12	0.045348	127.0.0.1	127.0.0.1	TCP	66	5000 → 48906 [FIN, ACK] Seq=61 Ack=7 Win=65536 Len=0 TSval=1085906468 TSecr=1085906468
13	0.047176	127.0.0.1	127.0.0.1	TCP	66	48906 → 5000 [FIN, ACK] Seq=7 Ack=62 Win=65536 Len=0 TSval=1085906470 TSecr=1085906468
14	0.047205	127.0.0.1	127.0.0.1	TCP	66	5000 → 48906 [ACK] Seq=62 Ack=8 Win=65536 Len=0 TSval=1085906470 TSecr=1085906470
15	0.047276	127.0.0.1	127.0.0.1	TCP	74	49142 → 5001 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM TSval=1085906470 TSecr=0 WS=128
16	0.047290	127.0.0.1	127.0.0.1	TCP	74	5001 → 49142 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM TSval=1085906470 TSecr=1085906470 WS=128
17	0.047300	127.0.0.1	127.0.0.1	TCP	66	49142 → 5001 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=1085906470 TSecr=1085906470
18	0.047325	127.0.0.1	127.0.0.1	TCP	78	49142 → 5001 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=12 TSval=1085906470 TSecr=1085906470
19	0.047334	127.0.0.1	127.0.0.1	TCP	66	5001 → 49142 [ACK] Seq=1 Ack=13 Win=65536 Len=0 TSval=1085906470 TSecr=1085906470
20	0.047357	127.0.0.1	127.0.0.1	TCP	76	49142 → 5001 [PSH, ACK] Seq=13 Ack=1 Win=65536 Len=10 TSval=1085906470 TSecr=1085906470
21	0.047366	127.0.0.1	127.0.0.1	TCP	66	5001 → 49142 [ACK] Seq=1 Ack=23 Win=65536 Len=0 TSval=1085906471 TSecr=1085906470
22	0.236518	127.0.0.1	127.0.0.1	TCP	7815	5001 → 49142 [PSH, ACK] Seq=1 Ack=23 Win=65536 Len=7749 TSval=1085906660 TSecr=1085906470
23	0.236545	127.0.0.1	127.0.0.1	TCP	66	49142 → 5001 [ACK] Seq=23 Ack=7750 Win=109568 Len=0 TSval=1085906660 TSecr=1085906660
24	0.236565	127.0.0.1	127.0.0.1	TCP	82	5001 → 49142 [PSH, ACK] Seq=7750 Ack=23 Win=65536 Len=16 TSval=1085906660 TSecr=1085906660
25	0.236573	127.0.0.1	127.0.0.1	TCP	66	49142 → 5001 [ACK] Seq=23 Ack=7766 Win=109568 Len=0 TSval=1085906660 TSecr=1085906660
26	0.236588	127.0.0.1	127.0.0.1	TCP	66	5001 → 49142 [FIN, ACK] Seq=7766 Ack=23 Win=65536 Len=0 TSval=1085906660 TSecr=1085906660
27	0.236695	127.0.0.1	127.0.0.1	TCP	66	49142 → 5001 [FIN, ACK] Seq=23 Ack=7767 Win=109568 Len=0 TSval=1085906660 TSecr=1085906660
28	0.236718	127.0.0.1	127.0.0.1	TCP	66	5001 → 49142 [ACK] Seq=7767 Ack=24 Win=65536 Len=0 TSval=1085906660 TSecr=1085906660
29	4.419056	127.0.0.1	127.0.0.1	TCP	74	44868 → 5000 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM TSval=1085910842 TSecr=0 WS=128

> Frame 14: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
 > Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)
 > Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
 > Transmission Control Protocol, Src Port: 48906, Dst Port: 5000, Seq: 0, Len: 0

