TLS-RSA-PSK

Channel Binding using
Transport Layer Security
with Pre Shared Keys

Christian J. Dietrich
dietrich [at] internet-sicherheit . de

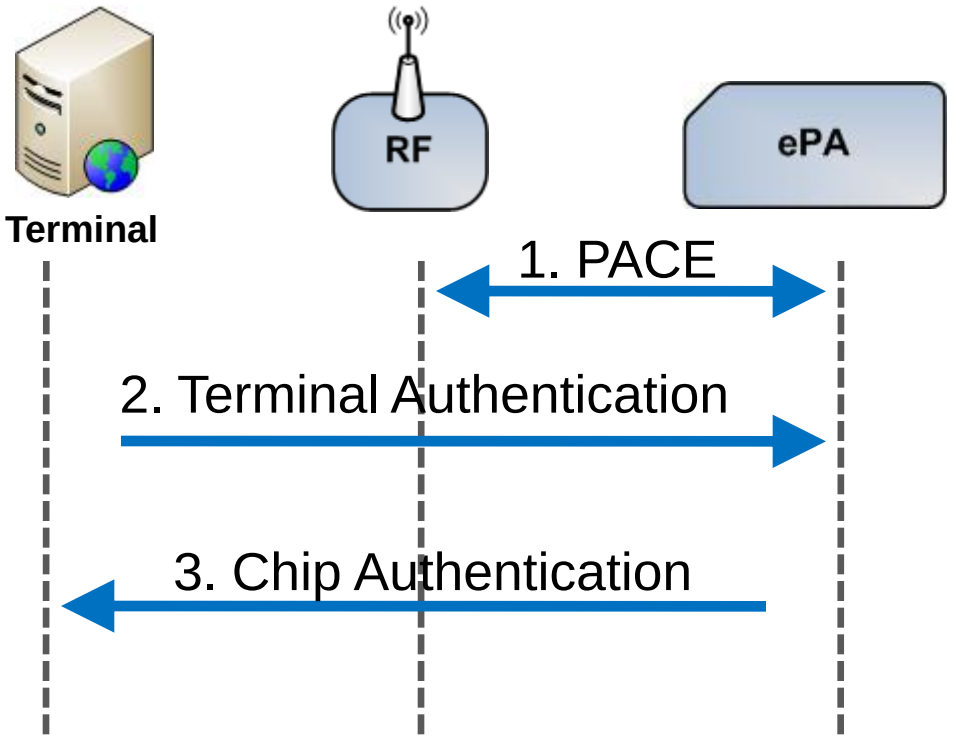
Institut für Internet-Sicherheit
https://www.internet-sicherheit.de
FH Gelsenkirchen

if(is) internet security.

Introduction

- Nov. 2010: New German ID card
- **eID functionality**
  - **authenticate** mutually between eID card and eID server
  - **transmit** personal data from eID card to eID server (securely)
- Extended Access Control (by BSI)
- 3 kinds of card readers
  - Basic (no pinpad)
  - Standard
  - Comfort

