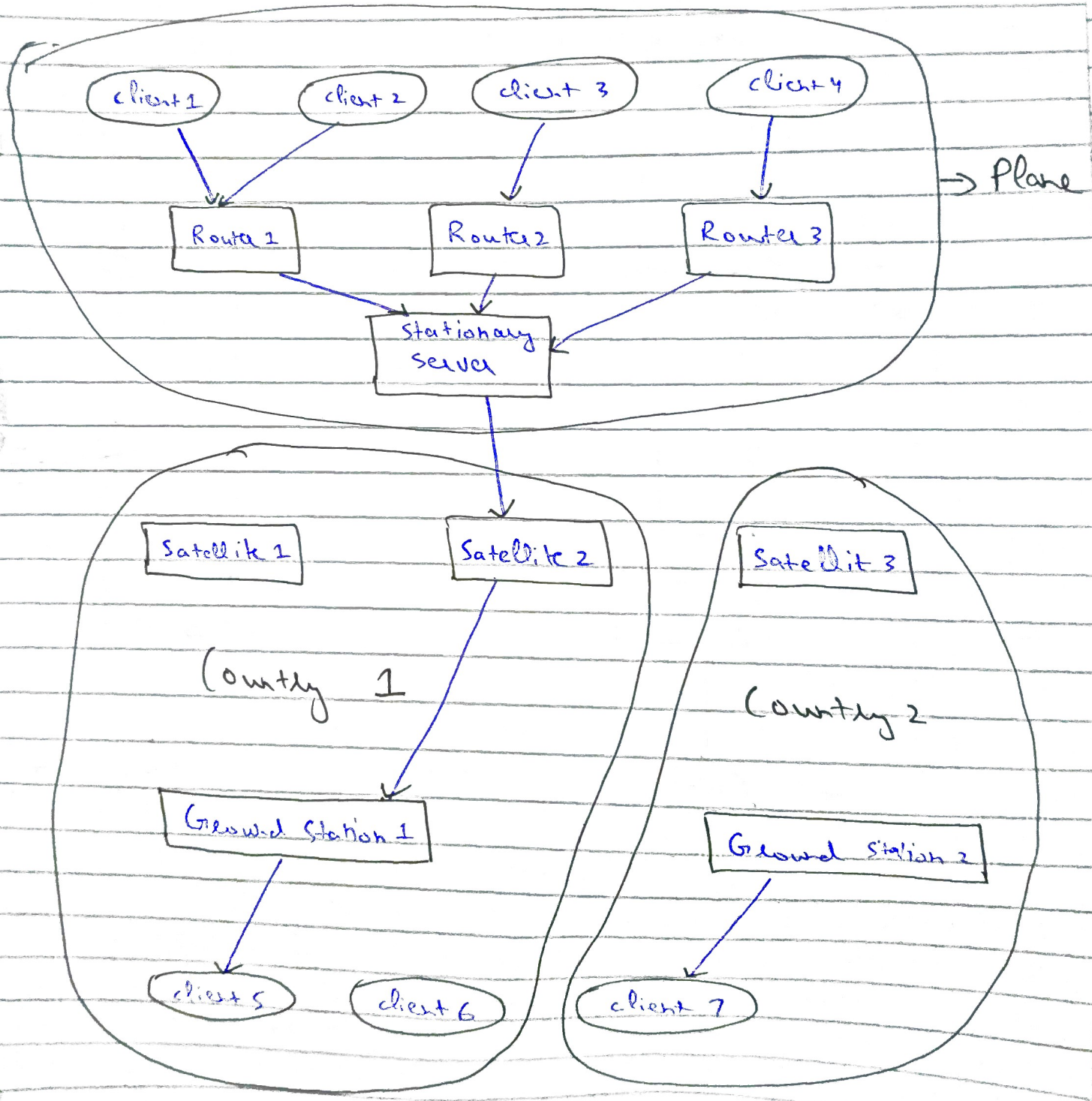
Page 1

16L-4120 Bilal Ashfaq

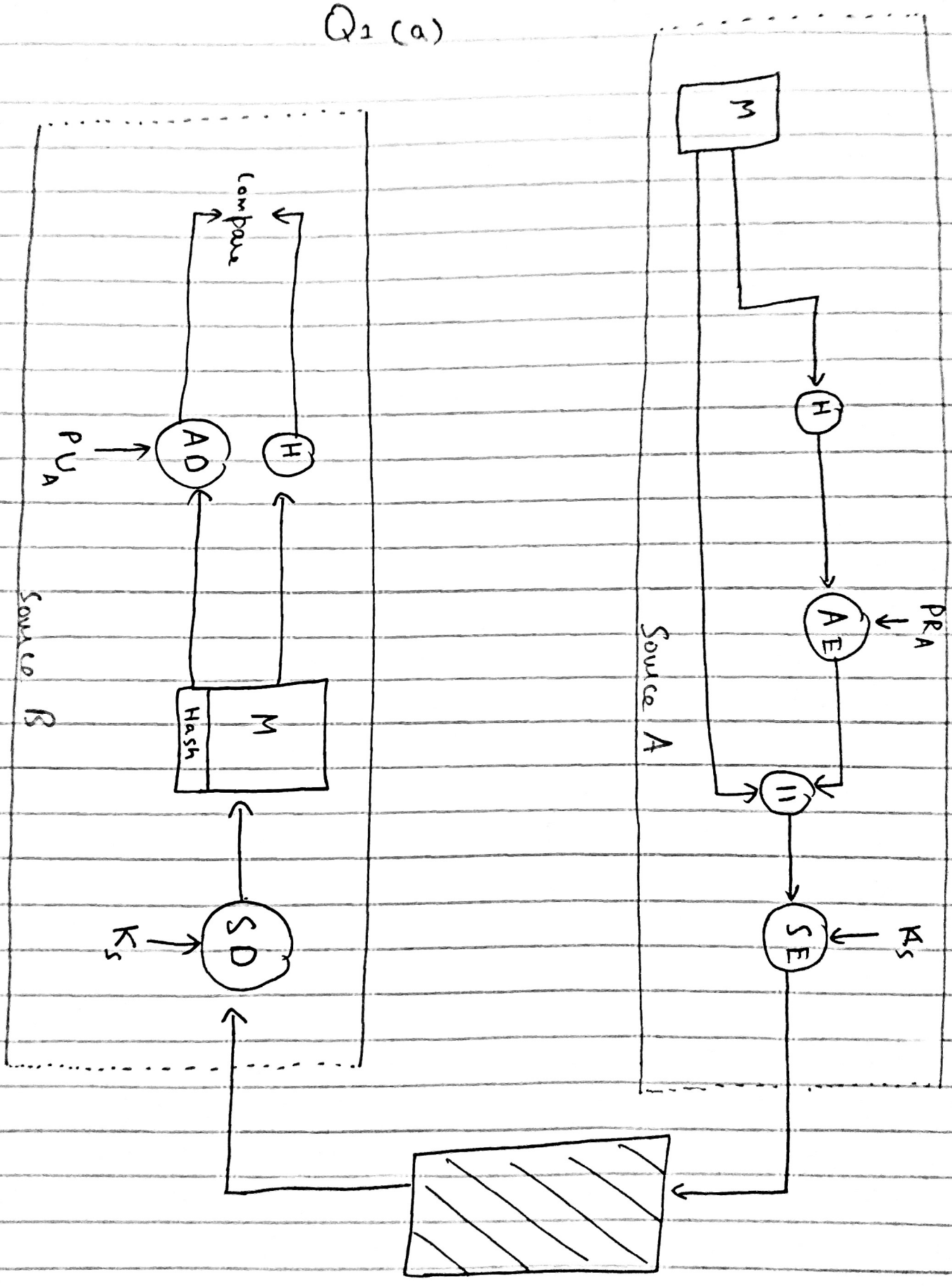
Q 1

Network Topology



16L-4120 Bilal Ashfaq

Q1 (a)



16L-4120 Bilal Ashfaq

Terms Used:

M → Message
H → Hash function
AE → Assymmetric Encryption
|| → Concatenate function.
SE → Symmetric Encryption.
SD → Symmetric Decryption
AD → Assymmetric Decryption
PR_A → Private key of source A
PU_A → Public key of source A

Justification:

Authentication and Integrity:

- Hash of the message is computed, so that reciever can very if the message is altered or authentic. This ensures the Integrity of message.
- Hash is then encrypted using the private key of sender as a signature. So, when reciever decrypts it using public key of sadat, he will be sure about the identity of sender.

161-4120 Bilal Ashfaq

Confidentiality:

- Finally Message and concatenated hash is being encrypted by a Symmetric key which ensures confidentiality because only Sender and Reciever share this key.

Symmetric Encryption:

- Algorithm used for symmetric Encryption is AES (Advanced Encryption standard)

- Reason for using Symmetric key Encryption is that it adds less overhead as compared to Assymetric Encryption. Hence, we are using Asymmetric Encryption only for encrypting hash and whole message is encryped through Symmetric Encryption

- AES is used because it is the most secure algorithem. DES lacks security because of its short key length. (56 bit). On the other hand, AES uses a key of 256 bit length.

Assymmetric Encryption:

- Algorithm used is RSA-3072.
- RSA is used because it the most secure Assymmetric Encryption algorithm.

16L-4120 Bilal Ashfaq

- We didn't use Deffie-Hellmen because it is easily vulnerable to man in the middle attack.

- Similarly, ECC adds extra overhead because it increases ciphertext length. Also, since it is new, there can be unknown weaknesses.

Hash Algorithm:

- Hash algorithm used is SHAKE (Secure Hash algorithm and KECCAK). Because, it lets us generate a hash of any variable length.
- SHA-1 is not used because it was broken and SHA-2 is also very doubtful.

