Secrecy, internal and external threats and related topics

An ***error state*** is a state of the form $(out\ net\ s \mid secret(s))$; that is, a state in which the assertion is made that s remains secret, while s is also sent over the network. A process P is called ***safe for secrecy*** if it cannot reach an error state, or, more formally, if we have that:

whenever $P \rightarrow^* Q$, this implies that Q is not an error state.

Now, this definition does not make a lot of sense. Consider a process in which the secret s is sent over the network in encrypted form, along with the decryption key, i.e. something like the following:

$$A \rightarrow B : (\{s\}_k, k)$$

According to the definition of safety for secrecy, such a process would be safe for secrecy, since the secret is never explicitly put onto the network. However, in practice, it is trivial for an attacker to find the secret from such a message. To capture this intuition more appropriately, we introduce the notion of ***robust safety for secrecy***. This notion works by introducing an opponent process O, which can take any form desired by an attacker[1]. If, under any possible opponent process, we have that $P \mid O$ is safe for secrecy, then we say that P is robustly safe for secrecy. In effect, this opponent process can always publish the secret s if it is capable of obtaining it, although the opponent process may also participate more actively in the protocol (e.g. like a man in the middle).

Next, we distinguish between external threats and internal threats. The main difference is that, in an ***external threat model***, the attacker cannot control any of the participants. In effect, this means that the attacker does not possess any of the keys which a valid protocol participant would have. However, in an ***internal threat model***, the attacker will have access to the keys of a participant. Usually, we call this compromised participant the Spy. Such a participant is usually modelled as follows:

$$\texttt{Compromised(k}_{spy}\texttt{) == out net k}_{spy}\texttt{;}$$

That is, the compromised participant publishes their keys; this effectively implies the attacker has access to those keys and can use them to communicate on behalf of the Spy.

In a ***type flaw attack***, the attacker manages to have the victim confuse one type of message with another. For example, consider a protocol similar to the following:

$$A \rightarrow B : (\{s, p\}_k, \{p', s'\}_k)$$
$$B \rightarrow A : (p, p')$$

In this protocol, if the attacker swaps the order of the elements of the first tuple, the victim will publish the secrets s and s' instead of the public values p and p'. There are several methods to prevent type flaw attacks:

1. Ensuring the message format is always the same. This sounds plausible at first, given that it is an easy workaround for the implausible example given above, but it must be implemented consistently across all protocols. This is impractical in reality, since it makes the security of one protocol depend on the other protocols used in the system.
2. Another way to prevent type flaw attacks is to tag messages with their types. This works more consistently, but has the disadvantage of requiring type checks whenever messages are received. In addition, it may increase the size of messages.

Finally, ***conditional secrecy*** is a concept that provides for a more useful notion of robust safety for secrecy in the context of an internal threat model. In conditional secrecy, the secrecy assertion made in the definition of an error state is changed slightly to the form $(out\ net\ s \mid if\ b \neq Spy\ then\ secret(s))$. This effectively means that the secrecy of the value

[1] With the exception that the opponent process may not contain assertions.

only has to hold when the value comes from the Spy. If we did not require this, we would find trivial attacks; note that the attacker can always simply publish secrets generated by the Spy, which leads to many protocols being insecure. By using the conditional secrecy assertion, we can limit ourselves to considering that secrets from honest parties need to remain secret.