Challenge Response Protocol & Secure Access

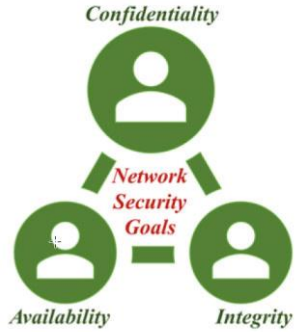
Vijay Kulkarni

Bosch Global Software Technologies
alt_future

Challenge Response Protocol

User Authentication

RECAP

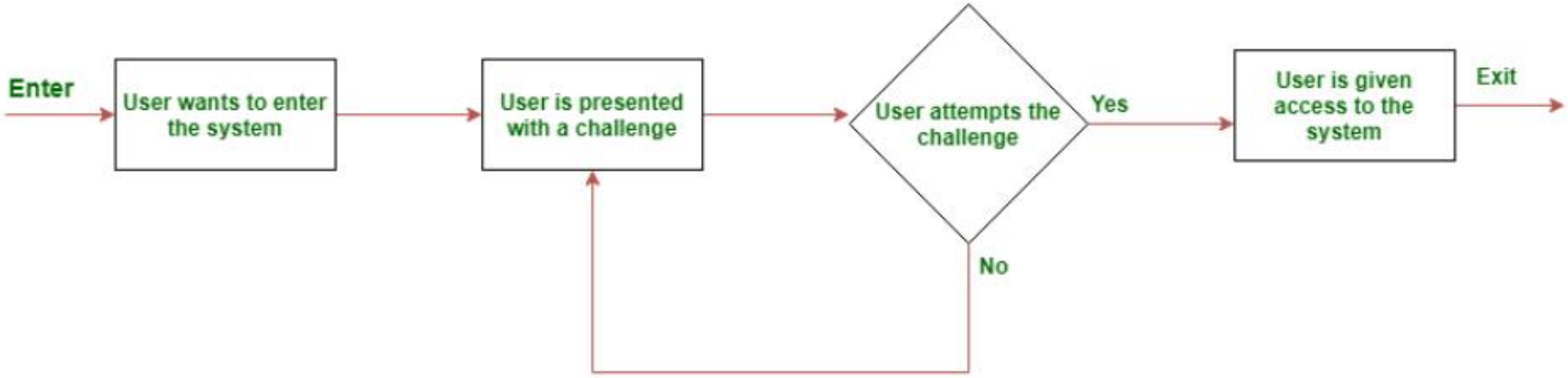


In general authentication is the mechanism that verifies the identity of a user. This guarantees the received message is really from the sender who is professing to be.

Cryptographic procedures, for example, the digital signature can be used for user verification. These are systems where the sender can sign the message utilizing a digital signature, while the receiver can confirm the message is really from the authorized sender or not.

