INSTITUT FÜR INFORMATIK
DER LUDWIG–MAXIMILIANS–UNIVERSITÄT MÜNCHEN

Masterarbeit

Expiration and Revocation within the Certificate Lifecycle Management of the BMW Vehicle PKI

Daniel Leimig

INSTITUT FÜR INFORMATIK
DER LUDWIG–MAXIMILIANS–UNIVERSITÄT MÜNCHEN

Masterarbeit

Expiration and Revocation within the Certificate Lifecycle Management of the BMW Vehicle PKI

Daniel Leimig

Aufgabensteller:	Prof. Dr. Helmut Reiser
Betreuer:	Sophia Grundner-Culemann Tobias Guggemos Dr. Marianne Busch (BMW Group) Lea Himbert-Mordhorst (BMW Group)
Abgabetermin:	26. November 2019

Hiermit versichere ich, dass ich die vorliegende Masterarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 26. November 2019

.....
(Unterschrift des Kandidaten)

Abstract

Certificates are used to secure the communication between software instances, such as a browser communicating with a website or two applications running on different servers. Most of these certificates are issued by a Certificate Authority (CA) of a Public Key Infrastructure (PKI) and have a validity lifetime and a revocation status. An invalid, e. g., expired or revoked certificate can cause a connection to be refused and make a service unavailable. Certificate Lifecycle Management (CLM) comprises actions like issuing, enrolling, invalidating, and reissuing certificates.
After analyzing the specific requirements for the BMW vehicle PKI and evaluating available CLM software on the market, this thesis introduces a concept for a customized CLM for BMW with focus on certificate expiration.
Additionally, a possible future challenge for this PKI is investigated in the context of identity based cryptography (IBC) regarding invalidation, which is Vehicle-to-everything (V2X) communication. V2X defines the communication of vehicles with the traffic infrastructure around them, e. g., another vehicle or a traffic light. IBC uses a publicly known string, e. g., an email address, as public key instead of generating the public key from the private key. Current approaches of securing V2X communication rely on PKIs and thus suffer from similar problems regarding invalidation as current PKIs. Therefore, using identity-based cryptography instead of a PKI in the context of V2X is investigated to find benefits regarding invalidation. The result is that monitoring of individual expiration times of entities and verifying the certificate chain for expiration and revocation are not needed.

Contents

1. Introduction

Certificate Lifecycle Management (CLM) describes the handling of certificates, which is often combined with a general monitoring of a Public Key Infrastructure (PKI). The CLM is a centralized system with a database which allows managing and monitoring of PKI related services and components. Managing and monitoring certificates can reach from issuing a certificate, revoking, renewing or informing if a certificate is expiring to a fully automated service where the certificate automatically gets renewed if it expires. Thus, Certificate Revocation Lists (CRLs) and their CRL Distribution Points (CDPs) and Authority Information Acess (AIA) need to be monitored and managed. Validation of a certificate fails, if the CRLs at the CDP or the Certificate Authority (CA) certificates needed for validating the certificate chain are not available or not valid. However, CLM can also cover managing and monitoring of critical components and services of a PKI like OCSP responders or Hardware Security Modules (HSMs). For example, HSMs can fail because of physical problems due to defective fans and resulting heat problems or having a too high delay by reason of insufficient computing power. In addition, the physical servers the CAs are running on and the CAs themselves can be monitored to ensure the satisfying availability needed for this critical infrastructure.

Another advantage of the CLM is the reduced need for manpower as the manual processes and the amount of work is lower. In a PKI without CLM, the expiring date of a certificate can only be seen in the CA, at the service where the certificate is installed or if another monitoring solution is in use. Not handling the renewal of expiring certificates can lead to various incidents, such as the expired SSL certificate for the Microsoft Windows Azure Cloud which resulted in a worldwide outage of the HTTPS traffic [Mic13], or an expired SSL certificate of Google's Gmail service which lead to problems sending emails for third-party email clients [PCW15]. In May 2019 an intermediate CA certificate of Mozilla expired, which was used for signing all Add-ons for the Firefox browser and caused all Add-ons in Firefox to stop working for three days [hei19]. An expired SSL certificate of Ericsson lead to a nearly 24-hour outage of O2's 3G and 4G data services in December 2018, which affected over 32 million customers [the18].
With the use of CLM these problems can be avoided, as an overview of certificates that will expire in the near future is given and an automated system can be used for, e. g., a certificate renewal. Additionally a better logging is supported, which can be very extensive through the centralized entry of the CLM and can include a monitoring of the services which use the certificates [Cry17a].
Instead of the manual process to request, retrieve and fetch certificates to a system, automated certificate enrollment through the CLM can be used to avoid mistakes like wrongly configured Certificate Signing Requests (CSRs) or imports of certificates in the wrong certificate store [Dig15]. Many CLM solutions increase security with the use of PIN Entry Devices (PEDs) and/or a four-eyes principle [Cry17b], or simplify the management of HSM-Clusters and thus enable a more efficient use of the HSMs [Cry17c].

Challenges for the CLM include, among other things, scaling millions of certificates and the necessity to determine policies for these certificates and to execute and log them. Policies in the context of certificates means all actions performed with the certificates, e. g., for what systems certificates are allowed to get issued or what minimal key length has to be configured. It can be necessary to revoke or renew many certificates on short notice caused by, e. g., a change of the policy, as it happened with the change from SHA-1 to SHA-2, a PKI migration or a compromise of certificates. An example of a necessary change of a policy that had a huge impact was caused by the Heartbleed-Bug. The Heartbleed-Bug is a security bug in the Transport Layer Security (TLS) protocol implementation of the cryptography library OpenSSL, which led to the possibility of reading private data of the systems using the TLS connection. As private keys could get compromised that way all private keys and their certificates of affected systems had to be replaced. [Wik19]
In these cases the reliability of the executed action is very important, but hard to check with many manual tasks if no CLM is in use.

The BMW Group offers digital services of their cars under the name ConnectedDrive. This started in 1994 with the car generation E38 of the 7 Series, which was the first time a BMW car had a GPS navigation system installed. After 1998, BMW Assist was introduced, which featured, e. g., emergency calls, breakdown services and traffic information. In 2003, the iDrive system was introduced in the E65 car generation, which allows the user to get information about the current status of the car, e. g., oil level and tire pressure, and the possibility to control certain functions, e. g., the heating system, using the so called IDrive controller and a monitor. The E65 introduced for the first time an internet connection in the car which can be used to, e. g., read or send emails [7-f08b]. In 2008, the 7 Series of the generation F01 established an integrated web browser which is used with the iDrive Controller. With services like “Map On Mobile” the position of the car can be sent to the owner’s mobile phone, while in case of a crash the service “Intelligent Emergency Call” automatically sends data like position, vehicle identity number, car type, exterior color, and data from the sensors, like type and intensity of the crash, to the BMW Call Center [7-f08a]. With the introduction of the X5 (G05) in 2018 the service “Digital Key” launched which allows opening the car with a Near Field Communication (NFC) supported smartphone and starting its engine.

At BMW, three main data centers, so-called hubs are responsible for the ConnectedDrive services, where all data from ConnectedDrive cars are processed. They are located in USA, Germany and China. Because of new regulations, e. g., due to new privacy protection laws, it is necessary to build further smaller hubs which are located in a specific country to keep the data inside the country.
The communication between the backend of BMW and the cars is encrypted and certificates get flashed on the Electronic Control Unit (ECU) in the manufacturing process of the car. An ECU is a controller which operates an electric system, e. g., the radio ECU of a car, which manages loudspeakers. These certificates are issued by CAs with the help of Hardware Security Modules. Manufacturers of ECUs are generating key pairs on their ECUs and provide a CSR which gets signed at the production plant and flashed to the ECU. This can lead to an End-to-end encryption from the ECU to the backend or even to the services behind the backend. Another aspect provided through the certificates is an internally encrypted communication between the ECUs on the car-internal BUS systems. Additionally, the exchange of an ECU would only be possible by an authorized dealer as the missing or

wrong certificate would prohibit the communication of the manipulated ECU in the car.

Due to further use-cases, like autonomous driving, Software Over-the-Air (SOTA) updates, or the communication with cars of another brand or, e. g., traffic lights, and due to various standards and regulations, the requirements for integrity and confidentiality and thus for certificate handling keep growing. The complexity of managing these certificates increases if certificates expire and need to get renewed. Additionally, if, e. g., the car gets stolen or a backend server gets hacked these certificates have to get revoked. This leads to the point where a Certificate Lifecycle Management solution is needed.

Certificates can be invalidated through expiration and revocation. This thesis will discuss how certificate invalidation can be handled within the Vehicle PKI of BMW and is specified through two Research Questions (RQ):

- RQ1: How can certificate expiration be monitored in the BMW Vehicle PKI?
- RQ2: How can the BMW Vehicle PKI benefit from using an alternative cryptographic mechanism like identity-based cryptography regarding invalidation?

To solve RQ1 the architecture of the Vehicle PKI needs to get probed to discover issued certificates and to determine the risk of expired, revoked or revocation failed certificates and the best way to monitor them. Revocation failed certificates are certificates that cannot be checked for validation, which can occur if no valid CRL is available to retrieve the revocation status of the certificate. Therefore, a Proof of Concept (PoC) for the CLM is run. The scope of RQ1 is defined for the certificates that are used in securing the backend of the BMW Vehicle PKI.
Answering RQ2 requires investigating a use case of future developments of the vehicle PKI on which alternative cryptographic mechanism could be applied. Therefore, one of the main future challenges of invalidation regarding vehicle certificates, which is Vehicle-to-everything (V2X) communication, is analyzed for the current development and evaluated if a different approach, like using identity-based cryptography, can deliver benefits. V2X communication describes the communication of vehicles to other vehicles or, e. g., traffic lights.

The remainder of this thesis is structured as follows: Chapter 2 contains background information about PKIs, invalidation methods and current work on CLM regarding revocation. Besides that, alternatives to the X.509 scheme like IBC are explained. Certificate invalidation requirements at BMW concerning the Vehicle PKI are covered in chapter 3.1. Chapter 3.2 handles the weighting of the requirements, while chapter 4 evaluates the requirements specified in chapter 3 through possible solutions for the Certificate Lifecycle Management based on a Proof of Concept (PoC). In chapter 5 current concepts for securing the V2X communication using PKIs are presented. Additionally, a concept using Identity-based Signatures (IBS) as an alternative cryptographic mechanism is presented.

2. Background and Related Work

This chapter describes the cryptographic standard X.509, the scheme of a PKI and concepts using it. Additionally, this chapter lists alternatives to the X.509 standard.

2.1. X.509

X.509 is a standard in cryptography that was introduced 1988 and defines public key certificates [ITU88]. These X.509 certificates are used in, e. g., the Transport Layer Security (TLS) protocol, which is described in chapter 2.1.2. An X.509 certificate includes information about the identity of the owner, its public key, the certificate's serial number, the validity period, the certificate's issuer, the signature algorithm, the X.509 certificate's version and the signature value. The issuer is the entity that signs the certificate, which means it signs all information of the certificate with the listed signature algorithm and adds the signature to it. X.509 certificates were introduced as version 1 in 1988, were extended to version 2 in 1993, and were introduced as current version 3 in 1996 [Kom04]. Version 2 adds to the above information two optional information, which include an unique identification of the certificate owner, the so-called subject, and the issuer. Version 3 X.509 certificates include optional extensions like the Subject Alternative Name, which specifies one or more alternative names for the subject, e. g., additional IP addresses or URLs. The X.509 standard is often used by PKIs.

2.1.1. Public Key Infrastructure (PKI)

A PKI consists mainly of three components:

- Certificate Authority (CA)
- Registration Authority (RA)
- Validation Authority (VA)

A user generates a private and a public key pair and requests with the use of a Certificate Signing Request (CSR) a certificate by the RA. The CSR includes information which the user wants to have in the certificate, like the subject and the public key. The RA verifies if the user is allowed to request a certificate and if the CSR matches the defined policy of the PKI. If this is successful, the CSR is submitted to the CA, which issues a certificate using the information of the CSR and submits it to the user. This certificate can now be used for, e. g., authentication of a server. If a client contacts a server, the server can show its certificate. To check if a certificate is valid, it has to be checked if the information in the certificate matches to the provided ones of the server, if the validity period of the certificate

is valid, and if the certificate is not revoked. The RA provides the necessary information to validate if a certificate is revoked. This is done by using Certificate Revocation Lists (CRLs), which include the serial numbers of all revoked certificates by this CA. CRLs can be downloaded by the client through defined CRL download locations, which can be included in the optional CDP extension of version 3 X.509 certificates. Additionally, the so-called certificate chain has to get validated, which means that the CA certificate has to get validated as well. The certificate chain is build by checking the issuer's name, decrypting the signature of the certificate with the public key of the CA, and then matching it with the certificate's hash. Instead of using the issuer's name, which is a name string, the Authority Key Identifier (AKI) and Subject Key Identifier (SKI) extensions can be used, which can be, e. g., the hash values of the corresponding keys. PKIs often have more than one CA, which means that all CA certificates up to the Root CA have to be verified. A so-called two-tier PKI consists of a Root CA and one or more Intermediate or Issuing CAs. The Root CA only issues the Intermediate CA certificate(s) and a CRL, and then can remain offline to prohibit getting compromised, while the Intermediate CA issues the certificates for end-entities. Root CAs issue their certificates themselves and often use the optional X.509 basic constraints certificate extension for defining it to be a CA certificate by inserting "CA:TRUE" in the certificate. By using this extension a Root CA certificate can be differentiated to a self-signed end-entity certificate. Self-signed end-entity certificates are not signed by a CA and are used, e. g., for authentication of a server, in contrast to Root CA certificates that are used to sign intermediate CAs and CRLs. PKIs that are operated, e. g., inside companies are called private or internal PKIs. Public or global PKIs issue certificates for, e. g., web servers and are commonly trusted through operating systems and browsers, while a private PKI of a company is not trusted outside the company.
If certificates expire or get revoked, they have to get renewed. While renewing certificates means to issue a new certificate, rekeying is more precise and describes the process of generating a new key pair and obtaining a certificate for them.

2.1.2. Two-Way TLS

TLS is a cryptographic protocol used for securing network communications and the successor of Secure Sockets Layer (SSL). If a client connects to a server via Hypertext Transfer Protocol Secure (HTTPS), TLS is used. The client retrieves the server's certificate and validates it for, e. g., validity time, revocation status, CN/acspSAN and the certificate chain.
If this check is successful and only TLS instead of two-way TLS is used, the client generates a shared secret, encrypts it with the public key of the server and sends it to the server. Now the client and the server can communicate encrypted, as only the server can decrypt the sent encrypted shared secret with its private key.
If two-way TLS is used and the validity check of the server's certificate is successful, the client sends its certificate to the server, which validates the certificate of the client. The server then validates the validity time, checks the revocation status, the chain, and if the certificate is issued by a CA the server trusts. As certificates are public, the server must confirm that the client is the owner of its presented certificate. This is done by sending the client a challenge, which the client has to sign with its private key. The signed challenge is then verified by the server with the client's certificate.

