Segurança de Redes e Sistemas de Computadores
Computer Network and System Security: 1º Sem. 2024/2025

Project Assignment 1 for Mandatory Frequency Assessment
(v 1.0)

A Secure Channel Abstraction Supported by a Datagram-Based Secure Transport Protocol

Abstract

The goal of this project assignment is to design, implement, analyze, and experimentally validate a secure solution, materialized by a secure datagram (UDP-based) transport protocol that provides message authenticity, integrity, and confidentiality using configurable cryptographic methods. The designed solution should be aimed at supporting a generic and parameterizable secure communication abstraction, providing a reusable solution that can be used by programmers to implement secure datagram channels for Java applications. Additionally, the solution can be reused to protect pre-existing unsecured applications using UDP and Datagram sockets. The idea is that pre-existing unsecured solutions can adopt the provided solution in a way that requires minimal changes to the original code.

1. Introduction

The goal must be addresed by designing a solution providing a generic abstraction and primitives to be used by programmers to implement datagram secure transport channels for applications developed in Java, according to the following architectural stack represented in Fig.1.

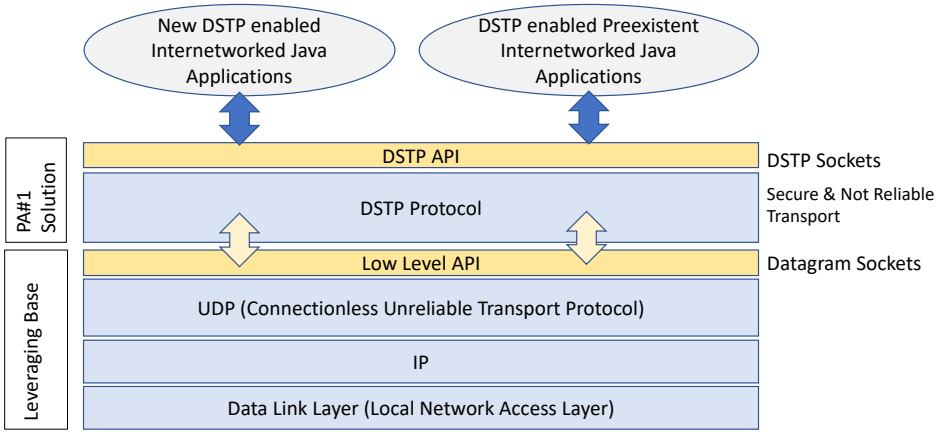

Fig.1. The system model, reference architecture and communication stack

The DSTP protocol and the provided API that must be developed will provide a programming abstraction based on DSTP Java Sockets, that must be leveraged by DSTP and UDP enabled Java Datagram Sockets.

1.1 Adversary model conditions and DSTP security properties

By using the developed solution, programmers must be able to write new DSTP enabled secure applications or easily porting pre-existent unsecure applications using Datagram sockets and UDP transport level communication. DSTP and DSTP sockets will provide security guarantees for the following properties (as defined in the OSI X.800 security framework):

- Message connectionless integrity without recovery to protect message tampering in UDP exchanged payloads and integrity of UDP packets
- Message authenticity preserving the authenticity of send/received messages
- Message connectionless confidentiality

The included integrity guarantees must provide traffic ordering integrity, allowing receivers to discard out-of-order packets (not recoverable packet losses and discarding of out order packets), only processing packets with right sequence order.

1.2 DSTP internetworking model

Fig.2 shows the interaction between distributed processes (in distributed applications), representing the secure DSTP link abstraction used for inter-process communication.

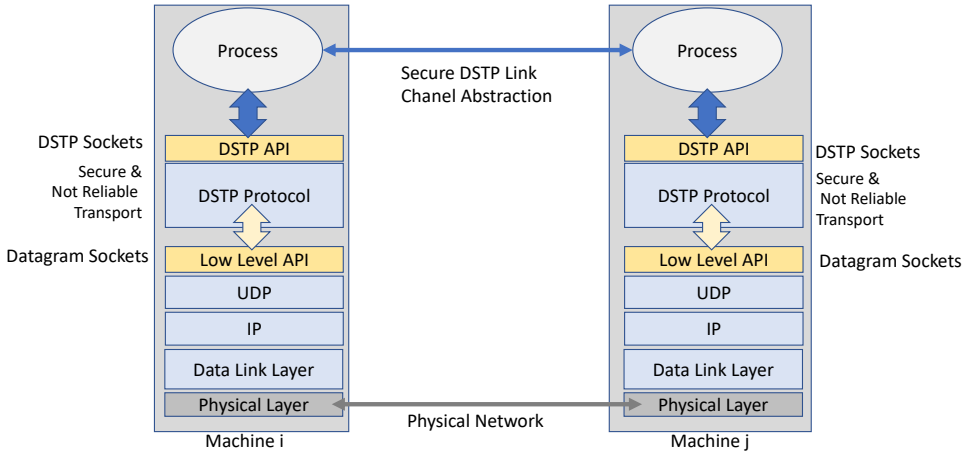

Fig.2 Distributed applications and internetworking processes using DSTP secure channels

