Network Architectures and Services
Informatics
Technical University of Munich



# Network Security

| | | | |
|---|---|---|---|
| **Module:** | IN2101 | **Date:** | Thursday 23rd February, 2023 |
| **Examiner:** | Prof. Dr.-Ing. Georg Carle | **Exam:** | Endterm |

| | | | |
|---|---|---|---|
| **First name:** | Roberto | **Registration number, UIDs:** | 03767095, E0076 / S0135 |
| **Last name:** | Castellotti | **Last update:** | Wednesday 8th March, 2023 |

| Σ (Endterm) | Grade (Endterm) | Grade intervall |
|---|---|---|
| 54.0 | **2.0** | [51.0; 54.5) |

**Notes:**

- Please make sure that the total amount of credits stated above is correct.
- Solely the second correction (green color) is decisive.

**Corrections:**
The table below lists all corrections (image recognition and complaints during review) that are already considered in the calculation of your grade. If a problem or subproblem is listed multiple times, the correction with the highest number (column “Correction”) takes precedence.

| Problem | Correction | credits | Annotations |
|---|---|---|---|

Network Architectures and Services
Informatics
Technical University of Munich

03767095 M 7

S0135

Note:

- During the attendance check a sticker containing a unique code will be put on this exam.
- This code contains a unique number that associates this exam with your registration number.
- This number is printed both next to the code and to the signature field in the attendance check list.

# Network Security

**Exam:** IN2101 / Endterm **Date:** Thursday 23rd February, 2023
**Examiner:** Prof. Dr.-Ing. Georg Carle **Time:** 08:00 – 09:15

| | P 1 | P 2 | P 3 | P 4 | P 5 | P 6 |
|---|---|---|---|---|---|---|
| I | | | | | | |
| II | | | | | | |





## Working instructions

- This exam consists of **16 pages** with a total of **6 problems**.
  Please make sure now that you received a complete copy of the exam.
- The total amount of achievable credits in this exam is 75 credits.
- Detaching pages from the exam is prohibited.
- Allowed resources:
  - one **analog dictionary** English ↔ native language
- Subproblems marked by * can be solved without results of previous subproblems.
- **Answers are only accepted if the solution approach is documented.** Give a reason for each answer unless explicitly stated otherwise in the respective subproblem.
- Do not write with red or green colors nor use pencils.
- Physically turn off all electronic devices, put them into your bag and close the bag.

Left room from _____ to _____ / Early submission at _____

☐ Exam empty

## Problem 1 Mixed Problems (17.5 credits)

0 1 2 3 **3.0** a)* Name the six *security goals* we have defined in the lecture.

CONFIDENTIALITY
INTEGRITY
AUTHENTICATION
AVAILABILITY
(AUDITABILITY) better: ACCOUNTABILITY
CONTROLLED ACCESS

✓

DNSSec is a technology that hardens DNS against various threats. To achieve this goal, various new records were introduced, such as *NSEC* and *NSEC3* records.

0 1 2 **1.5** b)* Explain the problem NSEC records have that NSEC3 solves. Also explain how NSEC3 solves it.

We need in some way to certify that some records do not exist. The simplest and safest (not distributing ZSK) solution are NSEC records.
Basically we have a way to show that no domains are present between intervals. ✓
This can lead to zone walking. To stop this (or at least make it slower) we have some lexicographic order. ✓ → Problem?
In NSEC3 records we use hashes instead of domain names. ✓
Now someone who wants to do zone walking needs to compute hashes. ✓

Another record introduced by DNSSec is the *Delegate Signer (DS)* record.

0 1 2 **0.0** c)* What is the content and purpose auf a DS record?

A CA can allow some other entity to sign certificates

✓

The TLS handshake allows an extension called *Server Name Indication (SNI)*.

0 1 **0.5** d)* What is the purpose of SNI and what privacy problems does it have?

An external intruder may monitor what sites we are visiting. ✓

0 1 **1.0** e)* What is TCP SYN Flooding, how does it work and what problems does it cause?

We can keep sending SYN packages and overload the server. It is a DoS attack

✓ how?

1.0 f)* Name and explain one defense against TCP SYN Floods. What is the drawback of this mechanism?