2.1.3. Certificate Pinning

HTTP Public Key Pinning (HPKP) allows to pin certificates to a server to only allow clients with a pinned certificate to connect. Certificate pinning can be used to omit the regular certificate chain validation check by checking the given certificate for revocation, expiration, and if it belongs to the entity. If this is successful then the certificate can be trusted, as it is pinned. If a certificate is being renewed, all applications and machines that store this certificate and use it for certificate pinning, have to replace the certificate with the current one. As a connection could immediately get refused if the old certificate is pinned, the replacement is time-critical to omit downtimes due to refused connections.

Server-sided certificate pinning:
The client connects to the server via TLS and validates the server's certificate, which is submitted in the server's HTTP response including the header information that the server uses HPKP. In case this validation is successful, the client contacts the server again with adding the client's certificate to the message and the server checks if the client's certificate is stored in its certificate store. This means that the server checks if the client's certificate is available in the stored set of pinned certificates. If, e. g., only the certificate of client A is pinned to a server, than client B cannot connect. If an intermediate or Root CA certificate is pinned to the server, all clients who own a certificate of the CA are allowed to connect.

Client-sided certificate pinning:
The client contacts the server over TLS and receives the server's certificate. The client now checks if the server's certificate matches one in the client's certificate store. If it matches a stored certificate, the client trusts the server and connects to it.
If two-Way TLS is enabled, then the server checks the client certificate for validity (see above). If the server uses certificate pinning, it checks if it has the client certificate pinned and lets the client connect or refuse the connection. If neither two-way TLS nor certificate pinning is enabled on the server the connection just gets established.

2.1.4. OCSP Responder

The Online Certificate Status Protocol (OCSP) is used to determine the revocation status of a certificate. The service that offers this functionality is called OCSP responder and runs on a server. The OCSP responder is contacted via HTTP by, e. g., a client that wants to validate the revocation status of a server's certificate. Therefore, the clients sends the serial number of the server's certificate to the OCSP responder. The OCSP responder contacts the corresponding CA or looks up in his own certificate database. The OCSP responder retrieves the CRL from the CA regularly and saves it to his certificate database. By contacting the CA directly the OCSP responder is always up to date, while using its CRL the OCSP responder is not always up to date. For better performance some OCSP responders use a caching function to store OCSP requests for a specific time to enable faster answering. If the OCSP responder uses a CRL and the server's serial number is not listed, the OCSP responder answers the client "good". If the serial number is listed the OCSP responder answers "revoked" and if it cannot get validated "unknown" is sent. Additionally, the OCSP responder can include in its answer the times of the up-to-dateness of its revocation information, when the revocation information gets updated the next time, when the OCSP response was signed, and in the case

the certificate is revoked the time of the revocation. The time stamps of the up-to-dateness and the next update correspond to the times of the underlaying CRL. If next update is not defined, than the OCSP responder does not rely on CRLs and therefore has up to date information.[Int13]

2.1.5. OCSP Stapling

OCSP stapling or TLS Certificate Status Request extension is a method to validate the revocation status of an X.509 certificate. The certificate holder, e. g., a server, contacts the OCSP responder of the CA its certificate was issued. The OCSP response, which contains the revocation status of the certificate, is then stored. The OCSP response cannot be faked by the web server, as it is signed by an OCSP signing certificate or the CA directly. If a client contacts the server using the Certificate Status Request extension in its TLS "client hello" handshake message, the server then returns the OCSP response as well in its "server hello" TLS handshake message. Through receiving the OCSP response the client can validate the revocation status of the server without needing to contact the OCSP responder or to download the CRL. Advantages of this concept are a shorter connection time and therefore a better performance, as the client does not need to contact the OCSP responder or to download the CRL. In case the CRL is several Megabytes large, the transferred data is also significantly decreased, which can be important for devices connected to a cellular network like LTE. An advantage for the PKI is the reduced load of OCSP status requests, which the OCSP responder has to manage and results in needing a less effective OCSP responder with a lower network bandwidth.

2.1.6. CLM for X.509

Certificate Lifecycle Management (CLM) describes handling the processes of issuing, enrollment, renewing, revocation and expiration of certificates. Digicerts describes CLM through four critical components: Installation, Monitoring, Inspection, and Management [Mar15]. Installation describes the process of creating a CSR, retrieving the certificate, and installing it on the machine. Monitoring describes the monitoring process of certificates. Inspection is described as the validation of security configurations, e. g., for SSL. Management describes the process of certificate discovery, a regular renewal of certificates, a short enrollment time of certificates, managing certificates with the help of software, and renewing certificates before expiration.
Gal Alton from Secure-ly states that not managing certificates lead to not knowing the installation location, the expiration date, the responsible contact, the key length, the issuer, and how many certificates were issued [Alt16]. The key features are described as certificate discovery, the import of certificates from a CA to the CLM software's database, alerts for, e. g., renewal, renewal of certificates, and administration through an easy-to-use GUI.
Entrust Datacard specifies CLM as follows [Ent15]: Issuance, where it has to be verified that certificates are issued from a trusted PKI, and the internal approval and administrative oversight has to be established. The inventory, where all information about certificates and responsible departments are logged. A monitor, which continually monitors the inventory. The refresh of certificates, where expiration is tracked and replacement is done before expiration. The retirement of certificates, where expired certificates are marked as not longer in

use or renewed and the revocation of certificates.

2.2. Alternatives to X.509

This section lists some alternatives to the previously described X.509 cryptography standard.

2.2.1. Identity-Based Cryptography

Identity-based Cryptography (IBC) was first introduced by providing an identity-based signature scheme in 1984 by Adi Shamir, the co-inventor of the Rivest-Shamir-Adleman (RSA) algorithm [Sha85]. His idea is to use information that uniquely identifies a user. It is used as its public key instead of creating a public/private key pair, like it is required for the X.509 scheme. This information could be, e. g., the user's email address. It is important to make this information easily available for other users, as they then know the user's public key. This leads to the point that a key or certificate exchange is not necessary if a well-known information is used, as it is with the user's email address.
As no public/private key pairs are used, the exchange of a public key or an X.509 certificate, which includes the public key, is not necessary. Instead, central instances, the so-called Key Generation Centers (KGCs), are needed to give the user, e. g., a smart card, which contains the corresponding secret key, when it first joins the communication network for secure communication with other users.
For signing a message, the sender uses its private key, while for encryption the sender uses the known information of the receiver, which acts as the public key. The receiver decrypts the message with its private key and verifies the signature with the public key of the sender. Instead of exchanging public keys, as, e. g., in the concept of a PKI, it must be shared which information is used for IBS, as the information scheme can be different. For example, different companies may send messages to each other and one company uses the user's email address, while the other company uses a username and the network address. Using, e. g., the email address of a person can become an issue if it changes, e. g., due to a marriage. Then, all parties involved have to be informed that the last name of the email address has changed and the secret key of the newly married person has to be changed. Within an X.509 PKI the key pair of the newly married person does not need to get renewed, only the certificate has to be reissued if the last name is, e. g., part of the CN.

Secret keys must be generated by the Key Generation Center (KGC) by using a secret information, because the entity's information is used as the public key and therefore the entity owns no non-public information that can be used for a private key. Note, that this secret information is the same for all generated keys by this key generation center. In general, any arbitrary string can be used as a public key and the key generation center generates the private key for it. Therefore, private keys must be handed out to the users, which can be a challenge, if the user is not capable of receiving it personally and the private key has to be exchanged over a non-secure connection. Issuing the private key and storing it until the private key is submitted must be done at the key generation center. This effects that all parties must have to fully trust the KGC. This is different to a CA of a PKI that never gets in contact to a private key of a different entity.

However, trust to the issuing CA is also required. Kiltz et al. notice that in a PKI signatures also have an issue, as the certificate and the signature are usually sent together for the first communication. Therefore, a cheating CA can generate a fake certificate for a known key pair and create signatures with it, where it intends to be another person [KNJ08].
Compared to a compromised KGC the impact is less severe, as all keys are definitely compromised within a compromised KGC, as all secret keys are generated there. If a CA is compromised, all certificates are compromised but the key pairs of the users are not compromised and could be further used. This matters if, e. g., the key pairs are stored on a read-only smartcard or an HSM, which can only be initialized and written once. As the private key is not compromised, all prior encrypted data are still encrypted and cannot get decrypted. This is different to a KGC, where all encrypted data can now get decrypted as the private keys are now known. In the concept of Shamir revocation is introduced by generating a new master secret key in the KGC that leads to issue new private keys to all end-entities.

Four main security issues in the identity-based scheme are noted by Shamir [Sha85]:

- The used cryptographic function has to be secure.
- The secret of the key generation center which is used to generate the secret keys has to be protected.
- The identity check of the users before they get their personal secret key has to be reliable.
- Users have to ensure that they do not loose their secret keys, get them duplicated, or get used by a third party.

Adi Shamir mentions that RSA cannot be used as is for this scheme and the paper only describes the signature process but no encryption and decryption processes [Sha85]. One of the first solutions for identity-based encryption, the so called Boneh-Franklin scheme, is presented by Dan Boneh and Matthew Franklin [BD01]. In their approach, an information is suggested as a tuple of an email-address and an expiration date for the private key. The expiration date defines the usage of the key and is, e. g., the current year, which assumes that the key can only be used for this year, while for the next year a new key has to be generated and leads to an annual private key expiration. Instead of the current year another time span could be chosen, e. g., the current day, which forces the receiver of an encrypted message to obtain a new private key every day. An explicit revocation scheme is not mentioned, instead the expiration of the keys is used and no further generation of new private keys is done. This leads to the point that the current key can still be used in its validity period and only after its expiration the user cannot use the key any more. This can be a problem, if the sender does not get informed that the key is, e. g., compromised and the key is getting misused.

2.2.1.1. Hierarchical Identity-Based Schemes

The KGC's function is to generate private keys and to transmit these private keys to the entities over a secure channel. In a scheme that only uses one KGC, there must be an absolute trust in this KGC, as all private keys are generated in it. To spread the duties of one KGC to many so-called domain KGCs allows the sharing of workload, establishes more trust through

more KGCs which than can be run by different parties, and reduces the damage if a KGC gets compromised. This reduced damage is due to only the private keys issued by this KGC are compromised and not all private keys. In a hierarchical identity-based scheme the root KGC only generates private keys for the domain KGCs, which then generates the private keys for the end-entities. However, this procedure has some drawbacks. First of all, the knowledge of the identity is not any longer enough for an encrypted communication or validation of a signature, the information from which KGC the opposites private key was issued has to be now known as well. Some schemes request the root KGC to be known, while others request the domain KGC. A second drawback is the higher complexity due to having many KGCs and generating a scheme which recalls a two-level PKI. Additionally, collusion resistance is a problem. [GS02, HL02] Collusion resistence describes a cryptograhpic property which is to avoid, as it means that two inputs in a cryptographic function deliver the same output. E. g. a hash function is collusion resistant, if it is unlikely that executing the hash function with two different inputs has the same hash value as result [GB08].

2.2.1.2. Multiple Trust Authorities

Multiple trust Authorities are an identity-based scheme which includes multiple KGCs instead of one to reduce the threat of keys getting compromised. Each KGC generates only a part of the private key and sends it to the end-entity, which then computes all parts to its private key. This has as result that only the end-entity is in possession of the private key, unlike to the former introduced identity-based scheme in chapter 2.2.1, where one KGC computes all private keys and is therefore in posession of them. With this scheme all KGCs have to collude if they want to get the private key of the end-entity and for this reason the hazard for key compromise is minimized. [Hes03]

2.2.2. Attribute-Based Cryptography

Identity-based encryption allows to send an encrypted message without needing to exchange a public key certificate or public key by using a known and standardized information, based on the identity of the receiver. Attribute-based Encryption (ABE), introduced by Sahai and Waters defines the identity as a set of descriptive attributes [SW04]. For example, if a sender wants to encrypt a file which only people in his department of his company are allowed to decrypt, then the file could be encrypted to the identity “company name”,“department name”. Only users whose identities have these two attributes can decrypt the file. This leads to needing a protection of collusion attacks as it has to be prohibited that a group of users has the ability of combining their keys in such a way that they can decrypt something they are not capable of alone. To face this problem the construction of private keys as a set of components for each attribute is suggested. An expiration scheme with validity times connected to attribute-based signatures is presented by Tate and Vishwanathan, which allows the generation of time-based keys [TV15]. Indirect revocation through expiration is therefore possible, which means that entities that get revoked can use their key until expiration, but are not capable of requesting a new key afterwards.

2.2.3. Web of Trust

Web of trust is a decentralized scheme, where trust between entities is established by signing the keys of each other. Each entity generates a public and a private key, where the public key gets, e. g., uploaded to a key server, from which other entities can download it. If an entity trusts the downloaded public key of another entity through, e. g., verification by a fingerprint provided by the owner entity, it signs the public key with his private key. This signature then gets uploaded to the key server. If an entity A trusts the entity B that signed the public key of entity C, then entity A trusts entity C as well. This concept relies on signing only trusted and verified public keys. If an entity A signs a public key of an entity B without validating the authentication of the key and the key is compromised, then every entity that trusts entity A trusts the compromised key of entity B.
This concept is used in Pretty Good Privacy (PGP) and OpenPGP. A revocation scheme for OpenPGP is presented in the RFC2440 [JCT98]. Revocation is done by signing the key that needs to get revoked with a so-called key revocation signature. It is possible to add the new ID in this process that tells what new key is used for this entity. This is then called the revocation certificate which is uploaded to a key server. If a user wants to download the revoked key, he notices the revocation certificate, knows that the key is revoked, and downloads the new key.

3. CLM requirements of the BMW Vehicle PKI

This chapter defines the requirements for the CLM of the BMW Vehicle PKI in section 3.1, while the requirements are weighted for importance and scored for the level of fulfillment in chapter 3.2.

3.1. Certificate Invalidation Requirements

BMW uses an internal Public Key Infrastructure for issuing certificates to its vehicles and the services communicating with them, which is called BMW Vehicle PKI. Beside the certificates from the Vehicle PKI, certificates from a second private PKI, a global PKI, and self-signed certificates are used to secure communication between servers of the backend of the BMW Vehicle PKI. The certificates used to secure the communication within the backend of the BMW Vehicle PKI are the scope of this thesis for Certificate Lifecycle Management. An automation of the certificate renewal process is due to, e. g., using certificate pinning, not in scope of the current plans for CLM. The financial aspects and the companies reputation are not covered by this thesis. The pricing models of the vendors differ much as some models are based on the exact amount of certificates, others offer defined certificate amounts for their pricing and some do not count the amount of certificates at all. Additionally, maintenance volume and costs are very different as well. This results in performing an invitation to tender for the vendors that are potential candidates after the PoC is executed to make the pricing comparable. The companies reputation are not rated through the PoC as this is done by the purchasing department.