To provide the required security properties, DSTP will use the adequate cryptographic methods and algorithms, that can be used in parameterizable cryptographic
The DSTP process links must offer the necessary guarantees for adversary model conditions related to the following threats and concretization of attacks:

- Message and UDP packets replaying.
- Attacks on message and packets connectionless integrity by tampering.
- Attacks against message authenticity beyond message integrity guarantees.
- Attacks for data-disclosure conditions and confidentiality breaks of exchanged UDP payload data.
- Attacks on out-of-order conditions (beyond possible out of-order conditions inherent to the UDP transport protocol characteristics.

Note that confidentiality must be guaranteed even if attackers conduct man-in-the-middle attacks by attacking per-authenticity guarantees on IP addressing or MAC-level addressing, as consequences of possible DNS name resolution attacks, DHCP Spoofing or ARP Spoofing and poisoning of ARP caches
You must consider only adversary model conditions defined above. For example, just for clarification, the following security properties and possible related attacks are out of scope of the DSTP requirements in the PA#1 assignment:

- Any other security property defined in OSI X.800 reference model not included above
- Attacks against peer-authenticity, including IP masquerading or
- Peer-Authentication and any related spoofing, sybiling or masquerading attacks
- Connection-Oriented Confidentiality or Connection-Oriented Confidentiality breaks
- Connection-Oriented Integrity or Connection-Oriented Confidentiality breaks

- Integrity guarantees with recovery guarantees and attacks against integrity recovery
- Traffc flow confidentiality guarantees and related attacks
- Availability guarantees protecting Denial of Service Attacks against processing nodes and against network
- Intrusions injecting malicious processing on host-based software stacks on involved computers (in any level of processing resources and process execution and data management) and all computer-level resources are in the trusted computing base assumptions.
- We consider that users running the applications in the hosting machines are trusted and operate the applications correctly, without the interference of intruders or intrusion-attacks.

2. DSTP Cryptographic parameterizations

The DSTP protocol must be designed with transparency assumptions, in order to be parameterizable not using any cryptographic parameterizations in implemented code or in the code of applications. For this purpose, each endpoint will be parameterized with configuration files and keystores not forcing the recompilation of source code, and the implementation must dynamically adapt, transparently to the provided configurations.

Fig. 3 shows the use of configurable cryptographic parameterizations established in th two represented files

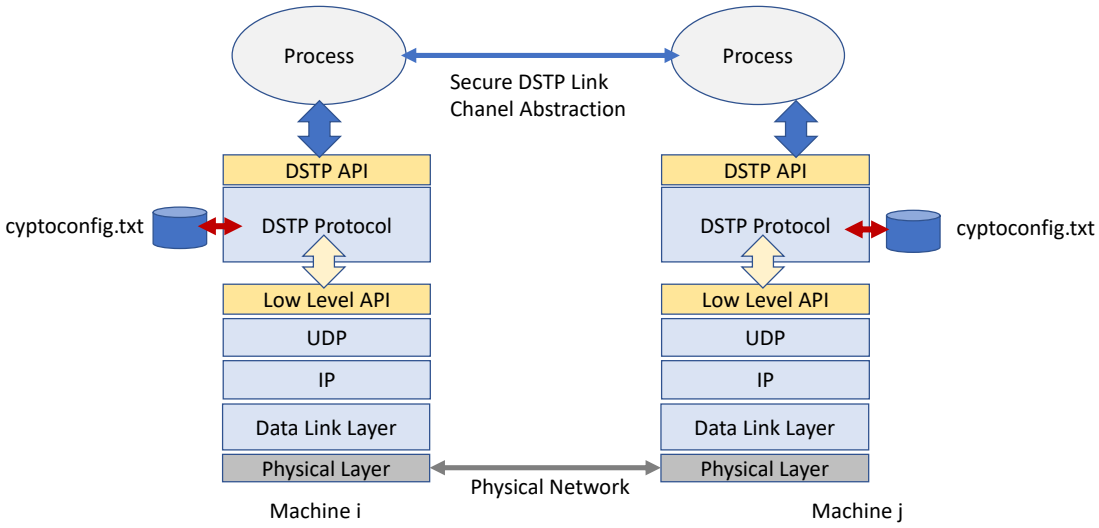

Fig. 3 – Configurable cryptographic parameterizations for DSTP endpoints

The parameterizations in the represented file are the following

- cryptoconfig.txt: is a text file contains the parameterizations defining cryptographic methods, algorithms and related ciphersuite compositions

