A Replay Attack in the TCG Specification and a Solution

Danilo Bruschi Lorenzo Cavallaro Andrea Lanzi
Mattia Monga

Università degli Studi di Milano
Dipartimento di Informatica e Comunicazione
{bruschi, sullivan, andrew, monga}@security.dico.unimi.it

Annual Computer Security Applications Conference 2005

Table of Contents

Trusted Computing Platforms

What are they?

According to the Trusted Computing Group (TCG) Specification, a Trusted Computing Platform (TP) is

- a Computing Platforms with built-in *trusted* hardware components endorsed by trusted third parties

These components, called *Roots of Trust*, provide secure services such as

- secure boot
- software integrity checking
- digital signatures
- ...

TCG-based Trusted Computing Platforms

Roots of Trust Components

A TP is composed by two main *trusted* hardware components

Core Root of Trust for Measurement (CRTM)

It starts the initial integrity check of every hardware and software components

Trusted Platform Module (TPM)

It provides cryptographic and protected storage facilities

TCG-based Trusted Computing Platforms

Main Functionalities

- **Identity**: any TP has an identity that cannot be forged
- **Measurement**: a TP can compute a *complete* integrity check of its software and hardware components
- **Protected Storage**: a TP can provide protection to *sensitive* data (i.e., passwords, cryptographic keys, passphrases, ...)

Authorization Protocols
General Concepts

Every time Alice wants to use a TPM-protected resource, she needs to use an *Authorization Protocol*. Thus, she *must*

- know the secret bound to the resource
- provide a proof of this knowledge to the TPM, during an existing authorization session

$\Rightarrow$ Authorization Protocols manage *authorization sessions* and verify subject's clearances for this purpose

Authorization Protocols
Existing Authorization Protocols

The TCG Specification defines two main Authorization Protocols

Object-Independent Authorization Protocol (OIAP)

A command can potentially be issued several times, in a single authorization session, acting on different protected resources

Object-Specific Authorization Protocol (OSAP)

Different commands can potentially be issued several times, in a single authorization session, acting on the same protected resource

Authorization Protocols

Protocol Threats and Countermeasures

According to the TCG Specification, Authorization Protocols have been designed in order to prevent the following threats

Replay Attack

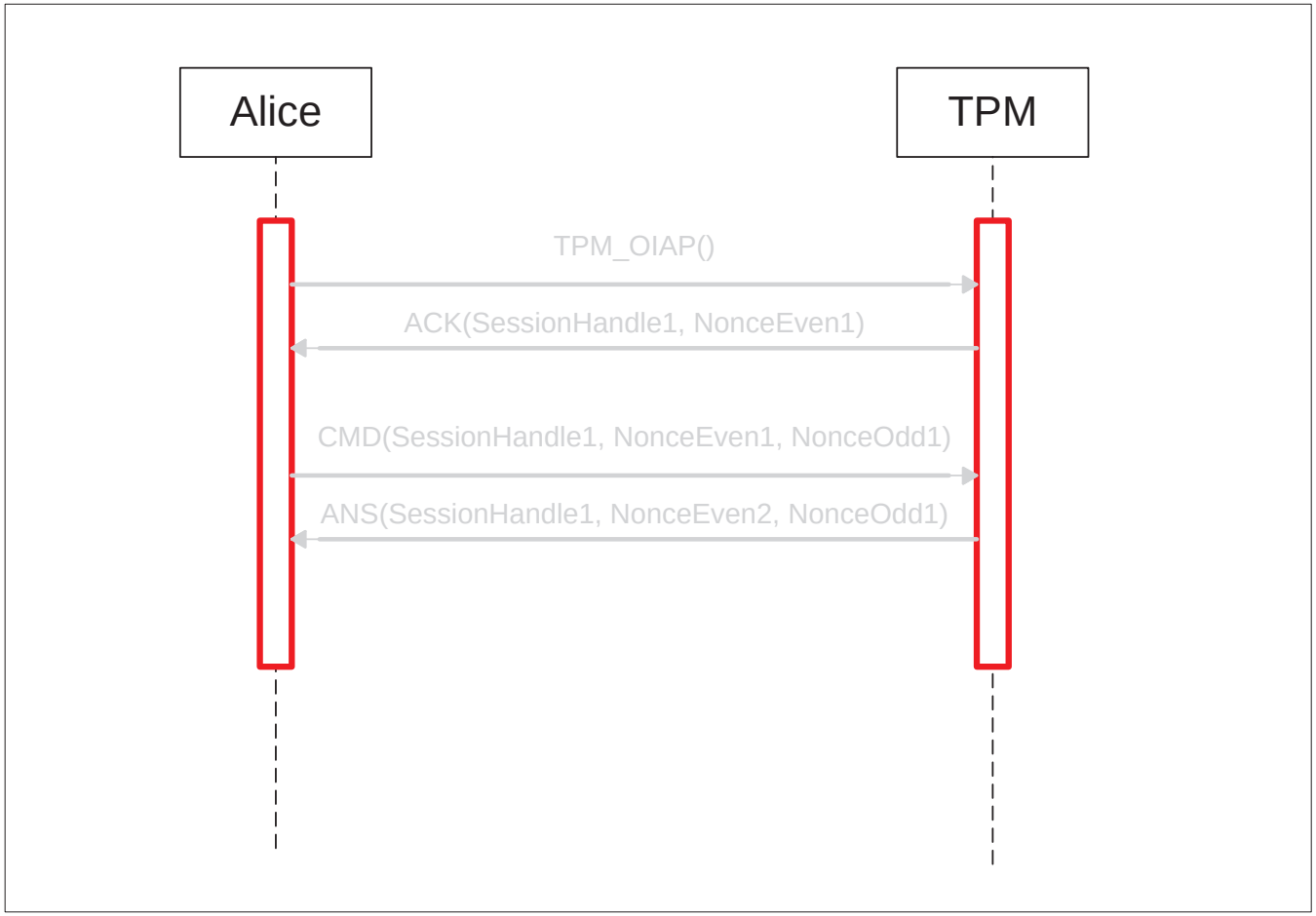
$\Rightarrow$ use of pseudo-random numbers, *nonces*, to provide a *freshness property*

Packet Mangling Attack

$\Rightarrow$ use of HMAC to provide authentication and integrity

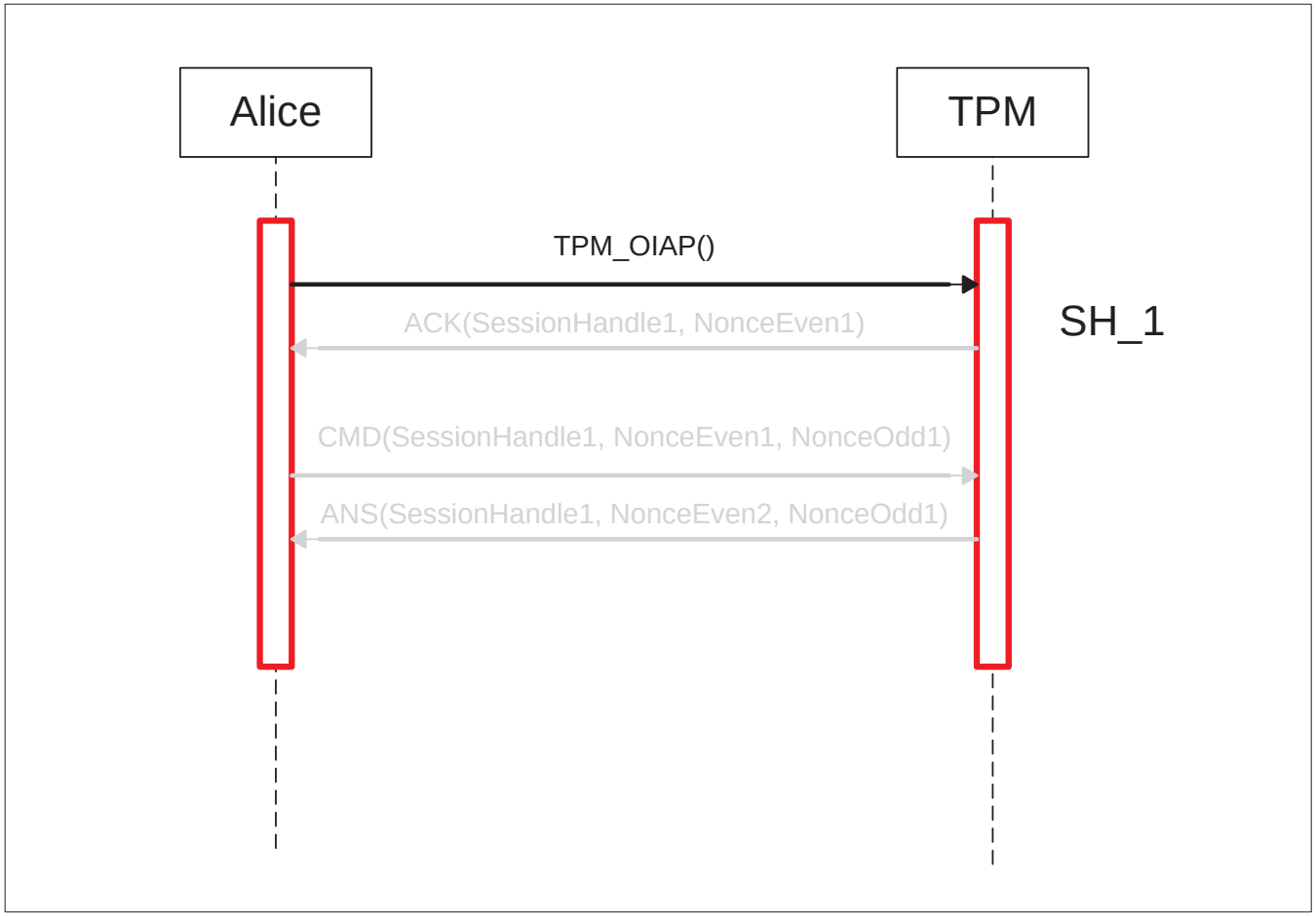
Object-Independent Authorization Protocol