The requirements for the Certificate Lifecycle Management solution are ordered into six domains with several requirements each:

- K.O. Criterion
- Discovery
- Certificate Inventory
- Monitoring and Alerts
- Software Operations
- Confidential Requirements

The domain “K.O. Criterion” is needed, as there is one must have requirement. “Discovery”, “Certificate Inventory”, and “Monitoring and Alerts” are often used for categorization of functionalities in CLM, as described in chapter 2.1.6. The discovery domain lists all

requirements that affect certificate import and discovery. The certificate inventory is the part of the Graphical User Interface (GUI) where all certificates imported into the CLM software are listed. The certificate inventory allows to get an overview of the certificates and if, e. g., the policy of the PKI changes due to an increase of the key length, it is possible to achieve the information which certificates do not fulfill this new policy and have to get reissued. The domain Monitoring and Alerts contains the requirements that affect, e. g., monitoring certificates for expiration and sending an alert if an certificate expires. Although "Software Operations" is not clearly named, it describes the requirements in this domain well, as this domain handles, e. g., user authentication. The "Confidential Requirements" domain includes all confidential requirements, which are special requirements for the BMW Vehicle PKI and are listed in the confidential appendix.

3.1.1. K.O. Criterion

This domain lists one must have criterion which has to be fulfilled.

KR1 On-premise Installation
The CLM software has to be installable and runnable on-premise in the BMW infrastructure without needing an internet connection. An off-premise solution is not sufficient as this is a must have requirement.

3.1.2. Discovery

This domain deals with the requirements which are affected by the process of discovering and importing certificates into the CLM software.

DR1 Manual import of certificates through the GUI
This functionality enables a user to import certificates through the GUI and upload them directly using a browser. This is an important functionality as it allows the user to import certificates without the need of copying the certificates on the CLM server or use a tool to import certificates.
A certificate should be importable from a local folder of the user's operation system (Windows, Unix and Linux) that has a Universal Naming Convention / Uniform Naming Convention (UNC) path directory via Common Internet File System (CIFS)/Server Message Block (SMB), and from a Network File System (NFS) directory. Multiple certificates and non password protected ZIP-archives should be importable with one step. If a folder is selected, all certificates in this folder should get imported, including certificates in subfolders of this folder. Certificates should be supported in Distinguished Encoding Rules (DER) and Privacy-Enhanced Mail (PEM) format with the file extensions crt, cer, pem and der. CA certificates should be importable as well.

DR2 Automated import of certificates from file shares
This process is needed to import certificates from a local directory on Windows and Unix/Linux, a UNC path directory via CIFS/SMB, and from a NFS directory through a process, which is configured by inserting the path to the certificate(s). For this process the certificates have to be accessible to the CLM software as the import does not happen using the user's browser like in the requirement "DR1 Manual import of

certificates through the GUI" and therefore allows the import of certificates which cannot be directly uploaded through the user's browser. These locations can be, e. g., a folder on a system which is not accessible to the user's computer. This accessibility can be achieved by using so called agents. These agents are installed on the system from which certificates should be discovered from and transfer the discovered certificates to the CLM software to import them into the database. The complete folders should get imported including subfolders. Non password protected ZIP-archives should be importable as well. Certificates should be supported in DER and PEM format with the file extensions crt, cer, pem and der. CA certificates should be importable as well.

DR3 Import a list of file shares which shall be discovered

A file, which contains a list of file shares which shall be searched for certificates, can get imported and executed. The credentials to access the file shares should be stored in a password vault of, e. g., the CLM software. This functionality improves the handling of the CLM software and is less error prone than changing everything in the GUI. Changes can occur quite frequently, as there are many distributed CAs located on different machines.

DR4 Scheduled tasks for certificate imports from a file share

An automation of the requirement "DR2 Automated import of certificates from file shares" is needed to reduce manual effort for the task running automatically on a specified interval or date, which has to be configurable.

DR5 Import certificates from an IP address

With a given IP address, all TLS certificates should be imported that are bound to it. Single or multiple ports to which the certificates are bound to should be configurable for scanning. Scanning only specified ports reduces network traffic, time and is less conspicuous than scanning a broad network section that might trigger an network intrusion detection software. However, a complete port scan should be offered additionally for systems that are unknown or not well documented. If there are multiple certificates sent by the server during the TLS handshake, e. g., the server certificate and the intermediate Certificate Authority certificate, all certificates should be imported.
The same requirements must apply for scanning an IP address range and for servers accessed by its Fully Qualified Domain Name (FQDN). To contact the server by its FQDN the CLM software must support the usage of a Domain Name System (DNS) server.
Scanning a server this way means that there is no authentication for the process required. This is different from, e. g., two-way TLS where the server always sends its certificate(s) bound to its addressed port before the client is requested to send its certificate to enable the secure communication channel.

DR6 Import a list of servers which shall be discovered

This process imports a file, e. g., an xlsx or csv file, which consists of a list with the IPs or FQDNs of the servers and the corresponding ports from which certificates shall be discovered. This list allows a more dynamic automation of adding and deleting servers.

DR7 Scheduled tasks for certificate imports from an IP address

An automation of the requirement "DR5 Import certificates from an IP address" is needed to reduce manual work.

DR8 Import certificates from Java Key Store

As some applications that run on Linux/Unix store their certificates in Java Key Store (JKS), an import of certificates from the JKS has to be supported as well as password protected JKSs. The credentials to access the JKS should be stored in the password vault of the software.

DR9 Categorize on import

As many certificates of different CAs are in use by different teams/departments at BMW, an automated categorization into certificate groups after their discovery has to be supported. Certificate groups are described further in the requirement "CIR5 Certificate groups with individual parameters". The categorization can be done by using the information of the certificate or from the place where the certificate was discovered from, e. g., the server name or file share. These certificate groups are then needed for individual alert schemes per team/department. This requirement is necessary for the operators of the CLM software to not just only know which certificates expire, but also to delegate the job to replace the certificates to the team which is responsible for them. Therefore, it is important to know from which system the certificate was discovered from and then record the responsible team for that system in the CLM software.

DR10 Customizable interface

To integrate the CLM into the issuing process of the two BMW PKIs, a process has to be established which allows to automatically import relevant issued certificates into the CLM software. An interface, like, e. g., an application programming interface (API) could be used for this process. The benefit of an API is that the CLM can be contacted directly without a detour using a file share or the need to implement a new custom interface.

3.1.3. Certificate Inventory

All requirements dealing with the certificate inventory are listed here.

CIR1 Dashboard

A dashboard is needed to get a quick overview of the current status of the CLM. To support multiple teams, each team should have their own dashboard showing only relevant information. To achieve this, configurable widgets on the dashboard have to be supported. These widgets should display the following overview information:

- Hash algorithms in use
- Signing algorithms in use
- Key lengths in use
- CA certificates in use and the amount of their issued certificates
- Amount of self-signed certificates
- Amount of certificates in one or more specified certificate groups
- Certificates that expired and will expire in a definable time
- Revoked certificates

- Status of the OCSP responders and CDPs

CIR2 Configurable Columns

Certificates store itself information, e. g., the subject's name, and additional information through the CLM software is connected to it. As an example, metadata like the responsible department and the system the certificate was discovered from, can be named. To display the most relevant information in the inventory GUI, the columns of the inventory should be configurable. This means to hide non relevant information and show the important ones. As the required columns to be displayed change depending on the use, this has to be dynamically adjustable. Besides, the columns should have a changeable order of ascending or descending.

CIR3 Search certificate inventory

It should be possible to search for certificates in the certificate inventory using all information the certificate contains, like the CN, basic constraints, or the SAN. Operators such as "EXIST", "NOT EXIST", "LIKE", "NOT LIKE", "STARTS WITH", "STARTS NOT WITH", "EQUALS", and "NOT EQUALS" should be supported. Furthermore it should be feasible to combine search parameters with "AND", "OR", and use parentheses. An option to save the search would be useful.
The list of certificates should be sortable by each column for, e. g., alphabetical order ascending or descending.

CIR4 Grouping certificates per CA

Due to using a high number of certificates of different CAs, it important to display all certificates issued by each CA separately. This is a functionality which can get high importance if a CA is getting replaced, is close to expiration or should be revoked and therefore all certificate owners need new certificates from another CA. This can be achieved by, e. g., supporting certificate groups where each CA acts as its own certificate group. A search of the certificate inventory is sufficient, but the option to save the search has to be given, as it is error-prone and time-consuming to create a new search pattern every time. This functionality should not only work for certificates of their issuing CA, but also for the certificates of the whole tree underneath the issuing CA. E. g., there should be a group listing all certificates issued by the Root CA and then a second group should exist, where all certificates issued by the Root CA and their Sub CAs are listed.

CIR5 Certificate groups with individual parameters

Due to the usage of the CLM software by different teams and departments at BMW each team or department should have the possibility to view only their certificates. Filtering could work with an individual parameter which could be a Metadata field including the team's or department's name, or matching parts of the certificate Subject's Common name (CN) or Subject Alternative Name (SAN). This defined certificate group has to be accessible for other users.

CIR6 Support of self-signed certificates

As self-signed certificates are used by BMW, the CLM should be able to list them separately in the certificate inventory and the dashboard, without assuming them to be as important as Root CA certificates. The distinction can be made, e. g., by the certificate's basic constraint extension that does not specify this certificate as a CA

certificate.

CIR7 Support of private certificates

The BMW Vehicle PKI is a private PKI and therefore issued private end-entity certificates must be supported.

CIR8 Support of private CA certificates

The BMW Vehicle PKI is a private PKI and therefore private CA certificates must be supported. Supported means that the CLM software should provide a functionality that offers the private CA certificates to be clearly recognizable and listed as CA certificates.

CIR9 Support of cross-signed end-entity certificates

Cross-signed end-entity certificates are used by BMW and therefore have to be supported. That means that they need to be correctly listed in the inventory with the correct CAs. Cross-signed certificates should be marked, so that they are quickly recognized as a cross-signed end-entity certificate.

CIR10 Display certificate chain

At the backend of the BMW Vehicle PKI are certificates from many different CAs and independent PKI trees in use. For better administration and overview purposed, the Certificate Chain up to the Root CA should be viewable. Cross-signed end-entity certificates should be supported as well. The software should build the chain with imported CA certificates no matter if the CA certificates were imported before or after the end-entity certificate. An automatic download of the CA certificates from the certificate's AIA extension is not required.

CIR11 Renewed certificates

When renewed certificates get imported, a logic is needed to handle the old and the renewed certificate. This could be done with a certificate history, which lists all predecessors and automatically moves the old certificate including its history from the inventory to the history of the new certificate.

Another approach would be to send an automated warning if a certificate gets imported with the same public key or the same CN or SANs. An automated deletion of the old certificate is too risky as it may not be replaced on all systems and a parallel addition without notification could lead to errors if the certificate is pinned.

If certificates are moved to the history section, it should be possible to manually do and undo this. This is required because of, e. g., a misclick.

CIR12 Special functionality for pinned certificates

For pinned certificates, a logic is needed that lists all systems that are affected by an invalidated or replaced certificate, as the certificates have to be synchronously replaced on several systems. A proposal for this requirement would be comment fields attached to the certificate, where all affected systems are listed, e. g., with the comment "System A is the owner of the private key, System B to E have the certificate pinned".

3.1.4. Monitoring and Alerts

This domain lists all requirements concerning monitoring and alerts.

MAR1 Check validity status of certificates

All certificates including CA certificates have to be checked for time validity and revocation status. If the certificate is about to expire or the revocation status turns to "revoked" a notification/alert has to be sent and the certificate should be graphically marked in the CLM software.

MAR2 Individual alert schemes

As the replacement and preparation time for certificate renewal differs significantly, individual alert schemes for certificates, certificate groups and CA certificates are needed. These alert schemes determine the time before an alert is sent prior to certificate expiration. The reason is that the rekeying of an Intermediate CA certificate has to be well planned and organized, as all system that use the certificate chain are affected and need to replace the certificate, while an end-entity certificate of a small system can just be replaced on the same day of the warning.

There should be multiple configurable warnings before a certificate expires, e. g., nine months, three months, one month, one week and two days for an Intermediate CA. For CRLs and OCSP tickets there should also be an individual alert scheme with multiple warnings. E. g., issuing a new CRL of the Root CA needs more time than to publish a new intermediate CA CRL, as the same security procedures as for a root key ceremony have to be executed.

MAR3 Inspection of the certificate chain

If certificates get imported with no previous imported chain an alert should be sent. This can help detecting missing/updated CA certificates in the certificate chain. For Root CA certificates and self-signed certificates an exception has to be made, as these certificates do not have a chain.

MAR4 Support OCSP responder monitoring

The PKI uses OCSP responders, whose availability and the validity of the generated OCSP tickets has to be monitored. The URLs of the OCSP responders should be entered in the GUI or imported through a list.

MAR5 Support OCSP stapling tickets

Some servers use OCSP stapling, which leads to a need for monitoring these OCSP stapling tickets for availability and validity. To monitor these OCSP stapling tickets, the CLM software has to contact the servers with an included Certificate Status Request extension in the TLS handshake.

MAR6 Support CRL monitoring

The time validity and availability of the CRLs have to be monitored. The URLs of the CRL download locations should be entered in the GUI or imported through a list. Checking the CRL download locations from the CDP extension of certificates is not sufficient enough, as this is an important infrastructure part from the PKI. An explicit check for time validity and availability for given CRL download URLs is therefore required.

3.1.5. Software Operations

Requirements regarding software operations of the CLM software are listed here.

SOR1 Authentication

As many users will use the CLM software, it has to offer support for the Lighweight Directory Access Protocol (LDAP) to access Active Directory (AD) groups and users. LDAP is the protocol used for contacting the AD, which is a directory service for administrating, e. g., users and computers.

SOR2 Fine-granular authorization concept

With the software being used by many different teams/departments, a fine-granular authorization management is needed to fulfill the need to know principle.
This requirement takes advantage of the requirement "CIR5 Certificate groups with individual parameters", as certificate groups are assigned to a user role. Three user roles are required: An admin user role for the admins of the CLM software, an operator user role for managing the user group and a user role for normal users. It should be possible to assign different roles to a user.
As an example the following model could be used:

These functionalities should be limited to a kind of administrator role:

- Create user roles
- Add new Active Directory users or groups to the admin user role
- Create certificate groups
- Modify the alert settings of OCSP and CDP locations
- Modify the log settings and the logs
- View Dashboard with all certificates
- View all certificates

These functionalities should be only available for the operators of a certificate group:

- Add new Active Directory users or groups to the user role "operator" and "user" of a certificate group
- Import/Add certificates to a certificate group
- Define alert settings for expiration of the certificates of a certificate group
- Create Dashboards and modify Widgets on a Dashboard

Users of the role "user" should only be allowed to see the certificates of their certificate group in the inventory and dashboard.

SOR3 Configurable GUI

The GUI should be configurable to display only relevant functions, menus and information according to the user roles. This means that buttons that contain blocked functionality to certain users are not visible to them. A simplified GUI reduces the effort to train the people who are supposed to use the CLM software. This requirement can overlap with the requirement "SOR2 Fine-granular authorization concept", as some

software automatically disable buttons if the functionality behind it are prohibited.