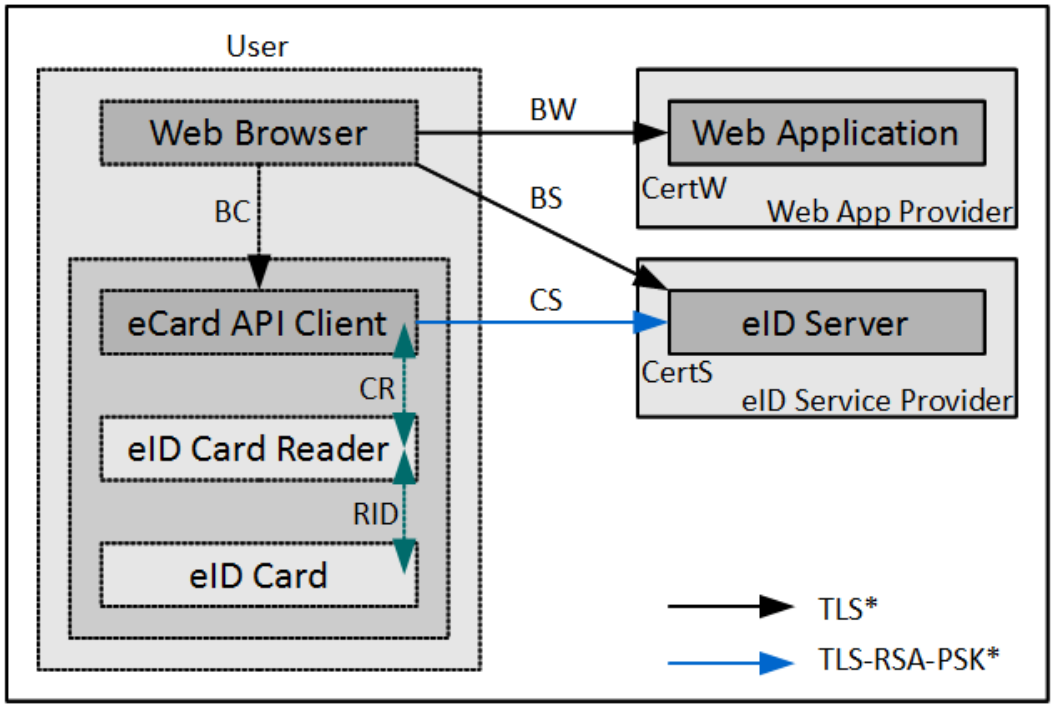
Terminal and Chip Authentication



1. **PACE**
   Bind eID to card reader (Diffie-Hellman)
2. **Term. Authentication**
   eID card authenticates the Terminal (Terminal Certificate)
3. **Chip Authentication**
   Terminal authenticates the eID card

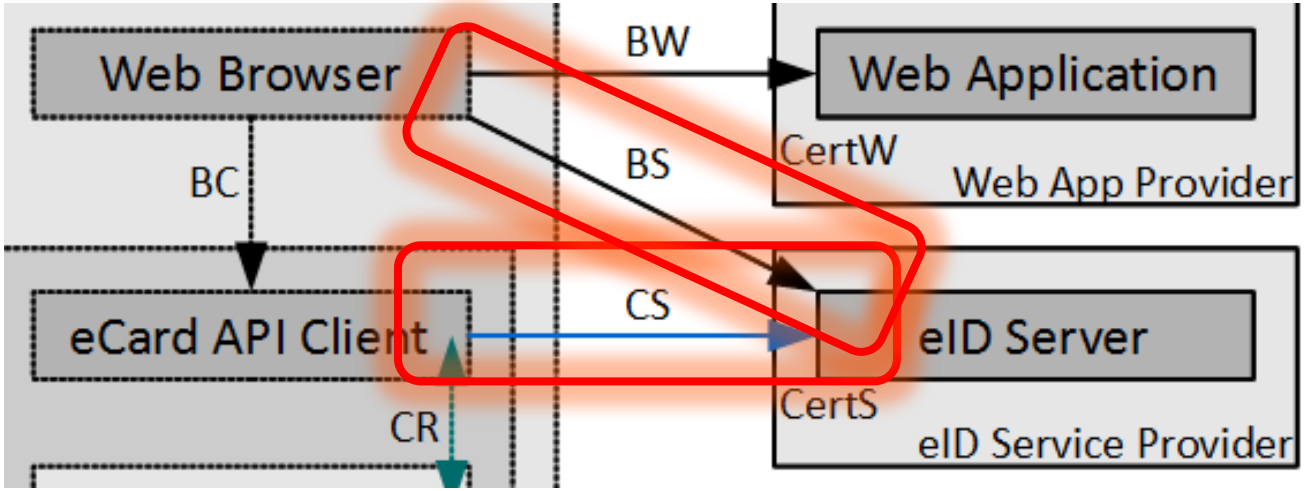
BSI Technical Guideline TR-03110

eID authentication in practice



- Lots of components involved
  - Browser, Browser plugin
  - 2 Servers
  - eID reader & card
- Several TLS connections
  - However: Connection management is browser‘s concern!