A Simple Protocol Sketch



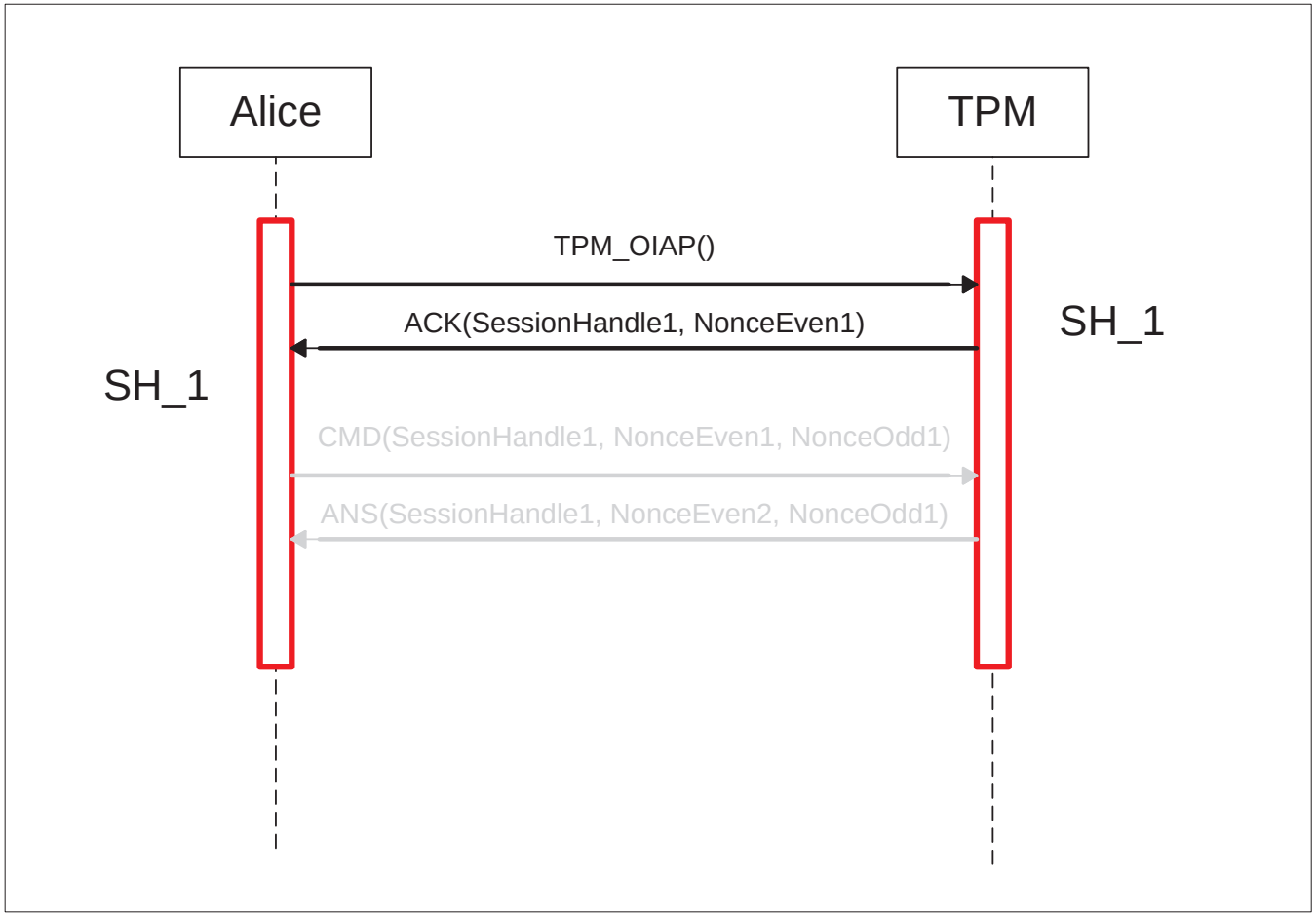
Object-Independent Authorization Protocol

A Simple Protocol Sketch



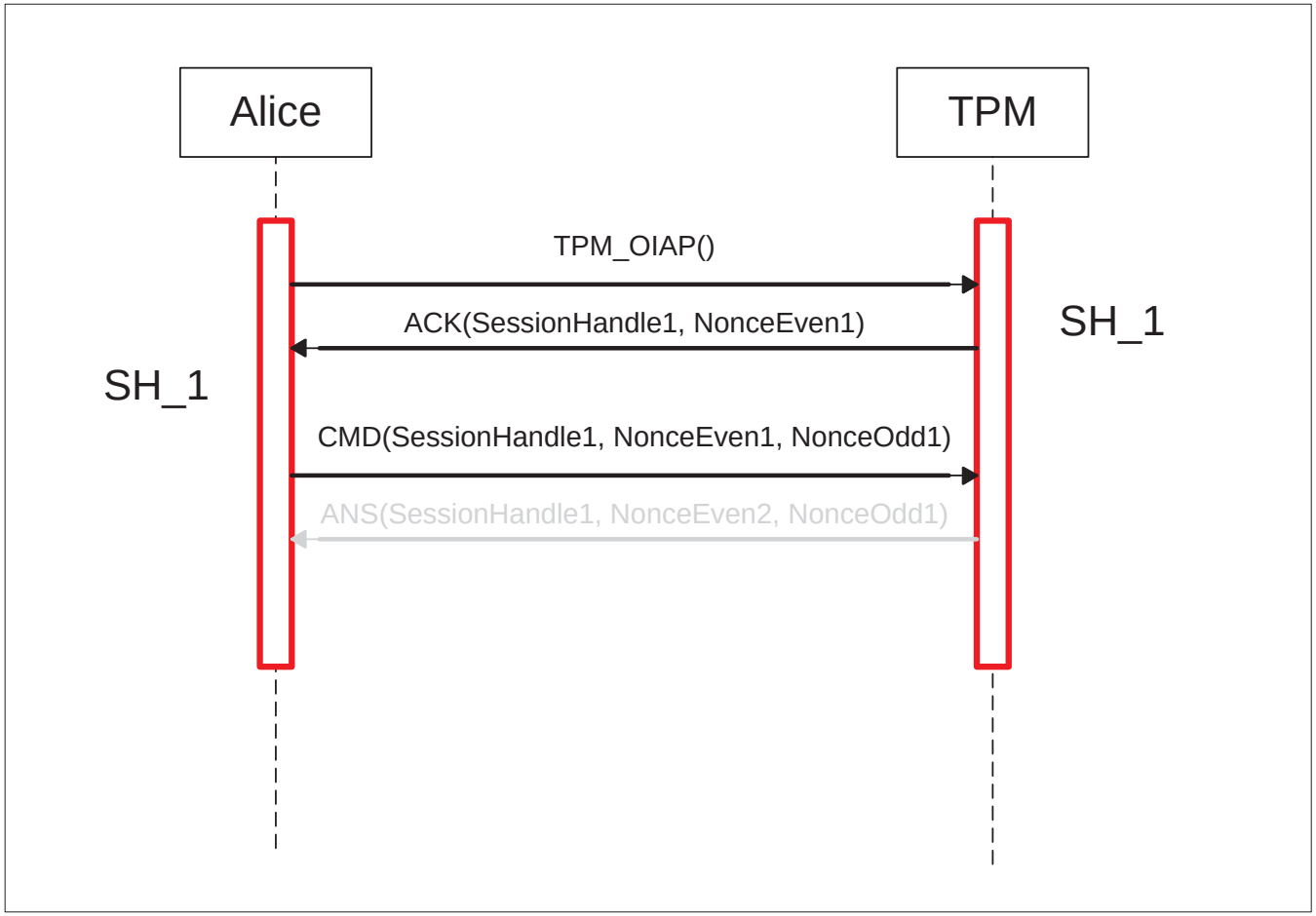
Object-Independent Authorization Protocol

A Simple Protocol Sketch



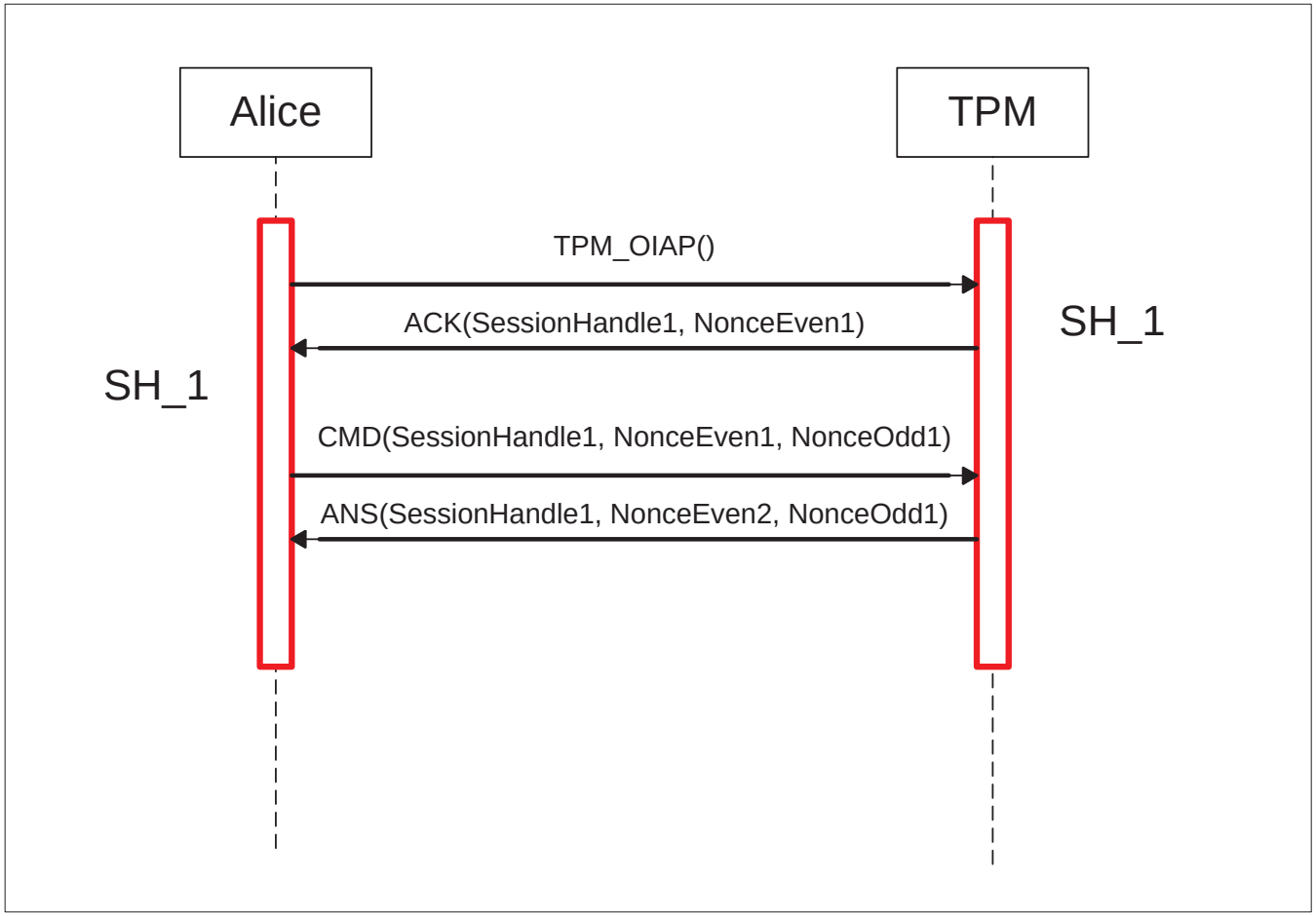
Object-Independent Authorization Protocol

