# Formalizing and Safeguarding Blockchain-Based BlockVoke Protocol as an ACME Extension for Fast Certificate Revocation

Anant Sujatanagarjuna [1,*], Arne Bochem [2] and Benjamin Leiding [1]

[1] Institute for Software and Systems Engineering, Clausthal University of Technology, 38678 Clausthal-Zellerfeld, Germany

[2] Institute of Computer Science, University of Goettingen, 37073 Goettingen, Germany

[*] Correspondence: anant.sujatanagarjuna@tu-clausthal.de

**Abstract:** Certificates are integral to the security of today's Internet. Protocols like BlockVoke allow secure, timely and efficient revocation of certificates that need to be invalidated. ACME, a scheme used by the non-profit Let's Encrypt Certificate Authority to handle most parts of the certificate lifecycle, allows automatic and seamless certificate issuance. In this work, we bring together both protocols by describing and formalizing an extension of the ACME protocol to support BlockVoke, combining the benefits of ACME's certificate lifecycle management and BlockVoke's timely and secure revocations. We then formally verify this extension through formal methods such as Colored Petri Nets (CPNs) and conduct a risk and threat analysis of the ACME/BlockVoke extension using the ISSRM domain model. Identified risks and threats are mitigated to secure our novel extension. Furthermore, a proof-of-concept implementation of the ACME/BlockVoke extension is provided, bridging the gap towards deployment in the real world.

**Keywords:** colored petri nets; blockchain; certificate revocation; formal verification; security; ACME; risk and threat analysis

## 1. Introduction

Certificates are one of the central building blocks of a secure Internet. They authenticate communication partners and are integral to trustworthy communications. However, in many cases, it becomes necessary to revoke valid certificates before their expiration date. Possible reasons include compromise of private keys, domain ownership changes, operational challenges at Certificate Authorities (CAs) and many more. Information about such revocations then has to reach end users to ensure they do not trust a no longer trustworthy certificate. For example, the non-profit CA Let's Encrypt (https://letsencrypt.org/, accessed 20 September 2022) found an issue with their authorization software, which required the revocation of 1.7 million certificates [1–3]. Especially in bulk revocation scenarios, it is important to ensure that the revocation process is reliable, secure and timely.

A standard method to ensure the security, reliability and trustworthiness of security protocols and processes is the use of formal methods [4,5]. By formalizing the protocol using methods like Colored Petri Nets (CPNs) [6], it becomes possible to verify the correctness of a protocol. Going beyond formal verification, it is also essential to analyze security protocols concerning potential risks and threats and mitigate them when detected [7,8].

Garba et al. [9] introduced the BlockVoke protocol, which was designed to allow cost-effective and reliable revocation of certificates; including CA root certificates, even in large numbers. At the same time, it allows near-instantaneous notification of users when revocations occur. Garba et al. also described the advantages of BlockVoke over other existing revocation mechanisms; such as Certificate Revocation Lists (CRLs), the Online Certificate Status Protocol (OCSP) and it's various extensions. In 2021, Sujatanagarjuna et al. [10] formalized and verified the BlockVoke [9] protocol using CPNs. However, the protocol formalization mainly considered the protocol on its own, without much regard

for the PKI ecosystem or consideration of how to embed it organically within the existing frameworks of CAs and other stakeholders.

One possible way of integrating BlockVoke with existing frameworks is to integrate it with the ACME protocol [11]. ACME is used by CAs such as Let's Encrypt to handle several processes in the lifecycle of an X.509 certificate in an automated manner, such as owner verification, issuance and revocation at no cost to end–users. Its introduction has fascilitated the widespread adoption of HTTPS, improving security for end–users. As a consequence, a majority of currently valid browser–trusted certificates on the World–Wide–Web were issued by Let's Encrypt [12]. This makes the ACME protocol a good choice for incorporating the BlockVoke revocation. Furthermore, integrating BlockVoke with the ACME protocol requires minimal changes to existing ACME servers and clients. Such an integration, would allow all stakeholders to benefit from the timely and secure revocation features of BlockVoke. However, such an extension of ACME can introduce new risks and security threats. Therefore, it is essential to formalize the adapted process, perform a formal verification, and secure the BlockVoke/ACME extension from risks and threats.

After BlockVoke's initial description by Garba et al. [9], and formal verification by Sujatanagarjuna et al. [10], this work connects the BlockVoke protocol and previous work with the real world by (1) specifying an extension of the ACME protocol to support BlockVoke as well as (2) formally verifying this extension, (3) mitigating eventual risks and threats and (4) providing a proof-of-concept implementation of BlockVoke integration with an ACME server. Specifically, this work addresses the following research questions:

**RQ** How to describe, formalize and secure the BlockVoke/ACME extension?

    **RQ1** What is the formalization of the BlockVoke/ACME extension?
    **RQ2** What are the security risks and threats of the BlockVoke/ACME extension?
    **RQ3** What are the required modifications to the BlockVoke/ACME extension to mitigate the identified security risks and threats?

To describe, formalize and secure the BlockVoke/ACME extension, we begin with the formalization, perform a risk and threat analysis of the extension and finally perform the required modifications to mitigate any identified risks or threats.

The paper is structured as follows: Section 2 presents supplementary literature and related works, while the BlockVoke/ACME extension is formalized in Section 3. Afterwards, the extension's risk and threat analysis is performed in Section 4, while identified risks and threats are mitigated in Section 5. The results of our evaluation are presented in Section 6. Finally, in Section 7, we give our conclusions and outline possible directions for future research.

## 2. Supplementary Literature and Related Work

This section provides background information, supplementary literature and introduces related work. Sections 2.1 and 2.2 provide a short overview on the BlockVoke protocol as well as ACME, while Section 2.3 describes the integration between BlockVoke and ACME. Sections 2.4 and 2.5 detail formalization approaches as well as security and risk modeling. Section 2.6 focuses on related works.

### 2.1. The BlockVoke Protocol

The BlockVoke protocol [9] was developed for decentralized certificate revocation and rapid, nearly instantaneous, distribution of certificate revocation information that allow certificate owners as well as CAs to revoke certificates. It also facilitates the revocation of CA root certificates which are usually difficult to revoke due to being self-signed. Still, the same technique as described below for regular certificate revocation using BlockVoke can be applied to CA root certificates as well.

As a blockchain-based protocol, BlockVoke utilizes an underlying blockchain to ensure revocation information's continued availability and immutability. While the remainder of the paper assumes a Proof-of-Work (PoW) based blockchain such as the Bitcoin or the

Ethereum blockchain (Before Ethereum switched to Proof-of-Stake (PoS) in September 2022) it is worth noting that the inner workings of the protocol mechanisms of BlockVoke are blockchain-agnostic. For simplicity, we will refer only to Bitcoin for the rest of this work.

Figure 1 presents and details BlockVoke's certificate lifecycle as explained subsequently.
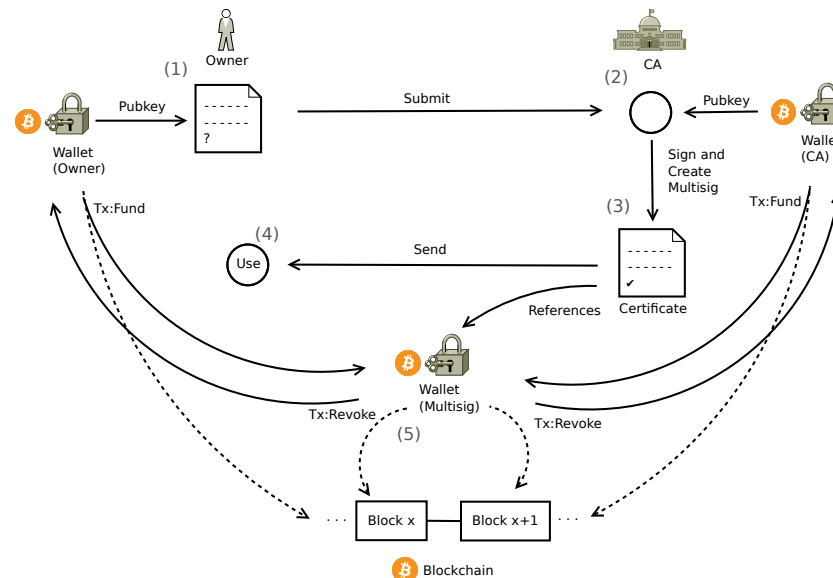


**Figure 1.** The BlockVoke certificate lifecycle—Based on [9,10].

### 2.1.1. Creating and Signing a Certificate

In the first step, the certificate owner (CO) sends a CSR (certificate signing request) to the CA, followed by which an SSL/TLS certificate is created and signed by the CA (step two). In addition to the conventional SSL/TLS certificate information, BlockVoke adds the public key of a Bitcoin address – controlled by the CO – as an additional attribute to the CSR. In step three, a CA-controlled Bitcoin address is used together with the address provided in the CSR and combined into a 1-of-2 multi-signature address, which is subsequently stored in an extension field of the certificate. The corresponding CA may also add the public key used to create the multi-signature address to the extension field to facilitate the combination of BlockVoke with Certificate Revocation Vectors (CRV) [13]. After adding this information, the certificate is signed. The resulting signed certificate may be used (step four) as any regular certificate by the CO.

### 2.1.2. Revoking a Certificate

A BlockVoke-associated certificate is revoked by sending a valid BlockVoke transaction from the multi-signature address specified in the certificate's extension field. A BlockVoke transaction originating from this address indicates a revoked certificate. The multi-signature approach enables both the CO and the CA to revoke the certificate. The certificate in question is revoked by creating a Bitcoin transaction that sends a small amount from the multi-signature address to any arbitrarily chosen other address after first funding the multi-signature address with the required funds to pay transaction costs in a separate transaction.

Moreover, the revocation transaction contains an additional output using Bitcoin's `OP_RETURN` script opcode. The opcode may contain a payload of up to 40 bytes and adheres to the structure presented in Table 1. A Bitcoin revocation transaction with a single input and a single output plus the full `OP_RETURN` payload has an estimated size of 283 Bytes. The `OP_RETURN` payload contains a fingerprint that enables users to verify that the revocation of a given certificate was intended. Apart from the data attributes described above, two more optional attributes allow for using BlockVoke in conjunction with CRVs, i.e., a user

processing the BlockVoke revocation confirms that the CA sent the transaction by matching the transaction's signature against the CA's public key specified inside the certificate.