Channel Binding

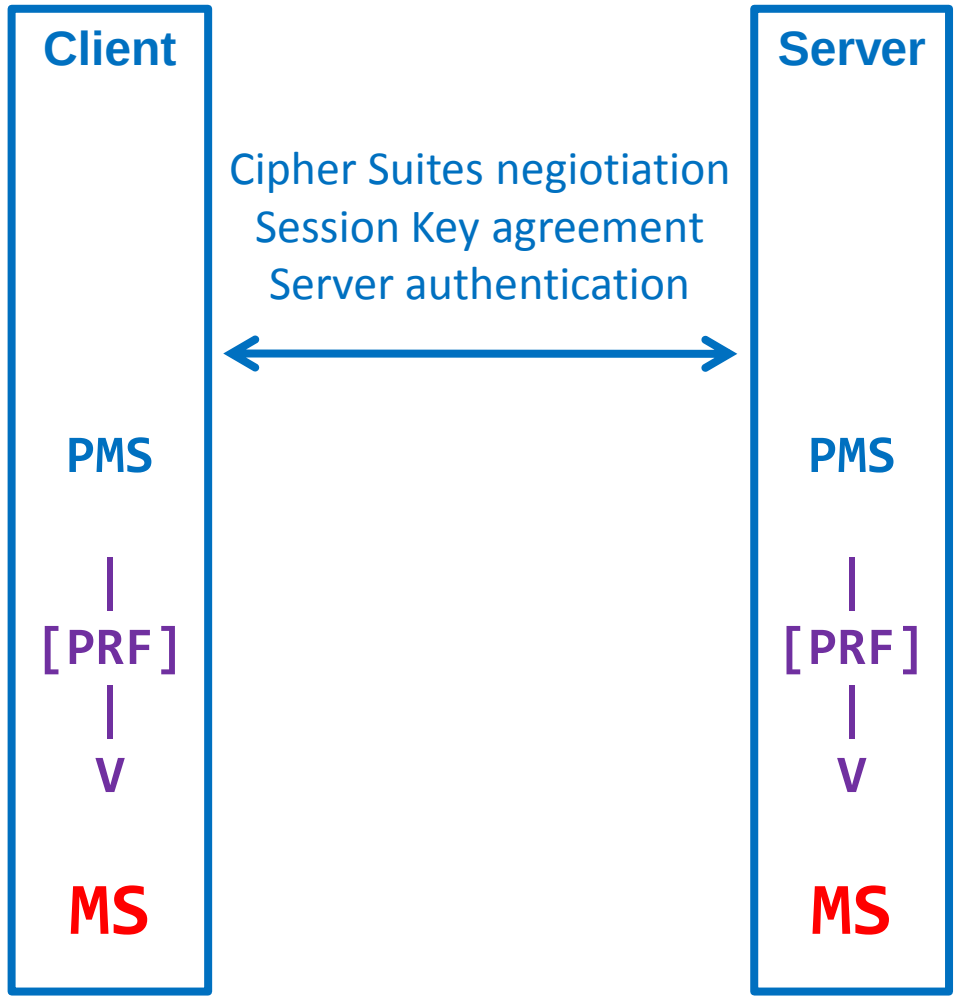


- Bind channel *BS* to *CS*
  - eID Server generates a Pre Shared Key and sends it via *BS* to the Web Browser
  - Browser plugin is triggered and provides the PSK to eID API client
  - PSK is then used to establish *CS*

Transport Layer Security (TLS/SSL)

- Negotiate Cipher Suite
- Session Key agreement
- Optional server authentication

- Pre Master Secret (PMS)
  - Input to the key derivation function (PRF) in order to get the Master Secret
- Master Secret (MS)
  - 48 byte fixed length output from PRF
  - Derive session keys for encryption and hashing (HMAC)