A Simple Protocol Sketch



Object-Independent Authorization Protocol

A Simple Protocol Sketch

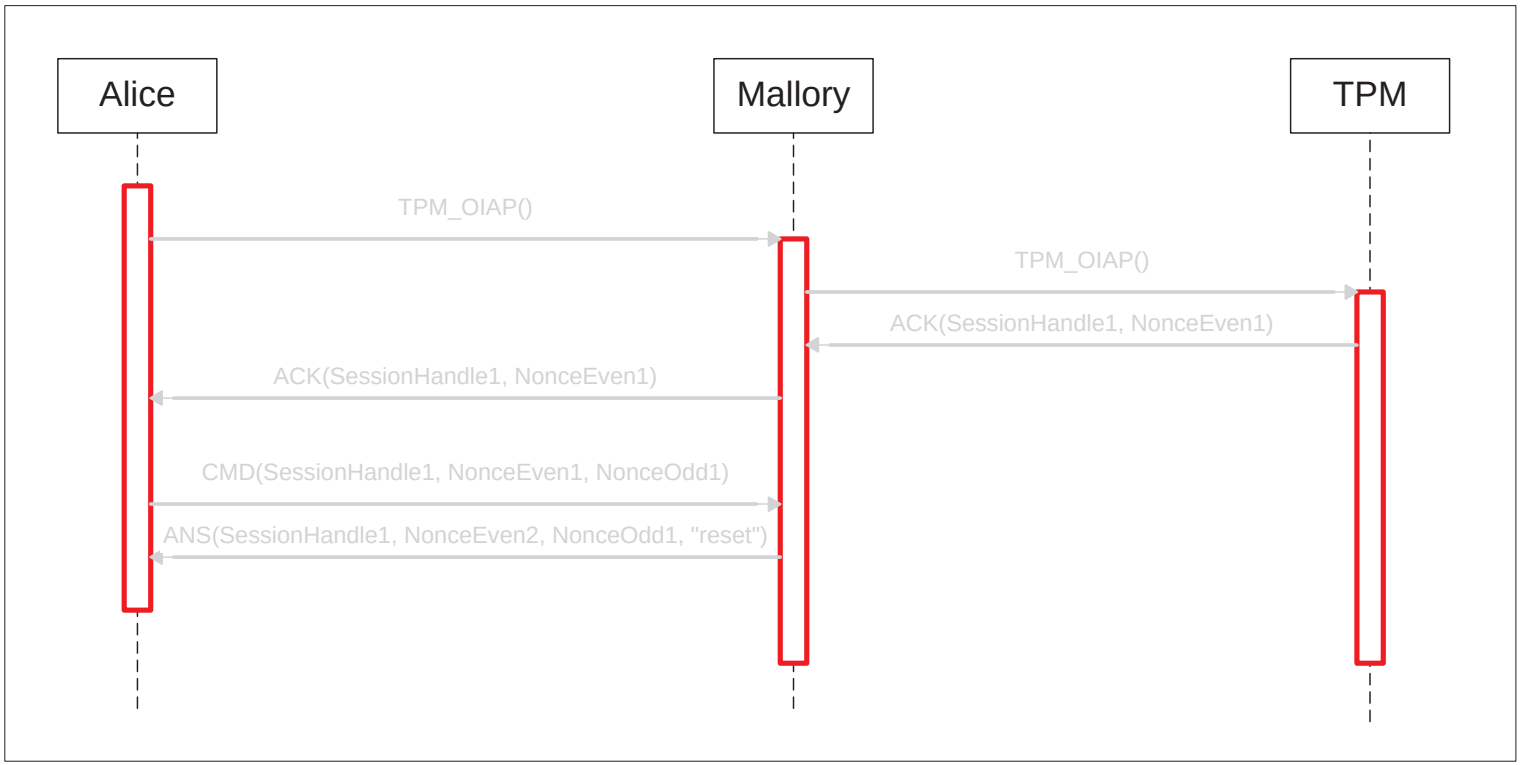


Replay Attack
OIAP Feature Leveraged by the Attack

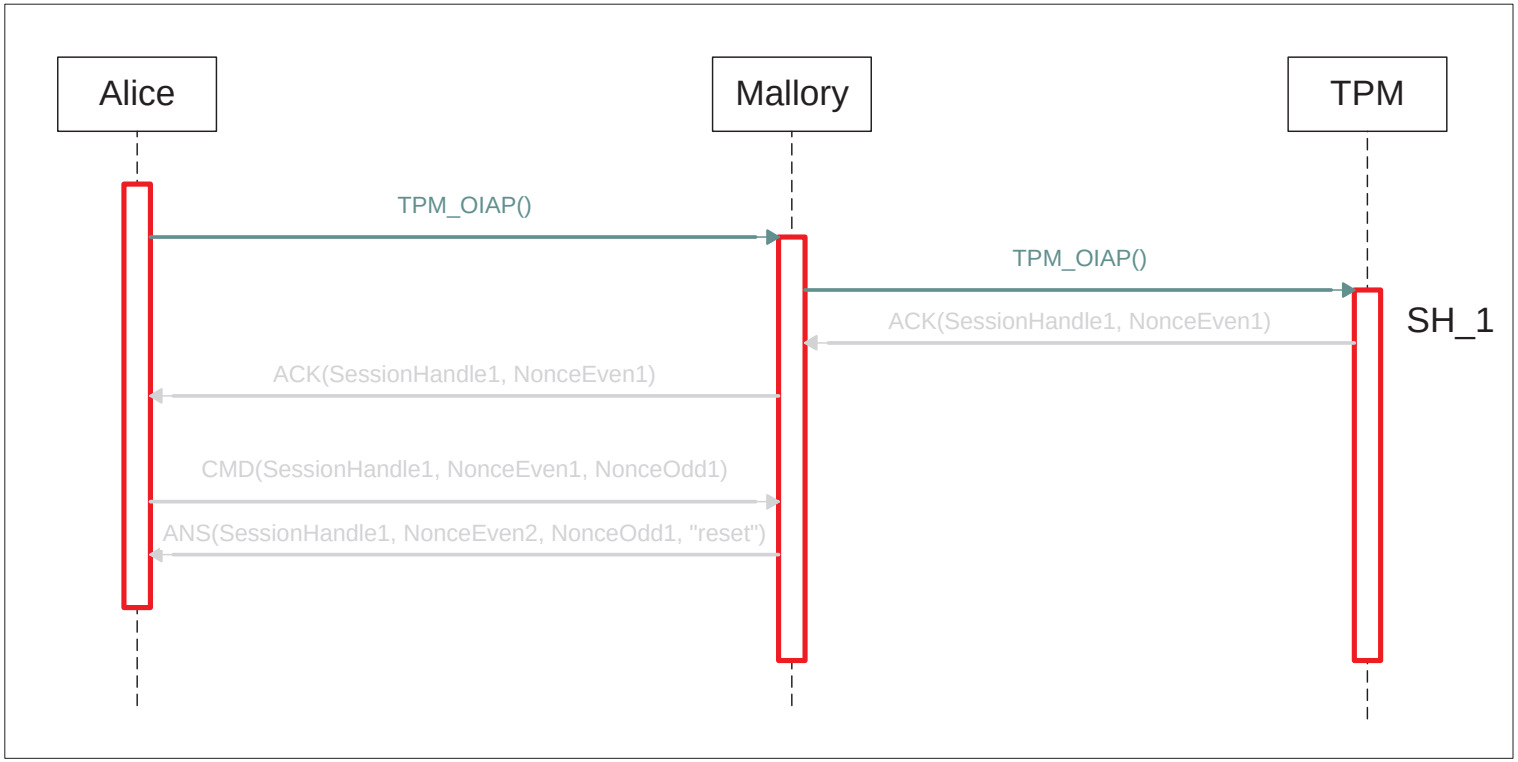
According to the TCG Specification, an authorization session is *kept open* indefinitely by a TPM, unless

- an erroneous message is received on an existing authorization session, i.e., wrong command arguments or invalid HMAC.