Challenge Response Protocol

Control Flow



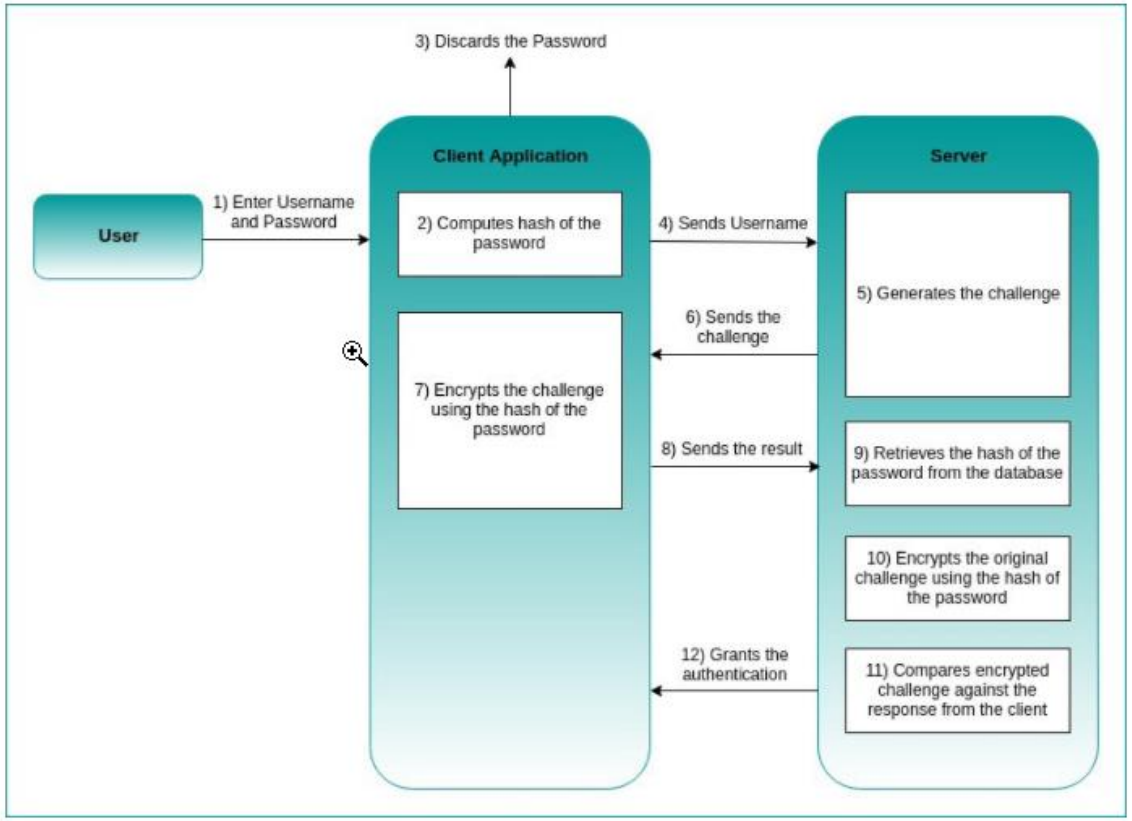
Challenge–Response authentication is a of protocol in which one party presents a question ("challenge") and another party must provide a valid answer ("response") to be authenticated..

The simplest example of a challenge–response protocol is Password authentication, where the challenge is asking for the password and the valid response is the correct password.

An adversary who can eavesdrop on a password authentication can then authenticate itself in the same way. One solution is to issue multiple passwords, each of them marked with an identifier. The verifier can ask for any of the passwords, and the prover must have that correct password for that identifier. Assuming that the passwords are chosen independently, an adversary who intercepts one challenge–response message pair has no clues to help with a different challenge at a different time.