SOR4 Usability
The software is expected to be in use by several people of different departments, so it should be usable with a short introduction and a custom operation manual written by BMW. Therefore an intuitive usability with, e. g., help notifications is required.

SOR5 Documentation
As the configuration of the CLM is quite complex, a comprehensive user documentation and operations manual is needed.

SOR6 Installation and Maintenance
The software has to be integrable into the BMW infrastructure. Therefore, the used database and the operating system has to be supported by one of the responsible departments of BMW. If the operating system or the database cannot be installed and maintained by a standard procedure at BMW, individual processes to run and maintain the software have to be considered.

SOR7 Backup
The software database and configuration needs to be backed up at defined intervals. This backup mechanism has to be integrable into the BMW infrastructure and standard processes. If this is not possible an individual method/workflow has to be considered.

SOR8 Events and Alerts
The alerts and event notifications of the software have to be integrable into the BMW ticket event handling process. As the ticket handling in the BMW IT is done with a custom solution, probably no software will be integrable out of the box and a custom solution needs to be developed. The BMW ticket event handling offers a REST API, where the CLM software has to be attached to. Using SMTP is not a solution as it is not considered reliable enough.

SOR9 Performance
The Performance of the software has to be good, there should be no lags or long database access times. For example the response time for a search should not be longer than ten seconds and the time to load a page should not be longer than one second.

SOR10 Software quality
The software should be reliable and well tested, e. g., a penetration test should have been executed. A qualified support model should be available and for critical defects bugfixes should get delivered in a short time span.

3.1.6. Confidential Requirements

Appendix A.1 lists all confidential requirements that are special to the BMW infrastructure.

3.2. Weighted Scores for the Requirements

This section lists the weighted scores for the requirements set up in chapter 3.1. These weighted scores are used to rank how much a CLM solution meets the requirements and

were set up by collaborating with the responsible operators of the BMW Vehicle PKI. The requirements are weighted for importance and defined by points for the stage of fulfillment. The total score reached for each requirement by the PoC participant is the weight of the requirement multiplied with the achieved points for it.

The weight is defined as follows:

Weight	Meaning
1	Very low importance
2	Low importance
3	Important
4	High importance
5	Very high importance

The points to achieve are more precise explained in each weighted score, while the base structure is defined as follows:

Points	Meaning
0	Functionality is completely unsupported.
1	Functionality is rudimentary supported.
2	Functionality is mostly supported.
3	Functionality is completely supported.

For sorting the weighted scores of the requirements, the same domains from chapter 3.1 are used:

- K.O. Criteria
- Discovery
- Certificate Inventory
- Monitoring and Alerts
- Software Operations
- Confidential Requirements

3.2.1. K.O. Criterion

This domain lists one must have criterion which has to be fulfilled.

KR1 On-premise Installation

As this is a must have criterion, there exists no weighted score. Not fulfilling this requirement has as result that the software is not a solution approach for the CLM of the BMW Vehicle PKI.

3.2.2. Discovery

This domain deals with the weights and points of the requirements which are affected by the process of discovering and importing certificates into the CLM software.

DR1 Manual import of certificates through the GUI

Weight:

4, as this is a needed feature to directly import certificates.

Points:

0: Certificates cannot get imported through the GUI.

1: Only the possibility to import some specific certificates is offered, e. g., only DER formatted certificate of the user's home directory can get imported.

2: The import functionality supports, e. g., DER and PEM formatted certificates from all file shares, but does not support ZIP archives.

3: All functionalities are supported as described in the requirement.

DR2 Automated import of certificates from file shares

Weight:

5, as this will be a highly used functionality.

Points:

0: No option to automatically import certificates is available.

1: Rudimentary functionality, e. g., only the possibility to import one DER formatted certificate of the user's home directory.

2: More but not all required functionalities are supported. This can be, e. g., the support for importing DER and PEM formatted certificates from all file shares, but no support of ZIP files.

3: All functionalities are supported as described in the requirement.

DR3 Import a list of file shares which shall be discovered

Weight:

1, as this will only improve comfort in a minor way and can lead to a better handling of configuring the certificates' locations.

Points:

0: Certificates from file shares cannot be imported.

1: Certificates can only be imported if they are, e. g., in a specific folder.

2: Complete functionality except of, e. g., ZIP archives.

3: All functionalities are supported as described in the requirement.

DR4 Scheduled tasks for certificate imports from a file share

Weight:

5, as an automation of importing certificates is necessary with many different locations and certificates.

Points:

0: Certificates from file shares cannot be imported automatically.

1: If the information of the task is running or has been executed is not clear and the current status cannot be retraced, or if the results of the execution are not logged properly.

2: If, e. g., the tasks execution time can only be configured very imprecise.

3: All functionalities are supported as described in the requirement.

DR5 Import certificates from an IP address

Weight:

5, as discovering and importing certificates from a server over IP is highly relevant.

Points:

0: Certificates cannot be imported through a TLS handshake.

1: If, e. g., ports cannot be specified regarding which one/ones to use or no definition of a port range is possible.

2: If, e. g., only IPs, IP ranges and ports can be specified, but no FQDNs.

3: All functionalities are supported as described in the requirement.

DR6 Import a list of servers which shall be discovered

Weight:

3, as the servers change often, a functionality to import a versioned list of the servers improves reliability.

Points:

0: No functionality is offered to import a list of file shares that shall be discovered.

1: Less functions are available in the list than in the discovery task definition.

2: If, e. g., the tasks execution time can only be configured very imprecise.

3: All functionalities are supported as described in the requirement.

DR7 Scheduled tasks for certificate imports from an IP address

Weight:

5, as an automation of importing certificates is necessary with many different locations and certificates.

Points:

0: An automated import of certificates through a TLS handshake is not available.

1: If the information of the task is running or has been executed is not clear and the current status cannot be retraced, or if the results of the execution are not logged properly.

2: If, e. g., the tasks execution time can only be configured very imprecise.

3: All functionalities are supported as described in the requirement.

DR8 Import certificates from Java Key Store

Weight:

1, as there are not many relevant JKS in use that have to be monitored.

Points:

0: No support for importing certificates from a JKS is offered.

1: If, e. g., only certificates in PEM format from the JKS get imported or it does not support password protected JKS.

2: If the JKS needs to have a special configuration.

3: All functionalities are supported as described in the requirement.

DR9 Categorize on import

Weight:

5, as this is a highly needed functionality to deal with many different teams and departments and therefore the use of certificate groups and the automated sorting into it is needed.

Points:

0: An automated categorization for imported certificates is not available.

1: Sorting into certificate groups after categorization is not fully supported.

2: If, e. g., the sorting categories are limited to predefined categories or cannot be specified enough.

3: All functionalities are supported as described in the requirement.

DR10 Customizable interface

Weight:

5, as this is a very important part to import certificates and add additional information, e. g., the responsible team/department, into the CLM.

Points:

0: No API or process to integrate the issuing process of the two BMW PKIs is possible.

1: It is only possible to import certificates through the detour of an import over a fileshare.

2: An API is available but not well documented or does not offer enough functionality.

3: A well documented API is available with the necessary commands for a certificate import and the later categorization into certificate groups.

3.2.3. Certificate Inventory

This domain deals with the weights and points of the requirements to the certificate inventory.

CIR1 Dashboard

Weight:

3, as this is an important feature to get a quick overview of the current CLM status.

Points:

0: There is no Dashboard available.

1: The dashboard only displays few information and is not configurable

2: The dashboard displays nearly all or all information from the requirement but it is not configurable.

3: All required information can be shown and adjusted in the configurable dashboard so that each team/department is shown only their relevant information.

CIR2 Configurable Columns

Weight:

4, as there are many different certificates and therefore information available in the certificate inventory, sorting of them is important to get the relevant certificates.

Points:

0: Columns are not configurable and very few information is available as column.

1: Columns are not configurable and only the most relevant information is available as column.

2: Columns are configurable, but not all information can be used as column.

3: Columns are configurable and all required information can be displayed.

CIR3 Search certificate inventory

Weight:

4, as there are many very different certificates and getting some of them together in an overview is important. Additionally, the aspect of manual renewal of certificates and touching one application or system can lead to the renewal of another certificate for this application or system. This is the case if another certificate is expiring as well on this application or system. Therefore, a configurable search

with a high degree of freedom is important.

Points:

0: The search supports only a limited number of information of the certificates.

1: The search supports all information of the certificates.

2: The search can be highly adjusted but is, e. g., cannot not saved.

3: All functionalities are supported as described in the requirement.

CIR4 Grouping certificates per CA

Weight:

4, as there are many CAs in use and the possibility to get a quick overview has to be given.

Points:

0: There is no option to display only certificates of one specific CA or the certificates of their Sub CAs.

1: Only certificates of one CA can be displayed, but not of their Sub CAs.

2: A search for certificate groups for the issuing CA and their Sub CAs is provided, but it cannot be saved.

3: A search can be saved or certificate groups are available that list all certificates of the CA and their Sub CAs.

CIR5 Certificate groups with individual parameters

Weight:

4, as grouping by CAs is not sufficient enough for different teams to see only their certificates.

Points:

0: No Metadata or certificate groups are supported.

1: Metadata for certificates are supported, but cannot be used as a search parameter.

2: Metadata are only rudimentary supported, e. g., cannot be filled with enough information, but are usable as search parameter.

3: All functionalities are supported as described in the requirement.

CIR6 Support of self-signed certificates

Weight:

5, as self-signed certificates are in use at BMW.

Points:

0: Self-signed certificates cannot be imported.

1: Self-signed certificates are processed wrong, e. g., they are handled as Root CA

certificates.

2: Self-signed certificates are not clearly listed as self-signed, but otherwise fully supported.

3: Self-signed certificates are treated like end-entity certificates, but clearly marked as self-signed.

CIR7 Support of private certificates

Weight:

5, as private certificates are the main scope through the BMW PKI being a private PKI.

Points:

0: Private certificates cannot be imported.

1: Private certificates can be imported, but can not be checked for expiration or revocation.

2: Private certificates are not fully supported, e. g., a functionality supported by the CLM software does not work for them that would otherwise work for public certificates.

3: Private certificates are fully supported.

CIR8 Support of private CA certificates

Weight:

5, as the PKI of BMW is a private PKI.

Points:

0: Private CA certificates cannot be imported.

1: Private CA certificates can be imported, but can not be checked for expiration or revocation.

2: Private CA certificates can be imported, but are handled as end-entity certificates.

3: Private CA certificates are clearly listed as CAs.

CIR9 Support of cross-signed end-entity certificates

Weight:

1, as these certificates are not widely used at BMW.

Points:

0: Cross-signed end-entity certificates are not supported, because e. g., only one certificate for the CN can be imported.

1: Cross-signed end-entity certificates can get imported, but are not treated right, e. g., moved to a history.

2: Cross-signed end-entity certificates are fully supported, but not marked for better identification.

3: Cross-signed end-entity certificates are fully supported and can be detected clearly.

CIR10 Display certificate chain

Weight:

2, as this functionality is for visualization only.

Points:

0: The chain cannot be viewed.

1: The chain is not available for all certificates, e. g., the chain cannot be build if the certificate extension "Authority Key Identifier" is not set in the certificate.

2: The chain is only viewable for end-entity certificates and not for CA certificates.

3: All functionalities are supported as described in the requirement.

CIR11 Renewed certificates

Weight:

3, as a logic is needed to handle renewed certificates.

Points:

0: A problem occurs with renewed certificates or the functionality for renewed certificates does not behave like described in the requirement and therefore can lead to problems.

1: A renewed certificate is treated like a normal certificate.

2: A functionality for this topic is available, but does not fulfill all requirements, like a history where certificates cannot get automatically moved into it.

3: A functionality for handling renewed certificates like describe in the requirement is available.

CIR12 Special functionality for pinned certificates

Weight:

4, as a logic is needed to handle pinned certificates.

Points:

0: No functionality is available to address this requirement.

1: E. g. a functionality to make notes is available, but not assigned to the certificates directly.

2: E. g. a comment field assigned to the certificates is available, but with limited usability. An example would be very limited character length or it cannot be named individually because it is for general notes.

3: A sufficient functionality to address the requirement is available.

3.2.4. Monitoring and Alerts

This domain deals with the weights and points of the requirements concerning monitoring and alerts.

MAR1 Check validity status of certificates

Weight:

5, as monitoring certificates for time validation and revocation is the main purpose of the CLM.

Points:

0: Functionality is completely unsupported.

1: Certificates are only checked for revocation or time validity, but not both.

2: Certificates are checked for time validity and for their revocation status, but there is, e. g., no alerts or no highlighting in the certificate inventory if certificates are about to expire, already expired or are revoked.

3: A sufficient functionality to address the requirement is available.

MAR2 Individual alert schemes

Weight:

5, as different alert schemes are necessary for individual time intervals, which are needed for certificate issuing and replacement.

Points:

0: Only a predefined time before certificates expire and an alert is sent is available. This time affects all certificates, no separation of certificate types or groups are possible.

1: The time before certificates expire and the alert is sent can be configured with custom values for different types like end-entity, CA certificates, and certificate groups. Multiple warnings are not possible.

2: Additionally to the former point multiple warnings are available for the certificates. However, multiple alert schemes for CRL and OCSP monitoring cannot be configured.

3: All functionalities are supported as described in the requirement.

MAR3 Inspection of the certificate chain

Weight:

1, as it should rarely happen that an imported certificate used in the BMW infrastructure has an unknown CA.

Points:

0: This functionality is completely unsupported.

1: This functionality is rudimentary supported. An example would be if only a hint is displayed but no alert can be sent.

2: Only the issuing CA of the certificate is checked and not the full chain.

3: All functionalities are supported as described in the requirement.

MAR4 Support OCSP responder monitoring

Weight:

5, as this is a critical infrastructure part of the PKI, the OCSP locations have to be monitored explicitly.

Points:

0: The software does not offer the possibility to monitor a given OCSP responder URLs.

1: OCSP responders are only checked for availability and not for time validity of an OCSP ticket.

2: The response message of OCSP responders is checked for validity, but no alert is sent if, e. g., the time validity is wrong.

3: All functionalities are supported as described in the requirement.

MAR5 Support OCSP stapling tickets

Weight:

3, as not all servers use this functionality, but monitoring it is still important.

Points:

0: No functionality for monitoring OCSP stapling tickets is provided through the software.

1: It is checked if an OCSP stapling ticket is provided through the TLS handshake, but no further verification for, e. g., time validity is done.

2: The validity of the server's OCSP stapling ticket is checked, but no warning is sent if, e. g., the time validity is wrong.

3: All functionalities are supported as described in the requirement.

MAR6 Support CRL monitoring

Weight:

5, as the CRL locations are an essential part of the PKI.

Points:

0: CRLs cannot be explicitly monitored.

1: CRL download locations are only checked for availability and not for time validity.

2: CRLs are checked for availability and validity, but no alert is sent if the CRL is not valid anymore.

3: All functionalities are supported as described in the requirement.

3.2.5. Software Operations

This domain deals with the weights and points of the requirements concerning monitoring and alerts.

SOR1 Authentication

Weight:

4, as an Active Directory simplifies user authentication.