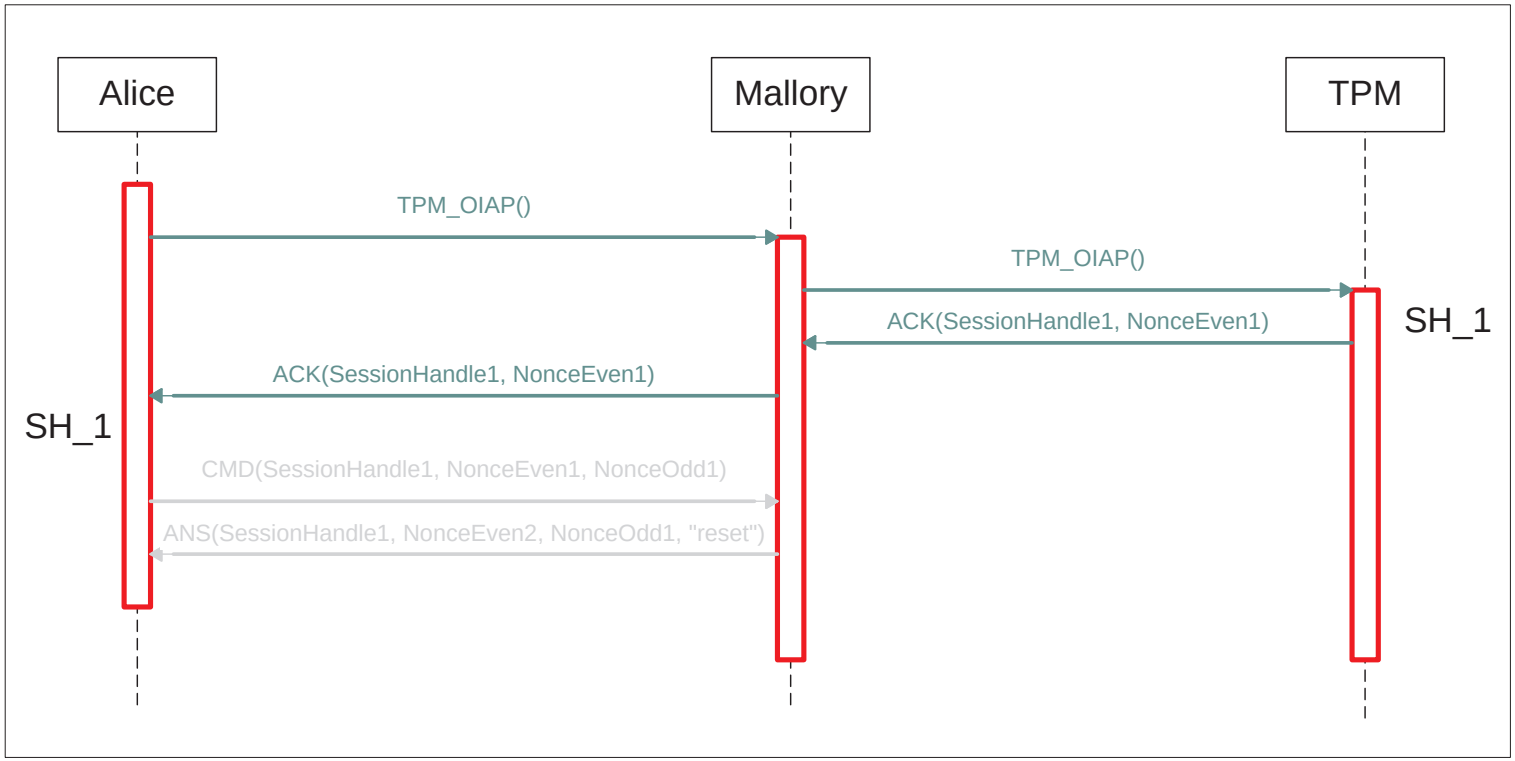
Message Storing Phase



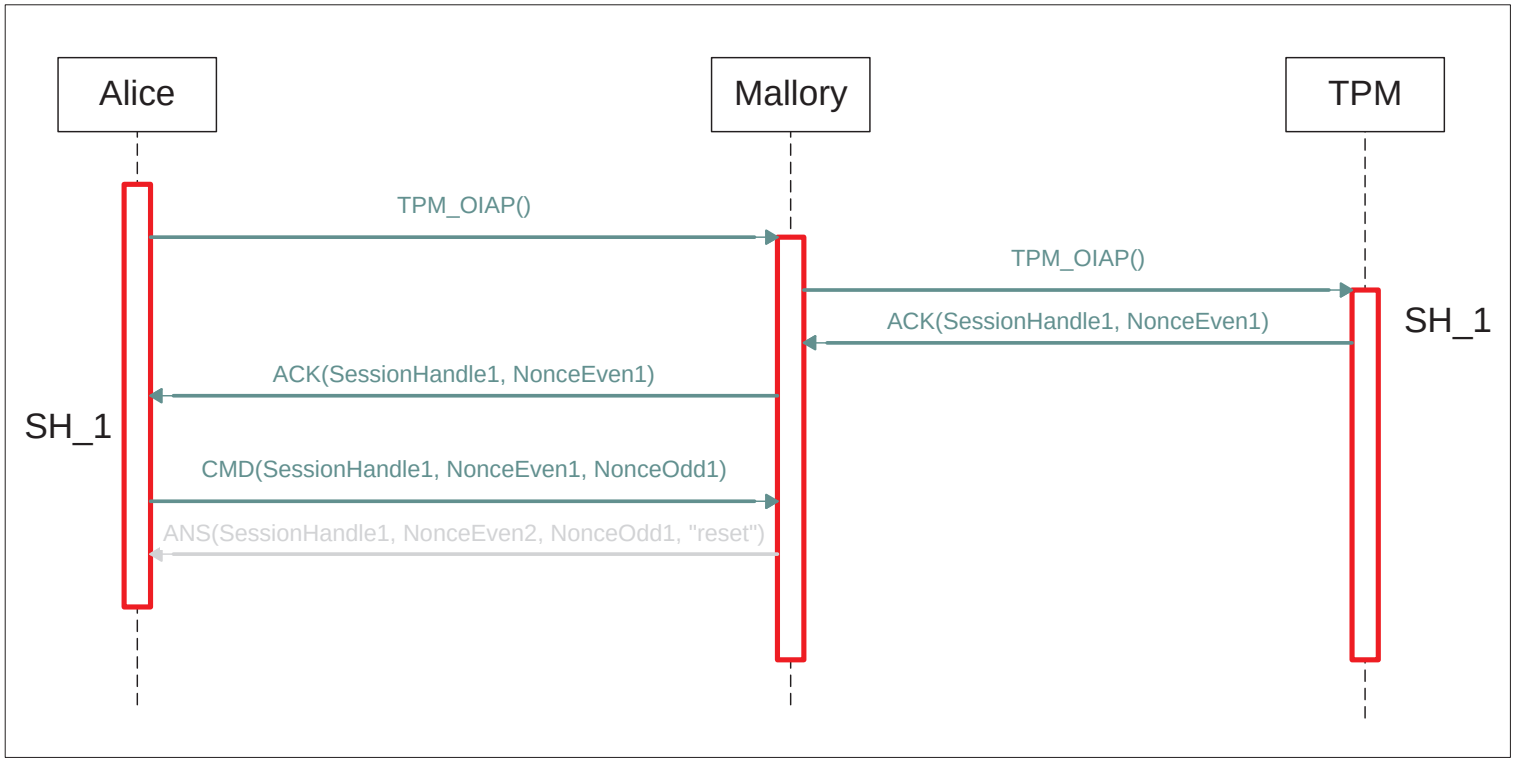
Message Storing Phase



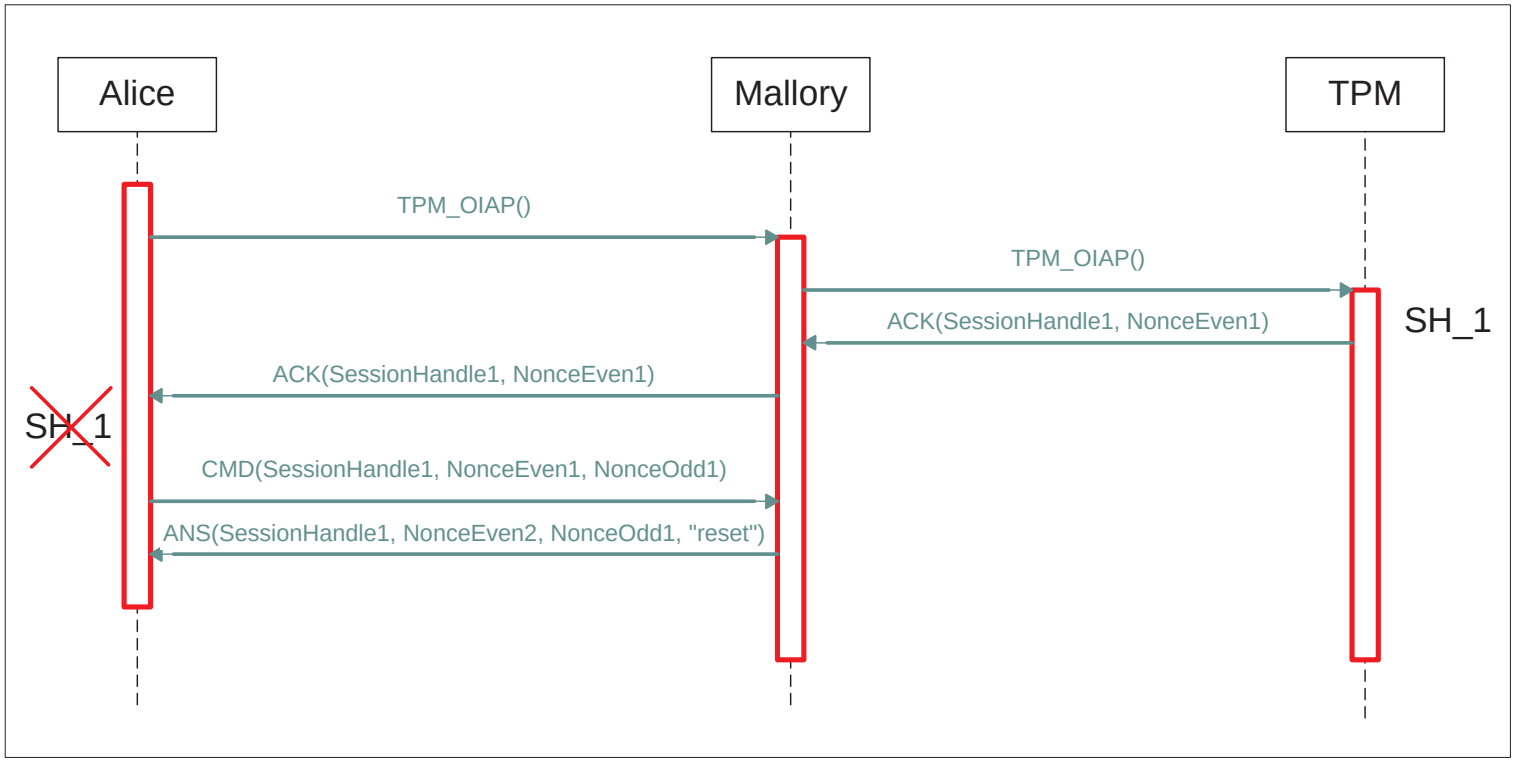
Message Storing Phase



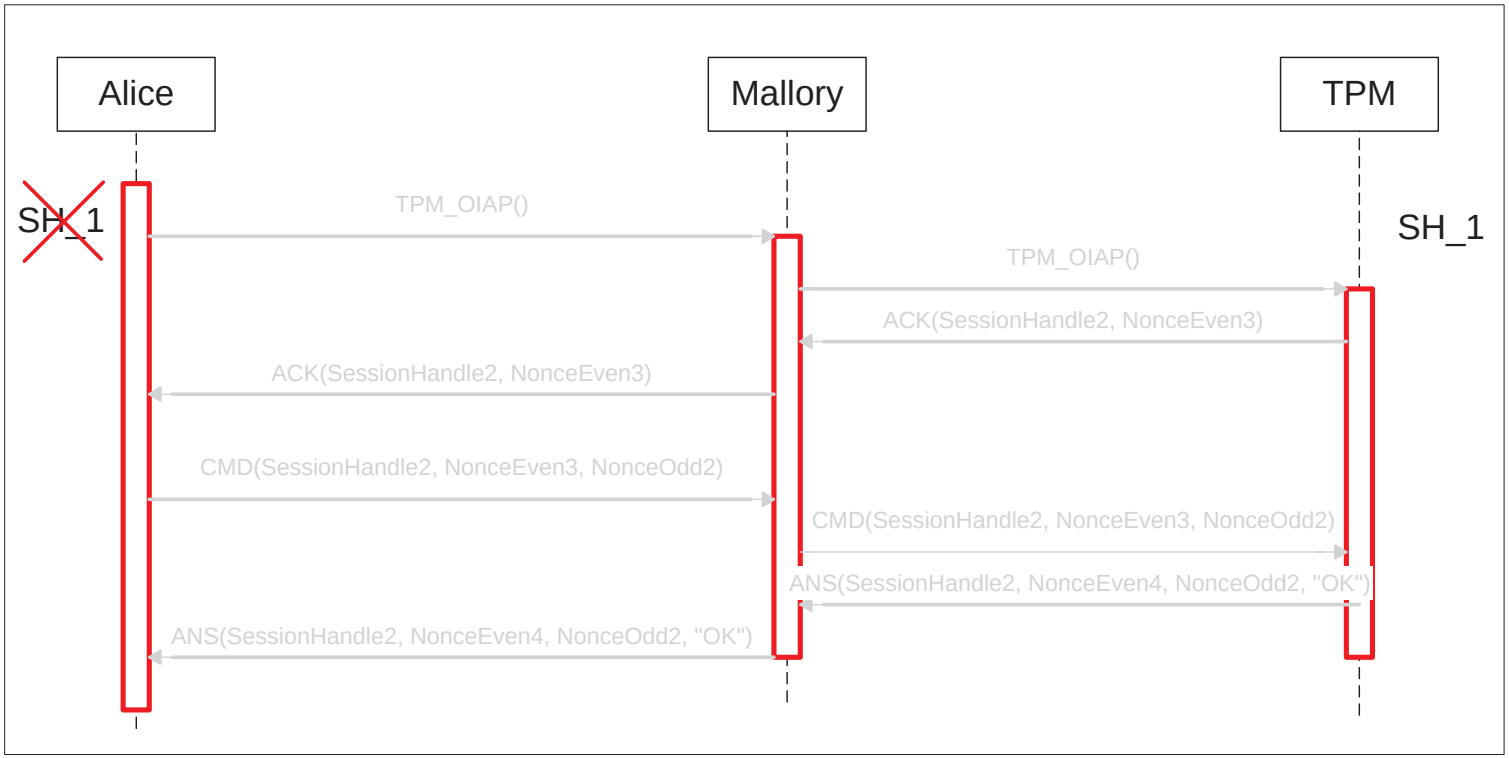
Message Storing Phase



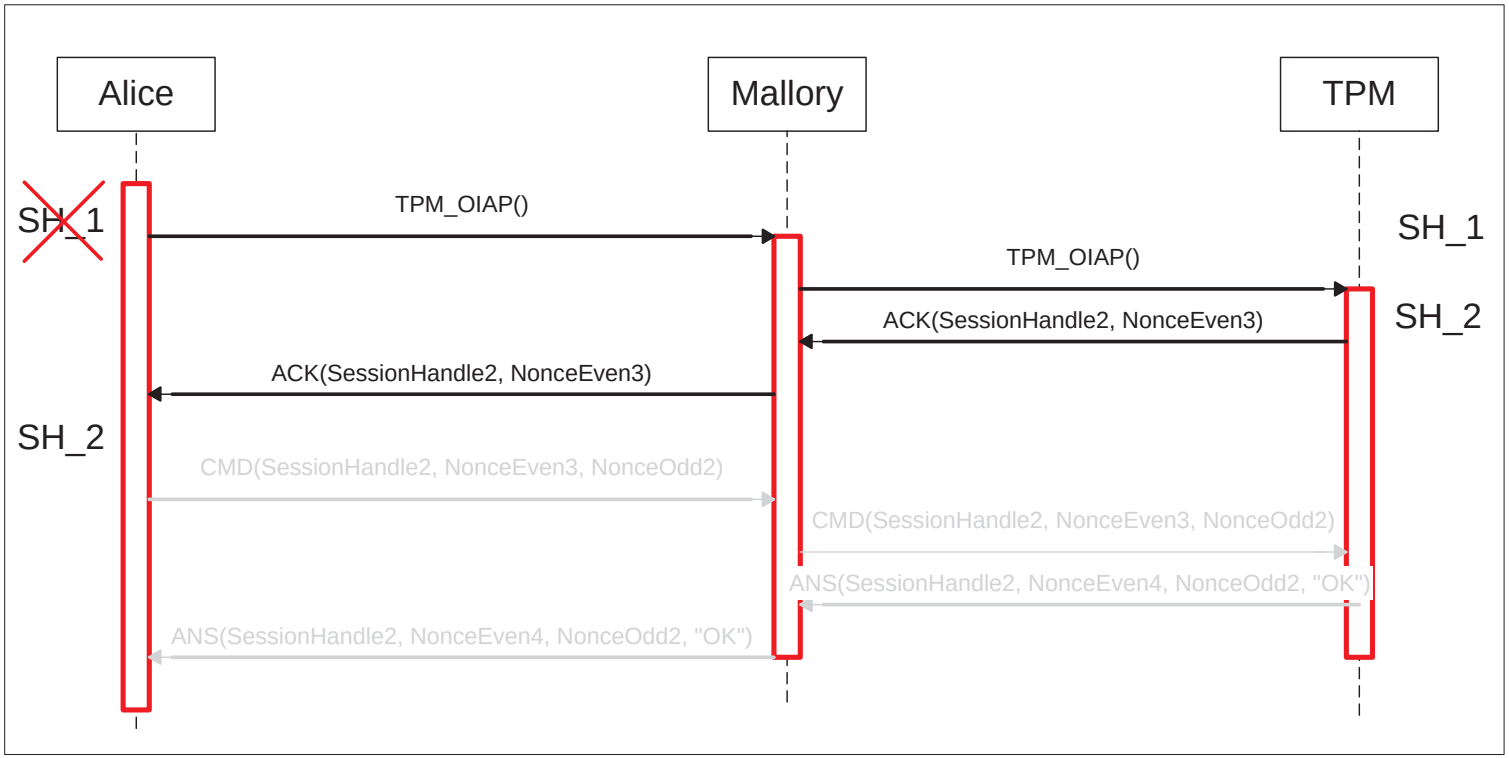
Message Storing Phase



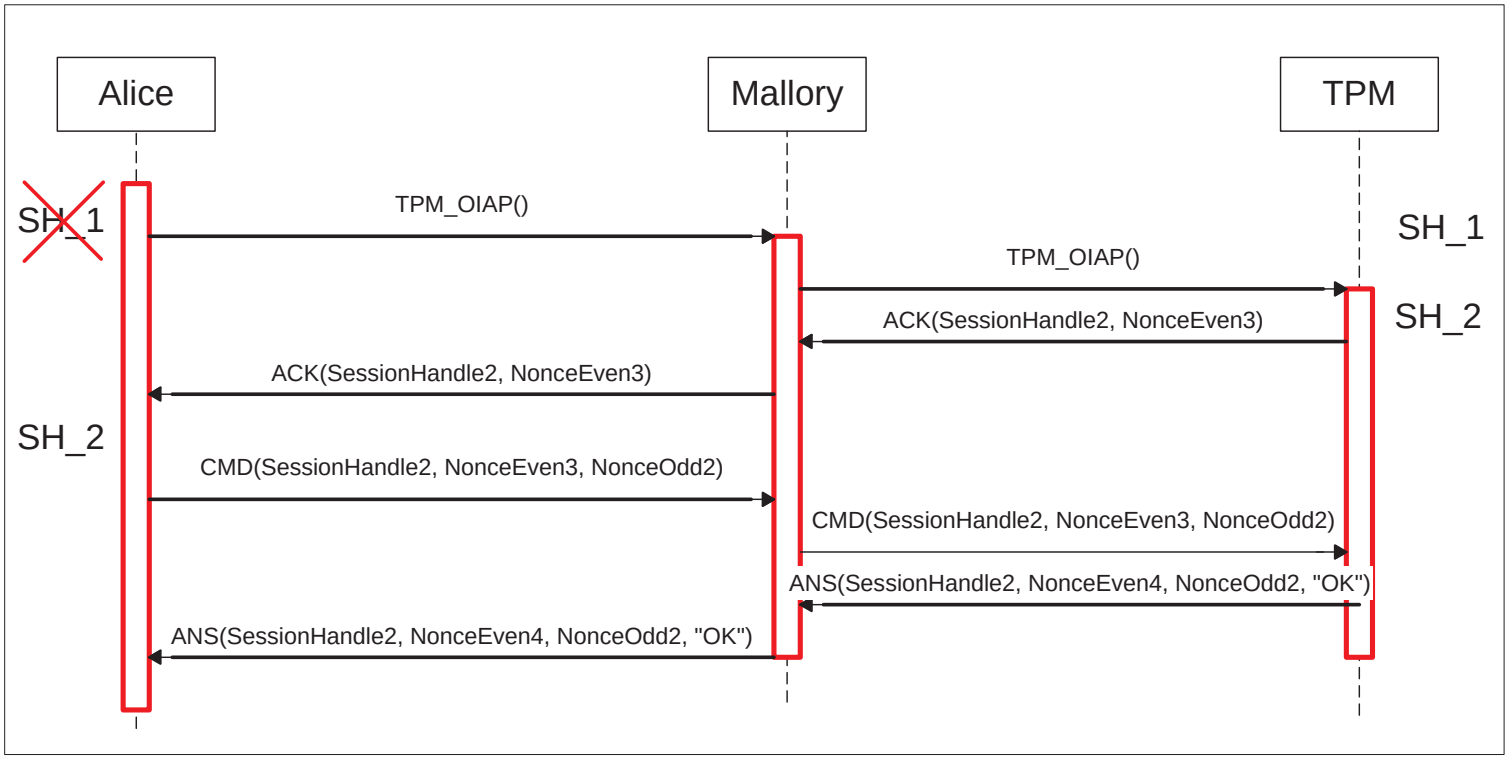
Message Resending Phase



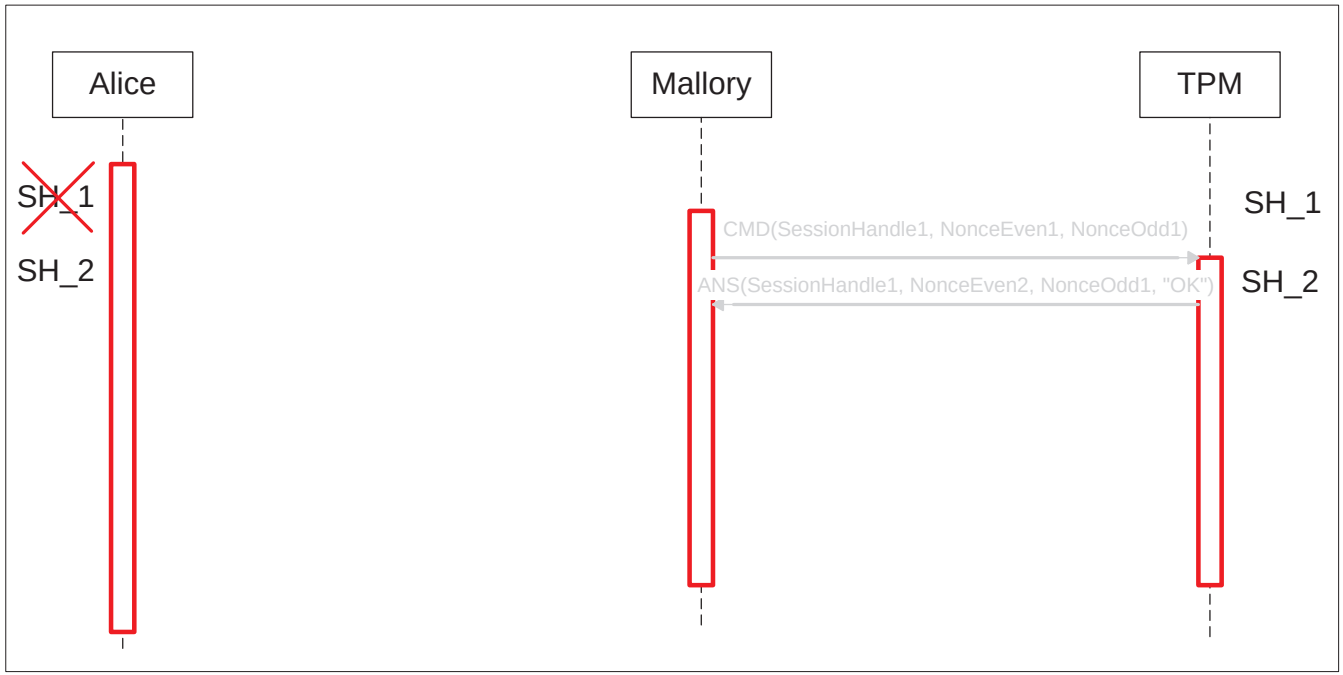
Message Resending Phase



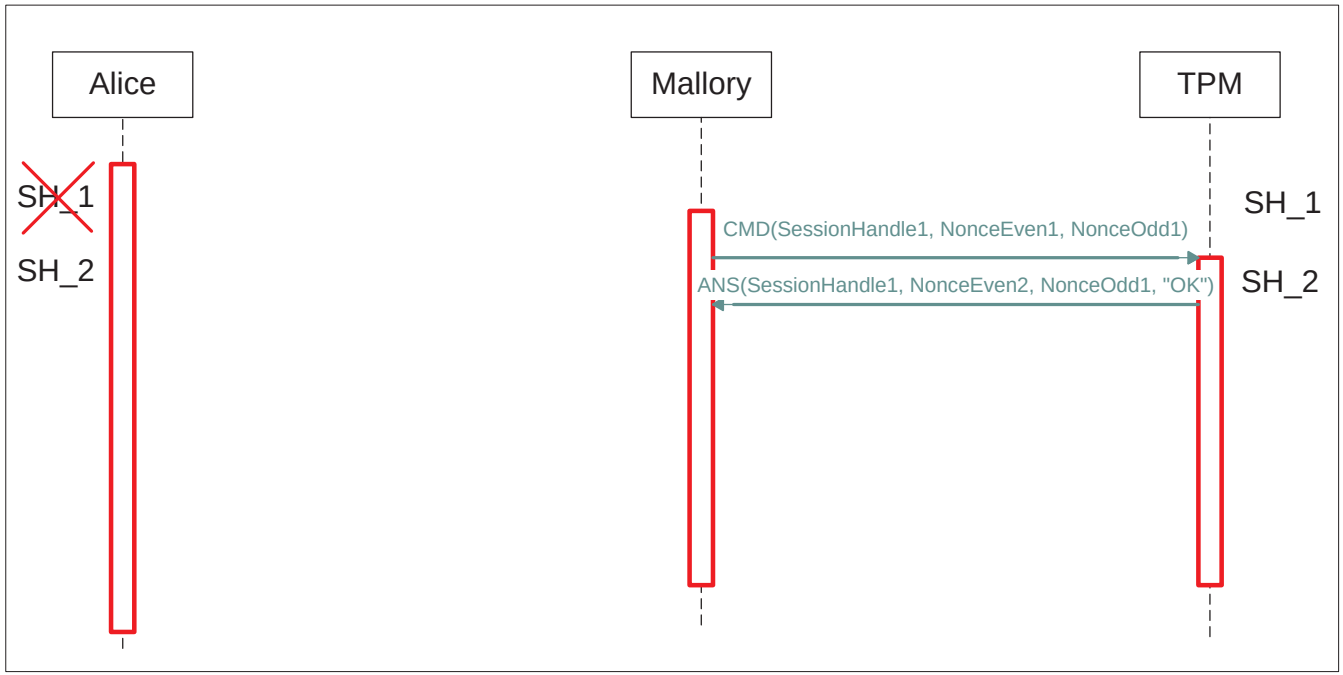
Message Resending Phase