Challenge Response Protocol

Control Flow

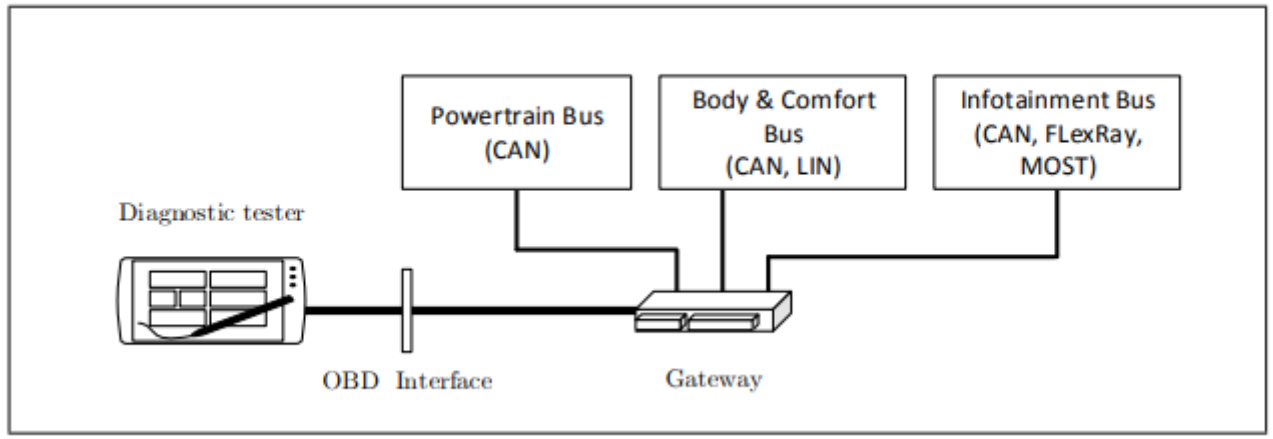


Challenge Response Authentication Protocol: Flow of actions

1.The user opens an Application that needs to be logged in and provide Username and Password.
2.The client application computes a cryptographic hash of the password.
3.Discards the actual password.
4.The application sends the Username to the server in plaintext.
5.The server generates a 16-byte random number called a challenge.
6.Sends the challenge to the client application.
7.The application receives the challenge and encrypts it using the hash of the password.
8.The application sends the resulting encrypted challenge to the server.
9.The server uses the previously received Username to retrieve the hash of the user's password from the database.
10.The server encrypts the original challenge using the password hash.
11.The server compares the encrypted challenge against the response computed from the application for the challenge.
12.If they are similar, then the authentication is granted.