16L-4120 Bilal Ashfaq

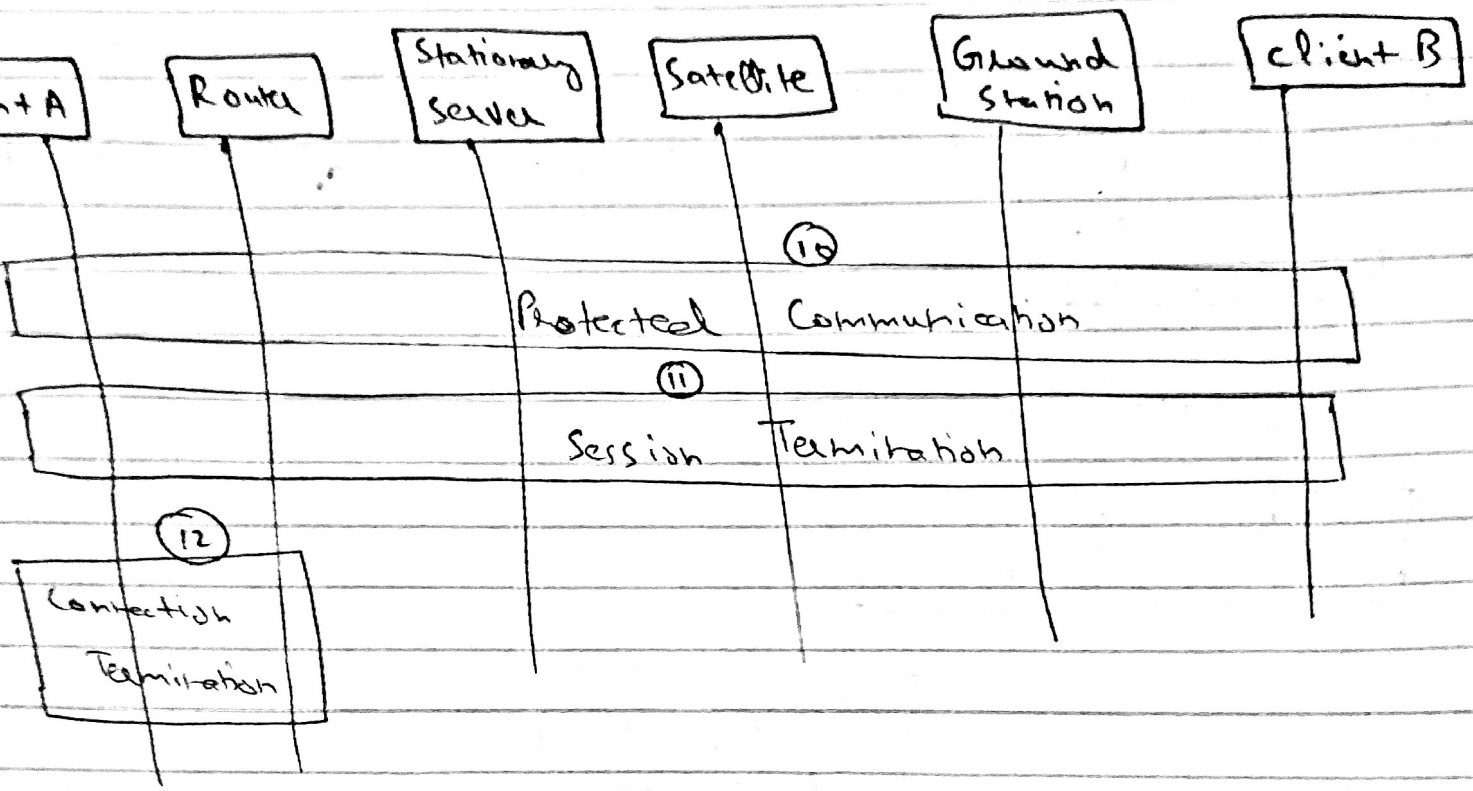
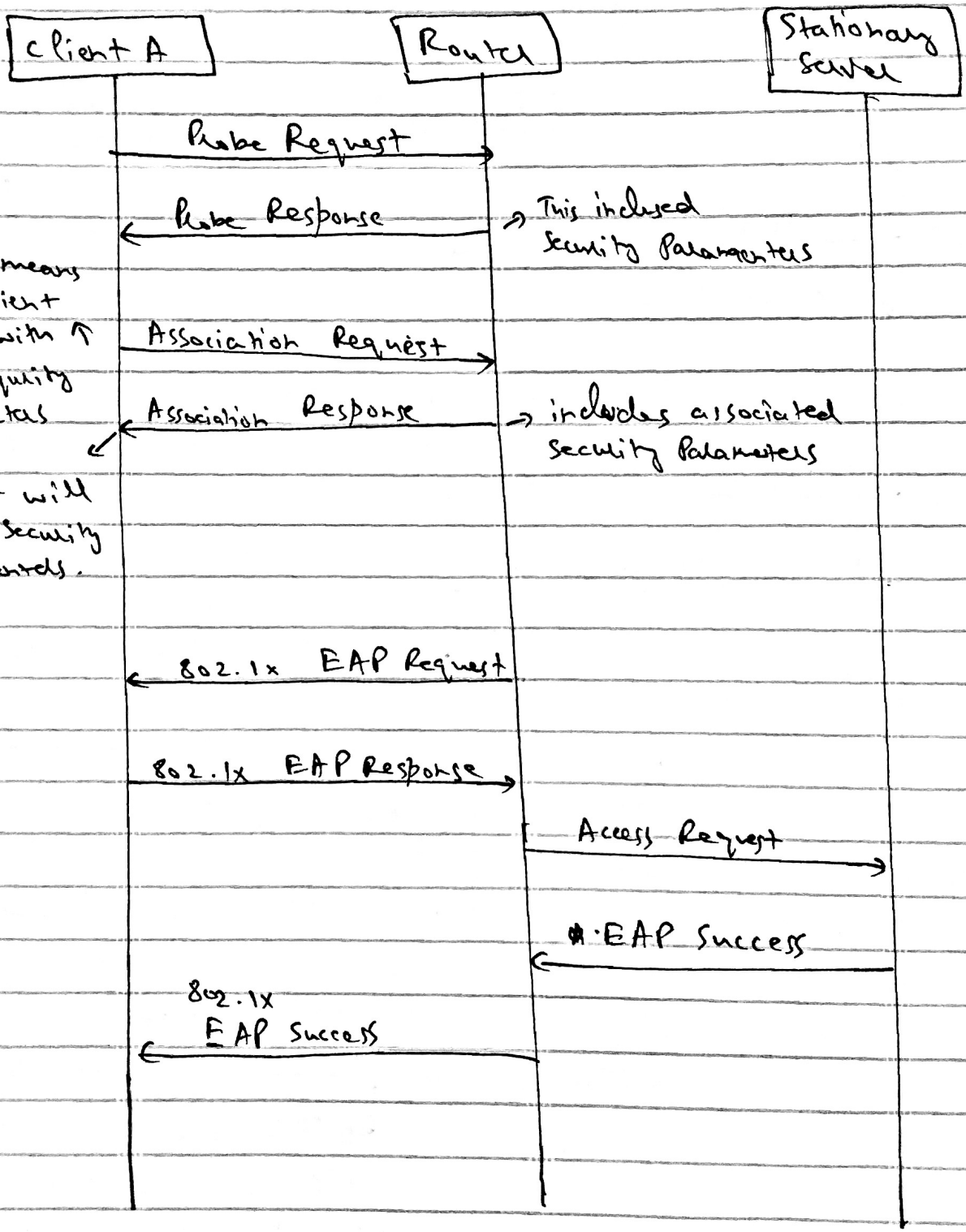


