HAPKIDO Deliverable D6.1

Overview of Archetype PKI Building Blocks

TNO Publiek ⟩ TNO 2024 R11266

3 April 2024

ICT, Strategy & Policy
www.tno.nl
+31 88 866 00 00
info@tno.nl

TNO 2024 R11266 – 3 April 2024

Overview of Archetype PKI Building Blocks

HAPKIDO Deliverable D6.1

Author(s)	Alessandro Amadori Ph.D., Michiel Marcus MSc., Dayana Spagnuelo Ph.D.
Classification report	TNO Publiek
Title	TNO Publiek
Report text	TNO Publiek
Number of pages	17 (excl. front and back cover)
Number of appendices	0
Sponsor	NWO
Programme name	Research programme cybersecurity
Programme number	NWA.1215.18.002
Project name	HAPKIDO
Project number	060.43667

Summary

The HAPKIDO (Hybrid Approach for quantum-safe Public-Key Infrastructure Development for Organizations) project is a 5-year research project funded by the NWO. The goal of the project is to study the migration to quantum-safe PKI from several tracks: technical, governance and evolution. Every track is divided into work packages.

Work Package 6 (WP6) focuses on identifying suitable migration strategies for the different sectors in scope of the HAPKIDO project. This document is the first deliverable of work package 6, which presents a structured overview of PKI components and their interdependencies from a technical perspective. This serves as a foundation for a technical approach to developing a migration strategy and serves as input to deliverables 6.2 and 7.2.

Contents

Summary....3

Contents 4

1 Introduction 5

1.1 Context....5

1.2 Methodology 5

1.3 Scope....6

1.4 Deliverable outline....6

2 Background 7

3 Archetype Building Blocks....8

3.1 PKI User 10

3.1.1 End entity 10

3.2 Components 10

3.2.1 Root Certification Authority....10

3.2.2 Intermediate Certification Authority....10

3.2.3 Bridge Certification Authority....11

3.2.4 Registration Authority....11

3.2.5 Validation Authority....11

3.2.6 Time Stamping Authority....12

3.2.7 Policy Authority....12

3.2.8 Archive 13

3.3 Resources 13

3.3.1 Policy and Regulations....13

3.3.2 Certificate Revocation List 14

4 Discussion and Conclusions 15

5 References 16

1 Introduction

1.1 Context

PKIs are a digital infrastructure to establish trust in the connection between a public key and an entity. The security of PKIs largely relies on the security of digital signatures. Currently, almost all deployed PKIs make use of classical digital signatures, that can be forged in a short amount of time by an adversary with access to a cryptographically relevant quantum computer. Quantum-safe digital signatures are conjectured to withstand attacks by such a quantum computer and therefore pose a solution. One downside of these quantum-safe digital signatures is that they have received much less cryptographic analysis than the ones that are currently deployed, so we cannot be certain that no efficient classical or quantum algorithm will be found in the near future that breaks these digital signature schemes. Another issue is the fact that migration to quantum-safe solutions is a multi-phase endeavour. It is unlikely that the entire infrastructure can be migrated in one go, so there will be issues regarding backwards compatibility. A solution to these issues would be to use hybrid cryptography, where both a classical and post-quantum digital signature are used every time a signature is needed. If the focus is on security, then a verifier only accepts a hybrid signature if both signatures verify. If the focus is on backwards compatibility, then a verifier accepts a hybrid signature if either the classical or quantum-safe signatures verify.

This deliverable is part of the HAPKIDO project, which aims to provide a sector-based plan for migration towards hybrid public-key infrastructures. The work package responsible for this deliverable is Work Package 6, which focuses on PKI architectures and architecture-based migration. In order to create sector-based migration plans, the technical interdependencies of the PKI components need to be investigated. There is some literature available that attempts to create an exhaustive overview of PKI components and their interdependencies, but they are often decades old and therefore outdated. The goal of this deliverable is therefore to provide an updated generic overview of PKI components and their technical interdependencies.

The overview resulting from this deliverable provides the foundation for the next deliverables planned for work package 6, which will focus on identifying the components of the PKIs of the four sectors in scope of the HAPKIDO project and defining migration strategies for these sectors based on the technical interdependencies.