Secure Access

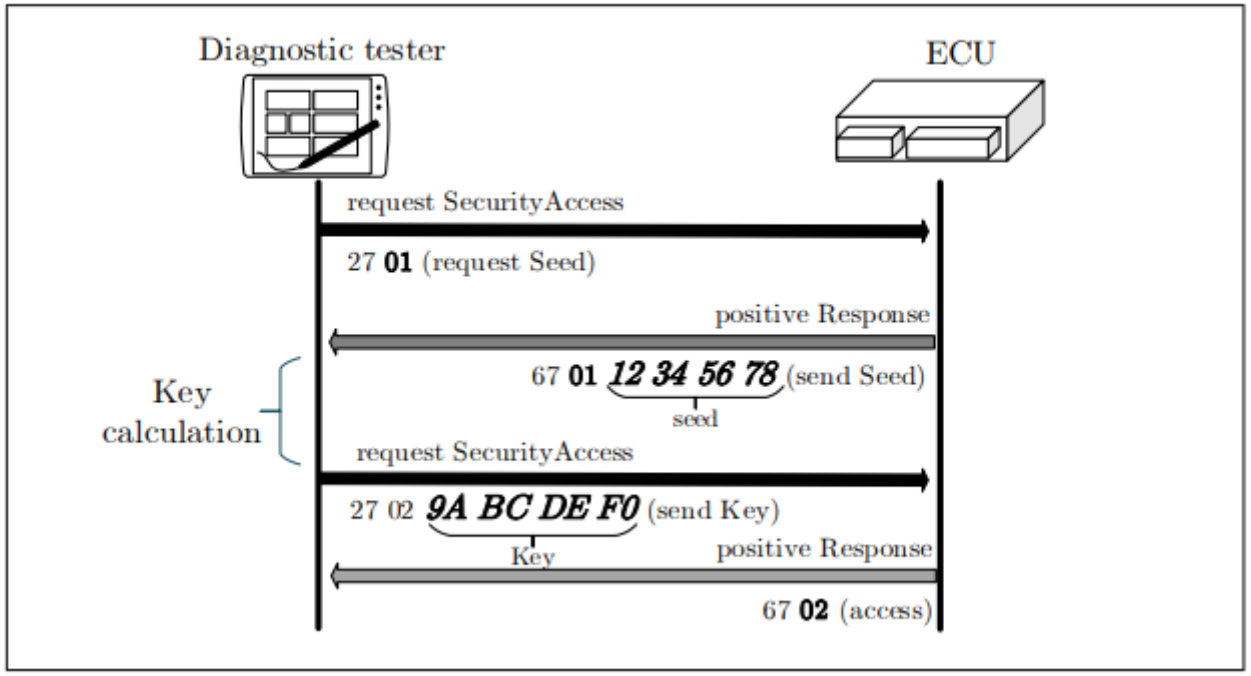
Automotive use case

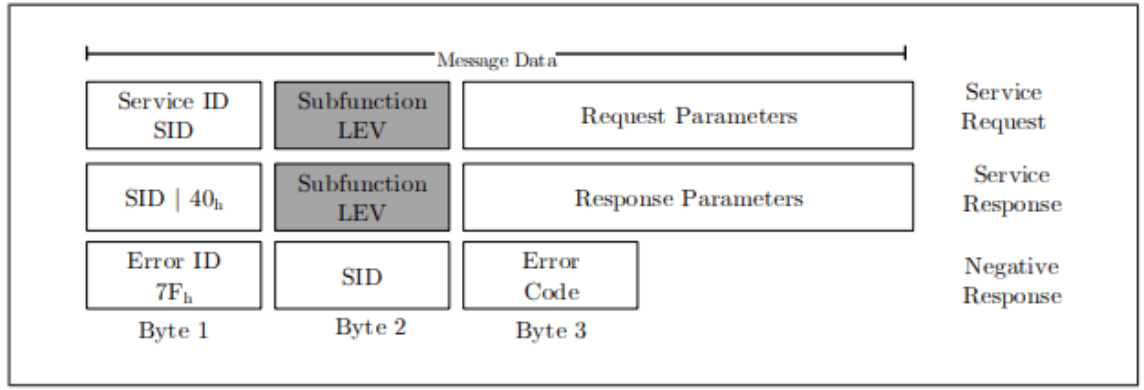


There are two popular diagnostic protocols: one is the Keyword Protocol (KWP) 2000 which is standardized in the ISO 9141 and ISO 14230; the other one is the Unified Diagnostic Services protocol (UDS) which is standardized in the IS0 14229. The operation of both diagnostic protocols is almost identical.