We can use TCP Syn cookies. ✓
A drawback is we have to use some (potentially expensive) cryptography. → hash
This might lead to some delays. ✓
Weakening the cryptography to be faster is not a good idea.
Fortunately Linux kernel uses some fast and strong crypto algorithm.

0 1 2 3

The HTTP standards specify a policy called *HTTP Strict Transport Security (HSTS)* which forces clients only to use HTTPS for specific sites.

1.5 g) What is the purpose of HSTS preloading and how does it work?

Browsers come preinstalled with lists of domains ✓ that have HSTS. ✓ Now it also happens for entire TLDs (.bank for example).
This means we are safer (first use might be a problem without preloading) ✓

0 1

In classical asymmetric cryptography, signing data requires the private key of the asymmetric key pair.

1.0 h)* In the lecture we discussed problems related to private keys. State *one* problem and explain this problem's possible outcome.

We must be really sure that noone can access it.
Failing to protect a private key allows anyone to sign in lieu of someone else
↑ Failing? — inaccurate

0 1

As a remedy of the problems discussed in Task h) we introduced *threshold cryptography*, specifically threshold signing.

0.5 i)* Outline the idea/approach of threshold signing!

We distribute a private key ✓ over several machines. → How do we distribute, key sharing not mentioned
We need to compromise at least a threshold of machines to steal the private key.
We need different servers to sign something. Allows to have some "consensus"
How many?

0 1

## Problem 2 Cryptographic Basics (10 credits)

Encryption functions like AES are *deterministic* (same input → same output). This property can be problematic for information secrecy in certain situations.

known PT/CT attack → att sees known ct → map to pt!

0 1 **0.5** a)* Name an attack that can exploit this property of encryption functions and briefly explain!

~~Known plaintext attacks~~

Chosen plaintext attack, we can send to an oracle a plaintext we know.
Read the ciphertext.
Try to gain info about key.

*Cipher Block Chaining (CBC)* is a mode of encryption for block ciphers that can mitigate determinism when implemented correctly. CBC encryption can be expressed mathematically as follows:

$$c_0 = IV$$
$$c_i = Enc_k(p_i \oplus c_{i-1})$$

(Explanations: $p_i$ / $c_i$ refer to the plaintext/ciphertext block $i, i \geqslant 1$, *Enc* refers to any block cipher, $\oplus$ is the XOR function, and *IV* is the initialization vector).

0 1 2 **1.0** b)* Explain why CBC encrypts identical plaintext *blocks* into different ciphertext blocks!

We are xoring the ith block with block $c_{i-1}$. Why does it solve problem?
Additionally to start this off we are using a FRESH Initialization Vector.
The fact it is fresh means same plaintext → different ciphertext
~~Freshness is very important~~. Detail: $c_{i-1}$ masks $p_i$ with "random looking bits" before encryption



0 1 2 **0.0** c)* Explain why and under what circumstances CBC encrypts identical plaintext *messages* into different ciphertext messages!

If we set IV to zero. This is used for example in CBC-MAC.
It is something that is actually desired for signatures. Only for signatures!
No!

Fresh random IV masks $p_0$, randomness propagated to $p_i$ due to $p_i$ XOR $c_{i-1}$

Alice and Bob have each generated an asymmetric RSA key pair and have exchanged their public keys.
Alice knows: $sk_{Alice}$, $pk_{Alice}$ and $pk_{Bob}$
Bob knows: $sk_{Bob}$, $pk_{Bob}$ and $pk_{Alice}$
(Explanations: $sk_i$ is the private key, $pk_i$ the public key of participant $i$)
Alice and Bob use the simple protocol shown in Fig. 2.1 to exchange a session key for a secure channel:

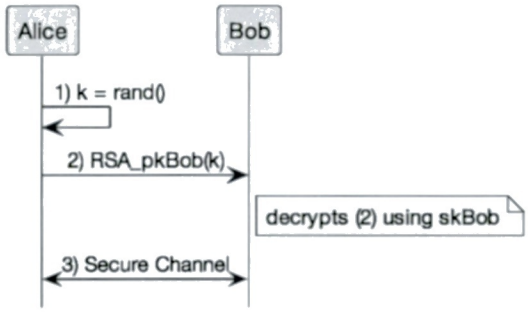



Figure 2.1: A simple key transport protocol

Explanations:

- **Message 1**: Alice generates session key $k$ using a cryptographic random number generator *rand*()
- **Message 2**: Alice sends $k$ RSA-encrypted with $pk_{Bob}$ to Bob
- Bob RSA-decrypts message 2 with $sk_{Bob}$. He now knows $k$
- **Message 3**: Alice and Bob use $k$ as key in a secure channel

3.0 d)* Explain why we encrypt $k$ with *asymmetric* encryption and why we encrypt the data exchanged between Alice and Bob with *symmetric* encryption! Also answer: What is the name of this approach?

Asymmetric cryptography has several benefits, but is is pretty computational intensive. We only use this to exchange symmetric keys. ✓
Symmetric keys are way faster and very secure as well. ✓
This is an hybrid approach and is very used. ✓

0 1 2 3

The key transport protocol shown in Fig. 2.1 does not satisfy *Perfect Forward Secrecy (PFS)*.

1.0 e)* Explain what PFS is and why the protocol in Fig. 2.1 does not satisfy this property!

PFS is Perfect Forward Secrecy → instead of using the longterm keys we exchanged we determine a session key each time. An attacker cannot decrypt messages he captured previously if he manages to steal a private key. Ok
The reason it is not enabled in this protocol is we are not computing a session key but we are directly using a longterm key. No

0 1 2

PFS: long term key compromised → session key still secure
⇒ $sk_{Bob}$ compromised → k compromised in our protocol!

## Problem 3 Cryptographic Protocols (10 credits)

Alice and Bob want to exchange messages $m$. They need to prevent the Dolev-Yao attacker Mallory from modifying their messages $m$. Therefore, they have already agreed on a cryptographic hash function $H$ that will be used to protect $m$. Alice sends the following data over the network: $m, H(m)$ (see also Fig. 3.1).

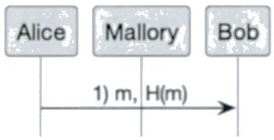



Figure 3.1: A very simple communication protocol

0 1 **1.0** a)* Explain briefly why Mallory can still modify messages exchanged between Alice and Bob!

We are not providing any authentication. An attacker can hash a message he wants to send just like Alice does.
Bob has no way to detect it. ✓

Now assume that Alice and Bob have also agreed on a shared secret symmetric key $k$.

0 1 2 **1.5** b)* Define a *basic* Message Authentication Code (MAC) construction based on $H$. You may ignore advanced attacks such as length-extension attacks. Use mathematical notation $MAC(m, ...) = H(...)$ for your answer. "..." indicates that something is missing here – add the missing information! Briefly explain why this construction prevents Mallory from changing $m$.

$MAC(m,k) = h(m \| k)$ -0,5

What this means is we are adding a key.
This works under the assumption that we know we exchanged the keys safely and only a certain person has the key. ✓
What I said applies if we want to provide some authentication.

Alice and Bob are happy that Mallory can no longer tamper with their messages $m, MAC(m, ...) = H(...)$. However, Mallory has found a new way to mess with Alice and Bob: Mallory replays old messages that were exchanged between Alice and Bob some time ago. So, a message flow as shown in Fig. 3.2 might occur.


Figure 3.2: Mallory annoys Bob by replaying old messages

In the lecture, we discussed two approaches to avoid replay attacks (c.f. Fig. 3.2): either adding a *nonce* or a *timestamp* to the protocol.

1.0 c)* Briefly discuss the pros and cons of using a nonce vs. using a timestamp!

Using a timestamp requires that clocks are synchronized. ✓
This is a pretty strong assumption.
Using nonces is way simpler.

Nonce: unique → need to track
Timestamp: simplify tracking, but require sync'd clocks

0 1 2

0.0 d) Refine message 1 from Fig. 3.2 ($m$, $MAC(m, ...) = H(...)$) using a nonce $n$ so that this message is *additionally* protected against being replayed! Your answer must specify what is sent over the network and how the values are computed!

m, MAC(m,k) = H(m,k), rand() ?