Diagram (A)

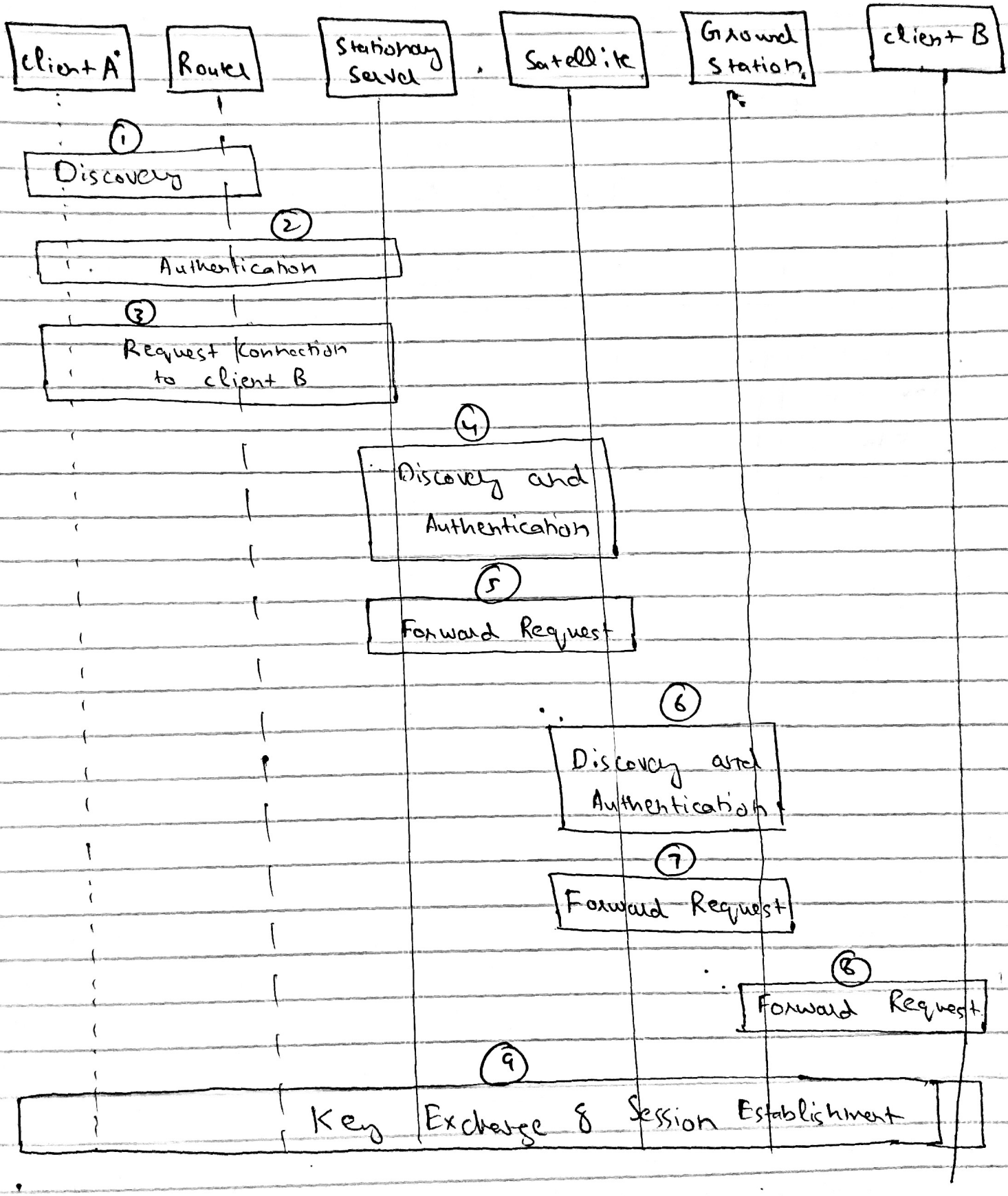
18L-4120 Bilal Ashfaq

Detailed Connection