Points:

0: There is no support of an Active Directory and therefore user management has to be done in the CLM software.

1: Some functionalities are not available for AD users, e. g., administrative privileges.

2: Only AD users are supported but no AD groups.

3: All functionalities are supported as described in the requirement.

SOR2 Fine-granular authorization concept

Weight:

4, as many users use the CLM software, different access levels are needed.

Points:

0: There are only one or two user roles available, or the user roles are predefined.

1: The customization possibilities of the user roles are very limited.

2: A configurable authorization concept is available but does not fulfill all functionalities.

3: All functionalities are supported as described in the requirement.

SOR3 Configurable GUI

Weight:

3, as this can reduce the time spent for training new users.

Points:

0: The GUI is not configurable.

1: Only a few modifications are possible.

2: There are still some features which should not be visible for some users.

3: The GUI can be highly configured as described in the requirements.

SOR4 Usability

Weight:

3, as a good usability is important, as it leads to less handling errors and a more efficient use of the software.

Points:

0: Usability is not very intuitive. E. g., nested menus with unclear naming.

1: Usability could be a bit more intuitive, e. g., workflows have to be remembered or looked up because no help notifications are offered.

2: Usability is sufficient, there are only some non intuitive functions that have to be looked up in the manual.

3: Usability is very good as described in the requirements.

SOR5 Documentation

Weight:

3, as the documentation is an important part of a complex software.

Points:

0: No documentation or a very short documentation, which acts more as an information brochure, is available.

1: The documentation is not complete or not well enough explained.

2: The documentation is sufficient but could be detailed.

3: All functionalities of the CLM software are well described and explained with the help of, e. g., screenshots.

SOR6 Installation and Maintenance

Weight:

4, as the CLM software has to be capable of being integrated into the BMW infrastructure.

Points:

0: The CLM software cannot be integrated.

1: The CLM software can be integrated, but needs many adjustments.

2: The CLM software can be integrated, but needs an individual process for, e. g., maintenance, as no maintenance solution offered by BMW is adjustable to work for the software.

3: The CLM software is capable of being integrated in the BMW infrastructure with no or only few modifications.

SOR7 Backup

Weight:

4, as a backup plan has to be set up that can be integrated into the BMW infrastructure.

Points:

0: Backup plans cannot be integrated into the BMW infrastructure.

1: A backup plan exists that can be integrated, but it needs many adjustments.

2: No standard BMW backup process fits, but an individual process with manageable effort can be integrated.

3: A backup plan can be integrated in the BMW infrastructure with no or only few modifications.

SOR8 Events and Alerts

Weight:

5, as this is a fundamental part and has to be integrated into the BMW infrastructure.

Points:

0: It is not possible to integrate the event and alert scheme of the CLM software into the BMW infrastructure.

1: An integration is possible but needs a lot of effort to develop it.

2: The event and alert handling can be integrated with a customized solution.

3: The event and alert scheme is compatible without or with minor changes to the interface of the ticket handling system of the BMW infrastructure.

SOR9 Performance

Weight:

3, as a slow or laggy software offers no good handling with the software.

Points:

0: The software is very slow and laggy.

1: Some important GUI operations show slow performance.

2: The CLM software could have a better response time.

3: The software has a good performance as described in the requirements.

SOR10 Software quality

Weight:

5, as the CLM software has to be reliable.

Points:

0: There are many bugs.

1: There are some bugs but without major bugs.

2: There are a few minor bugs, e. g., a browser incompatibility like a displaced button.

3: There are no known bugs.

3.2.6. Confidential Requirements

Appendix A.2 lists all confidential concept details for the confidential requirements, which are special to the BMW infrastructure.

4. Proof of Concept

This chapter describes the PoC that is used to evaluate CLM solutions for the BMW Vehicle PKI. At first, a market analysis is done in 4.1, where the CLM software, which will probably meet most of the requirements, are considered closer through a demo of it. Afterwards, a PoC of the most promising CLM software is executed. Additionally, an assessment of self implementing a CLM software, which fulfills the requirements, is done. The requirements are defined in 3.1 and scored in 3.2, while in 4.3 the tests are defined that are executed. Details to setup of the PoC are described in 4.2. The results of the PoC are presented in 4.4.

4.1. Market Analysis

The confidential chapter A.3 lists the market analysis of CLM software. Additionally, the costs for a self coded CLM software are estimated, which fulfills the requirements listed in section 3.1.

4.2. Setup of the PoC

The setup of the PoC is described in the confidential chapter A.4.

4.3. Validation Tests for the Requirements

While chapter 3.1 lists the requirements for the CLM of the BMW Vehicle PKI and in 3.2 the weights and scores for the requirements are defined, this section contains the tests that are made to verify if a requirement is met. The evaluation results are listed in the confidential appendix's chapter A.6.

All tests are run on a Windows Server 2012, 2016 or Windows 10 Build 16299. As Browsers, Microsoft Internet Explorer 11 and Google Chrome version 74 are used.

4.3.1. K.O. Criterion

This domain lists one must have criterion which has to be fulfilled.

KR1 On-premise Installation

The CLM software has to be installable and runnable on-premise in the BMW infrastructure without needing an internet connection.

4.3.2. Discovery

This section describes the validation tests for the certificate import and discovery process.

DR1 Manual import of certificates through the GUI

This requirement is tested through importing selected certificates in DER and PEM format with file extensions crt, cer, pem, and der. The used certificates are X.509 certificates with a 2048 bit RSA key and SHA-256 as Hash algorithm. Using the GUI of the CLM software, a standard Windows Explorer window is opened in which the following possibilities are successively executed to test the import functionality:

- select one certificate and try to import it
- select multiple certificates and try to import them
- select one ZIP-archive and try to import its certificates
- try to import all certificates from one folder by selecting the folder that contains certificates and one subfolder with certificates. The folder is placed on the local machine where the browser is installed and on a network storage accessible via CIFS.

As this test is executed only on Windows operating systems, NFS could not be tested.

DR2 Automated import of certificates from file shares

Importing certificates from file shares with an automated task is handled very differently by CLM software. Some software always use agents while others naively support an import of certificates on the same machine they are installed on. The agents of some software allow an import of certificates from other machines in the network, where the agent is installed on them and communicates with the CLM software. This functionality is therefore tested by the possibilities the software offers. The used folder for this test contains certificates and a subfolder with certificates. The certificates have 2048 bit RSA keys with SHA-256 as Hash Algorithm and are in DER and PEM format with file extensions crt, cer, pem, and der. The folder is located on a CIFS file share and on the machine where the software or an agent is installed. As this test is executed in an encapsulated network environment, NFS could not be tested.

DR3 Import a list of file shares which shall be discovered

This is a special functionality in which many various processes can exist. Therefore, the way it is executed depends on the CLM software. The list contains the link to the folder of the validation test of the requirement “DR2 Automated import of certificates from file shares” and the structure of the list is adapted to the specifications of the software.

DR4 Scheduled tasks for certificate imports from a file share

The time when the task is executed is set to five minutes after the current time. The process of how the task and the import mechanism works varies and therefore can not be covered by a standardized test. Therefore, the tests adapts to the possibilities offered by the CLM software. The folder of the validation test of the requirement “DR2 Automated import of certificates from file shares” is used for the test and is located on

a CIFS fileshare and on the machine where the software or the agent is installed.

DR5 Import certificates from an IP address

For discovery, a server is used that offers two TLS connections. One connection is secured using a self-signed certificate, while the second connection uses a certificate issued by the PKI of BMW and sends the server and the intermediate CA certificate during the TLS handshake. This test is done by connecting to the server via IP, IP range, and the FQDN of the server. Additionally, ports are specified in the CLM software as specific port, port range scan, multiple ports, and under a full port scan of the machine. In order to use the FQDN of the server the requirement is tested against, some CLM software require the DNS server of BMW to be configured, if, e. g., they are not installed on an operation system that has the DNS server of BMW configured. Firewall exceptions for the HTTPS connections have to be configured, if needed.

DR6 Import a list of servers which shall be discovered

The file containing the list of servers which shall be discovered over network is set up according to the requirements of the CLM software for this file. Two servers of BMW are inserted into the list, whereas one of the servers is the same server as used in the validation test of the requirement "DR5 Import certificates from an IP address". By connecting to the second server, a firewall exception has to be configured, if needed, to establish the HTTPS connection to retrieve a certificate.

DR7 Scheduled tasks for certificate imports from an IP address

A task is created to automate the execution of the functionality referencing the requirement "DR5 Import certificates from an IP address". To test this requirement a scheduled task is set up in the CLM software. The same server from the validation test of the requirement "DR5 Import certificates from an IP address" is used and shall be discovered. The execution time of the task is set to five minutes after the current time.

DR8 Import certificates from Java Key Store

The capability of importing certificates from a Java Key Store is tested by using a password protected JKS containing three certificates.

DR9 Categorize on import

A rule which moves discovered certificates into a defined certificate group is configured. Afterwards, one certificate is imported to check if the categorization works. As this functionality differs strongly between the different CLM software, no standardized test could be used. The specific tests are therefore described in the validation result of the requirement "DR9 Categorize on import".

DR10 Customizable interface

The requirement "DR10 Customizable interface" is too specific, complex, and time intensive to implement it for all CLM software. Therefore, a theoretical view on the specifications and possibilities of the CLM software is done to estimate and evaluate this requirement.

4.3.3. Certificate Inventory

All validation tests for the requirements of the certificate inventory are listed here.

CIR1 Dashboard

The test for this requirement consists of creating a dashboard and modifying it, if necessary and possible, so that it displays the information specified in the requirement "CIR1 Dashboard". Therefore, e. g., widgets on the dashboard and the necessary functions for them, like reports of specific information as the status of revoked certificates, are created. More details of the individual configuration of the dashboards are described in the validation result of the requirement "CIR1 Dashboard". To get an impression of the dashboards, around eighty certificates are imported, which offer a high variability of, e. g., issuing CAs, extensions, and algorithms. Some of these certificates are expired.

CIR2 Configurable Columns

The certificate inventory GUI is inspected on how the columns can be customized in terms of adding, deleting, and sorting. Additionally, the certificate information that are available as columns are evaluated, e. g., basic constraints.

CIR3 Search certificate inventory

The search of the certificate inventory is tested for the following search requests:

- Search for all certificates with the Subject Key Identifier extension.
- Search for all certificates without the Subject Key Identifier extension.
- Search for the certificates whose Key Usage extension is like "CRL Signing". This should display all certificates which have the Key Usage extension and at a minimum "CRL Signing" in it.
- Search for the certificates whose Key Usage extension is not like "CRL Signing". This should display all certificates which have a Key Usage extension but no "CRL Signing" in it.
- Search for all certificates whose CN starts with "DE".
- Search for all certificates whose CN does not start with "DE".
- Search for the certificate whose serial number equals "0a".
- Search for the certificate whose serial number does not equal "0a".
- Search for all certificates which use SHA-1 as signature hash algorithm and expire in the next twenty weeks.
- Search for all certificates whose CN starts with "DE" or whose serial number equals "0a".

Note that the search differs between CLM software, so the required search pattern is adjusted to the conventions of the software.

CIR4 Grouping certificates per CA

This is a requirement that is handled very differently by each CLM software. Therefore,

there is no standardized test and the details of the test are described together with the validation result of the requirement "CIR4 Grouping certificates per CA" for each CLM software.

CIR5 Certificate groups with individual parameters

This is a requirement that is handled very differently by each CLM software. Therefore, there is no standardized test and the details of the test are described together with the validation result of the requirement "CIR5 Certificate groups with individual parameters" for each CLM software.

CIR6 Support of self-signed certificates

Self-signed certificates are imported and checked if they can be displayed in the certificate inventory and if they are visible on the dashboard as self-signed certificates. Additionally, it is examined if these certificates get exclusively displayed in the certificate inventory through, e. g., a specific search for only self-signed certificates.

CIR7 Support of private certificates

Private certificates from the BMW PKI are imported and checked if they are treated exactly as public certificates in the CLM software.

CIR8 Support of private CA certificates

Private CA certificates from the BMW PKI are imported and checked if they are treated as public CA certificates in the CLM software.

CIR9 Support of cross-signed end-entity certificates

Cross-signed end-entity certificates are imported and checked if they are recognized, listed correctly, and specially marked to better notice them.

CIR10 Display certificate chain

The CLM software is inspected if it supports the visualization of the certificate chain. To test this, public, private, self-signed, and cross-signed certificates with the complete chain imported first are probed. As a second test, a private end-entity certificate is imported, while the certificate of the issuing CA is imported afterwards, to then check if the chain is built correctly as well.

CIR11 Renewed certificates

This test is for checking the behavior of importing a renewed certificate. Therefore, four certificates are imported. Two of them are issued from the same CA with identical information except of the serial number and the validity dates. This means, that, e. g., the CN, the used key pair, the used algorithms, and the extensions and their values are identical. Only the serial number, the validity dates, the thumbprint, and the signature differ between these two certificates. The other two certificates are issued from the same CA for the same subject CN but this time with different key pairs.
This test is divided into two sub tests. First the certificates with the same key pair get imported and it is checked how the CLM software reacts. Afterwards, it is examined what possibilities for them are offered, e. g., using a history mechanism. Then the second two certificates with different key pairs but an identical CN are imported and

checked for the same criteria.

CIR12 Special functionality for pinned certificates

For certificates that are pinned on different systems, a logic is needed to display these connections. Therefore, the CLM software is checked if it provides a solution to display the systems a certificate is pinned on.

4.3.4. Monitoring and Alerts

This section lists all validation tests concerning monitoring and alarming.

MAR1 Check validity status of certificates

Certificates which expire in the next 13 days, 29 days, four month, and in three years are imported. Additionally an expired certificate is imported. The certificate that already expired should be listed as expired, while the certificates that expire should trigger an alert or log entry and be listed as about to expire. It is only checked if the CLM software recognizes and lists the issues with the certificates and not if an alert is sent to a system, as this is treated in the test of the requirement "SOR8 Events and Alerts". A revoked certificate has not been tested because the PoC was set up in an isolated environment.

MAR2 Individual alert schemes

The alert functionalities of the CLM software are examined if they are configurable for specific certificates, certificate groups, and CAs. In parallel, the type of alert is inspected if it is capable to configure multiple steps of alerts, e. g., three month before a certificate expires the first alert and one week before the certificate expires the next alert is sent.
Additionally, the alert schemes for CRL and OCSP tickets are checked also for multiple alert steps.
This functionality is not tested with expiring certificates and an alert system like an SMTP server due to the required time effort. This test checks only the availability of these functionalities.

MAR3 Inspection of the certificate chain

To check if an alert gets send if a certificate without a previously imported certificate chain is imported, the software is searched for a mechanism to configure this procedure. Afterwards, a certificate is imported whose issuing CA certificate is unknown to the CLM software. Additionally, a self-signed certificate and a Root CA certificate is imported to prove if they can get imported without triggering an alert.