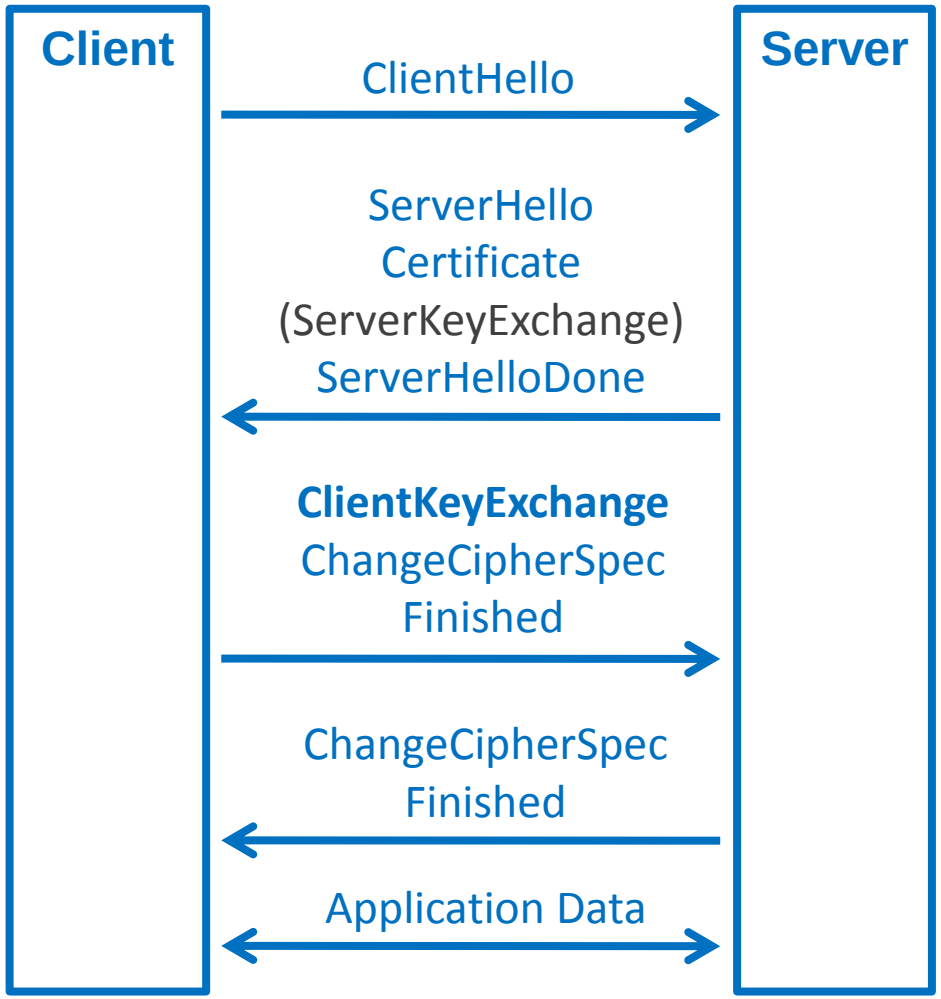


TLS with Pre Shared Keys

- Defined in RFC 4279 (Dec 2005)
- 3 variants of TLS with PSK
- Plain PSK
  - No server authentication
  - No Perfect Forward Secrecy
- **RSA-PSK**
  - Server authentication (certificate)
  - No Perfect Forward Secrecy
- DHE-PSK
  - No server authentication
  - Perfect Forward Secrecy