Note that that the overall migration strategy of an organisation encompasses more than the technical migration strategy. The sector-based technical migration strategies will serve as necessary input for the overall migrations strategy of an organisation, which takes into account other aspects as well, such as governance. The creation of the overall migration strategies falls under Work Package 7 of the HAPKIDO project. Deliverables 6.1 and 7.1 have been aligned for this purpose.

1.2 Methodology

As the current deliverable serves the purpose of creating a structured overview of PKI components, the research conducted for this deliverable consists of two activities:

1. An extensive literature overview of PKI's and their technical components.
2. Discussions with HAPKIDO partners to retrieve additional input.

The extensive literature overview ensures that our overview of components contains all technical components that have been identified from a theoretical perspective in literature. The discussions with HAPKIDO partners ensure that the final overview in this deliverable represents real-world contemporary PKIs within the scope of this deliverable.

1.3 Scope

The current report focuses on PKI components that can be identified in PKIs with a trust anchor (trusted party), such as hierarchical PKI's. There are other trust models that can be adopted by PKIs, such as the web of trust as used in PGP [1]. Since the HAPKIDO projects covers industry sectors where hierarchical PKI's are the *de facto* standard, decentralised trust models are considered out of scope. The focus of this report is on PKIs with a trust anchor and the components that they comprise. Therefore, when we refer to PKIs in the rest of the report, we specifically mean PKIs with a trust anchor. Specifically, we identify roles/services that can be delegated to separate entities, which introduce interdependencies that will be relevant during migration towards a hybrid PKI .

1.4 Deliverable outline

The current deliverables is structured as follows:

- Chapter 2 contains background information about PKIs that is necessary to understand the overview of components.
- Chapter 3 provides a diagram with the all the identified components and visually indicates their interdependencies. All the components in this diagram and their interdependencies are elaborated on in detail in the supporting text.
- Chapter 4 contains a brief discussion on the research process and findings and contains concluding remarks.

2 Background

A fundamental part of asymmetric cryptography is being able to establish a connection between a public key and an identity. In PKIs, this is done through the use of certificates. A certificate contains information about an entity and their public key and represents this binding. In order for PKIs to be useful in practice, there is a need for entities to trust the authenticity of certificates and the public keys that they contain [2]. This is achieved with the adoption of trust models, such as the ones covered in this work: flat single CA, hierarchical PKI, and cross-certification.

Flat single CA is the simplest form of PKI, it consists of one Root Certification Authority (3.2.1) which issues the certificates to all participants in the PKI. This model works well for a small number of entities, offering a simple and straightforward solution. However, by definition, no other Certification Authorities can be part of a "flat single CA" model, therefore the root CA becomes a single point of failure [3]. Another shortcoming of this model is the fact that it does not scale well for bigger organisations.

Hierarchical PKIs are the traditionally used model for PKIs. In this model multiple Certification Authorities (3.2.2) provide services for the participant entities, and have a hierarchical relationship among themselves, where each CA has only one superior CA [3]. This trust model requires participant entities to trust the Root CA only, which enables them to verify the authenticity of any given certificate issued by the same PKI by following the path between that certificate until their trust anchor, the Root CA. Hierarchical PKIs also have the advantage of handling the compromise of a CA more easily. In case a CA is compromised, its superior revokes the certificate [3]. Processes are needed to re-establish the CA and reissue all certificates below it, but in the meantime entities outside the compromised area can continue with their communication unaffected [3].

Cross-certification is a trust model that aims to establish trust relationships between two or more PKIs [3]. By default, an end entity in one PKI does not have a trust relationship with a PKI they are not involved. This model allows any two entities from different PKIs, which may have different trust models, to verify the authenticity of each other's certificates without the necessity of having another trust anchor, other than their own (root) CA. This is done by having a (root) CA of one PKI certifying a (root) CA of the other PKI, and vice versa, constituting a cross-certification. Cross-certification can also happen between multiple PKIs and one Bridge Certification Authority (3.2.3).

Regardless of the trust model, PKI services can expand beyond the generation of certificates for end entities as presented in section 3.1.1. In what follows we present the archetypical building blocks that most common PKIs are composed of, and their relationship with each other, spanning the various possible trust models.