Replay Attack Phase



Replay Attack Phase



Model Checker and Attack Property

What is wrong with the TCG Specification?

Model Checking techniques have been used to better understand the attack properties

- We modeled *Alice*, *Mallory* and the *TPM* using the SPIN model checker
- We noticed that a *coherent* and *consistent session knowledge* shared between the parties is missing from the TCG Specification

$\Rightarrow$ Hints about a solution just came up. . . :-)

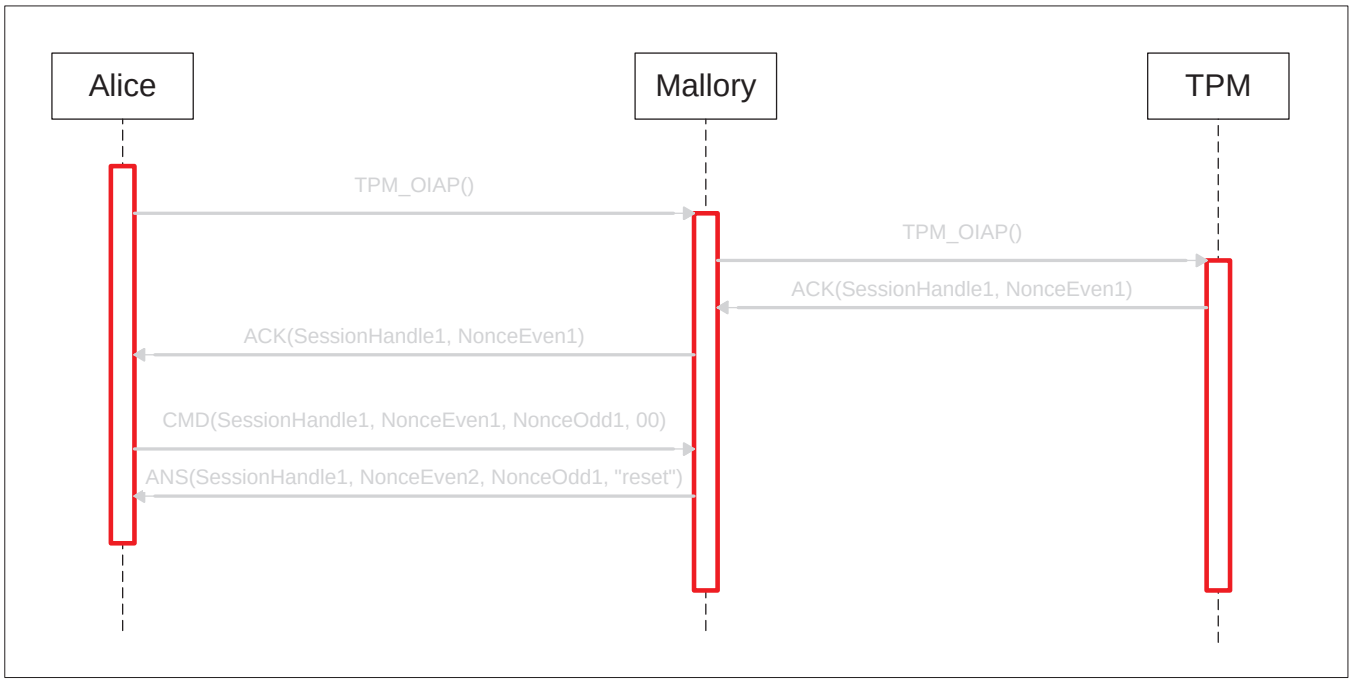
Proposed Solution