with the router:



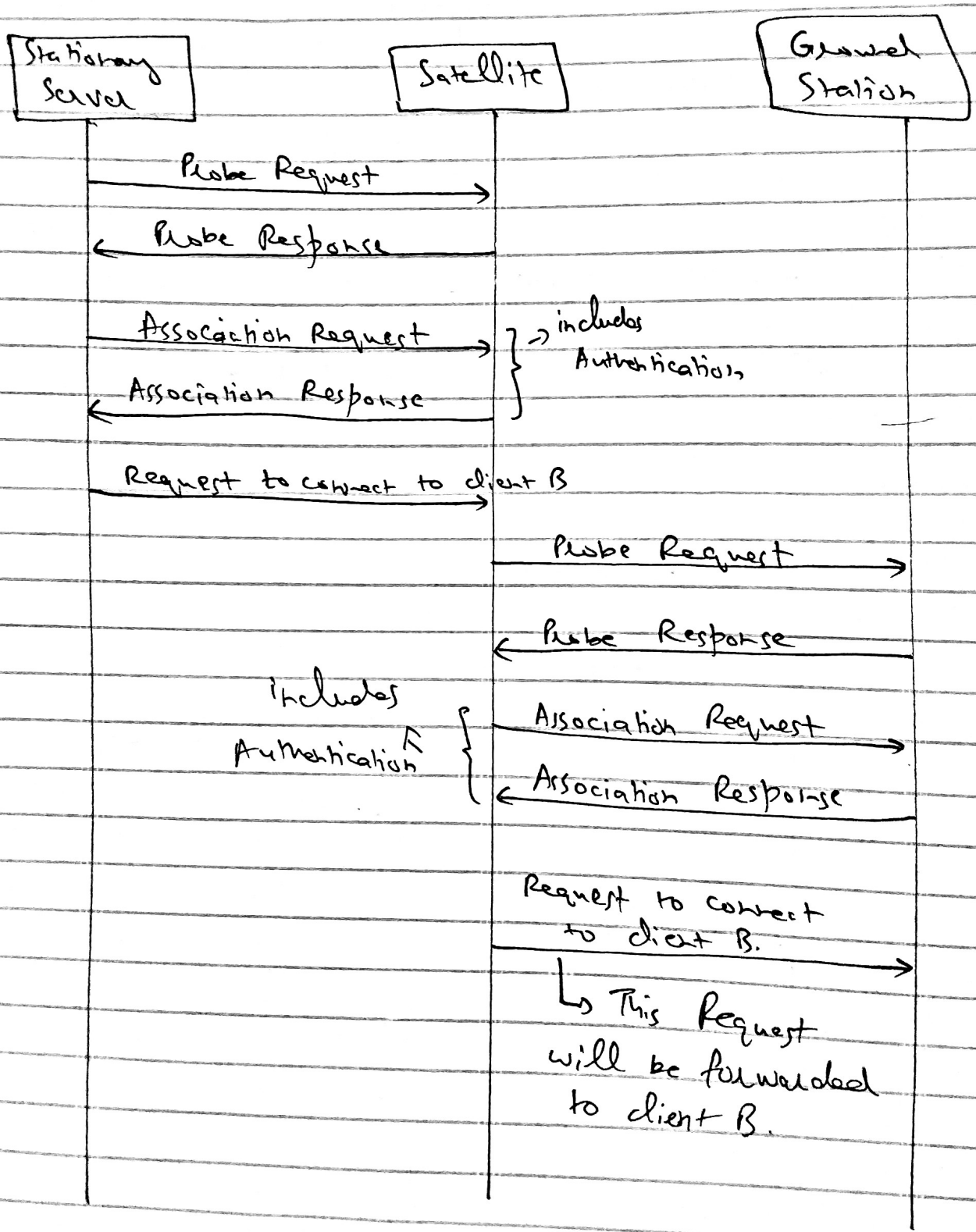
16L-4120 Bilal Ashfaq

Q1 (b)