**Table 1.** `OP_RETURN` payload of a BlockVoke transaction [10].

| Offset | Length | Content |
|--------|--------|---------|
| 0 | 10 | `BlockVoke` |
| 10 | 16 | First 16B of certificate fingerprint |
| 26 | 4 | Date of issuance in days since 2020-02-02 (uint32) |
| 30 | 1 | Revocation reason code according to RFC 5280 [14] (uint8) |
| 31 | 3 | Optional: If CA uses CRV, uint24 of revocation number RN |
| 34 | 6 | Optional: If CA uses CRV, unique CA identifier |

BlockVoke-related transactions are broadcast instantly via the underlying blockchain's P2P network, delivering BlockVoke revocation transactions to users. It is worth noting that consensus is not necessary for the revocations to become valid, i.e., no final transaction confirmation is required. Thus, distributing the transaction via the mempool of unconfirmed transactions is sufficient for users to be notified about the revocation. Once the transaction is mined, it will additionally be persistent and straightforward to look up for any user with access to the blockchain.

### 2.2. Introduction to ACME

The ACME protocol allows a client to use the protocol to request certificate management actions from a CA running an ACME server [11], using standardized JSON objects communicated over HTTPS. These aforementioned actions, such as ACME account creation, certificate issuance and revocation, are provided by an ACME server and can be requested by an ACME client. These two parties are henceforth referred to as the ACME CA and CO, respectively.

The ACME account creation process involves the CO generating a new asymmetric key-pair and requesting the creation of a new ACME account whilst providing information such as contact information and agreeing to the ACME CA's terms of service. Since the ACME CA associates this key-pair with the account, the CO must always sign their requests to the ACME CA with this key–pair.

The certificate issuance process requires the CO to first send an ACME Order to the ACME CA, which contains a list of identifiers, such as domain names, that the CO intends to include in the CSR. Next comes the ACME validation process, wherein the ACME CA issues various challenges, which are responded to by the CO. This process aims for the CO to demonstrate that they are in control of the aforementioned identifiers. Following successful validation, the CO send a CSR to the ACME CA, which subsequently triggers the issuance of the CO's certificate by the ACME CA.

A certificate revocation involves the CO sending a revocation request to the ACME CA, signed using the ACME key pair or optionally the certificate key-pair.

### 2.3. Description of BlockVoke/ACME Extension

The proposed BlockVoke/ACME extension modifies the ACME protocol's certificate issuance and revocation processes to accommodate BlockVoke. To make this possible, users requesting certificates from an ACME CA and the ACME CA itself, must additionally manage Bitcoin wallets; including generating new addresses and issuing transactions to the Bitcoin network .

The original BlockVoke protocol—as illustrated in Figure 1—requires the CO to add their Bitcoin address public key to the CSR. We propose that the CO adds this public key to the list of identifiers provided with the ACME order at the beginning of the certificate issuance. Following this, in addition to any existing ACME validations, the ACME CA creates another validation challenge that contains a securely generated random number. The CO signs this number using the private key associated with their Bitcoin address and

returns this signature to the ACME CA as the challenge response. The ACME CA validates that the number generated was indeed signed by the CO by verifying the signature in the challenge response. Successful completion of this validation process thereby proves that the CO has control of the private key associated with their claimed bitcoin address public key. The ACME CA accepts this public key with the CSR sent by the CO. The ACME CA then proceeds with the certificate issuance process, according to the BlockVoke protocol, as described in Section 2.1.1.

If the CO or ACME CA need to revoke a certificate, the BlockVoke/ACME extension proposes that they follow the BlockVoke revocation process, as described in Section 2.1.2, in addition to the standard certificate revocation process. Including such an extension allows participating ACME clients and servers to benefit from the fast and secure revocation process while still being backwards compatible with traditional PKI.

*2.4. Formalizing BlockVoke*

Colored Petri Nets in particular became popular for formalizing blockchain-based protocols, e.g., [15–17]. Both can be understood as state machines, and mappings from blockchain data structures (tokens) to coloured CPN tokens exists. Moreover, CPN-ML expressions are used to specify and implement data types and operations of the modelled system, which correspond to the functionalities of blockchain smart contracts, e.g., in [18,19] the Authcoin protocol [20] was formalized using CPNs. Similarly to the previous BlockVoke paper [10] that detailed the formalization of the protocol processes, this paper will also utilize CPNs for the very same reason focusing on the ACME/BlockVoke extension in particular. The subsequent paragraphs briefly iterate on the formalization process. For a detailed explanation, we refer the reader to the previous paper on BlockVoke's formalization [10].

Formalizing the BlockVoke protocol requires an appropriate modelling strategy, mapping the existing descriptions of the protocol, as outlined in Section 2.1, to the corresponding elements of a CPN model [6,21]. The resulting sound model allows the consideration of concurrency conflicts, the prevention of dependability issues, and the detection and mitigation of design flaws.

To do so, we first utilize the AOM (Agent-oriented Modeling) methodology to create goal models and corresponding behaviour interface models of BlockVoke. Subsequently, protocol semantics are defined, and the CPN models are derived and implemented using CPN Tools (https://cpntools.org/, accessed 20 September 2022). The AOM methodology allows technical- and non-technical stakeholders to model complex systems by capturing and understanding their functional- and non-functional requirements. A goal model describes the goal hierarchy of the system to be developed, starting with the purpose of the system [22].

Table 2 gives the notation used in AOM goal models. Goal models hierarchically describe the relationship between the various goals in multi-agent systems. As shown, functional goals, often referred to as goals, are represented using parallelograms, and non-functional/quality goals are represented using clouds. The stick figures represent the specific roles of various agents in the system that satisfy the goals that they must fulfil. The relationship between goals themselves and between goals and roles are represented by solid lines, whereas the quality goals associated with goals are connected using dotted lines. In addition to the notations listed in Table 2, a goal with a grey background denotes that its sub-goals are described in another figure.
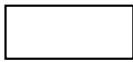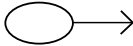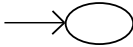
Goal models alone do not represent the various behavioural aspects of the activities and the associated roles that conduct them. Behavioural interface models (BIMs) tabularly represent the activities conducted by the various agents in a multi-agent system in the form of so-called *behavioural interfaces*. A behavioural interface is comprised of an *activity*; the behavioural unit, a trigger; an event that initiates the activity, preconditions; which need to be fulfilled for the activity to take place, and postconditions; which need to be fulfilled for the successful completion of the activity [22].

**Table 2.** Goal model notation (Adapted from [22]).

| Meaning | Goal | Quality Goal | Role | Relationship between goals | Relationship between goals and quality goals |
|---|---|---|---|---|---|
| **Symbol** | ▱ | ☁ | 🧍 | —— | · · · · · · · · · |

Mahunnah et al. [23] describe mapping heuristics from agent models to CPN models based on [22] socio-technical requirements-engineering methodology of agent-oriented modelling. Table 3 illustrates how the various notions of AOM are mapped to the CPN model. Figure 2 details how the components of a behavioural interface can be mapped to its CPN model representation.

**Table 3.** Mapping AOM to CPN (Adapted from [23]).

| Name | CPN Notation |
|---|---|
| Connecting arc | ⟶ |
| Sub–goal or activity | ▭ |
| Trigger or precondition | ⬭⟶ |
| Postcondition | ⟶⬭ |
| Goal | ▭ |
| Precondition(s) | [<conditions>] |
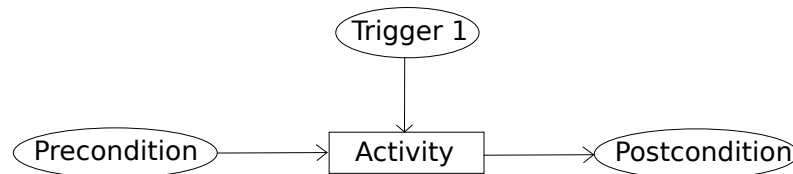


**Figure 2.** Mapping a behavioural interface to a CPN model (Adapted from [23]).

Subsequently, the CPN model of BlockVoke is created using CPN Tools [6] and further used to evaluate and verify the BlockVoke CPN model. "During the enactment of a CPN model, flow of control passes to the sub-goals or activities (in the AOM equivalent) associated with a parent goal represented as a module. This way, a CPN model represents a hierarchical structure of the goal model in AOM" [23]. The behavioural interface model describes the protocol's activities by identifying the activities' triggers, preconditions, and postconditions. The activities detailed in the behavioural interface model are mapped to the transitions of the CPN model. The triggers, preconditions, and postconditions are mapped by adding appropriate places and guards to the CPN model between the transitions.

### 2.5. Security Risk Management and Security Risk-Oriented Patterns

The formalized CPN models are subsequently used to undergo a risk and threat analysis using the ISSRM domain model [7] before mitigating identified risks using security risk-oriented patterns (SRPs) [24,25]. The ISSRM domain model [7,8] is shown in Figure 3. It is based on the extensive analysis [26] of security, risk, and security risk management standards, methods and frameworks and provides a methodology for identifying, evaluating, and quantifying security risks during information systems development. ISSRM considers three key concepts: asset-related, risk-related, and risk treatment-related concepts.

First, asset-related concepts are concerned with important assets that need to be protected and the criteria to guarantee the assets' security, e.g., business assets which characterise the information, processes and skills that have value to the organisation in terms of the business model and are inherently necessary for achieving an organisation's objectives [7,8].

Second, risk-related concepts pertain to risks themselves and their components. Thus, "the risk is a composition of a threat and one or more vulnerabilities that together (i.e., event), lead to a negative impact on two (at least one business asset and at least one information system asset), or more assets. The impact also negates one, or more security criterion. A threat is a potential attack, carried out by a threat agent, who is using an attack method to exploit the vulnerability and who targets information-system assets [7,8,19,26]".

Third, risk treatment-related concepts address corresponding decisions, security requirements and controls.
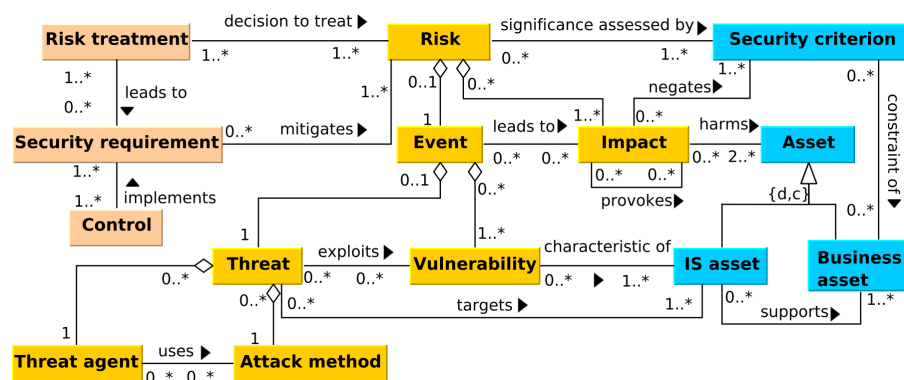


**Figure 3.** ISSRM domain model (adapted from: [7,8,19]).

According to Yoder and Barcalow [27] security patterns address recurring security problems [28] that occur in specific contexts and present well-proven generic schemes of solutions. Therefore, "a security pattern-system is a set of security patterns comprising guidelines for their implementation, combination and practical use in security engineering" [19]. Following the domain model for security-risk management [7,26], the authors of [24,25] list a set of security risk-oriented patterns (SRPs) which pertain to "a specific security context expressed with asset-related concepts, recurring security problems analysed with security-risk related concepts. Related security countermeasures we present with security-risk treatment concepts" [19]. The relevant SRPs for this paper will be introduced in Section 5.1.