MAR4 Support OCSP responder monitoring

It is checked if a functionality is offered to enter the URL of an OCSP responder, which is used to check its availability and the validity of the OCSP ticket. To test this functionality, an OCSP responder is entered, which gets blocked by a firewall rule, and an available and fully working OCSP responder is entered. The case of an invalid or soon expiring OCSP ticket is not checked, as setting up an OCSP responder to simulate

a not fully working OCSP responder is too time-consuming.

MAR5 Support OCSP stapling tickets

The CLM software is checked if it is capable of monitoring OCSP stapling tickets of a server for availability and validity. Therefore, the IP of a server is entered and a TLS handshake including a Certificate Status Request extension has to be performed in order to get the OCSP stapling ticket and to check it for validity. Validity is proved by checking the time validity and the signature of the OCSP stapling ticket.

MAR6 Support CRL monitoring

The CLM software is inspected if it supports entering the URL of a CRL download location. The URL has to be checked for availability and for validity of the downloaded CRL. To test this, a firewall blocked CRL download location and an available CRL location with a valid CRL is entered. An expired CRL is not tested as it is too time intensive to make the CRL available on a web server.

4.3.5. Software Operations

The validation tests concerning software operations of the CLM software are listed here.

SOR1 Authentication

The CLM software is checked if it offers support for Active Directory groups and users. The integration of a domain controller that is responsible for handling authentication requests, can be significant administrative effort. This is due to the need for establishing the connection to the AD server with a dedicated technical user. Therefore, the implementation is not done in the PoC and the software is only checked if it offers the possibility to integrate AD users and groups.

SOR2 Fine-granular authorization concept

To test the authorization concept of the CLM software, the defined role model from the requirement “SOR2 Fine-granular authorization concept” is getting configured. After the configuration, three users are created and assigned to the three role models. Then each of the three users log in and the configured functionalities that are allowed or prohibited are verified.

SOR3 Configurable GUI

The GUI of the CLM software is configured to be as minimalist as the user roles have permission, so, e. g., not allowed or unnecessary functionalities are made invisible. After that the three users configured in the test of the requirement “SOR2 Fine-granular authorization concept” are logged in to verify the changes.

SOR4 Usability

Several operations and configurations are done in the CLM software to get an impression of the usability. Presenting the software to colleagues and discussing the usability leads to the rating. The following topics are considered: How often the documentation is looked into, the reaction due to an unexpected process while navigating the menu, the time to change and inspect items, and the memorization of the menu navigation

and configuration after two weeks of not working with the software. Additionally, help notifications in the GUI are considered as well.

SOR5 Documentation

The documentation of the CLM software is inspected. This is done while using the software and looking for needed configuration instructions and explicitly for getting an impression of the documentation quality.

SOR6 Installation and Maintenance

This is tested, among other things, by setting up the PoC. As the PoC is not fully integrated like the final solution would be and not all functionalities are set up due to too much time effort, some of the required information for a full integration are handled theoretically. For the integration into the BMW infrastructure, standard procedures are viewed and the corresponding colleagues are contacted to get the information, if the software can get integrated or to find a way to integrate it if it does not match to standard procedures. There are two main points to consider: The operating system and the database, whereas the alert handling is covered in the requirement "SOR8 Events and Alerts" and the user authentication is handled in the requirement "SOR1 Authentication".

SOR7 Backup

The backup of the database and the configuration is considered theoretically as the CLM software is only set up as a PoC and therefore establishing a backup mechanism is too time intensive. To implement a backup procedure in the BMW infrastructure, standard procedures at BMW are inspected. If the software does not fit into these procedures, the corresponding colleagues are contacted to find a solution.

SOR8 Events and Alerts

This is handled theoretically due to the high time effort implementing this functionality. The BMW ticket event handling offers an API which is probably the best solution to send alerts from the CLM software. Therefore, the CLM software is checked if it offers mechanisms to integrate with the BMW ticket event handling software or API.

SOR9 Performance

To get an impression of the software performance various actions are performed, like certificate searches, refreshing or reloading of the GUI, and log or report viewing.

SOR10 Software quality

The quality of the CLM software is tested through interactions with the software and the reliability of tasks, tested through, e. g., the requirement"MAR6 Support CRL monitoring". If bugs are found they are affecting the score. The release times of bugfixes and product updates are considered from a theoretical standpoint as they cannot be verified through the PoC. However, as the theoretical release times for bugfixes are pretty similar between the vendors, they are rated the same and do not occur in the validation.

4.3.6. Confidential Requirements

All Tests for the confidential requirements are listed in appendix A.5.

4.4. Validation of the Requirements

The validation results of the requirements are listed in the confidential appendix A.6.

5. Vehicle-to-X Communication Security

In the previous chapters the monitoring of certificate expiration within the BMW Vehicle PKI is described to answer RQ1. This chapter processes RQ2 by evaluating certificate invalidation in a future development of the BMW Vehicle PKI. Because one of the main upcoming challenges for securing the communication of vehicles could be V2X communication, it is used to investigate benefits regarding invalidation by comparing identity-based cryptography as an alternative cryptographic mechanism to V2X PKIs with, e. g., X.509 certificates.

At first, a short overview of the current concept status for securing V2X communication is given, which is achieved by using PKIs. Next, a security concept for V2X communication is presented, which investigates if IBC can deliver advantages to prior concepts regarding invalidation.

Different systems for the intended V2X infrastructure are currently under development worldwide. Their first noticeable difference are based in the underlaying technology. There are curently two approaches: One relying on Dedicated Short Range Communication (DSRC), which is based on the Wireless Local Area Network (WLAN) standard IEEE 802.11p in the 5.9 GHz band. The other approach relies on Cellular V2X, which is based on 4G Long-Term Evolution (LTE) and 5G mobile cellular connectivity [GSM17]. While China is implementing C-V2X, the USA will use DSRC. In the EU the debate about using C-V2X or DSRC is still ongoing [Ofi19, SH19]. As the USA and the concept of the EU using DSRC differ in many aspects like, e. g., the WLAN frequency, all three approaches use for securing the communication a PKI [Wan19, HPY+14, BSS+11]. These PKIs are very different and the schemes of the PKIs for the Vehicle-to-X communication of the USA and the EU are explained in the following sections.

5.1. USDOT V2X PKI

This section describes the scheme of the V2X PKI for the United States Department of Transportation (USDOT) that focuses on privacy and security [HPY+14, WWKH13]. The PKI does not collect or store data that would allow to identify a vehicle or driver who is maybe driving erratic or speeding. Personal identifying information is only used to prevent misuse, leakage or an attack on the system. The PKI is depicted in figure 5.1.

The three-tier PKI uses multiple private Root CAs that are operated in an offline environment. Under the Root CAs there are multiple Intermediate CAs that have the function to issue certificates to issuing CAs, but not to end-entities like vehicles. The Pseudonym CA is the issuing CA for short-term certificates for end-entities, e. g., vehicles. In earlier designs, the lifetime of the short-time certificates was five minutes, whereas in the current design it is

a variable length of some minutes to make them less predictable and as a consequence more difficult to track.
Using these short-term certificates leads to a high issuing rate for the certificates of each device. For this purpose, a so-called Butterfly key expansion is introduced. Butterfly keys offer a device the functionality to request an arbitrary number of certificates with only one CSR. This is done by using a public key seed, an encryption public key seed, and two expansion functions instead of multiple CSRs. The CSR is sent by the device through the Location Obscurer Proxy, which is explained afterwards, to the Registration Authority.
The Registration Authority calculates an arbitrary amount of statistically uncorrelated public keys for the device, using the public key seed and one expansion function, and forwards them to the Pseudonym CA. The Pseudonym CA issues certificates for these public keys, calculates a random value, the so-called private key reconstruction value, and then sends the certificates and private key reconstruction values for it to the Registration Authority, which itself forwards them to the device. Using the certificates and the private key reconstruction values, the device can calculate the private key for each certificate. It has to be noted that only the device who sent the public key seed can calculate with the private key reconstruction value the private key belonging to the received public key in the certificate.
As the Registration Authority knows the original device from which the public key seed comes and sends the new certificates back to the device, a possible tracking by the Registration Authority has to be prevented. Therefore, the Registration Authority calculates the public encryption keys by using the second expansion function and the encryption public key seed. Afterwards, it forwards the public encryption keys to the Pseudonym CA, which uses them to encrypt the certificates and the private key reconstruction values.
To ensure that only one Registration Authority is responsible for a device during a specified time period and no multiple actions are performed, the Request Coordination is established.

Additionally, there are Linkage Authorities that provide the Registration Authority with so-called linkage values.Linkage values are used for the Pseudonym CA to issue certificates to a device and to enable the possibility to retrace all issued short-term certificates of that device in case of a revocation. This functionality enables the revocation of all short-term certificates of a device by just revoking the linkage value instead of listing all certificates in the CRL. The reduction in size of the CRL is of high significance. As the Linkage Authority could trace a device, they always operate as a pair of two, so that neither of them alone has enough information to track anyone.
For detecting misbehavioral uses of, e. g., a certificate, a so called Misbehavior Authority is responsible. This service detects when messages are not plausible or a malfunction or malicious action occurs. The Misbehavior Authority also issues all CRLs for the CAs and publishes them to a CRL download location.
Another part of the PKI is the Location Obscurer Proxy, which is responsible for obscuring location information of devices that are trying to contact one of the services of the PKI. Therefore, all contact attempts first contact the Location Obscurer Proxy before they are directed to the final service.
For establishing the bootstrapping process for the initial connection between the device and the PKI, the Device Configuration Manager, the Certification Lab, and the Enrollment CA are set up. The Enrollment CA issues long-term certificates.
The policy of the PKI and technical standards, like operating procedures are gathered in the SCMS Manager. [HPY+14, WWKH13]

The USDOT V2X PKI does not use X.509 certificates but instead uses so-called Wireless Access for Vehicular Environment (WAVE) certificates, defined in IEEE 1609.2. Compared to the X.509 standard, this standard uses a different structure for certificates, CSRs and CRLs. This results in certificates that are approximately half the size of a typical X.509 certificate. ECC is used for the signature algorithm. [IEE06] [RvD16]

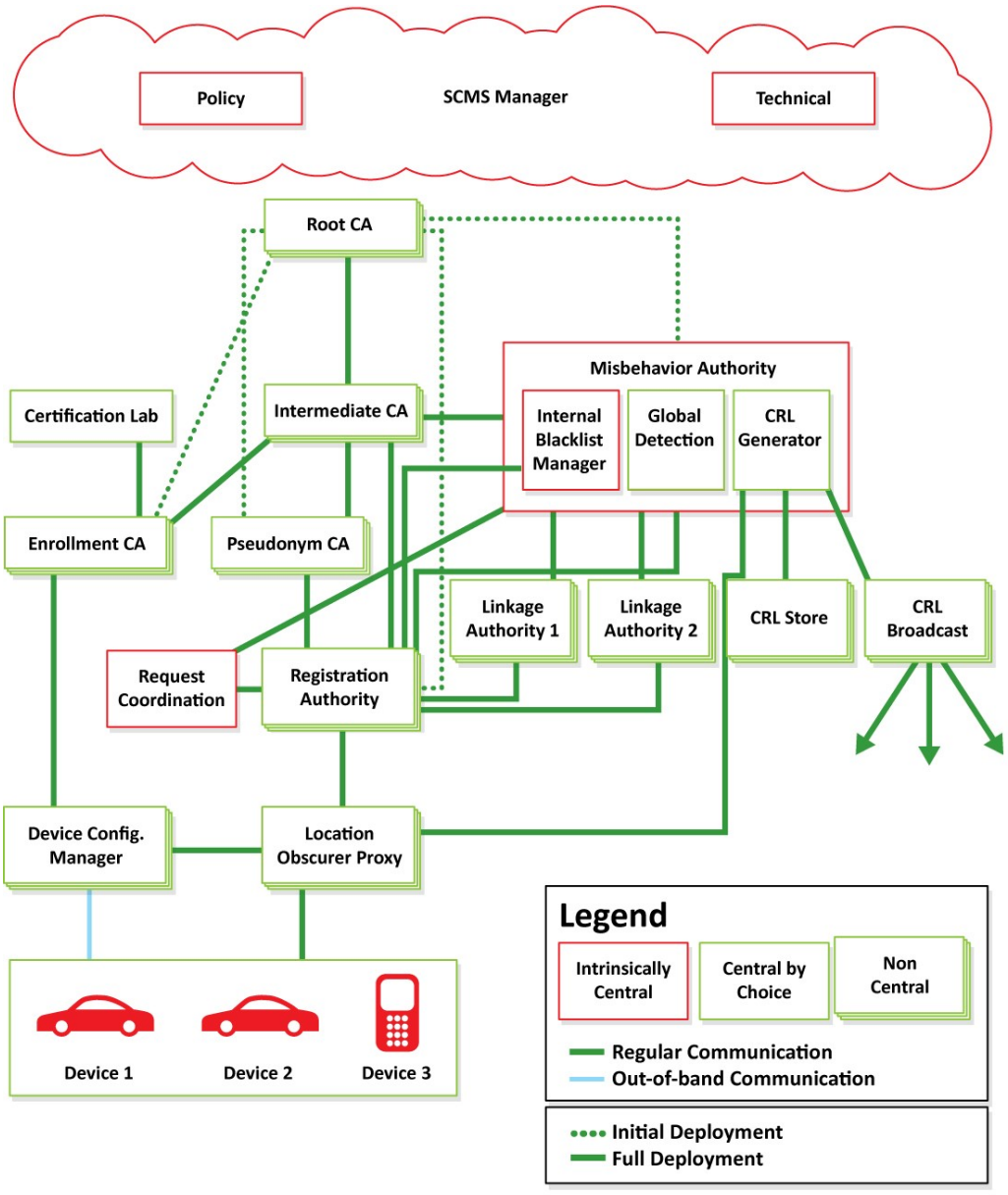

Figure 5.1.: V2X PKI concept of the USDOT [HPY+14]

5.2. C2C-CC V2X PKI

The Car2Car Communication Consortium (C2C-CC) develops the concept of the PKI for European V2X communication together with, e. g., ESCRYPT and the Bundesamt für Sicherheit in der Informationstechnik (BSI) of the EU. The C2C-CC V2X PKI is depicted in figure 5.2. This PKI is a two-tier PKI and can consist of multiple Root CAs, which are then cross-signed. Under the Root CA, so-called Long-Term and Pseudonym CAs are arranged. Every Intelligent Transport Systems station, e. g., a vehicle, gets one Long-Term Certificate issued by the Long-term CA, which is used for communicating with the services of the PKI, but never with other ITS stations for privacy reasons.
For the V2X communication the Pseudonym Certificates, issued by the Pseudonym CA, are used, which allows to sign messages without the possibility to identify the sender as a specific vehicle for other V2X communication members. Note that the verification of the signature is still possible due to the Pseudonym CA acting as trust anchor. The Long-Term and Pseudonym CAs could be operated by the C2C-CC or by, e. g., the vehicle manufacturers, but no organization should operate both for the same car fleet, as privacy could get compromised. For the Long-Term certificates, no special requirements for their size is needed, as they are not used for V2X communication. Instead, standard X.509 certificates with RSA keys could be used. For the Pseudonym certificates size matters and therefore WAVE certificates, as defined by IEEE 1609.2 and used in the USDOT V2X PKI, would be a presumptive solution.
Every vehicle has one Long-Term Certificate that has a lifetime of 10 years and is based on ECDSA p224. Bißmeyer et al. [BSS+11] define Pseudonym certificates and the underlaying key pairs to have a lifetime of one year and that they are used only for a short time span, which results in needing about 1500 certificates per year. To avoid constantly requesting these certificates, they are requested as big packets and are stored in a vehicle's certificate store. In the current PKI concept, vehicles store their Pseudonym Certificates in their certificate store up to three months prior the certificates get valid. These certificates are only valid one week as a maximum.
Currently, revocation is only possible for the Long-Term and Pseudonym CA certificates and not for certificates issued to the ITS stations like, e. g., vehicles, due to the difficult distribution of the CRLs. The revocation of the CAs is done using CRLs. An exclusion of vehicles from the V2X communication is only possible by prohibiting the request for new Pseudonym Certificates. [BSS+11, CAR19, ESC19]