This way we can

we send $m, n, H(m \mid n \mid K)$

0 1 2

Nonces can be simple counter variables that are monotonously incremented for each message. Challenges used in authentication protocols must originate from a high entropy source. Assume a challenge/response protocol in which a nonce (counter variable) is accidentally used in place of a proper challenge (random value)!

1.0 e)* Briefly explain how an attacker might exploit this weakness!

If it is easy to guess an attacker may choose to bruteforce it. ✓

0 1

*Enc-then-Mac* and *Mac-then-Enc* are two approaches to combining a cryptographic cipher with a MAC function.

1.5 f)* Which of the two approaches is generally more secure than the other? Briefly explain why! Also give an example of a problem that this approach prevents!

Enc-Then-Mac is safer. ✓ c-text auth'd
~~Mac does not~~ Enc does not provide integrity
If we detect tampering we can discard message without:
- wasting CPU cycles
- giving additional info to attacker.

0 1 2

## Problem 4 Public Key Infrastructures, X.509 and the X.509 landscape (14 credits)

0 1 2 **2.0** a)* Briefly explain what a certificate is.

A certificate is a mechanism to bind the verified identity of someone to a keypair. With a certificate we can be sure we are really talking with the person we wanted.

0 1 **0.5** b)* Briefly explain what a *root certificate* is and what its specific role is!

A Root certificate is a certificate for a CA. Since it is "on top of the chain" we need to embed them in OSes or browsers.
Self signed certificates as trust anchor

0 1 **1.0** c)* Briefly explain why root certificates are delivered to devices in *root stores* (i.e., as part of the operating system or included in software products)!

We have no way to go "upper" and verify a chain, since this is the root of the chain. Thus we need to have them locally stored.

0 1 2 **2.0** d)* Briefly discuss why the *root store vendor* (i.e., the entity that controls a root store) has such great power! Also, give an example of how a root store vendor might abuse its power, and describe the impact of that attack.

They have a great power because essentially they are telling end users who should be trusted.
If a Root Store Vendor were to decide that someone is not trustworth millions of end users for example may not connect safely to websites. This may be abused to prevent end users to browse safely the websites they want to visit.

0 1 2 **2.0** e)* Briefly explain the purpose of *Certificate Authority Authorization (CAA)* and how the required information is published!

Basically we can decide which Certification Authority should be able to issue certificates for our service.
This information is published in DNS records.

2.0 f)* When does CAA work and when does it not?

CAA only works if Certificate Authorities actually check the DNS Records before emitting a certificate.
Every CA should do this and it is in their interest, but technically speaking nothing prevents them to issue a certificate for a domain.
Reduces attack surface if attacker tries to trick lowest CA
Does not work if CA is compromised <= -1

0.5 g)* Briefly describe what *Certificate Transparency (CT)* is and why it is important!

Certificate Transparency is a public append-only log used for certificates.
Every CA should append certificates here.
Anyone can log of issued certs -> helps to detect rogue ones

1.0 h)* What is the function of a *Signed Certificate Timestamp (SCT)* in the context of CT?

Since we may have some delays appending the certificate to the CT log we include a timestamp to indicate when we signed a CSR.

1.0 i)* Why should your browser, as a public certificate validation instance, not report all the certificates it sees to a public CT log monitor?

They are already inserted by the CAs when they are issued.
There is no point in doing this.
This might also have some privacy drawbacks if implemented poorly.

1.0 j)* Why is CT an after-the-fact solution?

We cannot prevent misuse, we are only offering a-posteriori tooling to understand what went wrong. This is still very important and useful

## Problem 5 Firewalls and Middleboxes (16 credits)

After finishing your studies at TUM, you manage to get a job at a small Bavarian trading company for sustainable use of recycled wood furniture, *ShareWood4Rest*. The place is rather oldschool, like its employees. The company network is given in Figure 5.1.