2.1 cryptoconfig.txt format

```
CONFIDENTIALIY: ALG/MODE/PADDING
SYMMETRIC_KEY: key in hexadecimal value with the required key size
SYMMTRIC_KEY_SIZE: integer representing the  number of BITS
IV Size: integer or NULL
IV: hexadecimal value or NULL
INTEGRITY: HMAC or H
H: definition of secure ash Function or NULL
MAC: definition of MAC (HMAC or CMAC algorithms)
MACKEY: mackey value in hexadecimal with rquired keysize or NULL
MACKEY_SIZE: integer representing the size of the MACKEY in BITS
```

For development you can use and test different valid parameterizations. Note that you must know about valid parameterizations for the provided cryptographic configurations, according to the learned cryptographic constructions in Lab classes and when you need to define IV values or when fields must be defined as NULL when not needed or used. For the submission, you will receive a set of different reference configurations for the evaluation of your work.

Remember that for the use of Block ciphers, the Confidentiality must be defined with a valid configuration for block ciphers or stream ciphers, supported by Java crypto providers ex:

AES/CTR/NoPadding
RC6/CBC/PKCS5Padding
AES/GCM/NoPadding
BLOWFISH/ECB/PKCS5Padding
DESede/CBC/NoPadding
IDEA/CTR/NoPadding
RC4
CHACHA20-Poly1305
ChaCha20

For the definition of Secure Hash functions, use the field H, ex:
H: SHA224
H: SHA256
H: SHA512
H: SHA3-256
H:RIPEMD320

For the definition of Secure Hash functions, use the field HMAC ex:
RC6GMAC
HMAC-SHA512
HMACSHA384
HMAC-SHA256
AESGMAC
HMACSHA3-512

4. DSTP API

The implementation of DSTP must be done to provide a secure channel communication abstraction, implemented as DSTP sockets (leveraged by Datagram Sockets). Students must design the solution implementing the DSTP Socket abstraction (as an object class), with the primitives implemented by means of DSTP methods. Is part of the work assignment the definition of provided methods and DSTP class specifications (that can be supported after the implementation as Javadoc specifications for DSTP socket

See as reference the native specification for Datagram, sockets. Note that you only need to implement support for the required Dataram socket operations used in the targeted applications that will be used in testbench. (see section 8).

5. DSTP protocol message format and cryptographic elements

The following figure represents the reference format of DSTP encpsultion carried in base UDP datagram and IP pckets

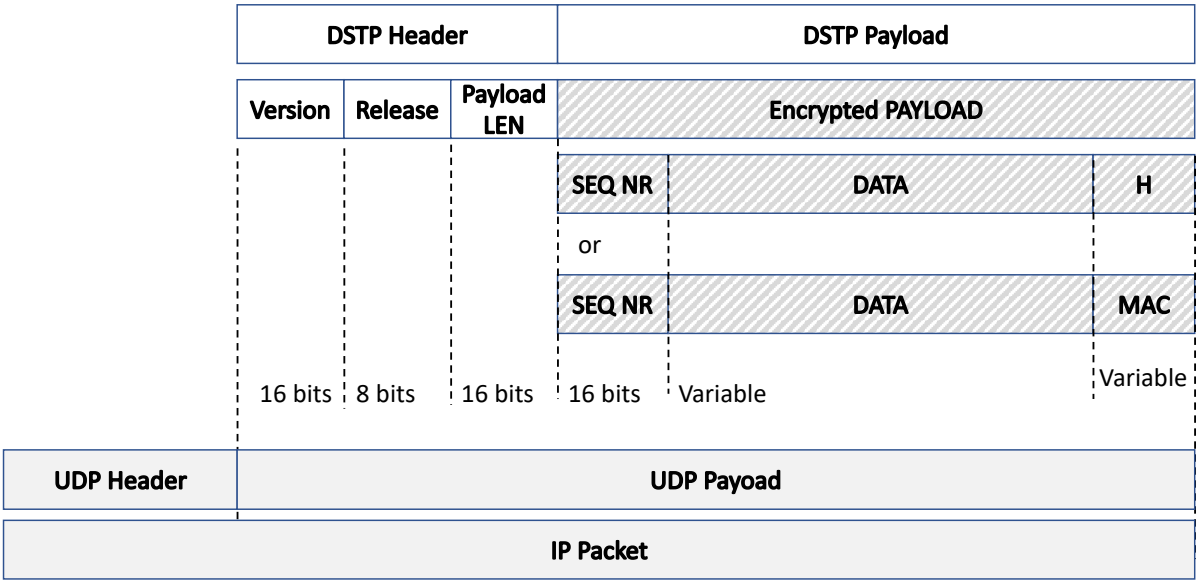

Fig.6. DSTP Cryptographic Format and DSTP Payloads encapsulated by UDP Packets and IP Packets