### 2.6. Related Works

New Internet security-related protocols are proposed regularly and analysed in terms of security and correctness using different types of formalisms, e.g., $\pi$-calculus [29], Communicating Sequential Processes [30], or Colored Petri Nets (CPNs) [31]. CPNs in particular became popular to formalize and verify blockchain-based protocols, smart contracts and blockchain-based applications, e.g., [15–17,32] and further previous research demonstrated their feasibility for formalising and analysing protocols [33–36]. Most recently, Pinna and Tonelli [37] analysed and classified the application of the Petri net formalism in the design and development of Smart Contracts for modelling as well as verification purposes.

In [35], the authors design and develop CPN models for an authentication protocol and show how CPN is instrumental in creating fully functional models. At the same time, Xu and Xie [36] exemplify the usability of CPN not only to model existing security protocols but also to identify potential attacks through model checking. Additional publications demonstrate the general feasibility of CPN for protocol analysis and verification, e.g., [34,38,39].

Furthermore, the authors of [18,19] first formalized the Authcoin protocol [20] using CPNs and subsequently performed a risk and threat analysis using the ISSRM domain model [7] before mitigating identified risks using SRPs [24,25].

Software patterns and security patterns, in particular, are commonly used in software engineering, e.g., [27,28,40]. In [24], Ahmed and Matulevičius show the applicability of security risk-oriented patterns in different scenarios of the aviation sector, e.g., in [41,42].

The formal verification and subsequent security analysis of security-related protocols using formal methods, as well as the use of security patterns, is a common practice based on the related work provided above. It exemplifies that formal analysis and verification facilitate the development and implementation of secure protocols, prevents incomplete specifications, demonstrates specific protocol properties, and identify and mitigate security and privacy issues identified through a subsequent security analysis of the formal models.

## 3. Formalization of the BlockVoke/ACME Extension

This section formally specifies the BlockVoke/ACME extension using CPNs to answer the research question **RQ1: What is the formalization of the BlockVoke/ACME extension?**

Section 3.1 details the modelling strategy used to arrive at the formal CPN model specification, using the top-level AOM model as an example, followed by Section 3.2 which gives the associated protocol semantics of the top-level CPN model. Finally, Section 3.3 describes the refined sub-modules of the BlockVoke/ACME CPN model.

This section builds on top of and extends the previous paper [10], which focused on the formalization of the BlockVoke protocol without the BlockVoke/ACME extension.

### 3.1. CPN Modelling Strategy

As explained in Section 2.3, the BlockVoke/ACME extension is formalized by mapping its CPN model from AOM models. Mahunnah et al. [23] provide the mapping heuristics required to formally map AOM models to CPN models. The two AOM model types introduced in Section 2.4, namely goal models and BIMs, are identified by the authors as essential tools to capture relevant sociotechnical behavioural features in multi-agent systems such as BlockVoke. In the interest of brevity, the explanation of the mapping process is limited to the top-level functional goals of the BlockVoke/ACME extension. To this end, Sections 3.1.1 and 3.1.2 respectively detail the top-level goal model and BIM of the BlockVoke/ACME extension, followed by Section 3.1.3 which describes the process of deriving the CPN model from these AOM models and finally, Section 3.1.3 gives the derived top-level CPN model of the BlockVoke/ACME extension.

#### 3.1.1. Top-Level Goal Model

The top-level goal model of the BlockVoke/ACME extension is given in Figure 4. The main goal is to *enable secure, timely and privacy-preserving certificate revocation*. It's sub-goals are *Register ACME Account*, *Generate Certificate*, *Verify Certificate* and *Revoke Certificate*.

The refining goal models of the BlockVoke/ACME extension are given in Appendix A.1.

#### 3.1.2. Top-Level Behavioural Interface Model

The top-level behavioural interfaces of the BlockVoke/ACME extension are given in Table 4. The first activity *Register ACME Account* is triggered when the CO wants to register a new ACME account. It requires that the CO has generated an ACME key-pair as a precondition, and its only postcondition is that the ACME CA registers the ACME account. It can also be inferred by the preconditions of the second activity, *Generate Certificate*, that the CO's ACME account must be already registered by the CA. Similarly, the interfaces for *Verify Certificate* and *Revoke Certificate* are also given in Table 4.

The refining behavioural interfaces of the BlockVoke/ACME extension are given in Appendix A.2.
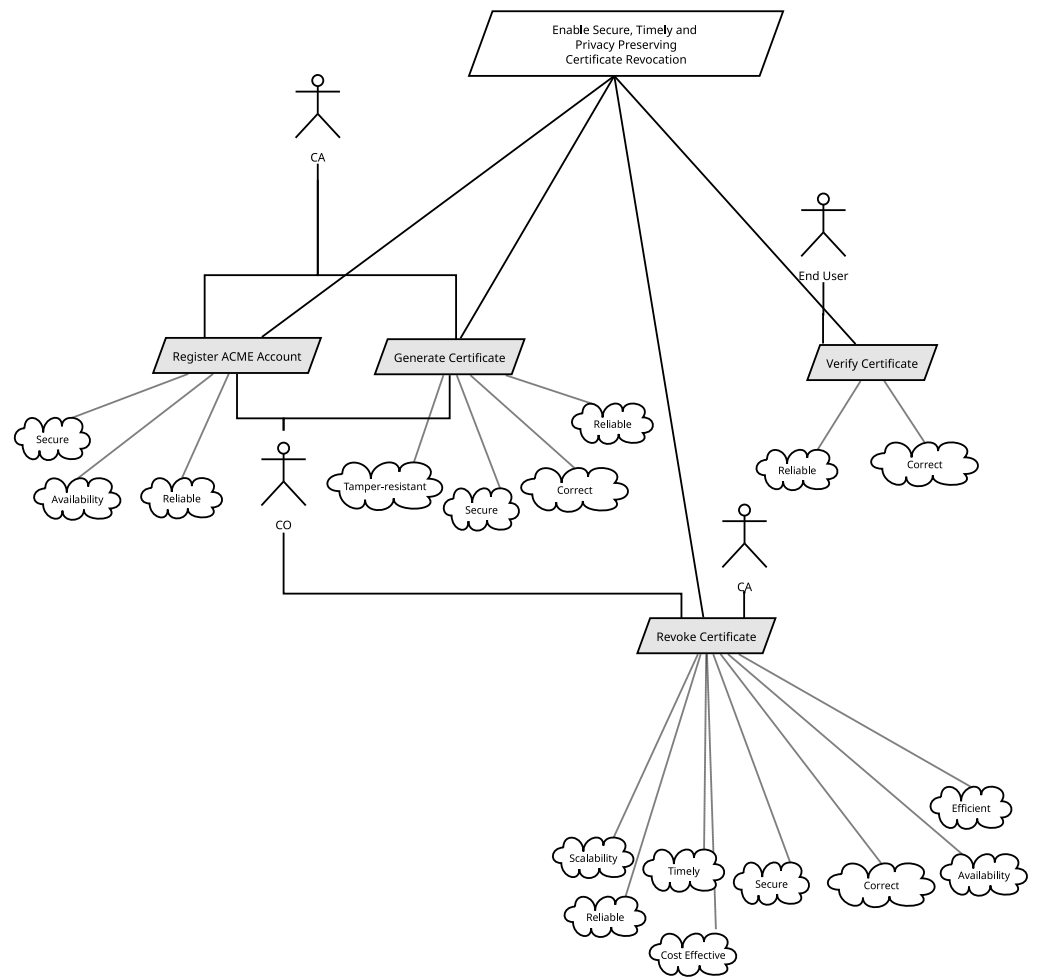
**Figure 4.** Top–Level goal model of BlockVoke/ACME extension.

**Table 4.** Top–level behavioral interface model of the BlockVoke/ACME extension.

| Activity | Trigger | Precondition(s) | Postcondition(s) |
|---|---|---|---|
| Register ACME Account | CO wants to register a new ACME Account | CO has generated an ACME key–pair | ACME Account registered |
| Generate Certificate | CA wants to generate CO's certificate | CO's CSR with information relevant to the certificate, CO's (wallet) public key, CA's signing key pair, CA's (wallet) public key, ACME Account Registered | Generated certificate with CA's signature ready to be verified by the end–user's organization members. |
| Verify Certificate | Certificate ready to be verified by the end user | Generated certificate, CA's public key | Certificate has been verified by an end user. |
| Revoke Certificate | CA or CO wants to revoke a certificate that they have signed/own respectively | Bitcoin wallet with small credit amount, signed certificate, certificate verified, RFC 5280 revocation code, optional CA identifier | Certificate has been revoked. |