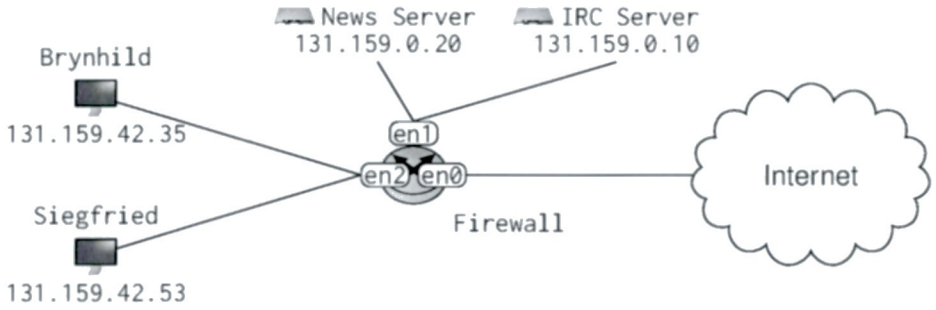



Figure 5.1: Network Topology for Firewall

- Your managed network is 131.159.0.0/16
- There is a server network at 131.159.0.0/24
- Clients are located in 131.159.42.0/24
- A firewall runs on the central router

Your new boss asks you to secure the company network, using a firewall. Today he saw an advertisement for the *KasperCloud AntiBacta* stateless firewall in the local newspaper. Now he wants to discuss the characteristics of firewall types with you.

0 1 2 **2.0** a)* Briefly outline how a stateful firewall works.

Keeps memory of flows (identified by the 5-tuple). We now have a way to detect whether a connection is established or not.

0 1 2 **2.0** b)* Outline one upside and one downside of a stateless firewall. Briefly justify your answers.

Upside: they require less memory to work

Downside: harder to configure and thus more error prone.

0 1 2 **2.0** c)* Briefly explain two configuration strategies for a stateless firewall.

BLACKLIST → we maintain a list of not allowed services, everything else is allowed

WHITELIST → everything is not allowed, we then allow only certain connections.

0 1 **1.0** d) Which of the strategies you explained in Task c) would you implement if security has the highest priority? Briefly justify your answer.

I would use a whitelist.

Higher security but more frustrated users. In this case priority is security

7.5 e)* In order to ensure security, the new firewall should protect the company networks, given in Figure 5.1. Intended behavior of the firewall:

- Only the specified connections are allowed
- Clients can access websites via HTTPS (TCP port 443) in the Internet
- Clients can resolve names with DNS (UDP port 53) in the Internet
- Clients use SSH (TCP port 22) to manage the servers in the server network
- Clients and Internet may access the IRC server (TCP port 194)
- Clients may access the News server (TCP port 119)
- News server can access the Internet (TCP port 119) to sync with other news servers

The implementation is done in a **stateful** firewall.

- In all fields, use * to match any value.
- You can write ! as a negation of the given IP set, i.e. "everything except ...".

NO, this shouldn't be allowed to go anywhere...

this permits answers from server!

| Rule | Incoming Interface | Src IP | Dst IP | Proto | Src Ports | Dst Ports | State | Action |
|---|---|---|---|---|---|---|---|---|
| A | * | * | * | * | 71023 | * | EST | Accept |
| B | en2 | 131.159.42.0/24 | * | TCP | 71023 | 443 | NEW | Accept. |
| C | en2 | 131.159.42.0/24 | * | UDP | 71023 | 53 | NEW | Accept. |
| D | en2 | 131.159.42.0/24 | 131.159.0.20/24 | TCP | 71023 | 22 | NEW | Accept |
| E | ~~en2~~ | ~~131.159.0.10~~ | | | | | | |
| F | ~~en0~~ | ~~131.159.0.10~~ | | | | | | |
| G | en2 | 131.159.42.0/24 | 131.159.0.20 | TCP | 71023 | 119 | NEW | Accept |
| H | en1 | 131.159.0.20 | internet | TCP | 71023 | 119 | NEW | Accept. |
| I | | | | | | | | |
| J | | | | | | | | |
| K | | | | | | | | |
| L | * | * | * | * | * | * | NEW | Drop. |

I amended rows E and F, check the table.

Additional table for this task in case you made mistakes. Strike out invalid solutions or leave this table blank.

