1.4. (updated) Any message that A transmits initially can potentially be compromised and known to the intruder. However, to communicate with s, A could encrypt an already encrypted message (albeit with a weak password) with the public key of s. Furthermore, the message can include a nonce generated by A and a request ID for this specific session. Even if the intruder gets a hold of this message and impersonates A later on, upon reception s can determine that the request ID is indeed from a previous session and the communicating entity is an intruder and not A. Moreover, another possible way to prevent the attack would require A to generate another symmetric key and encrypt the message (that is already encrypted with a weak password) with this key. Furthermore, the key and the encrypted message are encrypted again in a message with the public key of the server s.

1.5. The very first step in this protocol requires that both communicating entities support TLS. Therefore, when A tries to communicate with B, the first exchange of messages between these two entities would comprise the former requesting a TLS session to be established. If the latter does not provide a relevant response, which requires B to send its valid digital certificate, A can suspect an impersonator intercepting the communication and terminate the connection. This initial step is called the "hand-shake" and it is during this phase that a unique symmetric session key is generated by A. This key is encrypted using the public key of B, which is obtained from the certificate from the previous step, and sent to B. Thus, all the data during the entire session is encrypted using this key. Furthermore, similar steps will be followed during the communication that requires A to authenticate itself to B using the third party server s. However, if an intruder obtains a valid digital certificate that A might trust, a TLS session will still be established and communication compromised.