We propose to patch the hardware component TPM, by introducing a HMAC-protected **bitmask** in any authorized exchanged message, where

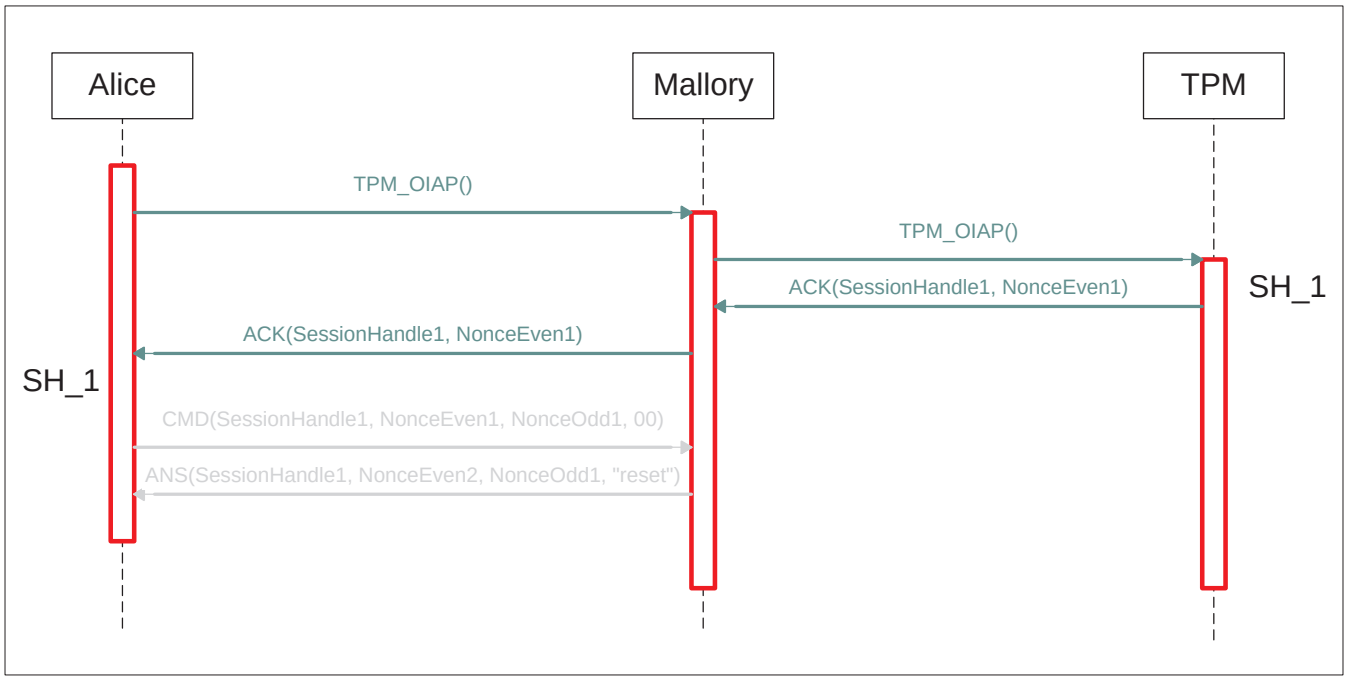
- the **i-th** bit is 0 if the **i-th** authorization session is considered either **open** or in an **unknown** state;
- the **i-th** bit is 1 if the **i-th** authorization session is considered **failed**

$\Rightarrow$ coherent and consistent shared session knowledge

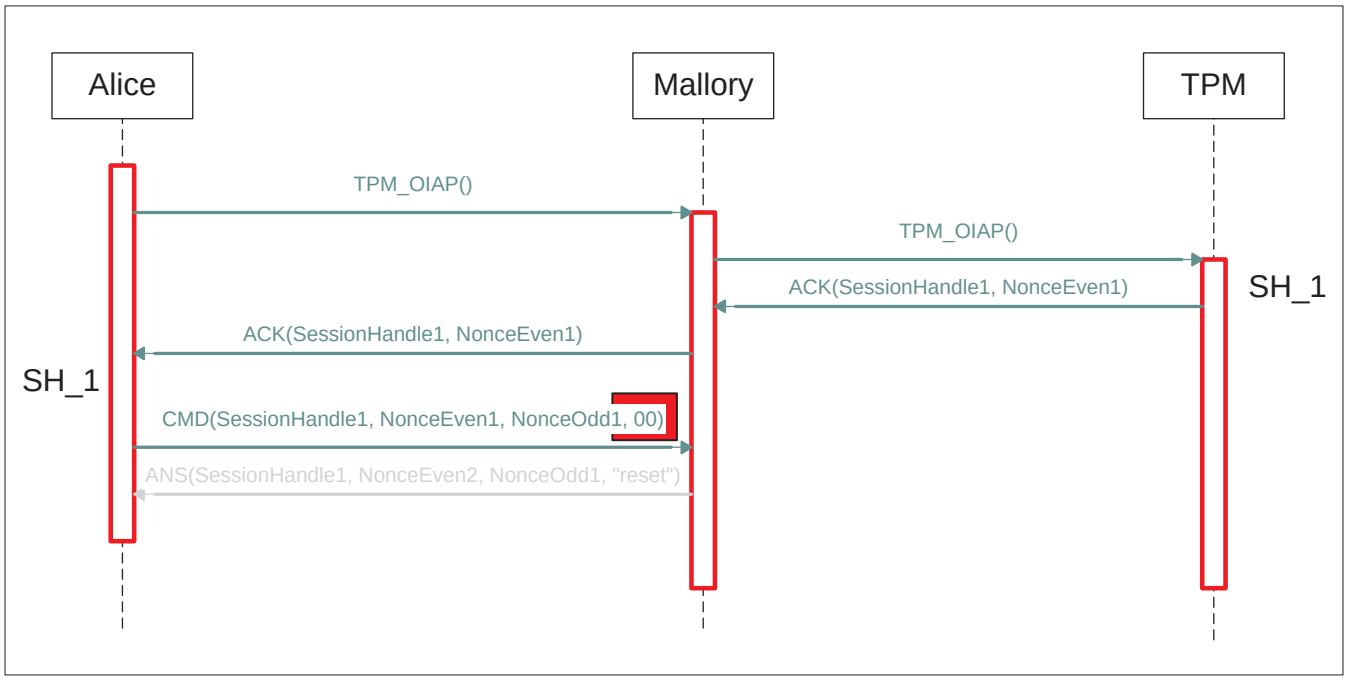
Proposed Solution
Solution Sketch (1)