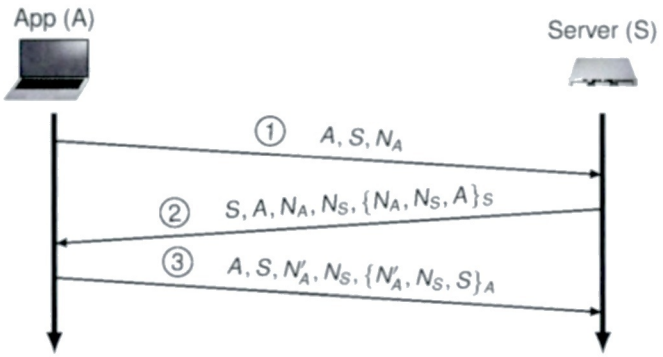
| Rule | Incoming Interface | Src IP | Dst IP | Proto | Src Ports | Dst Ports | State | Action |
|---|---|---|---|---|---|---|---|---|
| A | | | | | | | | |
| B | | | | | | | | |
| C | | | | | | | | |
| D | | | | | | | | |
| E | en0 | internet | 131.159.0.10 | TCP | * | 194 | NEW | ACCEPT |
| F | en2 | 131.159.42.0/24 | 131.159.0.10 | TCP | 71023 | 194 | NEW | ACCEPT |
| G | | | | | | | | |
| H | | | | | | | | |
| I | | | | | | | | |
| J | | | | | | | | |
| K | | | | | | | | |
| L | | | | | | | | |

## Problem 6 Protocol Breaking (7.5 credits)

# GambleHalt

In your free time you're earning some money on the side as a freelance security analyst for *GambleHalt*; a non-profit organization, aiming to reduce the prevalence of lootboxes in online gaming. The organization offers an app, which can be used by gamers to report lootbox statistics.

During the initializiation and connection phase of the App, the message exchange displayed in Figure 6.1 is conducted. Integrity protection (i.e. signature) by party E is denoted as $\{\cdot\}_E$. All parties know each others pubkeys.




① A sends the names of $A$ and $S$ as well as a nonce $N_A$

② S answers with both names, $N_A$ his own nonce $N_S$ and a signature of $N_A, N_S$ and $A$

③ A sends both names, $N_S$, a new nonce $N'_A$ and the signature over $N'_A, N_S$ and $S$

Figure 6.1: GambleHalt App Initialization Protocol

The protocol documentation claims:

*After the protocol is complete, S can be sure it is talking to A and vice versa*

Assume our default **Dolev-Yao attacker model**. Furthermore assume, that the protocol can be **initiated and answered by all parties**.

0 1 **0.0** a)* Which kind of protocol is displayed in Figure 6.1? Name the purpose and briefly justify your answer.

This is an insecure implementation of the Needham Schroeder secret key protocol.

none.

0 1 **1.0** b)* To cut costs, your boss proposes to replace integrity protection by party P in Figure 6.1, denoted as $\{\cdot\}_P$, by encryption. Does this modification reduce security? Briefly justify your argument.

Yes, of course, we should remind our boss that encryption does not provide authenticity nor integrity. ✓

Independent of your choice in Task b), the protocol in Figure 6.1 is flawed.

0 1 **1.0** c)* Outline the flaw of the protocol given in Figure 6.1. Briefly justify why this is a problem.

There is a problem because we can't be sure that message ② really comes from the server. An attacker may relay it. ✓

We should not use another different nonce in ③

3.0 d) **Provide a sequence of message exchanges** which conform to the protocol specification in Figure 6.1, yet violate the documentation claim. Provide the full message exchange, not just your modifications. Name the attacker M and his nonces $N_M$. If you don't want to draw, the following is also valid notation (i.e. third message of Figure 6.1): "`3.) A -> S: A, S, N'_A, N_S, {N'_A, N_S, S}_A`"

0 1 2 3

please see next page.
The main idea is that an attacker can start two connections and trick both parties that they are communicating with each other. We can do this by sending a payload to A and basically ask them to sign something for us.

After demonstrating the vulnerability of the protocol from Figureg 6.1 through a successful exploit, the *GambleHalt* president promises a bonus payment in case you can fix the protocol.

0.0 e) **Provide the sequence of message exchanges** of the corrected protocol.