Secure Access

UDS $27 service



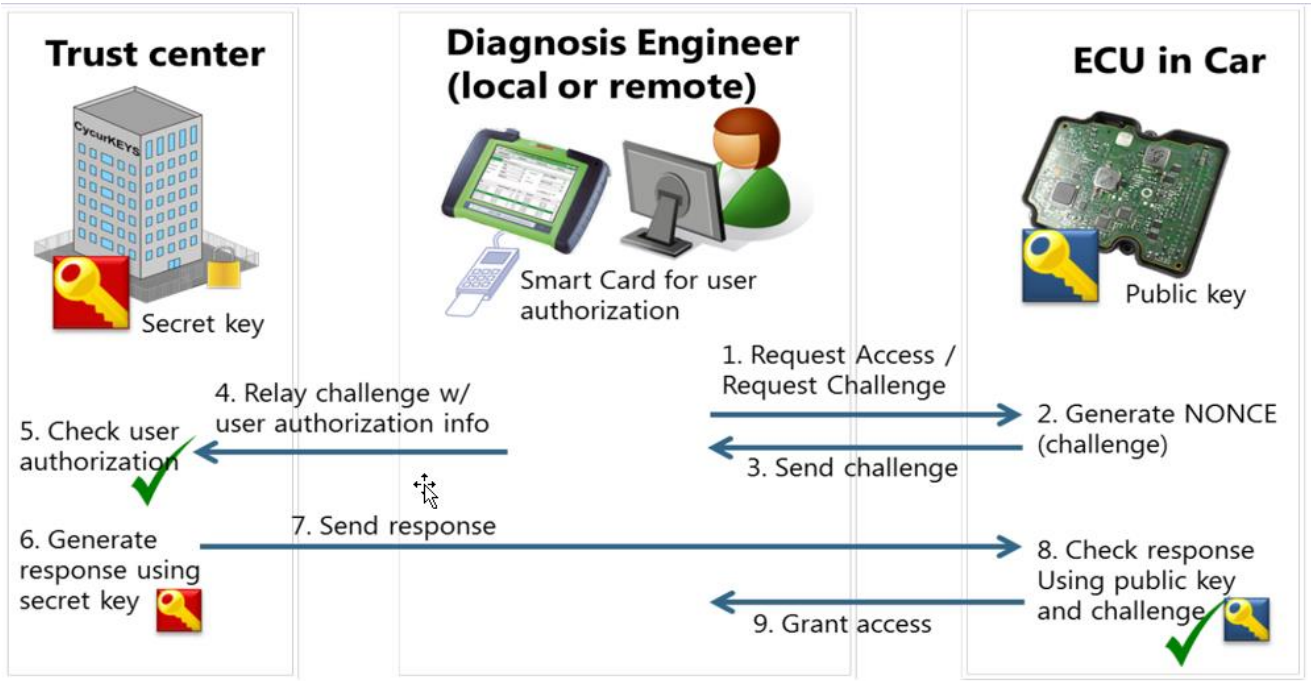


RequestSeed : LEV 01h, 03h, 05h, 07h – 5Fh
SendKey : LEV 02h, 04h, 06h, 08h – 60h

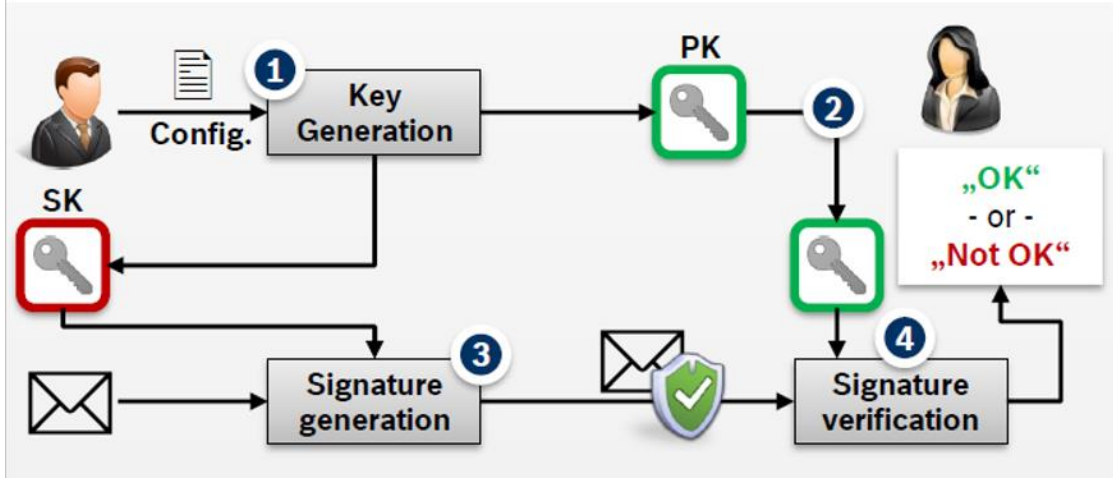
Bosch Global Software Technologies
alt_future