3 Archetype Building Blocks

Figure 1 presents a diagram with an overview of archetypical PKI Building Blocks. We distinguish between three main types of blocks, representing all possible (1) users, (2) components or (3) resources that can play a role in any given PKI. Users (green box) are the parties benefiting or making use of services provided by the PKI, normally people or devices which possess a certificate, they are further explored in 3.1. Components (blue boxes) are foundational to the functioning of PKIs, they provide services such as registration of users, or generation of new certificates. A few components appear only once per PKI, we call them “singleton”; they are distinguished from other components by the underscore. All components are further explored in 3.2. Finally, resources (yellow boxes) provide valuable supporting documentation or services which can be consulted in the operation of a PKI, and are further explored in 3.3.

All these building blocks interact with each other, sometimes in multiple capacities. We distinguish two types of interactions: technical and administrative. Technical interactions refer to a request or response that is automated and standardized. An example is the generation of new certificates. Technical interactions are represented by blue arrows in the diagram. Administrative interactions are normally performed by persons, or on behalf of legal persons/entities, they can take the form of a less structured interaction, such as the consultation to a policy document. These interactions are represented with red arrows in the diagram. We describe interactions when presenting the building blocks involved in each of them.

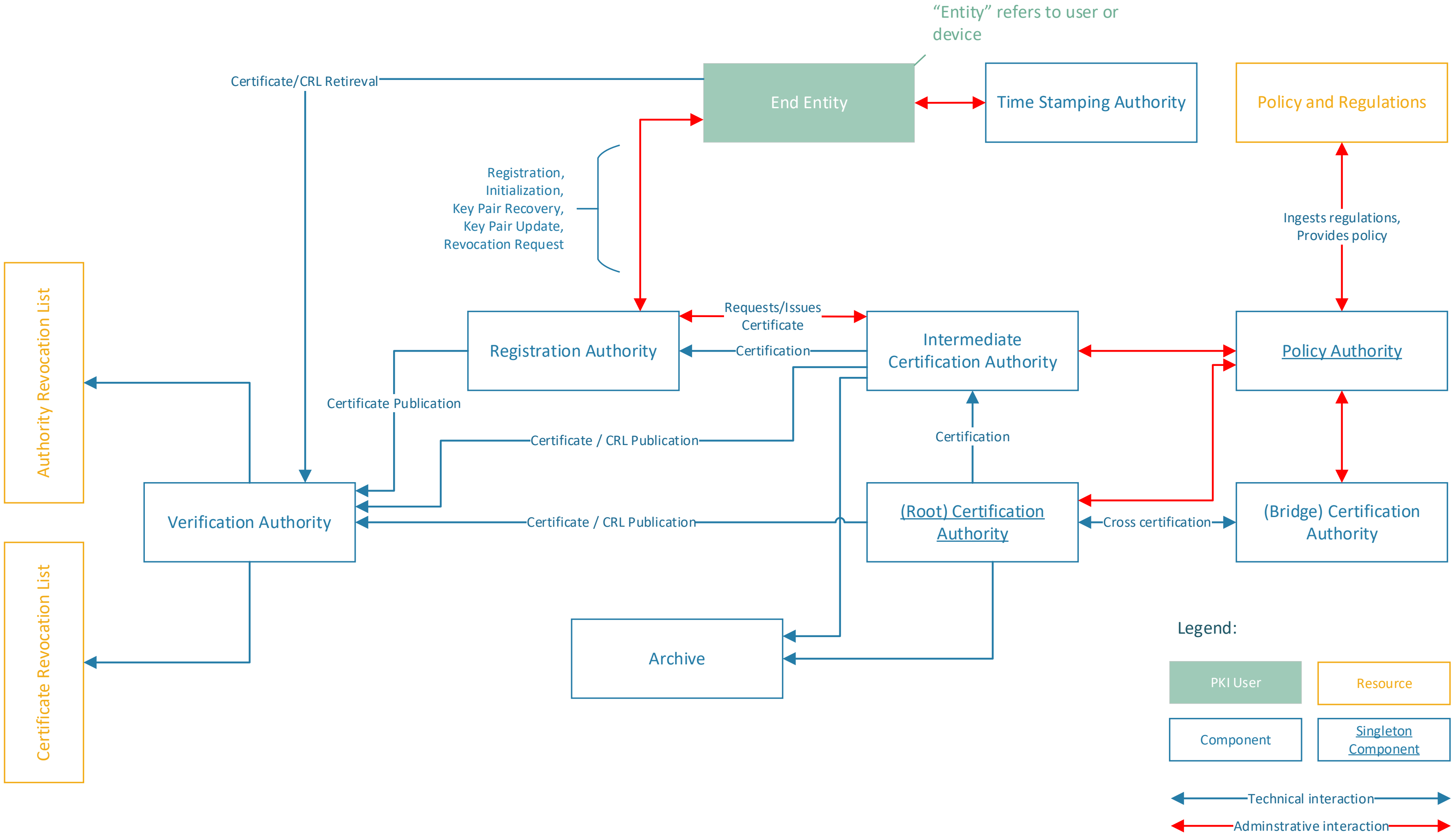

Figure 1 - PKI Archetype Building Blocks

3.1 PKI User

3.1.1 End entity

End entities are the users of PKI which are endpoints in the certification chain. That is, end entities consume or support PKI services [4], but do not (and cannot) issue new certificates [5]. The term "entity" is deliberately generic, because this type of user encompasses humans, devices such as routers or servers, as well as software processes [4]. From a technical perspective, the certificates of any of these entities are virtually identical [5], and end entities which are devices are so common that most people have one or multiple of these devices in their own homes which are part of a PKI. These devices operate autonomously, without people necessarily being aware of their operation.

One distinction is made for how the end entity interacts with the PKI. They can either be a certificate holder, using their certificates and private keys for generating digital signatures and encryption, or relying parties, which use the public key linked to the certificates of others to verify digital signatures and encryption [3]. In practice, several end entities assume both roles of holder and relying party when interacting with PKIs, for instance when authenticating towards a web server over TLS, or a VPN, or even signing and verifying emails [5]. However, in TLS when only authenticating the server (most https connections happen like this) the client's browser assumes the role of relaying party, and only consumes services of retrieval certificates and their statuses.

3.2 Components

3.2.1 Root Certification Authority