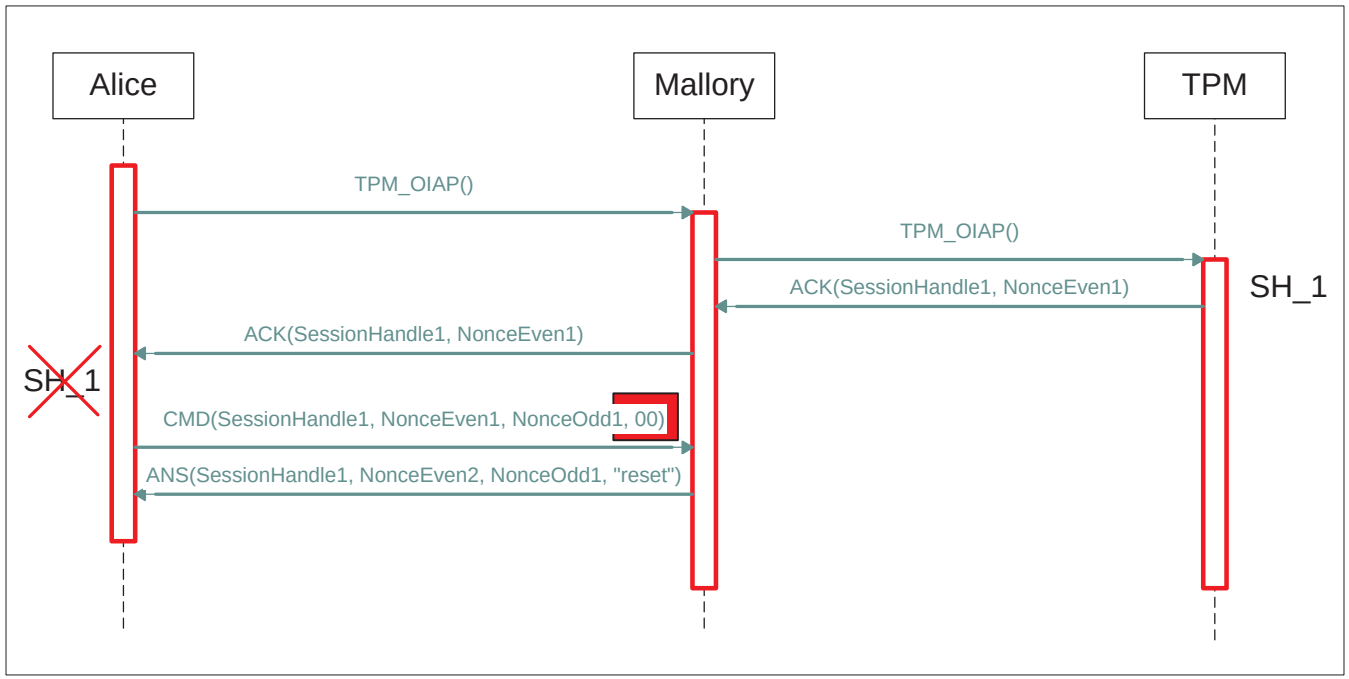
Proposed Solution
Solution Sketch (1)



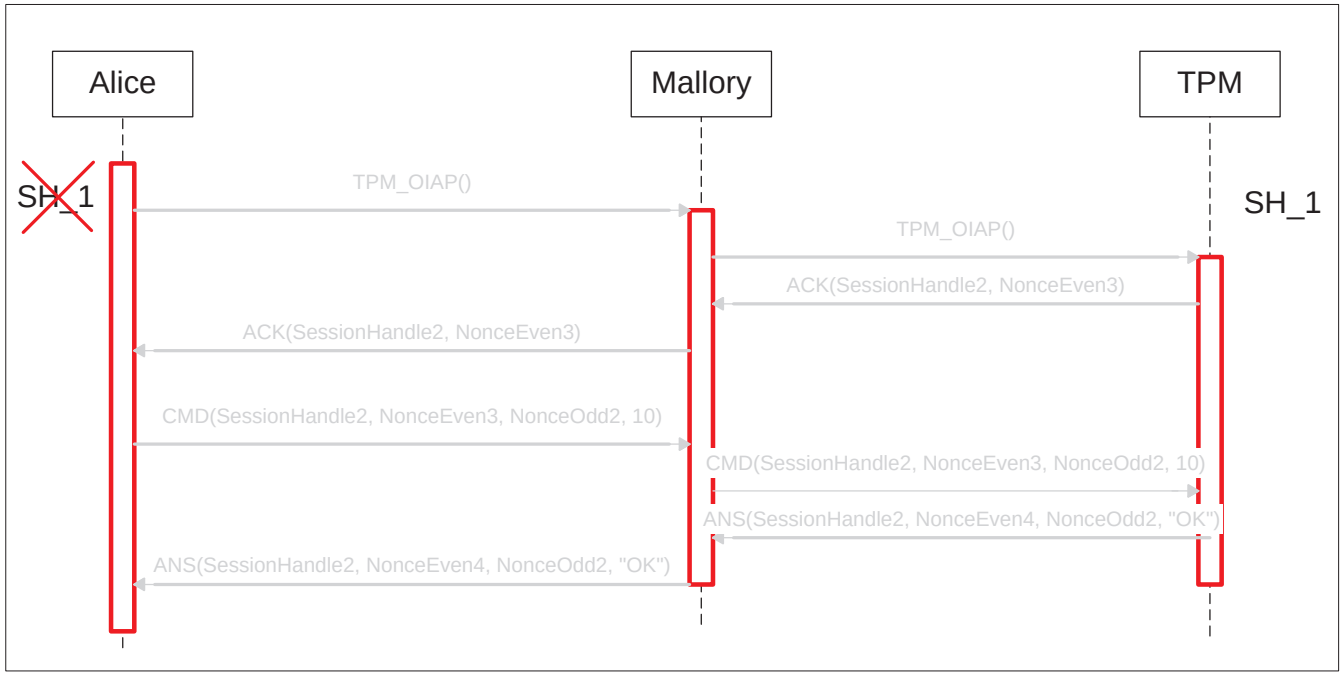
Proposed Solution
Solution Sketch (1)



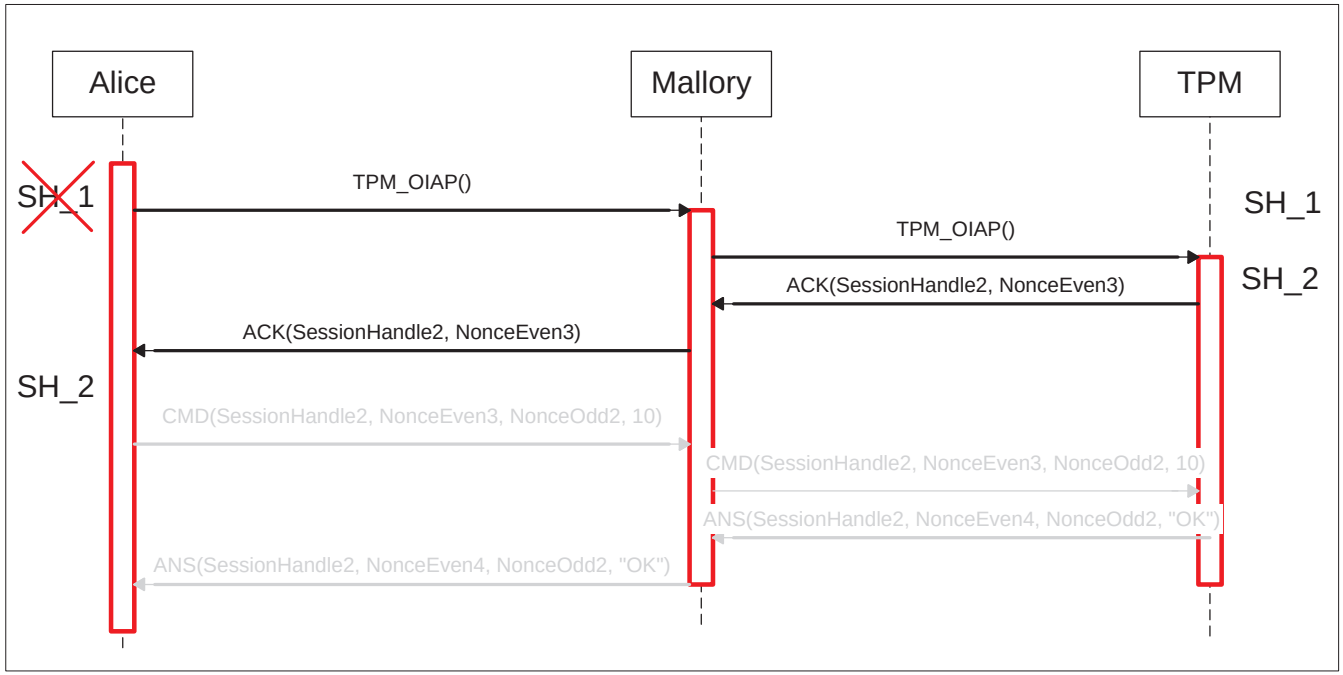
Proposed Solution
Solution Sketch (1)