Secure Access

Asymmetric Cryptography



RECAP



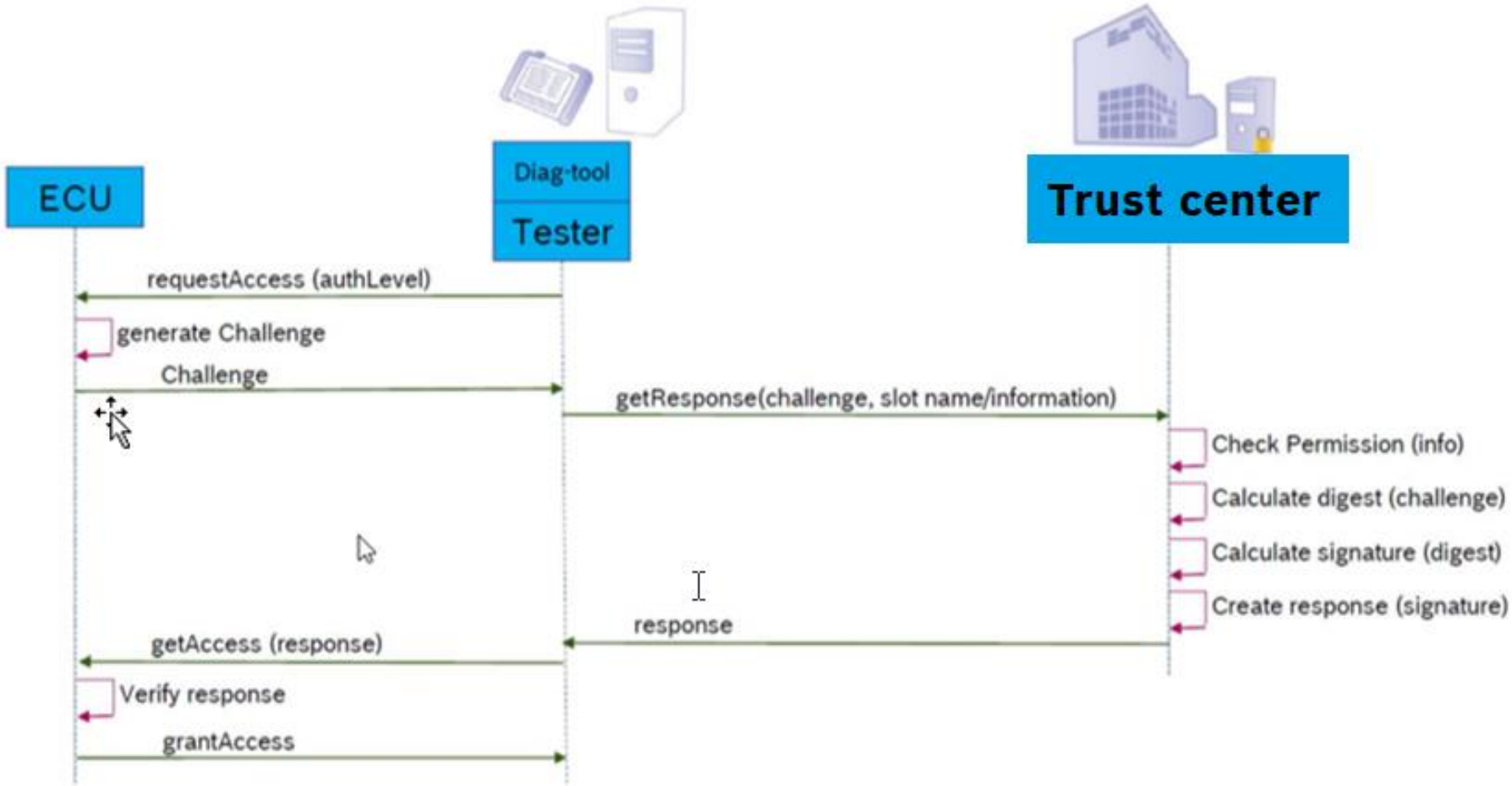
Application scenario & basic principle

- Scenario: **Bob sends a message to Alice and wants to prove the message's authenticity and integrity**

1. **Key generation:** Bobs generates a key pair (i.e., **public verification key PK** and **secret signing key SK**)
2. **Publish PK:** Bob makes his public verification key available to everyone, e.g., Bob sends the PK directly to Alice)
3. **Signature generation with SK:** Using the signing key, Bob computes a **digital signature** for the message
4. **Signature verification:** Alice uses PK to check the signature

Secure Access

Control Flow



References

https://www.geeksforgeeks.org/challenge-response-authentication-mechanism-cram/

Challenge–response authentication - Wikipedia

Challenge Response Authentication Protocol | by Nipuna Dilhara | Medium

Thank you!

Bosch
Global
Software
Technologies
alt_future