The Root Certification Authority is the anchor of trust. It is the one entity that needs to be trusted, otherwise the whole system falls apart. Usually the root CA's certificate is pre-installed in operating systems and browsers. A theoretically minimal PKI would consist of a root CA and end users, where the Root CA registers the end users and creates and manages end user certificates [4].

In a PKI with many users, the Root CA can get overloaded. In this scenario, certain tasks can be delegated to other entities, which will oversee specific responsibilities. The root CA then creates certificates for these entities, so that trust in the root CA translates to trust in the other entities. Another advantage of delegating responsibilities to other entities, is that the root CA does not need an open interface with end users, which reduces the attack surface of the root CA [4]. Therefore, virtually every practical PKI encompasses more entities than the theoretically minimal PKI.

3.2.2 Intermediate Certification Authority

An Intermediate Certification Authority can take up several responsibilities of the root CA. The most important responsibility is the creation and distribution of public key certificates for end users. The intermediate CA can also take the responsibilities of the Registration Authority [4].

There can be multiple intermediate CAs with their own hierarchy, depending on the size of the group of end users. The root CA provides a certificate for the CAs on the next level in the hierarchy. This CA provides a certificate to other CA's on the next level from there, etcetera. This creates a path that can be verified from the root CA [4].

3.2.3 Bridge Certification Authority

A bridge certification authority is a CA that facilitates the connection between different PKIs. For example, if two PKIs in different domains want to connect their domains, then the root CAs can cross-certify each other. For example, each root CA can issue a certificate to the other root CA. They are then bidirectionally cross-certified, which means that end-users from these two domains can now use their own root CA to verify the validity of certificates in the other domain. The two root CA's then act as bridge certification authorities for the two PKIs. Cross certification can also happen on lower levels of the hierarchy, so intermediate CAs can be used instead of root CAs. If a new CA is used that is not part of any of the two PKIs and cross-certifies with both PKIs, then this new CA is referred to as the Bridge CA, since it provides a bridge between two PKIs. The previous example illustrates interdomain cross-certification. It is also possible to have intradomain cross-certification, if there are multiple PKIs that are part of the same domain (for example a government). If the different departments want to cross-certify each department, then the number of cross-certifications that would need to be made would be the square of the number of departments. This introduces a lot of overhead and is not scalable, so instead, one department could be appointed as the intradomain bridge CA. Each department then cross-certifies that CA, which can be used as a bridge between any two departments [4].

3.2.4 Registration Authority

The Registration Authority (RA) is an optional component in PKIs that exists to offload some of the administrative tasks from the CAs, particularly regarding the identity check of end entities [5]. This relationship requires trust. For this reason, an RA is also a child under the same hierarchy of the CA for which it operates [5] [3].