needhamSchroeder.pcap

Packets: 77

Profile: Default

needhamSchroeder.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	127.0.0.1	127.0.0.1	TCP	74	48906 → 5000 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM TSval=1085906423 TSecr=0 WS=128
2	0.000027	127.0.0.1	127.0.0.1	TCP	74	5000 → 48906 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM TSval=1085906423 TSecr=1085906423 WS=128
3	0.000043	127.0.0.1	127.0.0.1	TCP	66	48906 → 5000 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=1085906423 TSecr=1085906423
4	0.000084	127.0.0.1	127.0.0.1	TCP	72	48906 → 5000 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=6 TSval=1085906423 TSecr=1085906423
5	0.000094	127.0.0.1	127.0.0.1	TCP	66	5000 → 48906 [ACK] Seq=1 Ack=7 Win=65536 Len=0 TSval=1085906423 TSecr=1085906423
6	0.045262	127.0.0.1	127.0.0.1	TCP	78	5000 → 48906 [PSH, ACK] Seq=1 Ack=7 Win=65536 Len=12 TSval=1085906468 TSecr=1085906423
7	0.045287	127.0.0.1	127.0.0.1	TCP	66	48906 → 5000 [ACK] Seq=7 Ack=13 Win=65536 Len=0 TSval=1085906468 TSecr=1085906468
8	0.045314	127.0.0.1	127.0.0.1	TCP	98	5000 → 48906 [PSH, ACK] Seq=13 Ack=7 Win=65536 Len=32 TSval=1085906468 TSecr=1085906468
9	0.045311	127.0.0.1	127.0.0.1	TCP	66	48906 → 5000 [ACK] Seq=7 Ack=45 Win=65536 Len=0 TSval=1085906468 TSecr=1085906468
10	0.045322	127.0.0.1	127.0.0.1	TCP	82	5000 → 48906 [PSH, ACK] Seq=45 Ack=7 Win=65536 Len=16 TSval=1085906468 TSecr=1085906468
11	0.045328	127.0.0.1	127.0.0.1	TCP	66	48906 → 5000 [ACK] Seq=7 Ack=61 Win=65536 Len=0 TSval=1085906468 TSecr=1085906468
12	0.045348	127.0.0.1	127.0.0.1	TCP	66	5000 → 48906 [FIN, ACK] Seq=61 Ack=7 Win=65536 Len=0 TSval=1085906468 TSecr=1085906468
13	0.047176	127.0.0.1	127.0.0.1	TCP	66	48906 → 5000 [FIN, ACK] Seq=7 Ack=62 Win=65536 Len=0 TSval=1085906470 TSecr=1085906468
14	0.047295	127.0.0.1	127.0.0.1	TCP	66	5000 → 48906 [ACK] Seq=62 Ack=8 Win=65536 Len=0 TSval=1085906470 TSecr=1085906470
15	0.047276	127.0.0.1	127.0.0.1	TCP	74	49142 → 5001 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM TSval=1085906470 TSecr=0 WS=128
16	0.047290	127.0.0.1	127.0.0.1	TCP	74	5001 → 49142 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM TSval=1085906470 TSecr=1085906470 WS=128
17	0.047300	127.0.0.1	127.0.0.1	TCP	66	49142 → 5001 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=1085906470 TSecr=1085906470
18	0.047325	127.0.0.1	127.0.0.1	TCP	78	49142 → 5001 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=12 TSval=1085906470 TSecr=1085906470
19	0.047334	127.0.0.1	127.0.0.1	TCP	66	5001 → 49142 [ACK] Seq=1 Ack=13 Win=65536 Len=0 TSval=1085906470 TSecr=1085906470
20	0.047357	127.0.0.1	127.0.0.1	TCP	76	49142 → 5001 [PSH, ACK] Seq=13 Ack=1 Win=65536 Len=10 TSval=1085906470 TSecr=1085906470
21	0.047366	127.0.0.1	127.0.0.1	TCP	66	5001 → 49142 [ACK] Seq=1 Ack=23 Win=65536 Len=0 TSval=1085906471 TSecr=1085906470
22	0.236518	127.0.0.1	127.0.0.1	TCP	7815	5001 → 49142 [PSH, ACK] Seq=1 Ack=23 Win=65536 Len=7749 TSval=1085906660 TSecr=1085906470
23	0.236545	127.0.0.1	127.0.0.1	TCP	66	49142 → 5001 [ACK] Seq=23 Ack=7750 Win=109568 Len=0 TSval=1085906660 TSecr=1085906660
24	0.236565	127.0.0.1	127.0.0.1	TCP	82	5001 → 49142 [PSH, ACK] Seq=7750 Ack=23 Win=65536 Len=16 TSval=1085906660 TSecr=1085906660
25	0.236573	127.0.0.1	127.0.0.1	TCP	66	49142 → 5001 [ACK] Seq=23 Ack=7766 Win=109568 Len=0 TSval=1085906660 TSecr=1085906660
26	0.236588	127.0.0.1	127.0.0.1	TCP	66	5001 → 49142 [FIN, ACK] Seq=7766 Ack=23 Win=65536 Len=0 TSval=1085906660 TSecr=1085906660
27	0.236595	127.0.0.1	127.0.0.1	TCP	66	49142 → 5001 [FIN, ACK] Seq=23 Ack=7767 Win=109568 Len=0 TSval=1085906660 TSecr=1085906660
28	0.236718	127.0.0.1	127.0.0.1	TCP	66	5001 → 49142 [ACK] Seq=7767 Ack=24 Win=65536 Len=0 TSval=1085906660 TSecr=1085906660
29	0.419056	127.0.0.1	127.0.0.1	TCP	74	48906 → 5000 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM TSval=1085910842 TSecr=0 WS=128
30	0.419085	127.0.0.1	127.0.0.1	TCP	74	5000 → 48906 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM TSval=1085910842 TSecr=1085910842 WS=128