### 3.1.3. Mapping AOM to CPN

Figure 5 shows the top-level CPN model of the BlockVoke/ACME extension. The methodology described in Section 2.4 is used to map the four main activities; *Register ACME Account*, *Generate Certificate*, *Verify Certificate* and *Revoke Certificate*, to transitions.
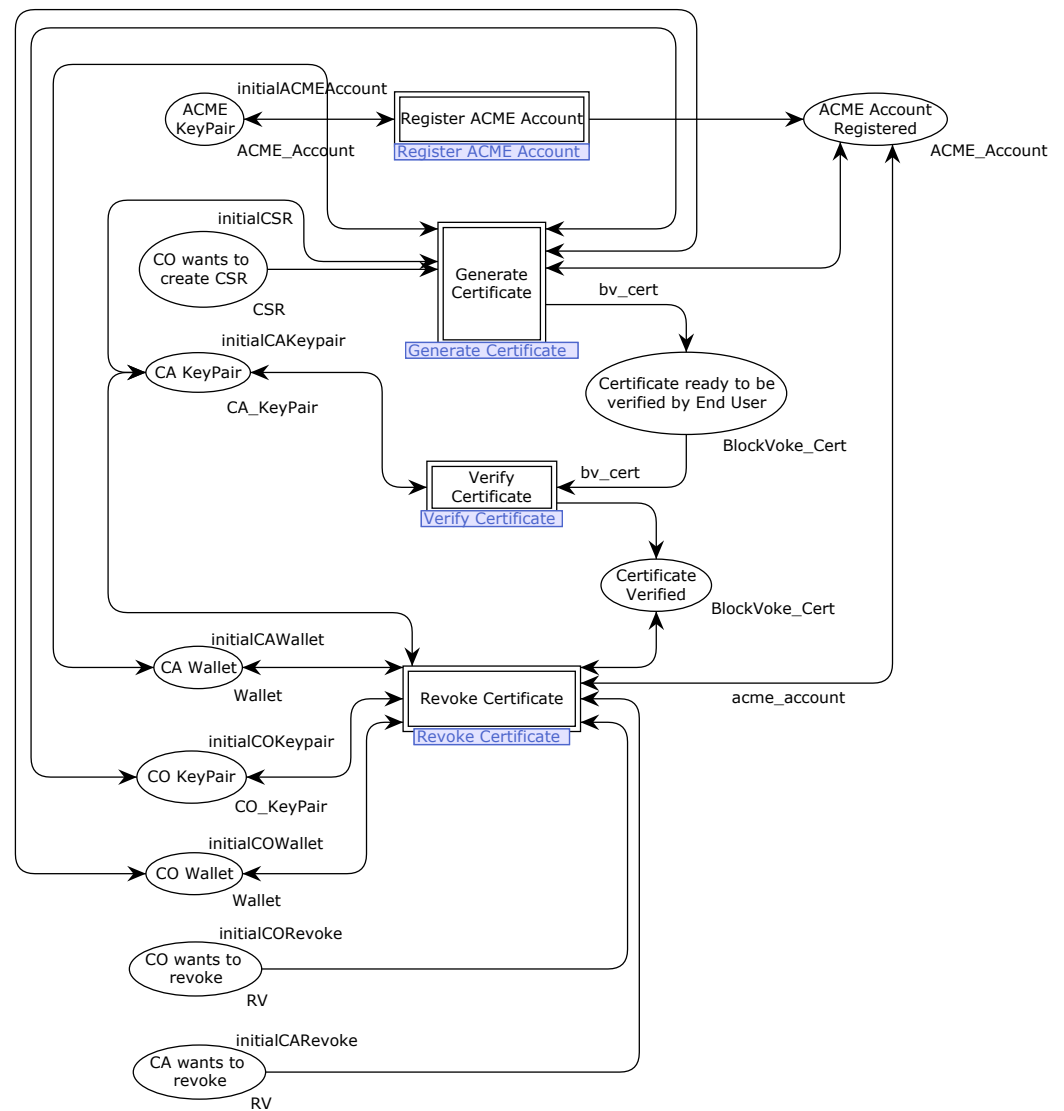
**Figure 5.** Top–Level CPN model.

### 3.2. Protocol Semantics of Top-Level CPN Model

The protocol semantics of the top–level CPN model of the BlockVoke/ACME extension are given in Table 5. Additionally, the initial markings used in the top–level CPN model are given in Table 6.

**Table 5.** Protocol semantics of the top–level BlockVoke/ACME extension CPN Model.

| Name | Type | Description |
|---|---|---|
| Wallet | (Wallet_Addr, Wallet_KeyPair, Wallet_Previous_Hash, Wallet_Balance) | A Wallet |
| BlockVoke_Cert | ColorSet (CO_CN, CO_PublicKey, CO_Key_ID, CO_CN, CO_PublicKey, CO_Key_ID, Cert_Valid_From, Cert_Valid_To, Cert_Multisig_addr, Cert_Sig, Cert_Fingerprint, Cert_DOI | SSL Certificate with extra BlockVoke fields |
| CSR | ColorSet (BlockVoke_Cert, Wallet_Addr) | Representation of a Certificate Signing Request |
| RV | ColorSet (Cert_Fingerprint, Is_CA, Funds, Wallet_Addr, Fees, RFC5280_RevocationCode, Cert_Multisig_addr, Cert_DOI, CA_Key_ID) | ColorSet with information required for a Revocation |

**Table 5.** *Cont.*

| Name | Type | | Description |
|---|---|---|---|
| ACME_KeyPair | ColorSet(RSAKeyPair) | | ACME key pair |
| ACME_Account | ColorSet(ACME_Contact, ACME_Status) | ACME_KeyPair, | ColorSet representing ACME account |
| bv_cert | Variable of color BlockVoke_Cert | | Variable |
| acme_account | Variable of color ACME_Account | | Variable |

**Table 6.** Initial markings of the top–level CPN model of BlockVoke.

| Value Name | Value Declaration |
|---|---|
| initialACMEAccount | 1'("www.co1-website.com", ((3127, 7), (3127, 431)), false)++<br>1'("www.co2-website.com", ((5767, 41), (5767, 137)), false)++<br>1'("www.co3-website.com", ((7387, 7), (7387, 1031)), false)++<br>1'("www.co4-website.com", ((4087, 17), (4087, 233)), false); |
| initialCAWallet | 1'("0x1", ("ca1_wallet_pubkey", "ca1_wallet_privkey"), "0x001", 100) |
| initialCAKeypair | 1'("CA1", (25877, 5), (25877, 20429), "ca1") |
| initialCOWallet | 1'("0x3", ("co1_pubkey", "co1_privkey"), "0x003", 100)++<br>1'("0x4", ("co2_pubkey", "co2_privkey"), "0x004", 100)++<br>1'("0x5", ("co3_pubkey", "co3_privkey"), "0x005", 100)++<br>1'("0x6", ("co4_pubkey", "co4_privkey"), "0x006", 100) |
| initialCOKeypair | 1'("www.co1-website.com", (33017, 7), (33017, 4663), "co1_website")++<br>1'("www.co2-website.com", (83767,13), (83767,6397), "co2_website")++<br>1'("www.co3-website.com", (69451,5), (69451,13781), "co3_website")++<br>1'("www.co4-website.com", (50299,3), (50299,33227), "co4_website") |
| initialCSR | 1'(("www.co1-website.com", (33017, 7), "co1_website", "CA1",<br>(25877, 5), "ca1", "02/02/2021", "02/02/2022", "", 0, 0, ""), "co1_pubkey")++<br>1'(("www.co2-website.com", (83767, 13), "co2_website", "CA1",<br>(25877, 5), "ca1", "03/03/2021", "03/03/2022", "",0, 0,""), "co2_pubkey")++<br>1'(("www.co3-website.com", (69451, 5), "co3_website", "CA1",<br>(25877, 5), "ca1", "04/04/2021", "04/04/2022", "", 0, 0,""), "co3_pubkey")++<br>1'(("www.co4-website.com", (50299, 3), "co4_website", "CA1",<br>(25877, 5), "ca1", "05/05/2021", "05/05/2022", "", 0, 0,""), "co4_pubkey") |
| initialCORevoke | 1'(1803, false, 10, "0x4", 1, "unused", "0xmultisig2",<br>"12.02.2021", "ca1") |
| initialCARevoke | 1'(1825, true, 10, "0x1", 1, "cACompromise", "0xmultisig4",<br>"12.02.2021", "ca1") |

### 3.3. Refining CPN Models

Figures 6 and 7 respectively give the CPN sub-modules for *Register ACME Account* and *ACME Validation*. Similar to the previous formalization of BlockVoke by Sujatanagarjuna et al. [10], the CPN model of the BlockVoke/ACME extension also uses non-cryptographic hashes and RSA signatures. In addition to their use in simulating the signing and subsequent verification of certificates, symbolic RSA keys are also used to simulate ACME key-pairs and the Bitcoin address key-pairs. The ACME key-pairs are used to sign and verify ACME registration requests and ACME orders. The Bitcoin address key-pairs are emulated via RSA key-pairs, in order to simulate the validation process in the *ACME Validation* CPN sub-module shown in Figure 7.
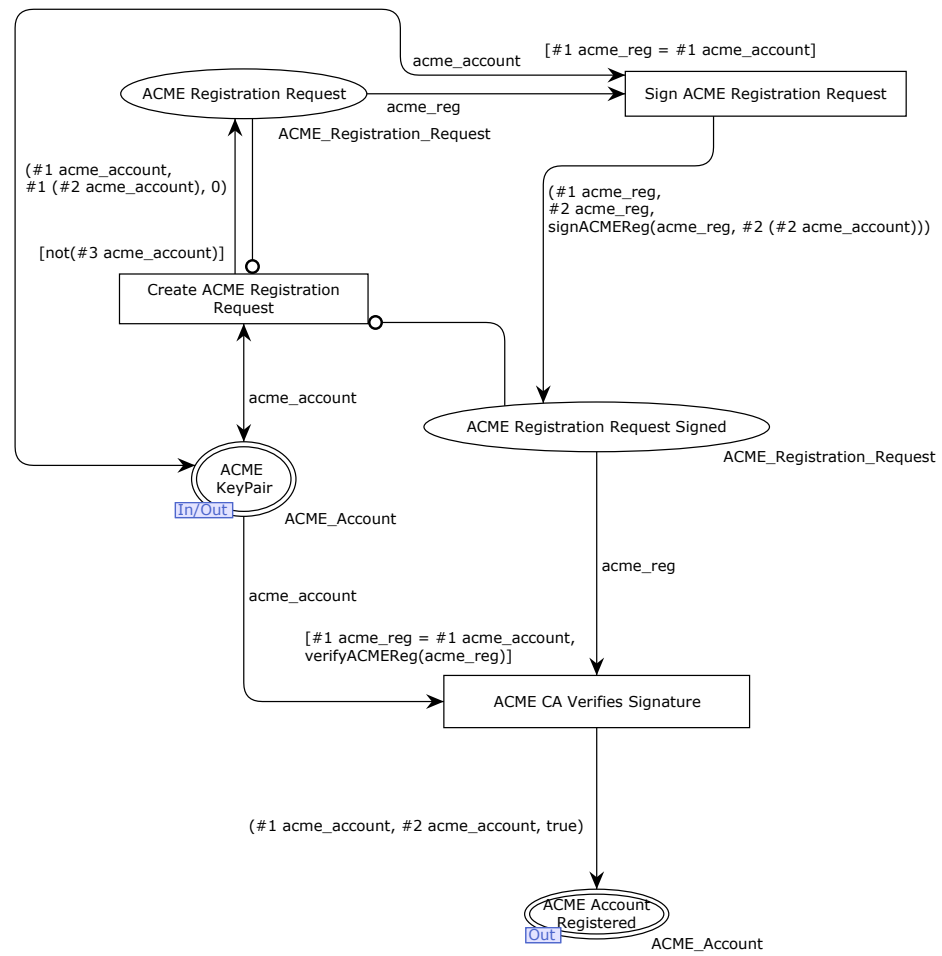
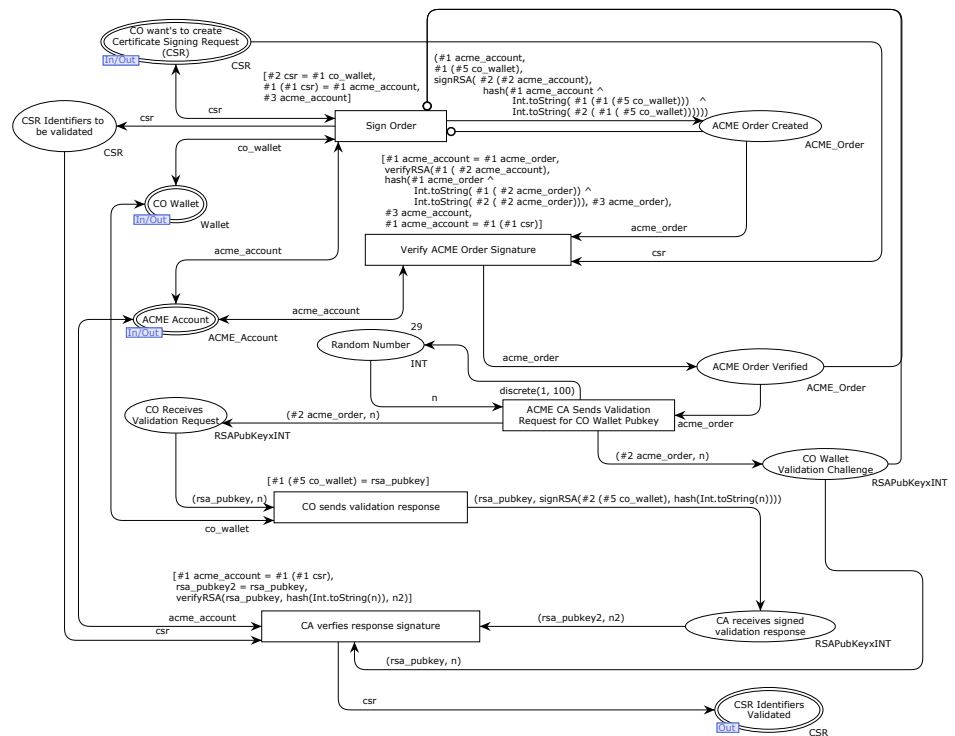**Figure 6.** *Register ACME Account* CPN sub–module.