Registration is an important task in which the information that goes into the certificate is verified. This includes information relates to the establishment of an identity, such as name, place of birth, email address, and biometrics, but also the preferences for the certificate, such as the key length, and possible pseudonym. Additionally, other information is collected for contact and billing, which may become relevant in case of legal disputes [2].
Because registration is the main task of the RA, it operates as the front end of the CA, interfacing with the end entities. For end entities who are natural persons (less relevant for end entities which are devices), the presence of multiple RAs in multiple geographical locations facilitates the registration process, as some information will need to be registered physically. For the CA, having RAs interfacing with end entities reduces the attack surface [4].

For each new request by an end entity, the RA generates a Certificate Request. The Certificate Request can be created by the RA on behalf of the end entity, or the end entity can create their own Certificate Request and sign the request with their private key to proof possession. The RA then signs this request and sends it to the CA. The certificate is then generated by the CA, and transmitted back to the responsible RA, which handles its delivery to the end entity [5].

3.2.5 Validation Authority

The Validation Authority (VA) is not a standardised authority in RFC 5280 [6]. However, since it is recognized by respected parties in the PKI field, such as the British NCSC, Entrust and Primekey [7] [8] [9], this concept has been incorporated in this report. The Validation Authority facilitates the process of determining whether an issued certificate is still valid. The VA is responsible for generating, maintaining, updating and publishing certificate revocation information. The CA that issues certificates, communicates certificate status updates to the VA, which in turn updates the revocation information. There are two lists that are tracked: one for the revocation status of certificates of end users, called the Certificate Revocation List (CRL), and one for the revocation status of Certificate Authorities, called the Authority Revocation List (ARL). The VA generally communicates revocation information in one of three ways. The first one is to send the CRL and ARL to any end user that wants to verify the status of a certificate. The VA makes sure they are signed so that the receiver can verify the authenticity of the lists. This is how revocation information was initially communicated, but nowadays it is usually a fall-back scenario for when the other two options are not available.

The second option is to use the Online Certificate Status Protocol (OCSP). Its purpose is to make certificate checking less bandwidth-demanding as the CRL can grow quite large in size. The OCSP is described in the standard [18].

In OCSP, the end user checks if the serial number of the certificate is present in the CRL by sending the serial number to the VA and requesting status information. The possible responses by the VA are:

- Good: there is no revocation entry in the CRL for this specific certificate[1],
- Revoked: the requested certificate is not valid,
- Unknown: the VA cannot process the certificate. This can happen for example if the certificate belongs to a different PKI.

The OCSP response is signed by the VA and sent to the requesting party. This mechanism severely reduces the bandwidth compared to the first option, but it still has the drawback that the VA needs to be online and respond to every single request.

The third option is called OCSP stapling, and it is described in RFC 6066 [19] [20]. In this case, a certificate holder will themselves request a timestamped and signed OCSP response for their certificate at regular intervals from the VA and send this along with their certificate when an end user asks for a certificate. The end user can check the time at which the OCSP response was made and decide whether the interval is short enough to accept the potential risk that the certificate has been revoked within that interval. This eliminates the need for the end user to contact the VA, which can be beneficial in cases where a large number of end users continually request certificate information, for example for websites with a large user base.

3.2.6 Time Stamping Authority

The Time Stamping Authority (TSA) is a type of trusted service provider which is tasked with attesting that a document existed at a certain point in time. This is particularly useful for documents that need long-term archiving [2]. This sort of attestation can be used, for example, to verify the validity of a document signed before the corresponding certificate was revoked, or whenever time of submission of a document is crucial [10].

TSAs are required to use a reliable source of time in order to provide trustworthy attestations. They produce a time-stamp token (with the time, unique identifier, a reference to the policy under which the token was generated, among others) upon receiving a valid time-stamping request. This token is associated with the hash of the data to be stamped. The TSA does not examine the data to be stamped, this means that there is no attestation about the validity of a given signature (other than verifying the request for stamping is valid– in the correct format accepted by the TSA), or the content of the stamped document [10]. TSAs only provide an anchor in time.