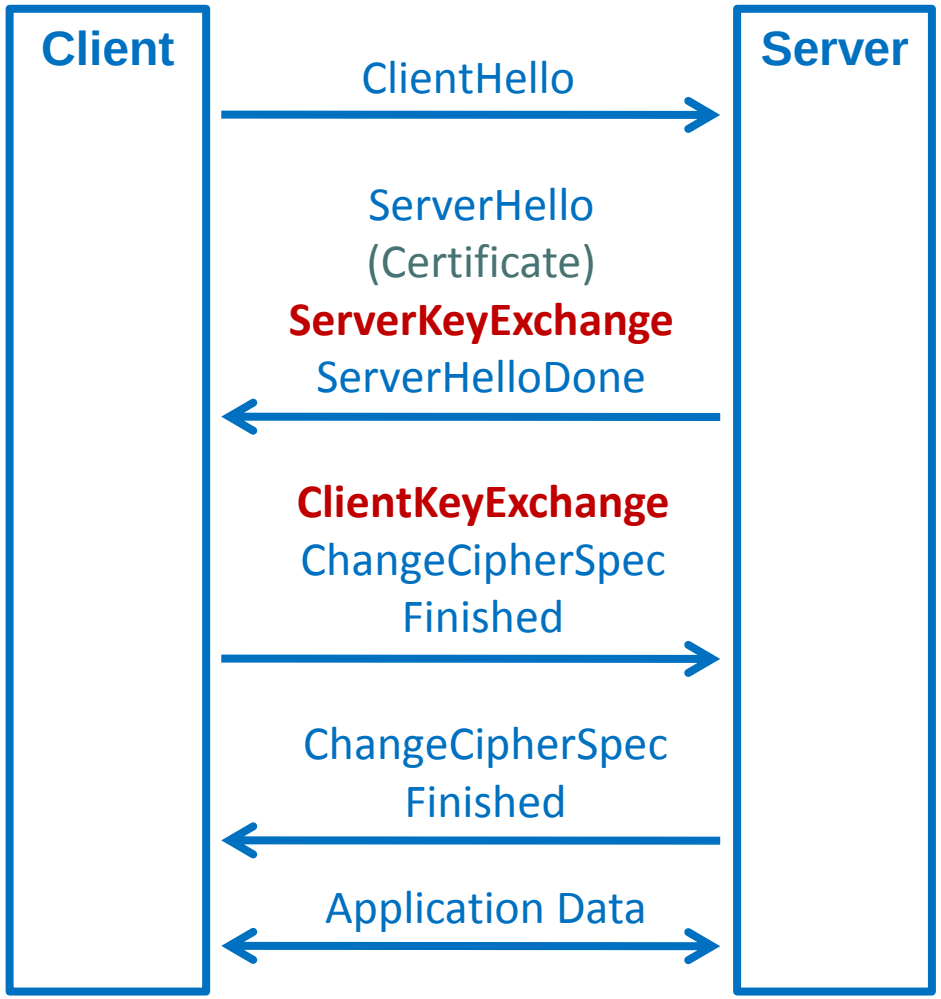
„usual“ TLS with plain RSA

- Certificate message required
- ServerKeyExchange is optional (not needed for plain RSA)
- ClientKeyExchange
  - Client builds premaster secret
  - `pms = (client_version, 46 random bytes)`
  - Transmit **encrypted premaster secret** to the server
  - `RSAEncrypt(pubkey`$_{Server}$`, pms)`