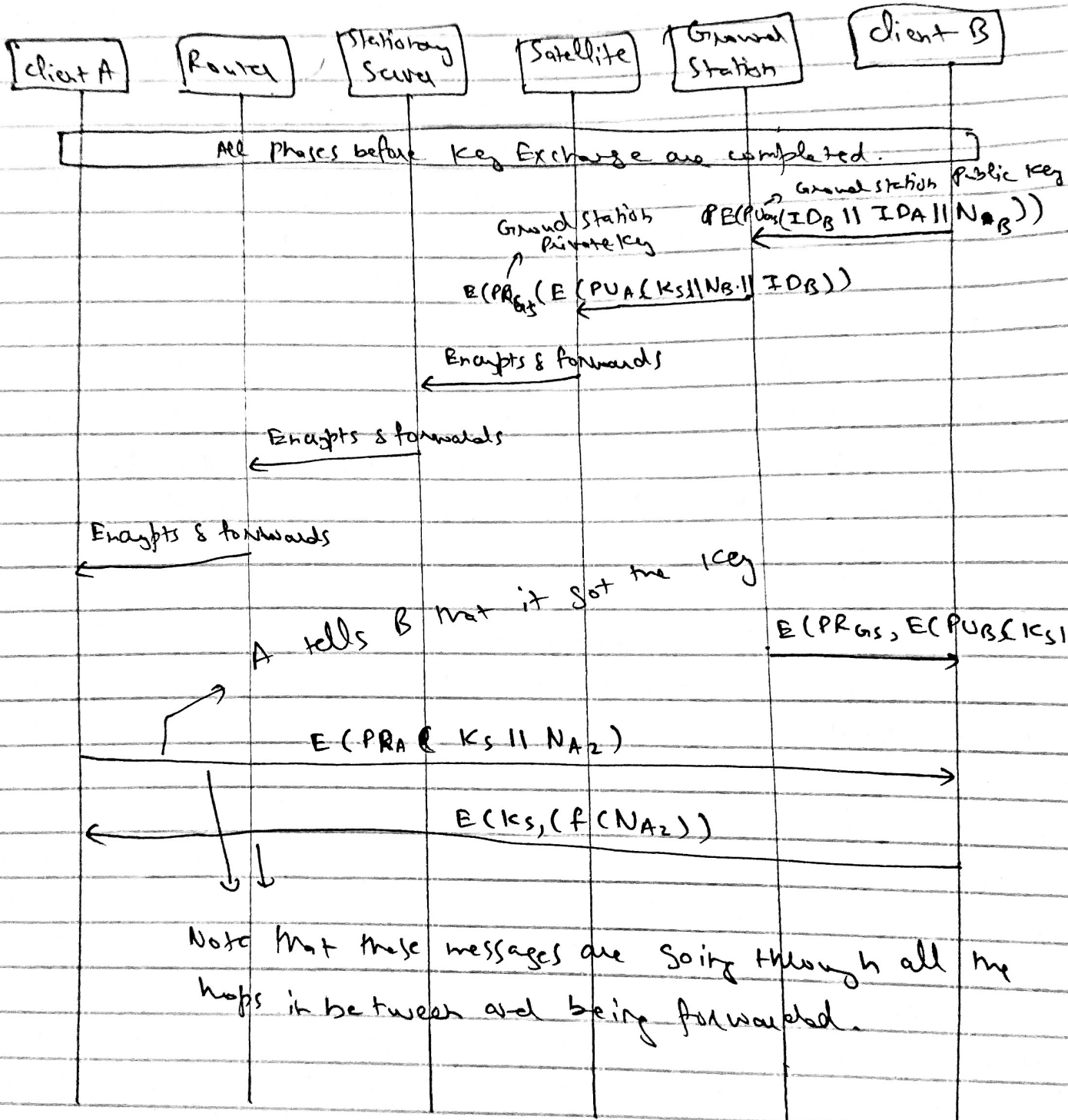
EL-4120 Bilal Ashfaq

Detailed connection of Stationary saval to satellite and satellite to Ground Station.



16L-4120 Bilal Ashfaq

Key Exchange & Session Establishment:



Note that these messages are going through all the hops in between and being forwarded.

Page 9

Page 11

16L-4120 Bilal Ashfaq

Why Nonce:

- Nonces are a very easy way to check the freshness of message. We have not used Timestamps because they require the synchronization of clocks.
- Similary, sequence numbers add extra overhead because they have to be remembered.

Why Probe:

- In the question statement it is not specified that all the hops are aware of Network topology. Hence, we have to perform discovery at each phase.
- Each hop in order to get information about next hop (to which it has to connect), will have to perform Probe Request.