3.2.7 Policy Authority

Policy Authorities (PA) are responsible of maintaining Certificate Policies and the Certificate Practice Statement which are the guidelines for issuing, managing and revoking digital certificates. The PA of a

[1] This does however not imply that the certificate is valid. It is still possible that the validity period has expired.

CA also documents if the CA is allowed to operate or interoperate with another PKI and issues procedures by which this determination is made.

An example of a Policy Authority in the Netherlands is PKIOverheid, managed by Logius, which establishes a common framework for trust within the Dutch government. PKIOverheid mandates that the PKI for the Dutch government is divided into two main RAs and ten intermediate CAs, each with specific capabilities and functionalities.

Trust Service Providers (TSP) that operate within PKIOverheid are mandated to follow ETSI standards ETSI EN 319 411-1, ESTI EN 319 411-2, NetSec and PKIOverheid's own Policy and Regulations [11].

3.2.8 Archive

3.2.8.1 Purpose

An archive documents the events that take place within a PKI that are relevant for audits and maintains documents that are required to determine the validity of certificates at a later time [12]. This archive can be maintained by a CA, or it can be delegated to a separate entity. Specifically, information is archived that is required for Webtrust audits. Every CA has to undergo regular audits to obtain a Webtrust seal. These audits need to be executed by a licensed Webtrust auditor. For example, the auditor for the government PKI of the Netherlands (PKI Overheid) is KPMG [13].

3.2.8.2 Audits

The most important themes of a PKI audit include business practices disclosure and management, physical security, and lifecycle management of keys and certificates [14]. The CA business practices disclosure and management part checks whether all required information according to the guidelines is made available and whether policy documents are effective and consistent. The physical security part checks whether only authorised personnel can get logical and physical access to the CA systems and whether continuity of operations and system integrity are sufficiently ensured. The lifecycle management of keys and certificates part checks whether proper authentication controls are used by the CA to protect the integrity of keys and certifications that it manages at generation time and throughout the rest of their life cycle.

3.3 Resources

3.3.1 Policy and Regulations

The goal of policies and regulations is to guarantee stability, minimise risks and increase the trust of the services. They consist of practices supporting PKI services, agreements between PKI parties, certificate and key management rules, responsibilities and managements. These policies are created and maintained by a Policy Authority.

There exist two types of policies within a PKI system and are explained in the memo [12].

The first are called Certificates Policies (CP) and provide the general policies for PKI certificates and guide users to determine if a certificate reaches the desired level of trustworthiness, while the second is the Certificate Practice Statement (CPS) and they provide more detailed description of the practices for certificate life-cycle management. An example of a CPS is provided by KPN [15].

A more in-depth overview on CP and CPS can be found in [16]. The standard mandates that CP and CPS maintain the same structure. CP and CPS can be layered on top of each other. An example is provided by the CA / Browser Baseline requirements for TLS certificates [17]. These requirements can be extended by the specific CA depending on their requirements.

An example of CP is given by the PKIOverheid's Programme of Requirements, which dictates how certificates and the CPS should be populated. For instance, for certificates, it provides a list of fields that must be populated (for example by disallowing the use of pseudonyms), and if some extension fields should be set to "critical" (for example the *extensions:keyUsage:critical* field is required to be set to *TRUE*).
A complete overview of this CP can be found at [11].

3.3.2 Certificate Revocation List

Revocation is part of the life cycle of a digital certificate. Certificates might be compromised or simply no longer needed (certificate updates, owner's private key is compromised, mistakes or bugs, etc.). The certificate revocation list (CRL) is a list of certificates that have been revoked before their expiration date and therefore should not be trusted. The CRL is described in the standard [6],
The CRL is managed by the VA and it is updated periodically with an integrated timestamp. CRL may be updated every hour, day, or week, depending on the frequency the CRLs are issued. To ensure authenticity, the CRL is signed by the VA. CRLs are distributed via CRL Distribution Points.
CRL lists containing revoked certificates of a CA are also called Authority Revocation Lists.
When an end-user wants to inspect the revocation status of a certificate, the end-user requests and downloads the CRL and checks that the serial number of the certificate is not present in the CRL.

.

4 Discussion and Conclusions

During the composition of this overview, the authors discovered that general information about PKIs and their components is very fragmented. In the first attempt to validate some of the literature that was found, another observation was that little information can be found about PKIs that are deployed by organisations. If information was available, it would be fragmented and it would take time to go through all the information to get an overview of the PKI. A diagram as presented in the current deliverable would have helped in interpreting this information in an efficient manner.