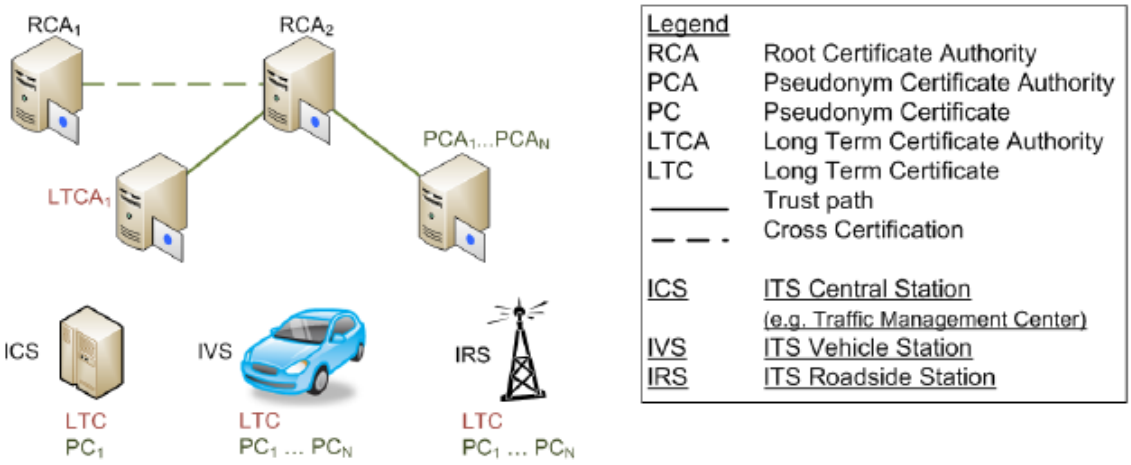

Figure 5.2.: PKI concept of the C2C-CC [BSS+11]

5.3. Identity-Based Cryptography Vehicle-to-X architecture

Instead of using a PKI for securing V2X communication, a concept using IBC is presented in this section. This approach has some advantages compared to a PKI relying on, e. g., X.509 which are explained in this section. This approach is used to discuss RQ2 by elaborating if the BMW Vehicle PKI can benefit regarding invalidation by using IBS for securing V2X communication.
For the IBC V2X architecture the following four main components are needed:

- Key Generation Center
- Anonymization Service
- Registration Service
- Revocation Service

Key Generation Center (KGC)
As the V2X architecture needs to get standardized for all devices using V2X communication, like the vehicles of all car manufacturers, an architecture has to be established that is trustworthy. If V2X communication does not get standardized it can happen that two cars of different brands cannot communicate with each other. Trustworthy is a problem for a single instance like the KGC which issues all private keys for all devices, as it has to be cleared who is responsible and trustworthy enough to operate it. In comparison to a Root CA of, e. g., the USDOT V2X PKI or the C2C-CC V2X PKI, the KGC is more critical, as the Root CA never gets in touch with the private keys of the devices while the KGC issues these for them. To solve this problem technically (and not politically) the IBS V2X architecture can use an hierarchical scheme or Multiple Trust Authorities similar to those explained in chapters 2.2.1.1 and 2.2.1.2.
A potential solution approach for the hierarchical scheme would be to put the root KGC under control of a political instance like the EU parliament and the domain KGC under

control of the car manufacturers. If Multiple Trust Authorities are used, these could be under the control of some countries or car manufactures. With this hierarchical scheme, a car manufacturer that does not operate their own KGC must trust the provider of the KGC by obtaining the private keys for its vehicles. With the Multiple Trust Authority scheme, the car manufacturer needs less trust as all KGC would have to collude to gain the private key of one of its vehicles.
On the other hand, some car manufacturers probably want to operate their own infrastructure without dependencies from the KGCs of other car manufacturers. Car manufacturers who operate their own KGC have the benefit, as most of the car manufacturers have their own secure communication to their cars, that they can generate and deploy the private keys for their cars using their secure communication channel to their cars.
Therefore, the author's recommendation is to use a hierarchical scheme with a root KGC operated by a department of a country, a domain KGC for infrastructure like traffic lights operated by the responsible traffic agency of each country and domain KGCs operated by each car manufacturer. The car manufacturers then can use their secure communication channel to the cars to deploy the private keys to their cars. For other devices using the V2X communication network, new domain KGCs should be added. Pseudonyms identifiers are used for vehicles, whereas stationary devices like traffic lights could use their GPS coordinates and are known to the vehicles through, e. g., the navigation map. The pseudonym identifiers are generated by the KGC of the vehicle manufacturers. Further information like validity periods are discussed in the section of the Revocation Service as these details highly dependent on the selected revocation method.

Anonymization Service
This service is needed to delete all unnecessary information out of messages sent by devices like vehicles to the V2X architecture. This is important to prohibit, e. g., tracking of vehicles. Main information that are not required are, e. g., location information. As it is recommended to use a hierarchical KGC scheme, only direct communication between end-entities in the V2X network and from end-entities to the Revocation Service have to be checked for anonymization. This is because the connection to the KGC is assigned to the car manufacturers and they already have all data of the cars and underlie regulations like the General Data Protection Regulation (GDPR) of the EU. Therefore, the Anonymization Service gets only contacted to check the revocation status of a member in the V2X communication. The Anonymization Service acts like a Proxy and passes the received information after deleting non important data to the Revocation Service.

Registration Service
The Registration Service is contacted if a device requests a new private key. The Registration Service then contacts the Revocation Service to check if the device is blacklisted due to, e. g., malicious behavior. If the device is not listed, then the request for a new private key is directed to the KGC. The Registration Service is a non-central instance and each KGC is operating one. Compared to one central Registration Service, this has the benefits of using the secure channel of the car manufacturers for their vehicles and to avoid the need for anonymization as tracking might be possible due to, e. g., knowing the periods of requests for new private keys. Depending on the used revocation scheme for this V2X architecture, the Registration Service is not necessary, if, e. g., keys are nor requested by the vehicles and are automatically send to them.

Revocation Service
This service is responsible for the revocation of compromised secret keys. Revocation of keys is necessary if, e. g., cars get stolen, or ECUs get demounted or misused. If cars get scraped ECUs are often demounted. An example for misuse could be a resident, who wants to have a BBQ in this garden without the noise of driving cars. Therefore, he uses an hacked ECU to send the information of a traffic jam effectuated to a road closure. Another example is a driver of a car who sends wrong data, e. g., being an emergency ambulance, to achieve a green wave. If a device behaves malicious this is reported by the devices noticing that to the Anonymization Service, which forwards this information to the Revocation Service. The Revocation Service then can revoke this device.

One big advantages of IBC compared to an X.509 PKI is the fact that private key revocation can be done mathematically by generating a new master secret key and therefore issuing new private keys for the devices. In an X.509 PKI the equivalent would be to issue a new CA and issue for all devices new certificates, while it has to be claimed that all entities do rekeying. This is much more overhead than in an IBC scheme, as in the IBC scheme the KGC issues new keys to the end-entities, while in an X.509 PKI all end-entities have to generate a new key pair and a CSR, which is transferred to the RA and then to the CA . The CA validates and signs the CSR before it sends the certificate back to the end-entity. Mathematically revocation results in not needing a revocation list where revoked CAs certificates or end-entity certificates are listed. However, the generation of new private keys for all devices is a complicated process in a dynamic network, where the devices are not always online, are supervised by different companies, mostly communicate over a short period of time, and change location frequently within a country and across borders like the EU. Short communication is, e. g., between a traffic light and a driving vehicle at 50km/h or a vehicle warning another vehicle on the other roadside that it will approach a stationary traffic jam. While revocation of single end-entities in an X.509 PKI is done through CRLs, revocation of single end-entities in the IBC scheme is more complicated.

In [Sha85] the generation of a new master secret key for revocation is described. Applying this method for the revocation of one malicious device has massive overhead costs as all members have to request for new private keys at the same time. If this update phase for new private keys is not successful or lasts too long for all devices, a non-successful communication is caused as the signature cannot be verified if one party already has a new private key while the other party still uses the old private key. Nearly simultaneously the V2X infrastructure will break down. To remember the devices from getting a new private key, the excluded devices have to be stored in, e. g., a database to prevent them getting a new private key by the KGC. With these drawbacks this cannot be considered as the standard revocation method for end-entity certificates. For revocation of a KGC it can be applied, as all issued key of that KGC are compromised and therefore have to get reissued in either case.
The authors of [CL02] present a so called dynamic accumulator scheme. This scheme allows the revocation of identities in an IBS scheme by sending a message to all members of a group. The non-revoked members can use the information of the message to update their secret keys, while the revoked members cannot do this. This prevents the need for generating new private keys at the KGC and to send them to the devices. This has the same drawback as above, that the revoked members have to be stored in, e. g., a database to prevent them getting secret key update messages in the future. Another problem is the point that if revocation is happening quite often, many update secret keys have to be send

and it can happen that one device has a new private key while another device still has the old one. Also each device has to generate a new private key with each revocation.
Verifier-local revocation is a revocation scheme where the revocation authority sends revocation messages to the verifiers in an identity-based group, without the need of generating new secret keys for the non-revoked members [ISJH10]. This concept is not applicable in this context, as every device can be a verifier, which happens if, e. g., two vehicles communicate with each other.
Another approach for revocation in IBC is the use of revocation lists that contain revoked identities [YWRL10]. This approach requires the KGC to issue new private keys for the non-revoked end-entities after an end-entity is revoked and added to the revocation list. In the revocation scheme of the authors the revocation list is kept at the KGCs to not issue new keys for the listed end-entities and is, in opposite to a CRL of an X.509 PKI, not used for end-entities to check themselves if another end-entity is revoked. However, the revocation list could be used by the end-entities as well if it gets signed by, e. g., the KGC. If there are many KGC in use, e. g., in an hierarchical identity-based scheme, this revocation list has the same following drawbacks as an X.509 CRL: CRLs are one of the biggest disadvantages of the X.509 PKI as they have the problem of growing to several Megabytes if many certificates are revoked, which can be a problem for the download of the CRL if the internet connection is not fast or reliable. Considering that the working range of WAVE in a line-of-sight is in best case 1000 meters [IEE06] and vehicles can cover this distance quickly there is not much time to download the required CRL if the vehicle did not download it before. Secondly, stored CRLs are not always up to date as it mostly lasts several hours or days before a new CRL gets downloaded. To temper the disadvantages of CRLs techniques from the X.509 PKIs could be used like OCSP responders and OCSP stapling, where the OCSP responses are signed by the KGCs. Both decrease the time to get information about the current status of a device regarding revocation and have less problems with bad connections. These systems do have drawbacks as well, like Denial-of-service attacks could make a CRL download location or OCSP responder unavailable.
An indirect revocation is possible by using, e. g., time stamps in the public identifier as suggested by Boneh and Franklin [BD01]. This time stamp then defines a date with a validity period, e. g., 12.05.19 21-22 CEST meaning that it is only valid on May 5th, 2019 between 9 and 10 PM Central European Summer Time (CEST). If these time stamps are only valid, e. g., one hour, an attacker could use a compromised identity only for this time span. If a device gets revoked it does not get new private keys. A drawback is that the revoked device has to be stored in, e. g., a database to prevent that new private keys for this device get issued by the KGC. This requires issuing new key quite often, but the time stamps allow to define, e. g., overlap times. Each end-entity requests every 30 minutes a new key that is valid an hour, so a 30 minutes overlap between the old and the new key exists where both are valid. E. g. an end-entity has a key which is valid from 4 PM to 5 PM and it requests at 4:33 PM a new key, which is then valid from 4:30 PM to 5:30 PM, the end-entity has two valid keys between 4:30 PM and 5 PM. This prohibits the problem of all keys needing to get simultaneously replaced with new keys and therefore each device has 30 minutes to receive the new key before it cannot communicate with other end-entities that already have the new key.
Another revocation approach is described in the paper of Boldyreva, Goyal, and Kumar [BGK08]. The paper focuses on identity-based encryption, but signatures are also possible. In this approach each identity has a private key and a so-called key update. The private key

is issued and transferred to the end-entity by the KGC. The key update is published by the KGC and is made public. For decryption an end-entity needs both of them. If the KGC wants to revoke the end-entity it stops publishing the key update. However, this scheme is not collusion resistant and as it is build on binary tree data structures it is more complicated than, e. g., a CRL or an OCSP responder of a X.509 PKI.
In the paper of Libert and Quisquater [LQ03] a revocation scheme for IBC is presented that uses a so-called security moderator that keeps a piece of the end-entities private key. Therefore, the end-entity has to receive a message-specific token of the security mediator to sign or decrypt a message. Revocation is done by the security moderator by not offering new message-specific token to the end-entity. This revocation scheme produces a massive overhead in the V2X communication context, if for each message a security moderator has to be contacted.

To solve RQ2, securing V2X communication using IBC is investigated as a use case regarding invalidation for a future development of the BMW Vehicle PKI. Related work, e. g., as listed above, introduces several different methods for revocation in IBC, but an efficient revocation of single end-entities is still a problem. Nevertheless, using IBC for V2X communication delivers some benefits regarding invalidation compared to the other V2X PKIs. As there are no certificates and the keys of the end-entities are generated with the same master secret key by the KGC, there is no expiration time that has to be monitored individually for each end-entity. Additionally, the verification of the certificate chain for expiration or revocation is not required. However, as revocation of single end-entities still lacks an applicable solution, using IBC for securing the V2X infrastructure is not recommended by the author.

6. Conclusion

This master's thesis evaluates how to monitor certificate expiration in the BMW Vehicle Public Key Infrastructure and introduces a concept for a future development of the PKI regarding V2X communication, where identity-based cryptography is used as an alternative cryptographic mechanism.