**Figure 7.** *ACME Validation* CPN sub–module.

The remaining refining CPN models are given in Appendix A.3.

## 4. Risk and Threat Analysis of the BlockVoke/ACME Extension

This section answers the research question **RQ2: What are the security risks and threats of the BlockVoke/ACME extension?**, following the ISSRM domain model. This is done by first identifying the specific assets in Section 4.1, including involved systems, processes and exchanged data objects, followed by identifying the risks and threats associated with these assets in Section 4.2.

### 4.1. Identification of Assets from CPN Model

The asset-related concepts defined in the ISSRM domain model were introduced in Section 2.5. The identification of security risks of the BlockVoke/ACME extension; as formalized in Section 3, first requires identifying the involved assets from the formalized CPN model based on the ISSRM domain modelling Figure 3.

The assets are divided into systems and processes; identified in Section 4.1.1, and the exchanged data objects; identified in Section 4.1.2.

#### 4.1.1. Identification of Systems and Processes

The systems involved in the BlockVoke/ACME extension, as identified from the CPN model as follows:

- **ACME Protocol**: This is the primary protocol which the CO and ACME CA use to communicate with each other at various points during the certificate's lifecycle. The ACME protocol has been described in Section 2.2.
- **CA Certificate Communication System**: The protocol(s) by which an end-user obtains the certificate of an ACME CA with whom they have implicit trust. These protocols are usually dependent on the end-user operating system, the end-user browser, etc.
- **CO Certificate Communication System**: The protocol(s) by which an end-user obtains the certificate of a CO with whom they do not have implicit trust. These protocols are also usually dependent on the end-user operating system, the end-user browser, etc.
- **Blockchain**: This term encompasses the various protocols of the blockchain network; used for communication of new transactions, blocks, etc.
- **User devices and software**: These systems can be classified as follows:
  - **Certificate-related cryptographic software**: These systems include software used for CSR or certificate generation and signing by COs and CAs and for certificate revocation by the CO. These systems also involve the ACME clients used by the COs to communicate with the ACME CA and manage their certificates.
  - **Blockchain-related software**: These systems include software used by COs and CAs; for instance, to generate addresses, create and sign transactions, etc. The end-user also uses similar software to operate a full node to be aware of new blocks and transactions.

Some of the aforementioned systems have significant implications in the BlockVoke protocol and the BlockVoke/ACME extension; but they cannot, however, be reasonably secured within the context of BlockVoke. For instance, compromised user devices and software can have a large impact on the security of various secrets stored on those devices, such as private keys. The underlying protocols of the blockchain systems are also not under the purview of the BlockVoke/ACME extension.

*User devices and software* and the *Blockchain* systems are hence excluded from this risk and threat analysis in favour of the *CA Certificate Communication System*, *CO Certificate Communication System* and parts of the *ACME Protocol* that have been modified/extended to support BlockVoke.

The processes involved in the BlockVoke/ACME extension are identified as follows:

1. CSR Generation

2.    Certificate Generation
3.    Certificate Verification
4.    Transaction Creation
5.    Transaction Scrutinization
6.    Mine Transactions
7.    Add Transactions to Mempool
8.    Mark Certificates as Revoked
9.    Register ACME Account
10.   ACME Order Sending/Receiving
11.   ACME Validation
12.   ACME Revocation
13.   Transaction Sending/Receiving
14.   Certificate Sending/Receiving
15.   Block propagation

Some of these processes, namely Register ACME Account, *ACME Order sending/receiving*, *Transaction Creation*, *Transaction Scrutinization*, *Mine Transactions*, *ACME Revocation*, *Add Transactions to Mempool* and *Block propagation*; similar to those earlier, cannot be secured within the context of BlockVoke since no changes are made to their standard usage scenarios for incorporating them for the BlockVoke protocol. These processes are: hence excluded from this analysis.

The processes subject to risk and threat analysis are: *CSR Generation*, *Certificate Generation*, *Certificate Verification*, *Transaction Sending/Receiving*, *ACME Validation*, *Certificate Sending/Receiving* and *Mark Certificates as Revoked*.

4.1.2. Identification of Exchanged Data Objects

Following the identification of relevant systems and processes, this section identifies the associated exchanged data objects from the BlockVoke/ACME extension CPN model and details them in asset identification tables. Each table lists the business asset (exchanged data object), IS assets (relevant processes), and the processes' descriptions and required security criteria.

The assets identified are: The *ACME Account*, *ACME Validation Challenge*, *ACME Validation Response*, *ACME Revocation Request*, *CSR*, *Certificate*, and *Transaction*. Detailed descriptions of these assets are relegated to Appendix B.

*4.2. Risk and Threat Identification*

The identification of the assets involved in the BlockVoke/ACME extension allows systematic identification and analysis of the risks that threaten these assets. This section identifies these risks and briefly describes them. More detailed descriptions, including the related threat agents, attack methods, threats, vulnerabilities, events, and impacts—all concepts whose definitions and relationships are defined by the ISSRM domain model in Section 2.5—can be found in Appendix B. The nine risks identified are as follows:

1.   *ACME Validation Challenge/Response Modified*
2.   *ACME Validation DDoS*
3.   *Malicious CO*
4.   *Malicious CA*
5.   *Certificate Modified*
6.   *Certificate DDoS*
7.   *Transaction Modified*
8.   *End–User new revocations modified*
9.   *End–User new revocations DDoS*

The risk titled *ACME Validation Challenge/Response Modified* pertains to the risk of both ACME validation challenges and responses being modified by a man-in-the-middle attack. At the same time, they are being requested by and sent to the CO from the ACME CA.

Similarly, *ACME Validation DDoS* is the risk of a DDoS attack on the ACME CA, preventing the ACME validation process from proceeding and consequently generating a certificate by the CA. While both these risks threaten the ACME validation process, it can be observed that similar attack methods can also be used to threaten other ACME processes.

The risks titled *Malicious CO* and *Malicious CA* describe risks relating to the CO or CA manipulating the Bitcoin address public key of the CO or the generated multi-signature address, respectively. While the latter can result in an un-revocable certificate generated by the CA, the former can also result in a remote code execution (RCE) attack.

The two risks, *Certificate Modified* and *Certificate DDoS* describe risks threatening generated certificates being communicated to the end-user's organization. These risks are similar to those of the ACME validation challenge/response objects.

The risk titled *Transaction Modified* describes the risk of the revocation transactions being sent by the CO/CA being intercepted and modified, making them invalid, or discarded. This could thereby prevent certificates from being revoked using the BlockVoke protocol.

Finally, *End-User new revocations modified* and *End-User new revocations DDoS* describe the risks that prevent accurate revocation information from being reliably communicated by the end-user to the members of their organization. These risks are also similar to their similarly named counterparts for the ACME validation challenges/responses and the certificate itself; as mentioned earlier.

## 5. Risk and Threat Mitigation of the BlockVoke/ACME Extension

This section answers the research question **RQ3: What are the required modifications to the BlockVoke/ACME extension to mitigate the identified security risks and threats?** This requires, firstly the identification of applicable SRPs in Section 5.1, secondly the identification of the security requirements and controls in Section 5.2, and finally, in Section 5.3, the application of the identified SRPs to achieve a formal specification of the BlockVoke/ACME extension with all identified risks being mitigated.

### 5.1. Identification of SRPs

From the SRPs developed by Naved Ahmed et al. [43], *SRP 1: Securing data transmission*, *SRP 2: Ensuring valid data entry* and *SRP 4: Ensuring availability of business service* are identified as applicable to the risks identified in Section 4.

*SRP 1*, summarized in Table 7, ensures secure data transmission of the various business assets; preventing the loss of data confidentiality and integrity [43]. The risks formaly identified that require such prevention, are *ACME Verification Challenge/Response modified*, *Certificate modified*, *Transaction modified* and *End–User new revocations modified*. Due to the public nature of the PKI, establishing a unique transmission medium for every pair of CO, ACME CA and End–User is not a viable option. As a consequence, these risks cannot be completely avoided, and can hence only be reduced by ensuring integrity of the exchanged data objects using an appropriate checksum.

**Table 7.** SRP 1—Mitigation.

| Treatment | Countermeasure | Applicable Risks |
|---|---|---|
| Reduction | Make the data unreadable before transmission | |
| | Ensure integrity using a checksum | *ACME Verification Challenge/Response modified*, *Certificate modified*, *Transaction modified*, *End–User new revocations modified* |
| Avoidance | Change the transmission medium to one that cannot be intercepted | |

As seen in risks *Malicious CO* and *Malicious CA*, invalid data can have a large impact to the ability to revoke a certificate using the BlockVoke protocol. Hence, *SRP 2*, as summarized in Table 8 is chosen to be applied, to ensure that appropriate validation of data occurs before it is subject to the various business processes.