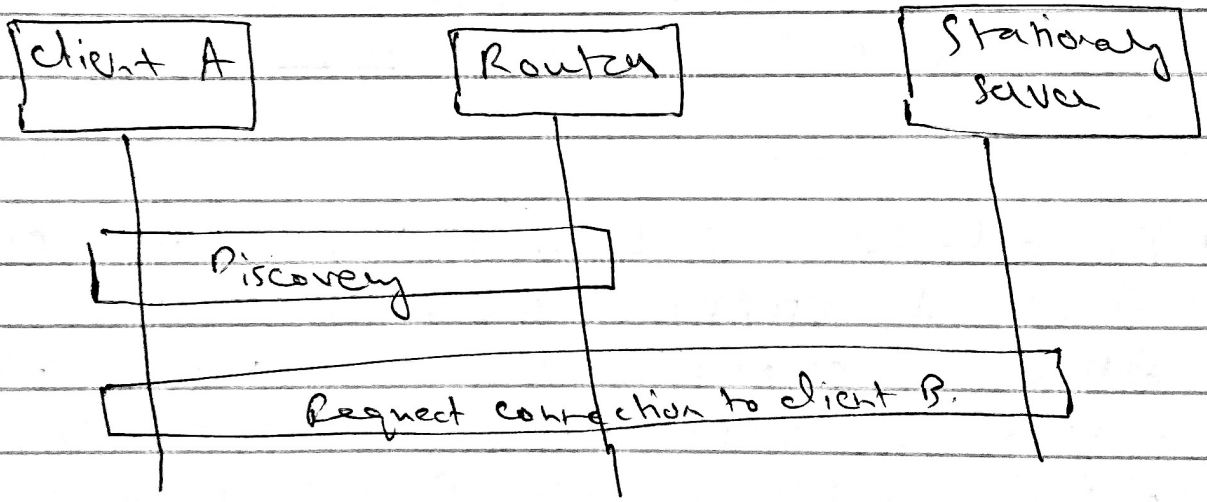
16L-4120 Bilal Ashfaq

Q2 (c)

Changes in Diagram (A)

Step 2 / Authentication b/w client A & Stationary server will be removed:

Since we can assume that all the passengers on the plane are authentic individuals, ~~we~~ and also, routers and directly attached to the Stationary Server, we can remove this step and let client make request after discovery



Step 11/ Session Termination will be removed:

If we were doing proper Session Termination in the previous approach. Because it is just a good way / practice and not neccassary, we can remove it and directly proceed with the

Page 13

16L-4120 Bilal Ashfaq

Connection termination.

Changes in Encryption Algorithms & Hash Algorithms

- Since AES is computationally very expensive we will now use modified DES. This DES will use 128 bit key rather than 56 bit key. This will ensure security and we will also get good performance. (fast performance because of less computational overhead)

- For calculating Hash we will now use SHA-3 instead of SHAKE. The reason is that although SHAKE is very secure but it is very complex too. Which basically means that it requires a lot of computations. So, to avoid overhead but still use a good hash function, we can use SHA-3. It is secure and also less expensive (computationally), it compared with SHAKE.

16L-4120 Bilal Ashfaq

Q2 (a)

Phase 1) Requirements Engineering:

① R1) Specify functional requirements using use cases:

Specification of functional requirements is most important because otherwise there will be inconsistencies during design and implementation phases.

② R2) Specify abuse cases and attack trees

This will help is determining the types of threats that can be faced by our System.

③ R3) Perform Risk analysis:

This ~~explores~~ tells that how secure is our model/ requirements right now and how much more use cases should be added to reach reasonable security

④ R4) Add Security requirements to mitigate threats:

This will finalize [redacted] all the security requirements associated with them. Functional requirements from R1

16L-4120 Bilal Ashfaq

(5) R5) Select security mechanisms for security requirements:

We need this activity because there are tons of security mechanisms available and we can not use them all.

(6) R6) Add security requirement "usecases" for selected mechanisms (in R5)

This will fitalize our usecases with the addition of security related usecases.

Phase 2) Design phase:

(7) D1) Construct detailed design:
This will help us understand our system better and will ultimately help in Implementation phase.

(8) D2) Perform design analysis
This will help in identifying any vulnerabilities related to our design.

(9) D3) Remove security vulnerabilities in the design:
This will make our design more secure and reluctant to vulnerabilities

(10) Assess if another round of design analysis is required:

This activity makes sure that while change design in previous activities, we have not introduced new vulnerabilities. We can identify them in this activity.

(11) D5) Design specification based embedded security monitor:

This will help our system perform intrusion detection on its own.