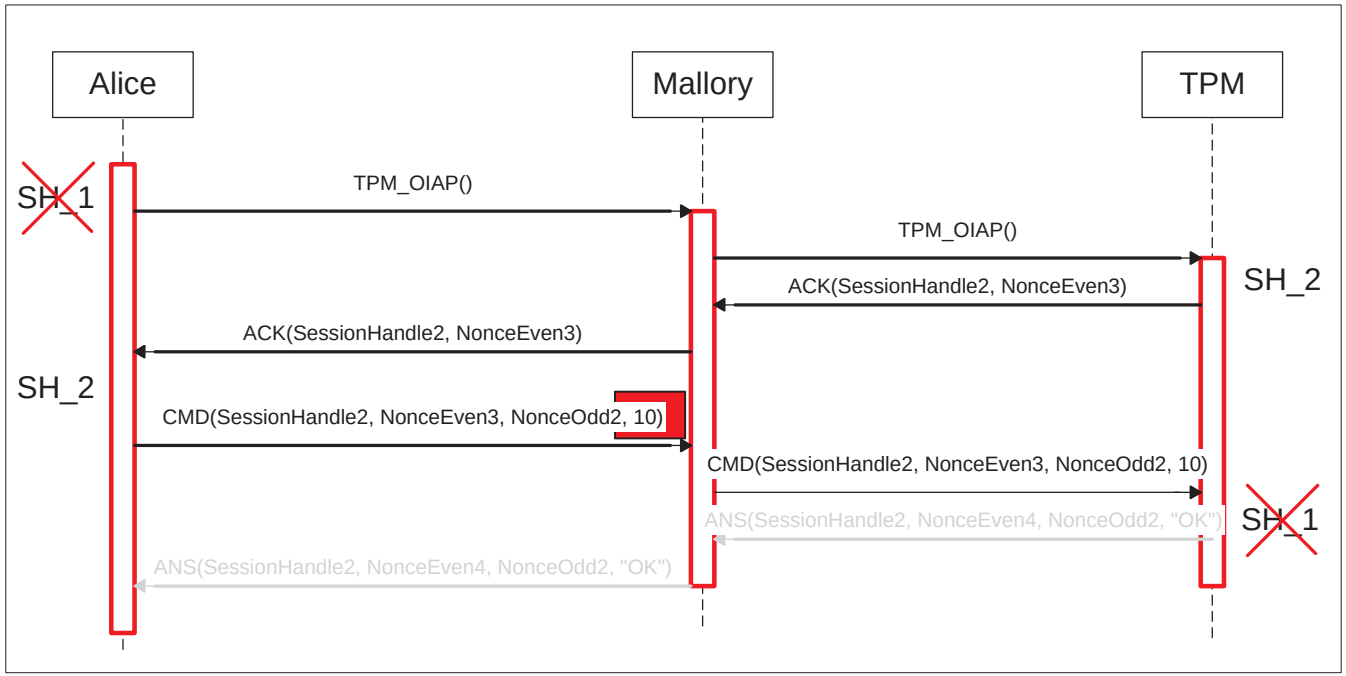
Replay Attack
Solution Sketch (2)



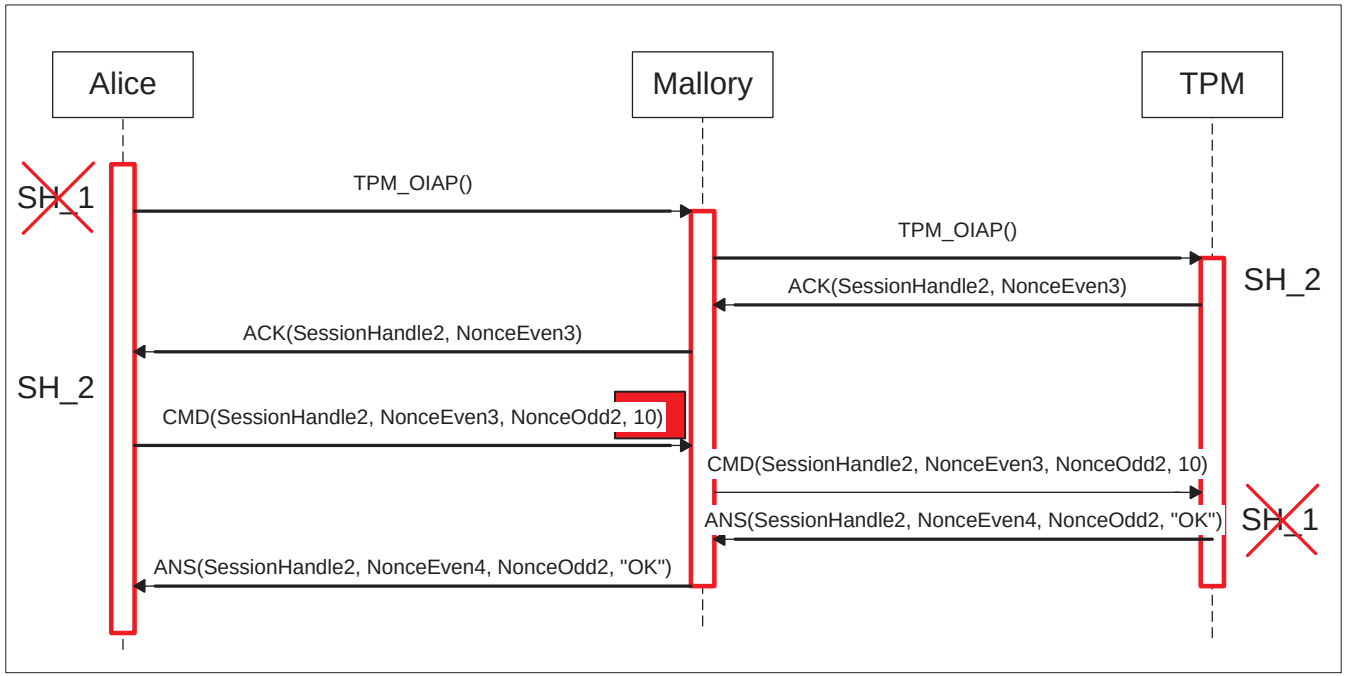
Replay Attack
Solution Sketch (2)



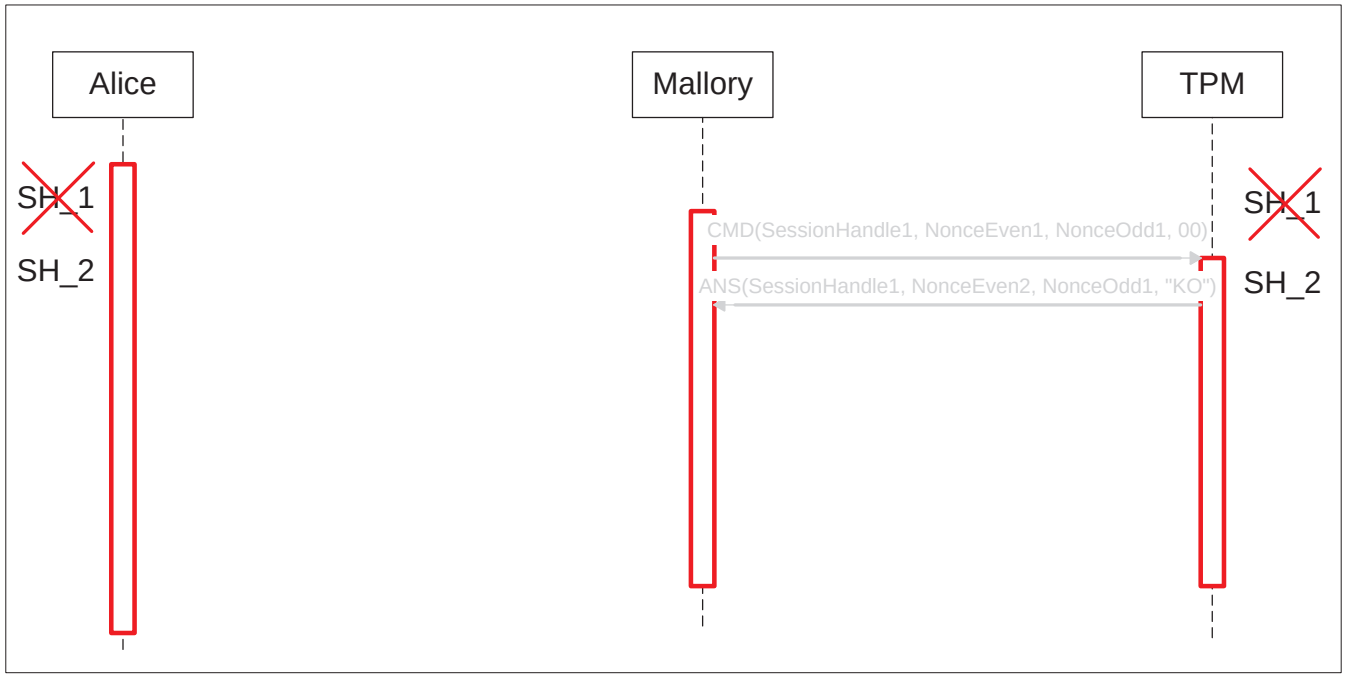
Replay Attack
Solution Sketch (2)



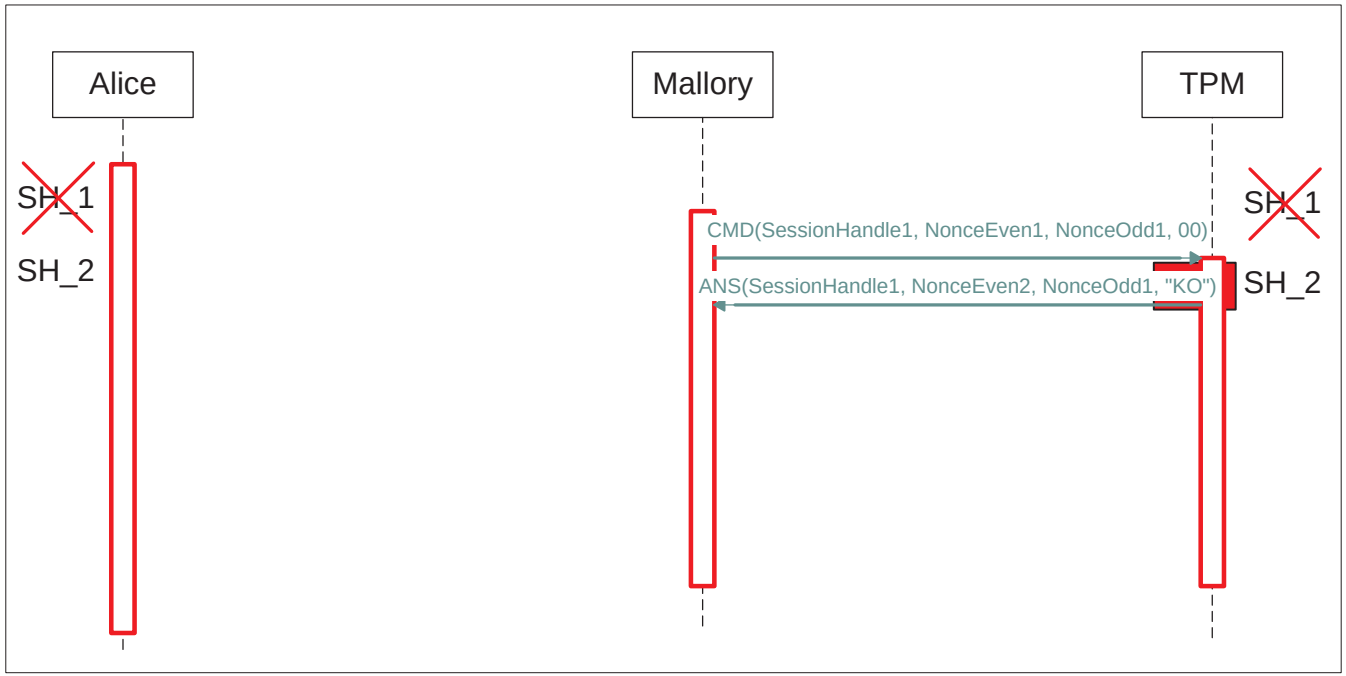
Replay Attack
Solution Sketch (2)



Replay Attack
Solution Sketch (3)



Replay Attack
Solution Sketch (3)



Conclusion and Future Works

- We recall TCG-based Trusted Computing Platforms
- Focus on TCG-based TPs *Authorization Protocols*
- We show a Straight Replay Attack against the Open-Independent Authorization Protocol, formally proved with the SPIN Model Checker
- We propose a solution based on the concept of shared session knowledge
- We are investigating a formal proof of the proposed solution

Thanks!

Thank you! :-)