In the research that will be conducted for deliverable 6.2, the presented diagram will be validated by applying it to the PKIs of HAPKIDO partners in the various sectors in scope. As an added benefit, this will yield visualisations of these PKIs, which addresses one of the current shortcomings mentioned in the previous paragraph.

5 References

[1] J. Callas, L. Donnerhacke, H. Finney, D. Shaw and R. Thayer, “RFC 4880: OpenPGP Message Format,” IETF Network Working Group, 2007.

[2] J. Buchmann, E. Karatsiolis, A. Wiesmaier and E. Karatsiolis, Introduction to Public Key Infrastructures, Springer, 2013.

[3] R. Housley and T. Polk, Planning for PKI: best practices guide for deploying public key infrastructure, John Wiley & Sons, Inc., 2001.

[4] S. Kiran, P. Lareau and S. Lloyd, “PKI Basics - A Technical Perspective,” PKI-Forum, 2002.

[5] A. Karamanian, F. Dessart and S. Tenneti, PKI Uncovered: Certificate-Based Security Solutions for Next-Generation Networks, Pearson Education, 2011.

[6] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley and W. Polk, “RFC 5280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile,” IETF Network Working Group, 2008.

[7] National Cyber Security Center, “Components of a PKI,” [Online]. Available: https://www.ncsc.gov.uk/collection/in-house-public-key-infrastructure/introduction-to-public-key-infrastructure/components-of-a-pki. [Accessed 11 07 2024].

[8] PrimeKey, “What is a Validation Authority?,” [Online]. Available: https://www.primekey.com/wiki/what-is-a-validation-authority/. [Accessed 11 07 2024].

[9] Entrust, “PKI Validation Authority,” [Online]. Available: https://www.entrust.com/products/pki/validation-authority. [Accessed 11 07 2024].

[10] C. Adams, P. Cain, D. Pinkas and R. Zuccherato, “RFC3161: Internet X. 509 public key infrastructure time-stamp protocol (TSP),” IETF Network Working Group, 2001.

[11] Logius, “PKIoverheid Programme of Requirements 5.2,” Logius, 16 01 2024. [Online]. Available: https://cp.pkioverheid.nl/pkioverheid-por-v5.2.html. [Accessed 24 07 2024].

[12] S. Chokhani, W. Ford, R. Sabett, C. Merrill and S. Wu, “RFC 3647: Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework,” IETF Network Working Group, 2003.

[13] Logius, “Wat is een PKIoverheidcertificaat?,” [Online]. Available: https://www.logius.nl/domeinen/toegang/pkioverheid/wat-een-pkioverheidcertificaat. [Accessed 03 07 2024].

[14] Chartered Professional Accountants CANADA, “Webtrust® for Certification Authorities: Webtrust Principles and Criteria for Certification Authorities v2.2.2,” 2021.

[15] KPN B.V., “Certificate Practice Statement PKIoverheid,” 2020.

[16] P. Consortium, “Policies and Documentation,” PKIConsortium, [Online]. Available: https://pkic.org/pkimm/categories/policies-and-documentation/. [Accessed 27 05 2024].

[17] CA / Browser Forum, “Baseline Requirements for TLS Server Certificates,” [Online]. Available: https://cabforum.org/working-groups/server/baseline-requirements/documents/. [Accessed 24 07 2024].

[18] S. Santesson, M. Myers, R. Ankeny, A. Malpani, S. Galperin and A. C., “RFC 6960: X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP,” IETF Network Working Group, 2013.

[19] DigiCert; QuoVadis, “What is OCSP Stapling?,” [Online]. Available: https://knowledge.digicert.com/quovadis/ssl-certificates/ssl-general-topics/what-is-ocsp-stapling. [Accessed 22 07 2024].

[20] D. Eastlake, “RDC 6066: Transport Layer Security (TLS) Extensions: Extension Definitions,” IETF Network Working Group, 2011.

[21] Chartered Professional Accountants CANADA, “WebTrust Seal Program,” [Online]. Available: https://www.cpacanada.ca/business-and-accounting-resources/audit-and-assurance/overview-of-webtrust-services. [Accessed May 2024].

ICT, Strategy & Policy

Anna van Buerenplein 1
2595 DA Den Haag
www.tno.nl

TNO innovation for life