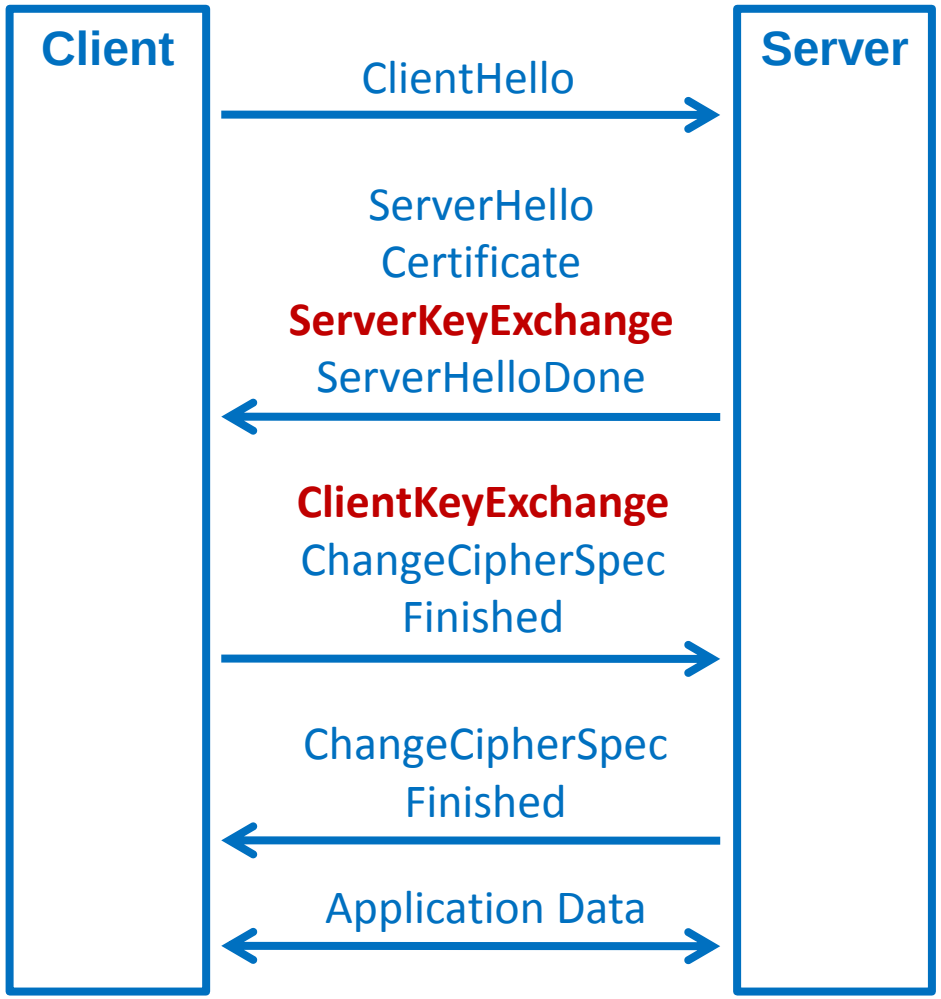
TLS with plain PSK

- ServerKeyExchange
  - Provide a hint for the client identity (username)
- ClientKeyExchange
  - Transmit **client_identity** to the server
- Both parties build the premaster secret as follows
  - `pms = (len(PSK), 0x00 * len(PSK), len(PSK), PSK)`



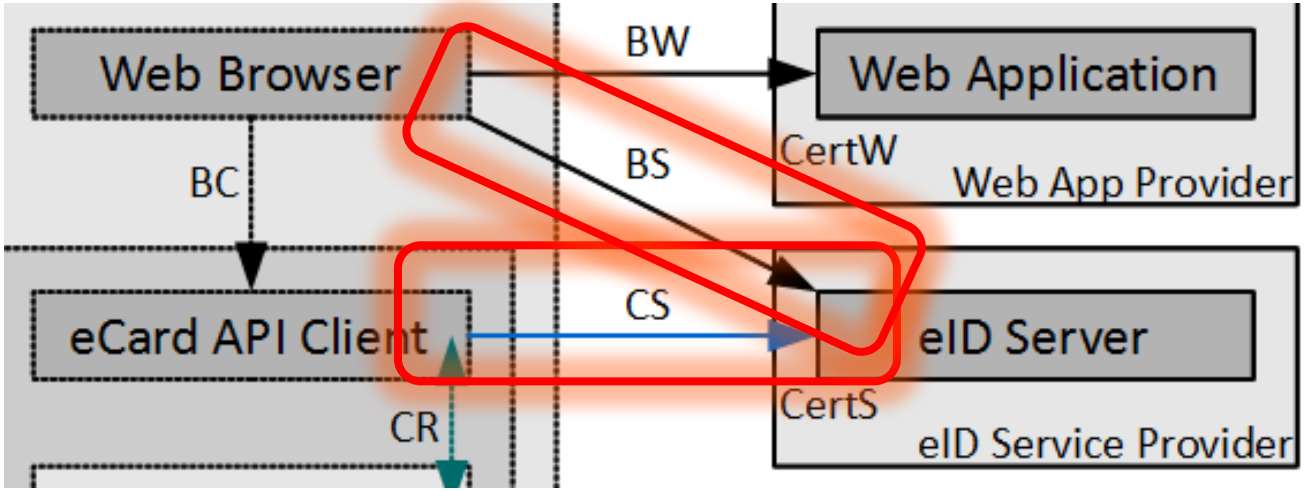
TLS with RSA-PSK

- ServerKeyExchange
  - Provide a hint for the client identity (username)
- ClientKeyExchange
  - Client builds premaster secret
  - `pms = (0x30, client_version, 46 random bytes, len(PSK), PSK)`
  - Transmit **client_identity** and **encrypted** premaster secret to the server
  - RSAEncrypt(pubkey$_{Server}$, pms)