**Table 8.** SRP 2—Mitigation.

| Treatment | Countermeasure | Applicable Risks |
|---|---|---|
| Avoidance | Filter incoming data against validity | *Malicous CO, Malicious CA* |

The risks that remain; namely *ACME Validation DDoS*, *Certificate DDoS* and *End–User new revocations DDoS*, all threaten the availability of various business processes of the BlockVoke/ACME extension. For this reason, *SRP 4*; which prescribes that network packets are restricted by proper router configuration, decentralization and load distribution; as shown in Table 9, is chosen to mitigate these risks.

**Table 9.** SRP 4—Mitigation.

| Treatment | Countermeasure | Applicable Risks |
|---|---|---|
| Reduction | Decentralisation, load distribution and balancing | *ACME Validation DDoS*, *Certificate DDoS*, and *End–User new revocations DDoS* |

*5.2. Identification of Security Requirements and Controls*

Following the identification of SRPs, appropriate risk treatment methods; i.e., security requirements and controls are identified along with the appropriate CPN modules where the controls must be applied.

The use of a checksum to protect the integrity of data in transmission, is the prescribed risk avoidance method for *SRP 1*. Since certificates and Bitcoin transactions are secured with digital signatures, the risks of *Certificate modified* and *Transaction modified* already satisfy these security requirements. Hence, no additional security controls are proposed for these risks. As modelled in the ACME Validation CPN sub–module in Figure 7, the ACME validation process already includes the signature of the CO in challenge response objects. Furthermore, all communication between the CO and any ACME CA, following the ACME specification [11] is secured using SSL/TLS layer encryption. Consequently, for similar reasons as in the previous case, the risk of *ACME Validation Challenge/Response modified* does not require any additional security requirements or controls.

The risk, *End–User new revocations modified* is hence the only remaining risk, for which the identified security requirements and controls, are given in Table 10. The security requirement of ensuring integrity of the new revocations must be fulfilled using the security control of the End–User by adding their signature to the revocation information before forwarding them to the members of their organization. The members, in verifying the signature, can ensure the integrity of the revocation information transmitted to them.

**Table 10.** Risk Treatment: *End–User new revocations modified.*

| Risk Treatment | Risk Reduction |
|---|---|
| Security Requirements | Ensure integrity using a checksum. |
| Controls | End–User signs the revocation information before transmitting to their organisation. |

*SRP 2*: chosen for the risks of *Malicious CO* and *Malicious CA*, can be applied by filtering the data against existing standards of validity. The CO's Bitcoin address public key, is already verified by the ACME Validation process. Specifically, the proposed BlockVoke/ACME extension proposes to have the CO prove that they control the public key so

claimed in the initial ACME Order. Hence, no additional security requirements or controls are necessary for this particular risk.

The security requirements and controls for the remaining risk, *Malicious CA* are given in Table 11. The validity of the multi–signature address in the certificate extension field is proposed to be validated by the CO, prior to the subsequent use of the certificate. In the event that a malicious CA generates a certificate that the CO cannot revoke using the BlockVoke protocol, the CO can simply choose not to use that certificate, while opting for another CA.

**Table 11.** Risk Treatment: *Malicious CA.*

| Risk Treatment | Risk Avoidance |
|---|---|
| Security Requirements | Filter incoming Certificate for validity. |
| Controls | CO Validates Bitcoin multi–signature address in Certificate extension field before use. |

The final SRP, *SRP 4*, deals with risks that threaten the availability of various processes of the BlockVoke/ACME extension; namely *ACME Validation DDoS*, *Certificate DDoS* and *End–User new revocations DDoS*. Although the proposed BlockVoke/ACME extension adds another validation method; namely validation of the CO's address public key, no modifications to the specification governing the protocol by which the validation challenge and response objects are exchanged, is proposed. Hence, no additional security requirements and controls are applied with respect to this risk, since the availability of the ACME CA server is not an aspect that can be secured within the context of the BlockVoke/ACME extension. The security requirements and controls for the risks of *Certificate DDoS* and *End–User new revocations DDoS* are given in Tables 12 and 13 respectively.

**Table 12.** Risk Treatment: *Certificate DDoS.*

| Risk Treatment | Risk Reduction |
|---|---|
| Security Requirements | Reduction of certificate communication medium disruptions. |
| Controls | Decentralisation, load distribution and balancing of the certificate communication medium. |

**Table 13.** Risk Treatment: *End–User new revocations DDoS.*

| Risk Treatment | Risk Reduction |
|---|---|
| Security Requirements | Reduction of disruption in the communication medium used for communicating new revocations from an end–user to their organisation. |
| Controls | Decentralisation, load distribution and balancing of the communication medium used for communicating new revocations from an end–user to their organisation. |

*5.3. Application of SRPs*

The AOM methodology used in Section 3 is used to derive the required modifications to the CPN formalization of the BlockVoke/ACME extension, in order to apply the identified SRPs.

The required modifications to the goal model sub–hierarchy of the BlockVoke/ACME extension that are necessary to accommodate the application of the SRPs, is given in Appendix A.6.

The updated BIM affecting the activities *Communicate Newly signed certificate* and *Communicate Revocation Transactions to Users* is given in Table 14.

The updated AOM goal model and BIM are mapped to the associated CPN sub–modules of the BlockVoke CPN model, shown in Figures 8 and 9. To apply the security

controls listed in Table 11, a symbolic guard condition; `validateCertMultisig(bv_cert)`, is added to the *Communicate Newly Signed Certificate* transition. The security controls listed in Table 10 are applied by ensuring that the end–user uses their own key–pair to sign new revocation information before their transmission to the rest of their organization. These key–pairs are also implemented using the previously mentioned non–cryptographic RSA keys, which allow the simulation of the end–user's organization verification of their signature with every new revocation.
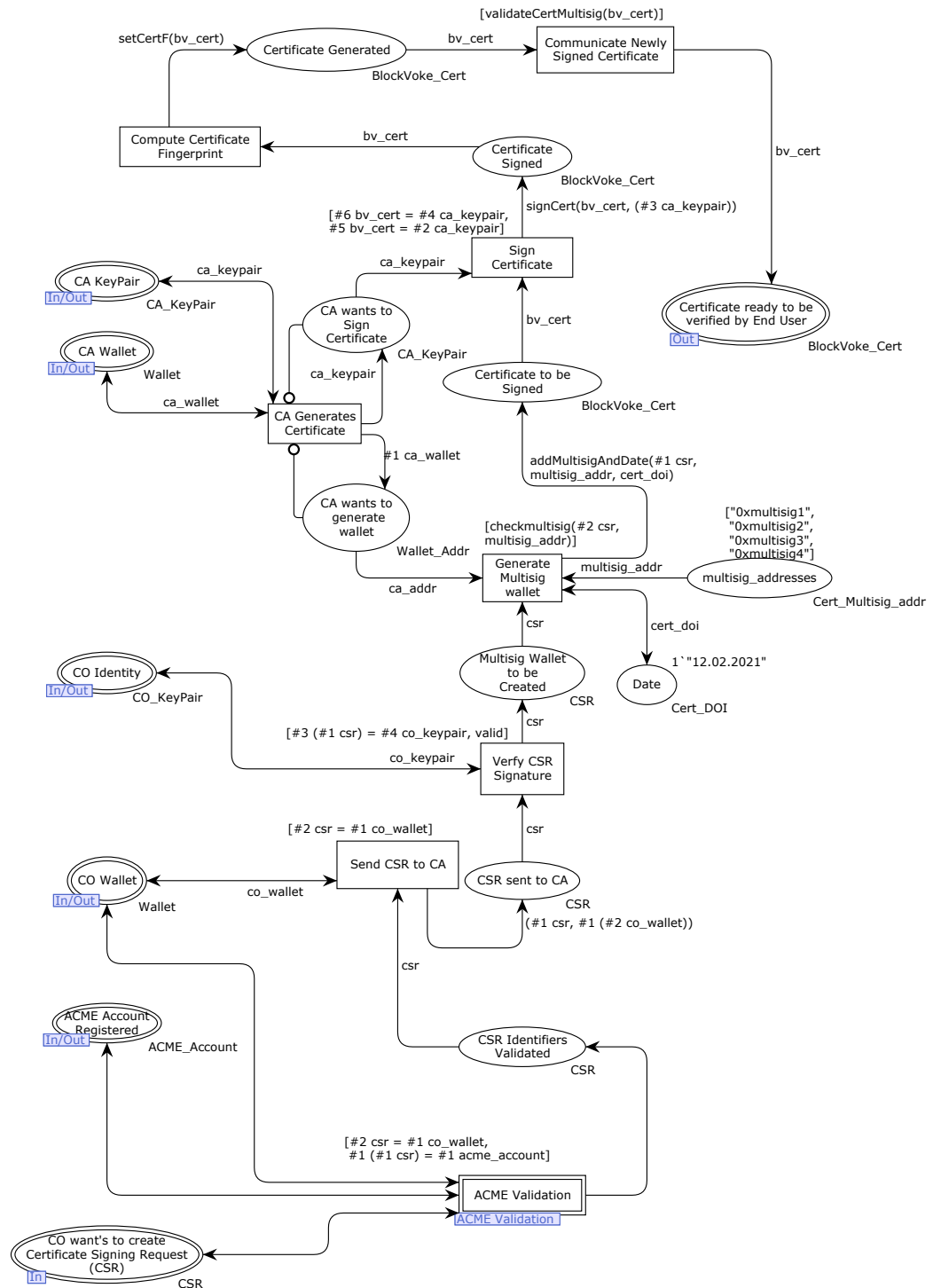