In this thesis seven different CLM solutions and the concept for a custom implementation are investigated for the CLM of the BMW Vehicle PKI. There are 46 requirements defined with weighted scores and categorized into six domains, e. g., into "Discovery" and "Monitoring and Alerts". Out of those 46 requirements seven are confidential as they are very specific to this PKI. As result of the demos of all CLM solutions through the market analysis, three are discarded because of, e. g., failing a K.O. criterion. The custom implementation is discarded because it is too time-consuming in implementation and the expected costs are too high.
Four CLM solutions participate in the PoC that contains defined validation tests for each requirement. By fulfilling all requirements 501 points can be achieved. The results of the PoC state that none of the participated CLM software fulfills all requirements. The best CLM software scores 84%, the second 67%, and the third 51% of the requirements of the BMW Vehicle PKI. The reason is that no CLM software through the market analysis is found that is mainly designed to get attached to an existing private PKI. Most of the CLM software are designed to manage a CA, offer many functionalities that are not required if a pure CLM software is needed, and lack functionalities for already operating private PKIs as proven through the PoC. Besides, all of the CLM software, examined through the market analysis and the PoC, are supporting commonly used X.509 certificate configurations. However, having some aberrations in the certificate's extensions lead to problems, although the aberrations are allowed by the X.509 standard. The CLM market is very small although X.509 PKIs are a concept of the late 80s and since then are a critical part of IT security. The author recommends to invite the first two CLM solutions to tender, as both require manageable effort for implementing the unmet requirements. BMW follows the authors recommendation, which results in an implementation in early 2020. The author implemented three scripts, which monitor the worldwide and critical infrastructure of the PKI. These scripts are run until the final CLM solution is in place.

For securing future V2X communication a concept using IBC is introduced and compared to the concepts of the USDOT and the C2C-CC regarding invalidation. This concept uses the already established and secured connection between the car manufacturers and their vehicles to transfer the private keys, which are used for IBC. The private keys are generated in Key Generation Centers, which are operated by the car manufacturers. To establish trust between different car manufacturers an hierarchical identity-based scheme is suggested to be used, where the root KGC is operated, for example, by the government. As a revocation concept for the used private keys is necessary, revocation concepts for IBC were examined

for applicability in the context of V2X communication. As a result no applicable revocation concept for single entities is found that delivers benefits to, e. g., a CRL. Therefore a solution is needed for a proper revocation for single entities in IBC, which does not have drawbacks like being not collusion resistant or needing to request a part of the private key for each message. However, monitoring of individual expiration times of entities and verifying the certificate chain for expiration and revocation are not needed.

As an outlook a CLM software can fill a market gap if it has its focus on being attached to an existing private PKI, offering the abilities to monitor critical infrastructure parts like OCSP responders, and fully supports the X.509 standard.

Abbreviations

ABE	Attribute-based Encryption
ABS	Attribute-based Signatures
AIA	Authority Information Acess
AD	Active Directory
AKI	Authority Key Identifier
API	Application programming interface
C-V2X	Cellular V2X
CA	Certificate Authority
C2C-CC	Car2Car Communication Consortium
CDP	CRL Distribution Point
CentOS	Community Enterprise Operating System
CIFS	Common Internet File System
CLM	Certificate Lifecycle Management
CN	Common name
CRL	Certificate Revocation List
CSR	Certificate Signing Request
DER	Distinguished Encoding Rules
DNS	Domain Name System
DSRC	Dedicated Short Range Communication
ECC	Elliptic Curve Cryptography
ECU	Electronic Control Unit
ECDSA	Elliptic Curve Digital Signature Algorithm
FQDN	Fully Qualified Domain Name
GPS	Global Positioning System
GUI	Graphical User Interface
HPKP	HTTP Public Key Pinning
HSM	Hardware Security Module

HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IBC	Identity-based Cryptography
IBE	Identity-based Encryption
IBS	Identity-based Signatures
IIS	Internet Information Services
IP	Internet Protocol
ITS	Intelligent Transport Systems
JKS	Java Key Store
KGC	Key Generation Center
LDAP	Lighweight Directory Access Protocol
LTE	Long-Term Evolution
NDES	Network Device Enrollment Service
NFC	Near Field Communication
NFS	Network File System
OCSP	Online Certificate Status Protocol
OVA	Open Virtual Appliance
OVF	Open Virtualization Format
PED	PIN Entry Device
PEM	Privacy-Enhanced Mail
PKCS	Public-Key Cryptography Standards
PKI	Public Key Infrastructure
PoC	Proof of Concept
RA	Registration Authority
REST	Representational State Transfer
RHEL	Red Hat Enterprise Linux
RSA	Rivest-Shamir-Adleman
SAN	Subject Alternative Name
SHA	Secure Hash Algorithm
SMB	Server Message Block
SOTA	Software Over-the-Air
SSL	Secure Sockets Layer

SKI	Subject Key Identifier
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Language
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UNC	Universal Naming Convention / Uniform Naming Convention
URL	Uniform Resource Locator
USDOT	United States Department of Transportation
V2X	Vehicle-to-everything
VA	Validation Authority
VMDK	Virtual Machine Disk
WAVE	Wireless Access for Vehicular Environment
WLAN	Wireless Local Area Network

List of Figures

List of Tables

Bibliography

[7-f08a] 7-FORUM.COM: *BMW ConnectedDrive: Netzwerk für mehr Sicherheit und Komfort.*, September 2008. `https://www.7-forum.com/news/2008/7er_f01/connecteddrive.php`.

[7-f08b] 7-FORUM.COM: *Das iDrive System im neuen 7er*, March 2008. `http://www.7-forum.com/modelle/e65/idrive.php`.

[Alt16] ALTON, GAL: *Certificates are an enterprisesecurity asset.* Secure-ly, April 2016. https://www.secure-ly.com/wp-content/uploads/DCM_WhitePaper.pdf.

[BD01] BONEH D., FRANKLIN M.: *Identity-Based Encryption from the Weil Pairing.* In *Advances in Cryptology*, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[BGK08] BOLDYREVA, ALEXANDRA, VIPUL GOYAL and VIRENDRA KUMAR: *Identity-based Encryption with Efficient Revocation.* In *Proceedings of the 15th ACM Conference on Computer and Communications Security*, CCS '08, pages 417–426, New York, NY, USA, 2008. ACM.

[BSS+11] BISSMEYER, NORBERT, HAGEN STBING, ELMAR SCHOCH, STEFAN GTZ and BRIGITTE LONC: *A Generic Public Key Infrastructure for Securing Car-to-X Communication*, October 2011.

[CAR19] CAR 2 CAR COMMUNICATION CONSORTIUM: *CAR 2 CAR Journal Issue 22 March 2019*, March 2019. https://www.car-2-car.org/fileadmin/downloads/PDFs/car-2-car-journal/Journal_22_C2C-CC_Mar_2019.pdf.

[CL02] CAMENISCH, JAN and ANNA LYSYANSKAYA: *Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials.* In *Advances in Cryptology — CRYPTO 2002*, pages 61–76, Berlin, Heidelberg, August 2002. Springer Berlin Heidelberg.

[Cry17a] CRYPTOMATHIC: *CKMS Case Study Swedbank*, September 2017. `https://www.cryptomathic.com/hubfs/docs/Case_Studies/CKMS_Case_Study_-_Swedbank.pdf`.

[Cry17b] CRYPTOMATHIC: *Crypto Key Management System Product Sheet*, June 2017. `https://www.cryptomathic.com/hubfs/Documents/Product_Sheets/Cryptomathic_CKMS_-_Product_Sheet.pdf`.

[Cry17c] CRYPTOMATHIC: *CSG Case Study Barclays*, November 2017. `https://www.cryptomathic.com/hubfs/Documents/Case_Studies/Cryptomathic_CSG_Case_Study_-_Barclays.pdf`.

[Dig15] DIGICERT: *Four critical Components of Certificate Lifecycle Management*, July

2015. https://www.digicert.com/blog/four-components-certificate-lifecycle-management/.

[ECR12] ECRYPT II - European Network of Excellence for Cryptology II: *ECRYPT II Yearly Report on Algorithms and Keysizes(2011-2012)*, September 2012. https://cordis.europa.eu/docs/projects/cnect/6/216676/080/deliverables/002-DSPA20.pdf.

[Ent15] Entrust Datacard: *Certificates are an enterprisesecurity asset*, 2015. https://www.entrust.com/wp-content/uploads/2014/11/Six-Steps-SSL-Management-WEB-Nov15.pdf.

[ESC19] ESCRYPT GmbH: *CAR 2 CAR: Pilot-PKI gemäß neuesten Sicherheitsstandards*, May 2019. https://www.escrypt.com/de/news-events/car-2-car-pilot-pki-gemaess-neuesten-sicherheitsstandards.

[GB08] Goldwasser, S. and M. Bellare: *Lecture Notes on Cryptography*, July 2008. http://cseweb.ucsd.edu/ mihir/papers/gb.pdf.

[GPSW06] Goyal, Vipul, Omkant Pandey, Amit Sahai and Brent Waters: *Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data*. Cryptology ePrint Archive, Report 2006/309, 2006. https://eprint.iacr.org/2006/309.

[GS02] Gentry, Craig and Alice Silverberg: *Hierarchical ID-Based Cryptography*. Cryptology ePrint Archive, Report 2002/056, 2002. https://eprint.iacr.org/2002/056.

[GSM17] GSMA: *Cellular VehiCle-to-eVerything (C-V2X) Enabling Intelligent Transport*, December 2017. https://www.gsma.com/iot/wp-content/uploads/2017/12/C-2VX-Enabling-Intelligent-Transport_2.pdf.

[hei19] heise.de: *Zertifikat abgelaufen: Firefox deaktiviert Add-ons*, May 2019. https://www.heise.de/newsticker/meldung/Zertifikat-abgelaufen-Firefox-deaktiviert-Add-ons-4413170.html.

[Hes03] Hess, Florian: *Efficient Identity Based Signature Schemes Based on Pairings*. In Nyberg, Kaisa and Howard Heys (editors): *Selected Areas in Cryptography*, pages 310–324, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[HL02] Horwitz, Jeremy and Ben Lynn: *Toward Hierarchical Identity-Based Encryption*. In *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques: Advances in Cryptology*, EUROCRYPT '02, pages 466–481, London, UK, UK, 2002. Springer-Verlag.

[HPY+14] Harding, J., G. Powell, R. Yoon, J. Fikentscher, C. Doyle, D. Sade, M. Lukuc, J. Simons and J. Wang: *Vehicle-to-vehicle communications: Readiness of V2V technology for application*. In *Report No. DOT HS 812 014*. Washington, DC: National Highway Traffic Safety Administration, August 2014.

[IEE06] *IEEE Trial-Use Standard for Wireless Access in Vehicular Environments - Se-*

curity Services for Applications and Management Messages. IEEE Std 1609.2-2006, 2006.

[Int13] Internet Engineering Task Force (IETF): *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*, June 2013. https://tools.ietf.org/html/rfc6960.

[ISJH10] Ibraimi, Luan, Nikova Svetla, Willem Jonker and Pieter Hartel: *An Identity-Based Group Signature with Membership Revocation in the Standard Model*, December 2010.

[ITU88] CCITT THE INTERNATIONAL TELEGRAPH AND TELEPHONE CONSULTATIVE COMMITTEE: *X.509 : The Directory - Authentication framework (11/88)*, November 1988. https://www.itu.int/rec/T-REC-X.509-198811-S/en.

[JCT98] J. Callas, L. Donnerhacke, H. Finney and R. Thayer: *OpenPGP Message Format*, November 1998. https://www.ietf.org/rfc/rfc2440.txt.

[KNJ08] Kiltz, E., Gregory Neven and M. Joye: *Identity-Based Signatures*. Journal of Cryptology - JOC, 2, January 2008.

[Kom04] Komar, Brian und das Microsoft PKI-Team: *Microsoft Windows Server 2003 PKI und Zertifikatsicherheit*. Microsoft Press, Unterschleißheim, 2004. ISBN: 3-86063-973-0.

[LQ03] Libert, Benoît and Jean-Jacques Quisquater: *Efficient Revocation and Threshold Pairing Based Cryptosystems*. In *Proceedings of the Twenty-second Annual Symposium on Principles of Distributed Computing*, PODC '03, pages 163–171, New York, NY, USA, 2003. ACM.

[Mar15] Martins, Flavio: *Four Critical Components of Certificate Lifecycle Management*. Digicert, July 2015. https://www.digicert.com/blog/four-components-certificate-lifecycle-management/.

[Mic13] Microsoft Azure: *Windows Azure Service Disruption from Expired Certificate*, February 2013. `https://azure.microsoft.com/en-us/blog/windows-azure-service-disruption-from-expired-certificate/`.

[Ofi19] Ofinno LLC: *5G Cellular Technology for Connected Cars Receives Boost from European Legislators*, October 2019. https://ofinno.com/news/5g-cellular-technology-for-connected-cars-receives-boost-from-european-legislators/.

[PCW15] PCWorld: *Expired Google certificate temporarily disrupts Gmail service*, August 2015. `https://www.pcworld.com/article/2906216/expired-google-certificate-temporarily-disrupts-gmail-service.html`.

[RvD16] Russell, Brian and Drew van Duren: *Practical Internet of Things Security*. Packt Publishing, June 2016.

[SH19] Schulzki-Haddouti, Christiane: *Keine Entscheidung im Wettlauf um die Autovernetzung*. heise.de, September 2019. https://www.heise.de/ct/artikel/Keine-Entscheidung-im-Wettlauf-um-die-Autovernetzung-4505861.html.

[Sha85] Shamir, Adi: *Identity-Based Cryptosystems and Signature Schemes.* In *Advances in Cryptology*, pages 47–53, Berlin, Heidelberg, 1985. Springer Berlin Heidelberg.

[SW04] Sahai, Amit and Brent Waters: *Fuzzy Identity Based Encryption.* Cryptology ePrint Archive, Report 2004/086, 2004. `https://eprint.iacr.org/2004/086`.

[the18] thesslstore.com: *Ericsson Outage: Expired Certificate knocks millions of UK mobile phones offline*, December 2018. `https://www.thesslstore.com/blog/expired-certificate-ericsson-o2/`.

[TV15] Tate, Stephen R. and Roopa Vishwanathan: *Expiration and Revocation of Keys for Attribute-Based Signatures.* In Samarati, Pierangela (editor): *Data and Applications Security and Privacy XXIX*, pages 153–169, Cham, 2015. Springer International Publishing.

[Wan19] Wangcheng, Jiang: *C-V2X Enables Intelligent Transportation.* Huawei, February 2019. https://5gaa.org/wp-content/uploads/2019/02/5GAA_Huawei_Presentation-Feb-27-2019.pdf.

[Wik19] Wikipedia: *Heartbleed*, September 2019. `https://en.wikipedia.org/wiki/Heartbleed`.

[WWKH13] Whyte, William, Andre Weimerskirch, Virendra Kumar and Thorsten Hehn: *A security credential management system for V2V communications.* In *2013 IEEE Vehicular Networking Conference*, pages 1–8, December 2013.

[YWRL10] Yu, Shucheng, Cong Wang, Kui Ren and Wenjing Lou: *Attribute Based Data Sharing with Attribute Revocation.* In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '10, pages 261–270, New York, NY, USA, 2010. ACM.