6. Validation of DSTP using preexistent applications and demonstration testbench environments

The implementation must be tested using preexistent applications supported by Datagram Sockers that must be used for tests and proofs-of-concept. Tests can be done initially in testbench environments using local hosts but tests with the following testbench distributed environments will be given more value:

- Demonstrations supported on docker and downloadable docker-solutions (from Docker Hub accounts) beyond, the source code in delivered GitHub repository
- Demonstrations supported and tested using virtual machines with virtualization/hypervisor technology
- Demonstrations involving different distributd physical machines

7. Experimental critical analysis

Students must conduct testing observations, to conclude about the correctness of delivered implementations. In the submission form students will be asked to report the tests and observations related to the evaluation criteria that will be confirmed later in the detaild evaluation of delivered solutions

8. Evaluation Criteria

The following table is the reference for the evaluation criteria of project assignment 1

TABLE 1: Evaluation criteria		
Criteria		Ref. evaluation
Correctness and compliance of DSTP implementation		
Provided API based on DSTP Sockets and implemented primitives (DSTP Sockets and provided operations as Java primitives to support DSTP enabled applications)		
Support of adequacy of the solution for the dynamic adaptation to different cryptographic parameterizations		

Correct operation of provided Multicasting Application using DSTP and provided DSTP Sockets		
Correct operation of provided TFTP implementation using DSTP and provided DSTP Sockets		
Correct operation of provided Streaming Service implementation using DSTP and provided DSTP Sockets		
Analysis of comparative observed throughput in sent/received messages Kbytes/Second in original and DSTP enabled application		
Analysis of comparative observed throughput in sent/received messages Kbytes/Second in original and DSTP enabled Streaming		
Optimization for the minimization of code modifications in provided code of original applications and DSTP enabled applications		

The table is a reference for the specific implementation evaluation. The final evaluation of PA#1 will follow the remaining evaluation conditions of students, as described in the course evaluation rules for project assessment and practical evaluation

9. Initial materials and project delivery conditions

a. Provided materials

As initial materials, there are three Java applications, ready to compile and run. Tese applications use Datagram Sockets:

Test Multicast, with the following components:

- MulticastSender: a process that sends messges using IP multicast addresses
- MulticastReceiver: a process that receives multicast messages sent by MulticastSender
- README file shows how to run MulticastSender and MulticastReceiver

TFTP Master: a client/server application implementing file transfer based on the TFTP protocol (https://en.wikipedia.org/wiki/Trivial_File_Transfer_Protocol), Ref. in RFC 1350.

- TFTP Client supports TFTP uploads/downloads files to/from the TFTP Serer
- README file explains how to use TFTP Client and TFTP Server

Streaming Service: composed b a Streaming Server and a Proxy

- Streaming server: a server for real-time media streaming (based on MPEG-leveraged movies)
- Proxy: a proxy that receives media streaming sent by the Streming server and can redirect the streamed movies to media player applications (such as VLC: https://www.videolan.org/vlc/ or MPV https://mpv.io/), freely downloadable from the related sites
- README file explains how to use the Streaming Server, Proxy and how to test the use of VLC or MPV

The above materials are provided in: **https://asc.di.fct.unl.pt/~hj/srsc2425/**

In Lab class, the use of these initial provided applications will be demonstrated.

The mission in the work assignment will be the use of the DSTP solution to provide the transformation of such applications, to be protected by your DSTP implementation. The transformed applications will be used to test the correctness of your implementation. The

minimum changes you need in the provided code of the provided applications, the best is you solution as addressed above.

b. Project delivery conditions

Implementations (source code) must be in a Github repository for the submission process.

Project submission involves:

- Project delivery on due date using a Google Form for submission. Th URL of the Google Form will be provided until 24/Oct/2024 and announced in a CLIP message that will be sent for all registered students
- In the Form students will be asked the include the URL of the GitHub Repository that must be shared with “henriquejoaolopesdomingos”
- The Submission Form includes questions with required answers covering the characterization and achieved challenges related to the project specifications nd implementation results. Some included questions will target answers on obtained experimental results covering qualitative and quantitative metrics that students must obtain when testing the implementation, its completeness, and compliance with the specifications. The answers in the submission form and the confirmation and validity of answers checked with the project evaluation will be considered as an evaluation element.

Delivery deadline and important dates:

- Delivery date for the Form submission and GitHub Repository access and availability
  - Submission form will open on 24/OCT/2024, 00h01m
  - Submission can be done from 24/OCT/24 to 27/OCT/24
- Penalizations for late delivery
  - Submissions from 28/OCT/24 to 31/OCT/24 have penalizations of 1 point (0-20 scale) per late day
- Deadline date (limit for PA1 delivery): 31/OCT/24