**Figure 8.** Updated *Generate Certificate* CPN sub–module.
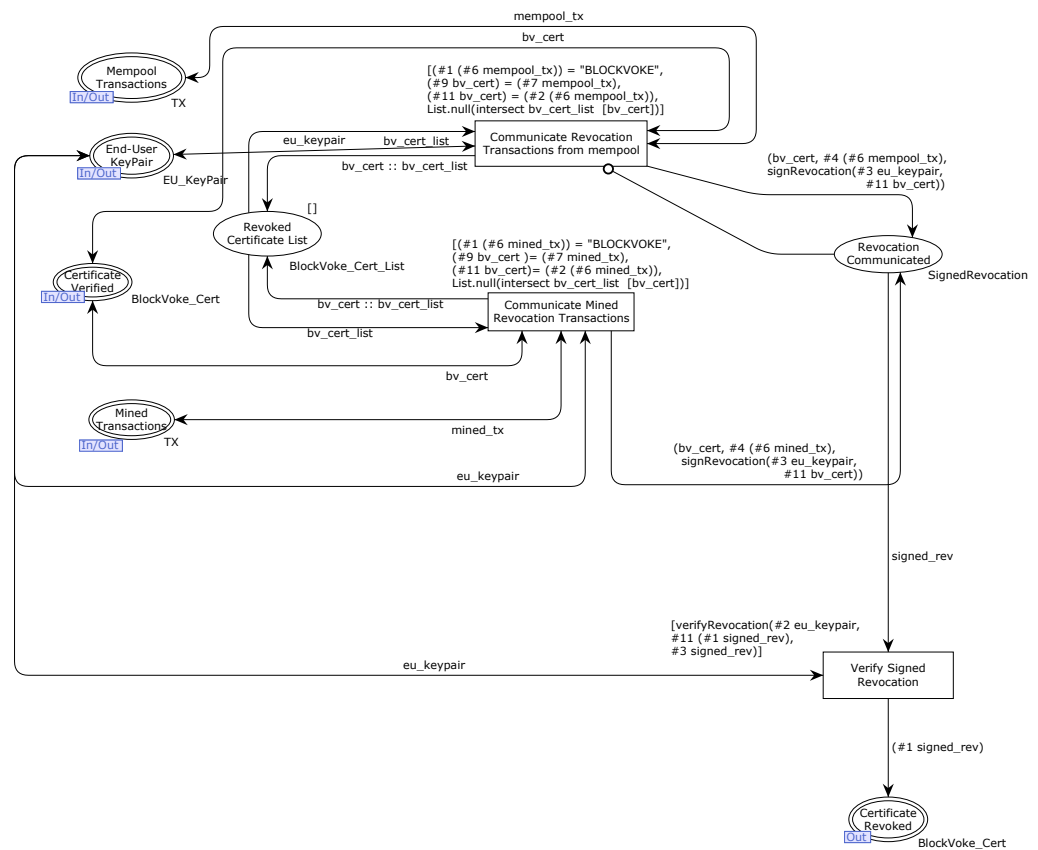
**Figure 9.** Updated *Mark Certificate as Revoked* CPN sub–module.

The updated protocol semantics of the BlockVoke/ACME extension are given in Appendix A.8.

**Table 14.** Updated Behavioural interfaces for new activities, *Validate Certificate Multisignature address*, *Sign new Revocations*, *Verify signed revocations*.

| Activity | Trigger | Precondition(s) | Postcondition(s) |
|---|---|---|---|
| Validate Certificate Multisignature address | Certificate Generated by CA | Certificate multisignature address is valid | Certificate ready to be verified by end–user. |
| Sign new Revocations | End–User wants to send new revocations, such as CRLite filter updates, to their organisation | New Tx:Revoke transaction(s), End–User's private key used to sign new revocation information | New revocation information signed by End–User and communicated to their organisation |
| Verify signed revocations | End–User's organisation receives the signed revocation information | Signed revocation information, Organisation members have End–User's public key | End–User's organisation verify End–User's signature and mark the certificate as revoked |

## 6. Evaluation

In this section, the formalized specification of the BlockVoke/ACME extension given in Section 3 is evaluated using a state–space analysis on the derived CPN model. It is compared to a similar analysis of the CPN model obtained in Section 5, which was derived after applying the required modifications prescribed by the SRPs. Following this, a proof–of–concept (PoC) implementation of the BlockVoke/ACME extension is introduced, along with experimental results obtained by revoking certificates using the BlockVoke protocol over the Bitcoin Testnet.

### 6.1. State-Space Simulation Evaluation

CPN Tools is used to compute and analyse the state–spaces of the CPN models derived in Sections 3 and 5. "The basic idea underlying state–spaces is to compute all the reachable states and the state changes of the CPN model and represent these as a directed graph where nodes represent states and arcs represent occurring events" [6]. "From a constructed state–space, it is possible to answer a large set of verification questions concerning the behaviour of the system such as the absence of deadlocks, the possibility of always being able to reach a given state, and the guaranteed delivery of a given service".

Due to the increased complexity of the CPN model, it is necessary to compute the state–space graphs of some parts of the model independently to prevent a state–space explosion. When a state–space explosion occurs, exponentially large amounts of time and memory are required for the computation. The CPN models are hence divided functionally into two sections. The first section simulates all possible states until all generated certificates are in the *Certificate ready to be verified by End User* place, whereas the second simulates the remaining certificate verification and revocation processes.

#### 6.1.1. Evaluation of BlockVoke/ACME Extension CPN Model before Application of SRPs

Selected state space analysis results for the CPN model derived in Section 3 are discussed in this section. The results for part–1 of the model, including only the top–level transitions *Register ACME Account* and *Generate Certificate*, are given in Table 15 and the results for the remaining model is given in Table 16.

**Table 15.** Selected State–Space Analysis Results of CPN model derived in Section 3—part–1.

| Loops | Home Markings | Dead Markings | Dead Transitions | Live Transitions |
|---|---|---|---|---|
| No | Yes (1) | Yes (1) | No | No |

**Table 16.** Selected State–Space Analysis Results of CPN model derived in Section 3—part–2.

| Loops | Home Markings | Dead Markings | Dead Transitions | Live Transitions |
|---|---|---|---|---|
| No | No | Yes (2) | No | No |

The non–existance of loops implies that no infinite occurrence sequences exist in the computed state–space. This is a desirable property, since it guarantees the protocol's eventual termination. The presence of dead markings in Tables 15 and 16, which are markings where no binding elements are enabled [6], are deliberate measures to prevent indefinite execution of the model. Live transitions, which are transitions for which a containing occurrence sequence can always be found, are also absent from any reachable marking. This is also a desirable quality of the CPN model formalization. The final aspect, namely home markings, are absent from part–2 of the model. A home marking is one that can be reached from any other reachable marking, meaning that it is impossible to have a sequence occur that cannot be extended to reach the home marking. The existance of one such marking in part–1, is a by–product of splitting the CPN model into two for purposes of preventing an exploding state–space.

#### 6.1.2. Evaluation of BlockVoke/ACME Extension CPN Model after Application of SRPs

Tables 17 and 18 give the selected state–space analysis results calculated for the CPN model derived after applying the SRPs in Section 4. The results obtained are identical to those previously listed in Tables 15 and 16. Hence, similar arguments about the quality of the CPN model derived after the application of SRPs can be made.

**Table 17.** Selected State–Space Analysis Results of CPN model derived by applying SRPs in Section 4—part–1.

| Loops | Home Markings | Dead Markings | Dead Transitions | Live Transitions |
|---|---|---|---|---|
| No | Yes (1) | Yes (1) | No | No |

**Table 18.** Selected State–Space Analysis Results of CPN model derived by applying SRPs in Section 4—part–2.

| Loops | Home Markings | Dead Markings | Dead Transitions | Live Transitions |
|---|---|---|---|---|
| No | No | Yes (2) | No | No |

The identical results reported by the state–space simulation indicate that the application of the SRPs does not negatively affect the desired CPN model properties. The complete state–space reports and the partitioned CPN models are given in Appendix C.

6.1.3. Limitations of State-Space Simulation Results

The size of the state–space graphs computed by CPN Tools is heavily dependent on several factors, including, but not limited to, the number of initial markings in the various Places. While, the initial markings were modelled to allow the generation of four certifiates, only two are revoked. However, two simultaneous revocation processes are sufficient for simulating the two distinct methods in which revocation transactions can be witnessed by the end–user—namely, the mempool or the transactions in newly mined blocks. Furthermore, the division of the CPN model, as described in Section 6.1, while inconvenient, does not disturb the simulation of the major sub–processes of certificate generation and certificate revocation. Since, for any given certificate, the processes of certificate generation and revocation are unlikely to overlap in time, this limitation is inconsequential.

*6.2. BlockVoke/ACME Extension Proof–of–Concept Implementation*

A proof–of–concept implementation of the BlockVoke/ACME extension is developed to experimentally determine the average time required for BlockVoke revocation transactions to be witnessed in the mempool, and mined permanently into the blockchain. An overview of the implementation is given in Section 6.2.1, followed by a discussion of the results in Section 6.2.2. The implementation is released under the AGPLv3 free–software license, and can be accessed via https://github.com/ETCE-LAB/BlockVoke-Lets-Encrypt-PoC, accessed 20 September 2022.
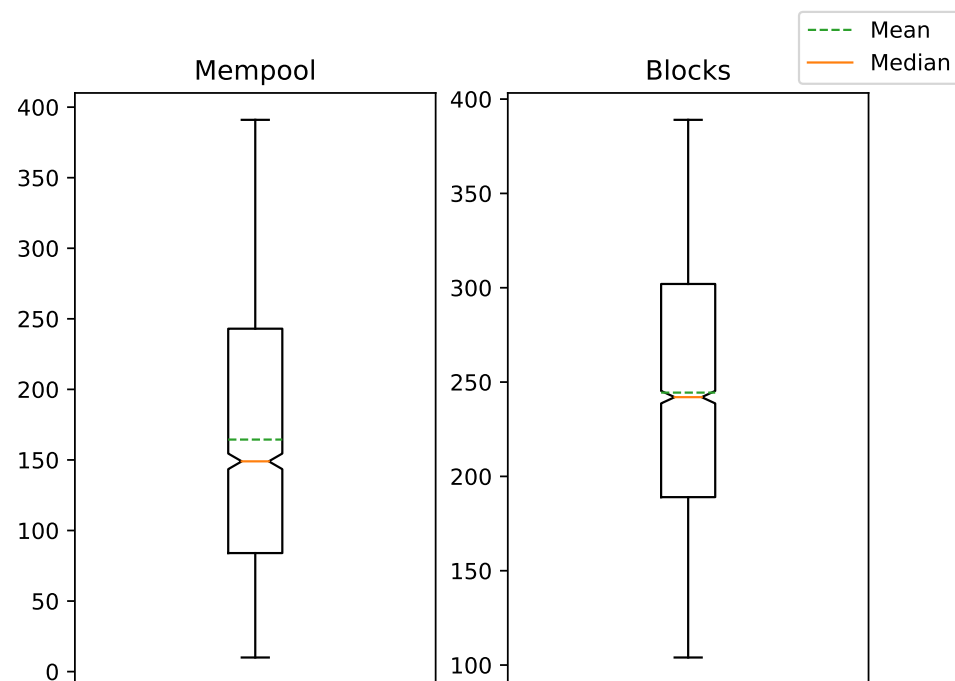
6.2.1. Overview of Proof-of-Concept Implementation

The proof–of–concept implementation was developed as a collection of scripts that allow to facilitate the testing various aspects of the proposed BlockVoke/ACME extension. In addition to these scripts, a modified version of Pebble (https://github.com/ETCE-LAB/pebble/, accessed 20 September 2022)—a miniature version of an ACME server meant for testing purposes—is used to implement the functions of the ACME CA as pertaining to BlockVoke. While Pebble is implemented in Go (https://go.dev/, accessed 20 September 2022), the test scripts are implemented in Python (https://www.python.org/). The Bitcoind (https://en.bitcoinwiki.org/wiki/Bitcoind,accessed20September2022) RPC client is used to manage the various Bitcoin related operations, such as address generation and communicating with the Bitcoin Testnet (https://en.bitcoin.it/wiki/Testnet, accessed 20 September 2022).