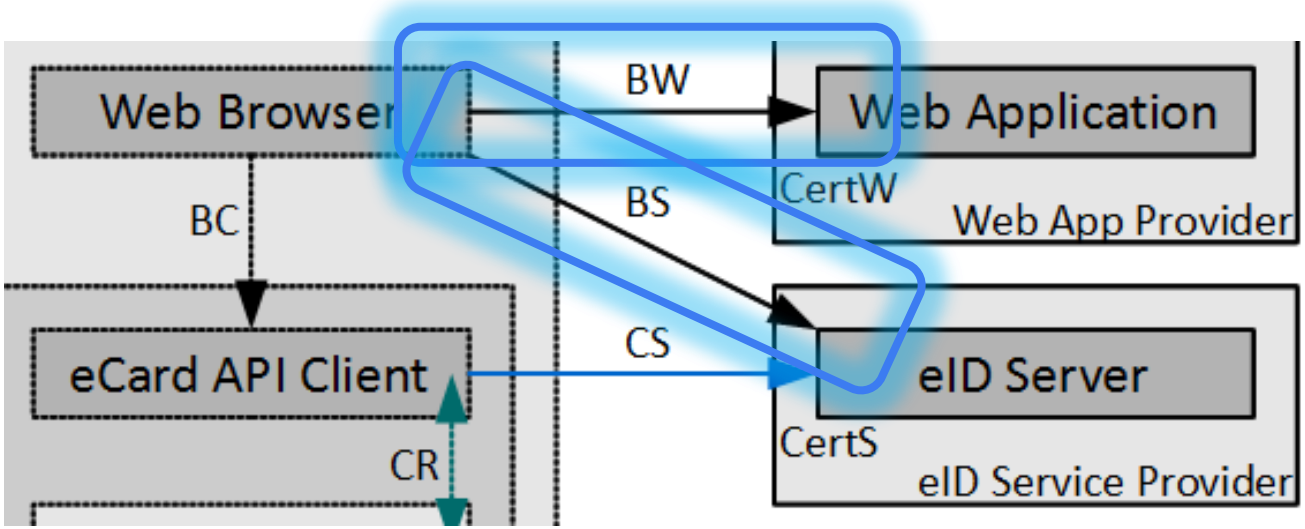
eID authentication: why RSA-PSK?

- PSK is required in order to bind channel *BS* to channel *CS*



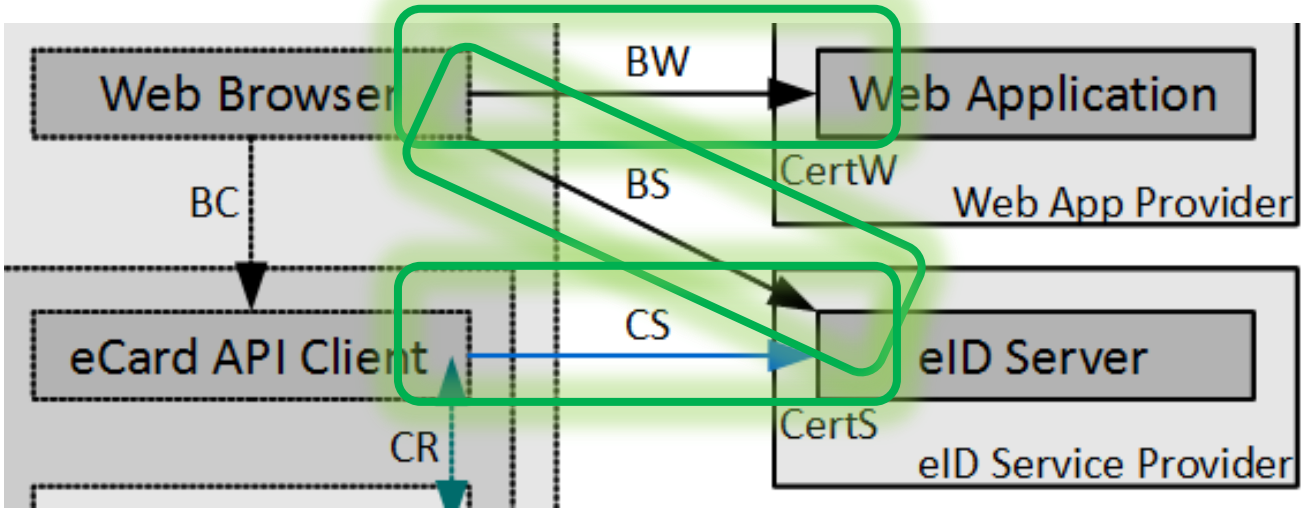
- Why is RSA Server Authentication required?
  - The Terminal Authentication is bound to a certain eID Server!
  - The Hash of the eID Server Certificate is stored in the Terminal Certificate
  - The Hash of the Web Application Server is stored in the Terminal Certificate, too!

Binding eID to certain Servers



- RSA Server Authentication
- Terminal Authentication is restricted to certain Servers
  - eID Server
  - Web Application Server

Extending eID trust to BS and BW



- Sequence of trusted channels is extended to cover the channels *CS*, *BS* and *BW* (in theory)
- Renders Man-in-the-middle attacks useless

Transport Layer Security with Pre Shared Keys

Thank you.

Questions?

Christian J. Dietrich
dietrich [at] internet-sicherheit . de

Institut für Internet-Sicherheit
https://www.internet-sicherheit.de
FH Gelsenkirchen