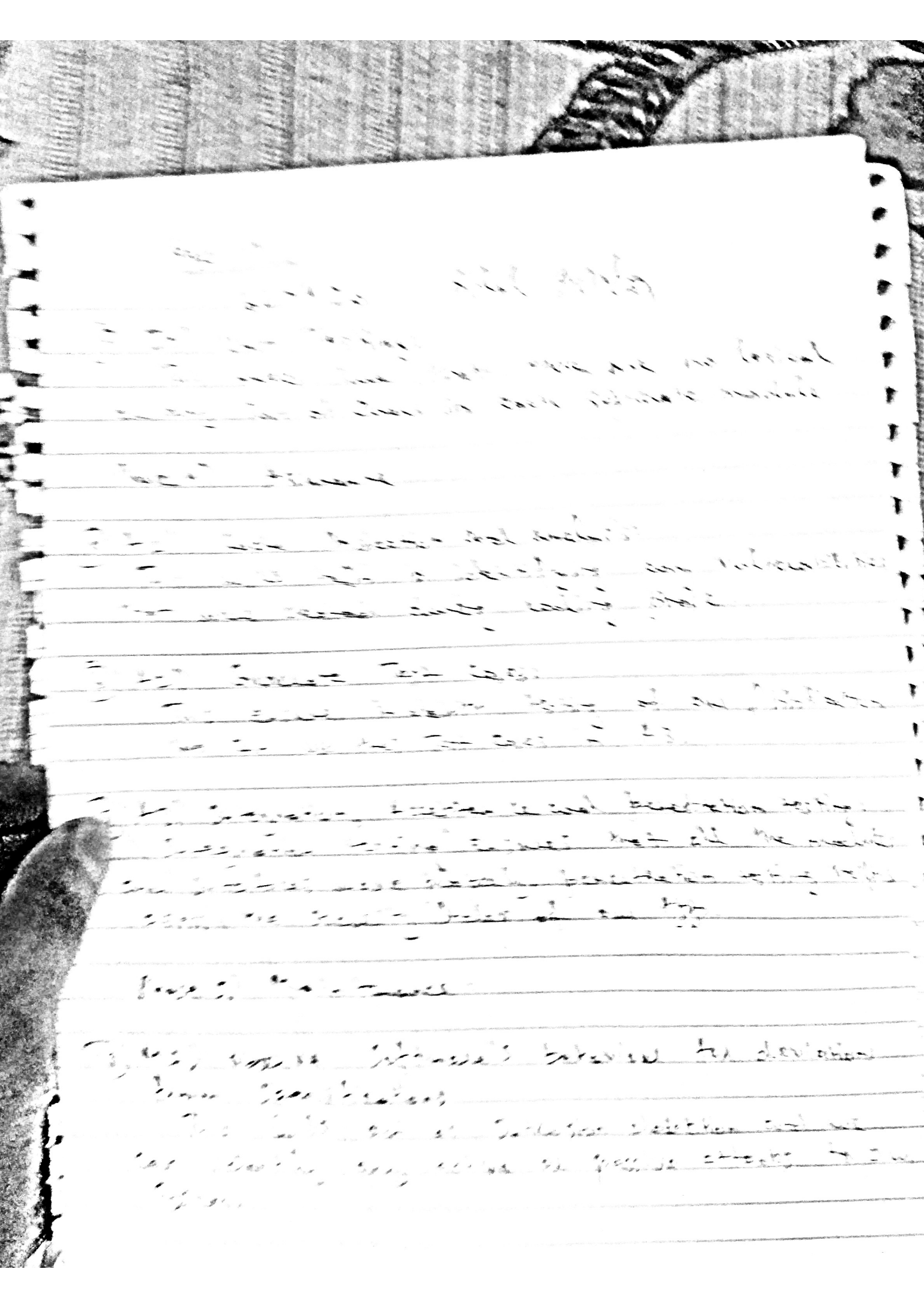
Phase 3) Implementation phase:

(12) I1) choose a secure programming language:

Many vulnerabilites arrive because of using low level languages. If we use a secure high level language, we can avoid all those attacks.

(13) I2) Follow secure coding standards & guidlines:

This will help us avoid any mistakes that were made by others. We can use consult vulnerability databases for this.



16L-4120

(14) I3) Unit Testing:
This makes sure that there are no logical or any sort of errors in each separate module

Phase 4) Assurance

(15) A1) Code Inspection and analysis:
This will help in identifying any vulnerabilities that were created during coding phase.

(16) A2) Generate Test cases:
This ensures in depth testing of our Application we can use these Test cases in A3.

(17) A3) Integration, Acceptance and penetration testing:
Integration testing ensures that all the modules and Interfaces work properly. penetration testing helps assess the security Index of our App.

Phase 5) Maintenance:

(18) M1) observe Software's behaviour for deviations from specifications
This will act as Intrusion detection and we can identify any active or passive attacks to our System.

(19) M2) Locate Underlying Vulnerabilities for identified deviations:
This activity will help us pin point the measures needed to secure our system in case of deviations.

(20) M3) Perform Rework to Remove Vulnerabilities:
Finally, this will make our System even more Secure and we can get rid of the attacks.

Q2 (b)

Removed activities are as follows:

(10) / D4): Assess if another round of design analysis is required:
This activity just acts as an additional check and adds extra work overhead. Hence it is removed.

(11) / D5): Design Specification based embeded Security Monitor:
This activity requires a lot of time and cost to be accomplished. Since, we are performing Intrusion detection in Maintenance phase. This can be removed.

Page 19

16L-4120 Bilal Ashfaq

~~Ans~~

Acceptance testing is removed from A3 / (17)

Acceptance testing will add extra overhead because all other testings ensure that system runs properly.

All other Activities are necessary and they can't be removed.