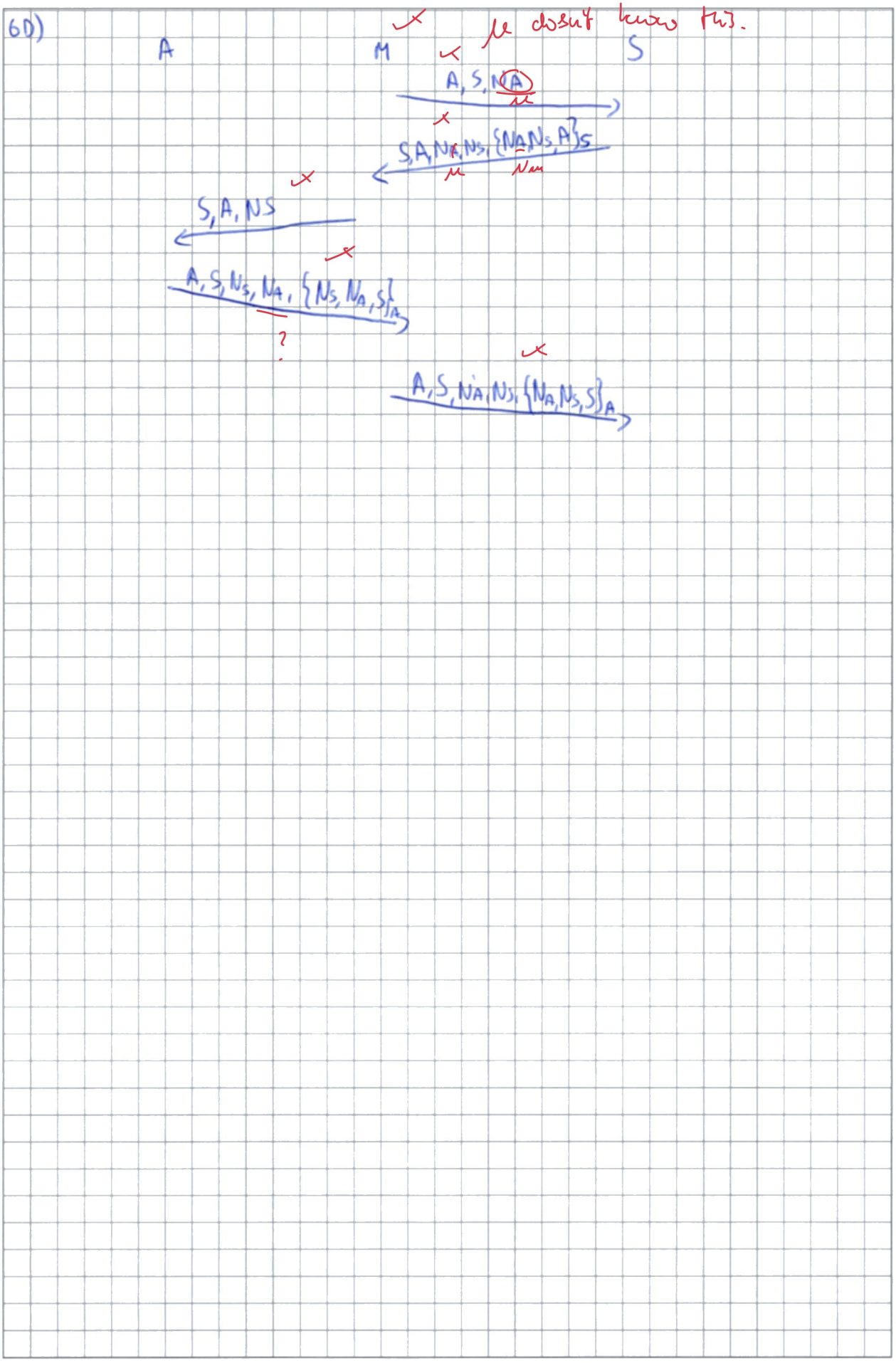
0 1

① $A, S, N_A$
② $S, A, N_A, N_S, \{N_A, N_S, A, S\}_S$
③ $A, S, N'_A, N_S, \{N'_A, N_S, S\}_A$ same problem ..

This might not be sufficient, in addition, think we should not use a different nonce ($N_A'$) because this allows us to send messages that should be answers to something independently.

Additional space for solutions–clearly mark the (sub)problem your answers are related to and strike out invalid solutions.

6D)



Page empty