> Frame 26: 76 bytes on wire (608 bits), 76 bytes captured (608 bits)

> Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)

> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

> Transmission Control Protocol, Src Port: 49142, Dst Port: 5001, Seq: 13, Ack: 1, Len: 10

> Data (10 bytes)

Data: 68656363662e74787400
[Length: 10]

needhamSchroeder.pcap

Packets: 77

Profile: Default

needhamSchroeder.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
49	4.469902	127.0.0.1	127.0.0.1	TCP	66	5001 → 34012 [ACK] Seq=1 Ack=24 Win=65536 Len=0 TSval=1085910893 TSecr=1085910893
50	5.224308	127.0.0.1	127.0.0.1	TCP	32834	5001 → 34012 [ACK] Seq=1 Ack=24 Win=65536 Len=32768 TSval=1085911647 TSecr=1085910893
51	5.224438	127.0.0.1	127.0.0.1	TCP	66	34012 → 5001 [ACK] Seq=24 Ack=32769 Win=94848 Len=0 TSval=1085911648 TSecr=1085911647
52	5.224476	127.0.0.1	127.0.0.1	TCP	32834	5001 → 34012 [PSH, ACK] Seq=32769 Ack=24 Win=65536 Len=32768 TSval=1085911648 TSecr=1085910893
53	5.224466	127.0.0.1	127.0.0.1	TCP	66	34012 → 5001 [ACK] Seq=24 Ack=65537 Win=94848 Len=0 TSval=1085911648 TSecr=1085911648
54	5.224960	127.0.0.1	127.0.0.1	TCP	62146	5001 → 34012 [PSH, ACK] Seq=65537 Ack=24 Win=65536 Len=62080 TSval=1085911648 TSecr=1085911648
55	5.225037	127.0.0.1	127.0.0.1	TCP	66	34012 → 5001 [ACK] Seq=24 Ack=127617 Win=94848 Len=0 TSval=1085911648 TSecr=1085911648
56	5.225479	127.0.0.1	127.0.0.1	TCP	47490	5001 → 34012 [ACK] Seq=127617 Ack=24 Win=65536 Len=47424 TSval=1085911649 TSecr=1085911648
57	5.225523	127.0.0.1	127.0.0.1	TCP	47490	[TCP Window Full] 5001 → 34012 [PSH, ACK] Seq=175041 Ack=24 Win=65536 Len=47424 TSval=1085911649 TSecr=1085911648
58	5.225538	127.0.0.1	127.0.0.1	TCP	66	34012 → 5001 [ACK] Seq=24 Ack=175041 Win=225792 Len=0 TSval=1085911649 TSecr=1085911649
59	5.225544	127.0.0.1	127.0.0.1	TCP	66	34012 → 5001 [ACK] Seq=24 Ack=222465 Win=356736 Len=0 TSval=1085911649 TSecr=1085911649

> Frame 57: 47490 bytes on wire (379920 bits), 47490 bytes captured (379920 bits)

> Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)

> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

> Transmission Control Protocol, Src Port: 5001, Dst Port: 34012, Seq: 175041, Ack: 24, Len: 47424

> Data (47424 bytes)

Data [..]: 74e9554e60fa86b0f7625dcab3aa6c91e1794f940aaab144f9d14b80eb7a6529783496c29ddc91ffe12423562
[Length: 47424]

Bytes 66-47489: Data (data.data)

Packets: 77

Profile: Default

Expected Outcome: Data should be encrypted, showing no plaintext communication.

6. Security Considerations

- **Replay Attack Prevention:** Nonces ensure each session is unique.
- **Authentication Security:** Uses PBKDF2 for strong key derivation.
- **Data Integrity:** AES-GCM ensures confidentiality and integrity.
- **Denial-of-Service (DoS) Protection:** Multi-threading ensures performance...