6.2.2. Results of Proof-of-Concept Implementation

A total of 4900 certificates were generated for the purpose of testing the time required for their revocation by the BlockVoke protocol. In the interest of unbiased measurements,

two Bitcoin full nodes were connected to the Testnet; the first being used for sending the revocation transactions to the Testnet, while the other listened and parsed new transactions from the mempool and new blocks for BlockVoke revocation transactions. Furthermore, both nodes were connected to the blockchain network from two different geographical locations. While 2036 certificates were revoked from the mempool, the remaining 2864 were detected as revoked via blocks mined within two new blocks on the blockchain, during the course of the test.

Figure 10 gives the time required for a certificate revoked using BlockVoke to be revoked via the mempool or via new blocks. As shown, the majority of certificates were marked as revoked within the first 300 s of their respective pair of revocation transactions being sent to the Testnet. This is a noticeable improvement over CRLs, which often do not expire at a frequency greater than once every 24 h [44]. On the other hand, while OCSP queries have been measured to have a very small median latency of 20 ms [45], a one–to–one comparison with the BlockVoke revocation time cannot be made, since this evaluation measures the time interval between the revocation transactions being transmitted by the CO or CA, and the certificate being marked as revoked by the end–user. Furthermore, OCSP's short latency times are a result of querying the CA's directly, which has the drawback of having significant privacy issues to end–users. The complete test results, including the transaction IDs on the Testnet are given in Appendix D.



**Figure 10.** Time elapsed (in seconds) between transmission of revocation transactions and a certificate to be marked as revoked by the BlockVoke protocol.

### 6.2.3. Limitations of Proof-of-Concept Implementation

While the PoC implementation demonstrates that the proposed BlockVoke/ACME extension has the potential for fast and reliable certificate revocation, one cannot directly infer the expected transction costs on the Bitcoin main network. The Testnet used a fee rate of 1 sat/vByte; the minimum possible rate for a valid transaction. While this does not prevent the certificates from being revoked on the Testnet very quickly, similar guarantees cannot be made for the Mainnet, or for other blockchains.

### 7. Conclusions

An initial description of BlockVoke was introduced by Garba et al. [9] followed by an in-depth CPN-based formal verification by Sujatanagarjuna et al. [10]. Subsequently,

this work makes four novel contributions to the BlockVoke protocol: First, we specify an extension of the ACME protocol to support BlockVoke. Second, we formally verify this extension using the same modelling approach used in previous papers resulting in an extended CPN model of BlockVoke. Third, we conduct and present the risk and threat analysis results of BlockVoke and its ACME extension using the ISSRM domain model to secure BlockVoke and its extension from possible risks and threats. Moreover, we perform the required modifications to mitigate any identified risks or threats. Finally, we provide a proof of concept implementation of the BlockVoke/ACME extension, which integrates it into a working ACME server.

The proposed BlockVoke/ACME extension modifies the ACME protocol's certificate issuance and revocation processes to accommodate BlockVoke, thereby allowing participating ACME clients and servers to benefit from the fast and secure revocation process while still being backwards compatible with traditional PKI. Subsequently, we defined goal models and behavioural interface models of the BlockVoke/ACME extension and the protocol semantics in the form of token colours representing the used data structures. Next, the CPN model of BlockVoke/ACME is derived from the AOM models and the defined protocol semantics using CPN-Tools.

The developed formal CPN model specification is used for a systematic risk and threat analysis of the BlockVoke/ACME extension. This includes the identification of the relevant assets, including systems, processes and exchanged data objects, followed by the subsequent identification of the risks that threaten these assets. This process has resulted in the identification of nine risks that require mitigation.

The identified risks are mitigated by identifying appropriate SRPs, followed by the security requirements and controls prescribed by the identified SRPs. These security requirements and controls are then applied using the AOM methodology to derive the required modifications to the BlockVoke/ACME extension.

The formalised BlockVoke/ACME extension is evaluated using state-space analysis via CPN Tools. The state-space reports of the formal CPN models are used to analyse and compare various characteristics to verify that the derived CPN models satisfy specific desirable properties. A PoC implementation of the proposed BlockVoke/ACME Extension is also proposed. Experimental observations of the PoC implementation in a test scenario have also shown to demonstrate the fast and reliable nature of the BlockVoke and the BlockVoke/ACME extension protocol.

The results of our work have some limitations caused by simplifications and intentionally limiting the scope of the formal BlockVoke model pertaining to the socio-technical nature of the protocol and the modelling process itself, e.g., various limitations of CPNs force the use of symbolic representations of real-world processes. For instance, the generation of the various RSA keys is omitted from the CPN models. The use of these RSA keys is purely symbolic and for simulation purposes. Furthermore, the performed risk- and threat analysis does not guarantee the absence of other undetected risks and security flaws. Further security-related analysis methods and penetration testing might uncover additional risks, threats or incomplete risk mitigations. Moreover, there remain some further limitations to the applied mitigation actions, e.g., some risks, such as those pertaining to the threat of DDoS attacks, are excluded from mitigation due to their being partly irrelevant to the BlockVoke/ACME extension. In addition, the applied security control used to mitigate the risk, *Malicious CA* is only modelled symbolically; due to the computational limitations of CPN–Tools. Finally, the manual and complex pattern detection process requires a good comprehension of the modelled system and thus also poses a challenge.

Future work will focus on further development and integration of BlockVoke and move from an academic proof-of-concept into production-ready certificate management and revocation protocol which can be used in conjunction with services like Let's Encrypt. Besides this, we plan to obliterate the limitations of the CPN model, such as the missing consensus and mining mechanisms, thereby improving the overall quality of the CPN model. Other potential topics of research pertaining to the risk and threat analysis as well

as risk mitigation, e.g., research on the automated occurrence detection of SRPs in a given system model, is preferable to the manual, labour-intensive and error-prone process as described above. Therefore, at least a partially automated support for detection is desirable.

**Author Contributions:** A.S.: Conceptualization, methodology, software, validation, formal analysis, writing—original draft preparation, visualization; A.B.: Conceptualization, methodology, validation, writing—review and editing, visualization; B.L.: Conceptualization, methodology, validation, writing—review and editing, visualization. All authors have read and agreed to the published version of the manuscript.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data presented in this work are openly available via https://github.com/bleidingGOE/2021_BlockVoke-CPN-Files and https://github.com/ETCE-LAB/BlockVoke-Lets-Encrypt-PoC (accessed 20 September 2022).

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A. BlockVoke/ACME Extension Formalization

*Appendix A.1. Goal Model*

https://github.com/xenomorph1096/2022_BlockVoke-Appendix-Files/blob/main/2022-09-20_BlockVoke-Journal-Paper--MDPI-Cryptography/GM.pdf (accessed 29 November 2022).

*Appendix A.2. Behavioural Interfaces*

https://github.com/xenomorph1096/2022_BlockVoke-Appendix-Files/blob/main/2022-09-20_BlockVoke-Journal-Paper--MDPI-Cryptography/BIM.pdf (accessed 29 November 2022).

*Appendix A.3. CPN Model Figures*

https://github.com/xenomorph1096/2022_BlockVoke-Appendix-Files/blob/main/2022-09-20_BlockVoke-Journal-Paper--MDPI-Cryptography/CPNModel.pdf (accessed 29 November 2022).

*Appendix A.4. CPN Model*

https://github.com/xenomorph1096/2022_BlockVoke-Appendix-Files/blob/main/2022-09-20_BlockVoke-Journal-Paper--MDPI-Cryptography/BlockVoke-v06.cpn (accessed 29 November 2022).

*Appendix A.5. Protocol Semantics*

https://github.com/xenomorph1096/2022_BlockVoke-Appendix-Files/blob/main/2022-09-20_BlockVoke-Journal-Paper--MDPI-Cryptography/CPN-Protocol-Semantics.pdf (accessed 29 November 2022).

*Appendix A.6. Updated Goal Model*

https://github.com/xenomorph1096/2022_BlockVoke-Appendix-Files/blob/main/2022-09-20_BlockVoke-Journal-Paper--MDPI-Cryptography/Updated-GM.pdf (accessed 29 November 2022).

*Appendix A.7. Updated CPN Model*

https://github.com/xenomorph1096/2022_BlockVoke-Appendix-Files/blob/main/2022-09-20_BlockVoke-Journal-Paper--MDPI-Cryptography/BlockVoke-v06-AppliedSRPs.cpn (accessed 29 November 2022).

*Appendix A.8. Updated Protocol Semantics*

https://github.com/xenomorph1096/2022_BlockVoke-Appendix-Files/blob/main/2022-09-20_BlockVoke-Journal-Paper--MDPI-Cryptography/Updated-CPN-Protocol-Semantics.pdf (accessed 29 November 2022).

## Appendix B. Risk and Threat Analysis

https://github.com/xenomorph1096/2022_BlockVoke-Appendix-Files/blob/main/2022-09-20_BlockVoke-Journal-Paper--MDPI-Cryptography/RnT.pdf (accessed 29 November 2022).

## Appendix C. State–Space Simulations

*Appendix C.1. State–Space Simulation Report—Part–1*

https://github.com/xenomorph1096/2022_BlockVoke-Appendix-Files/blob/main/2022-09-20_BlockVoke-Journal-Paper--MDPI-Cryptography/ss-v6-p1 (accessed 29 November 2022).

*Appendix C.2. State–Space Simulation Report—Part–2*

https://github.com/xenomorph1096/2022_BlockVoke-Appendix-Files/blob/main/2022-09-20_BlockVoke-Journal-Paper--MDPI-Cryptography/ss-v6-p2 (accessed 29 November 2022).

*Appendix C.3. State–Space Simulation (Updated CPN Model) Report—Part–1*

https://github.com/xenomorph1096/2022_BlockVoke-Appendix-Files/blob/main/2022-09-20_BlockVoke-Journal-Paper--MDPI-Cryptography/ss-v6-SRP-p1 (accessed 29 November 2022).

*Appendix C.4. State–Space Simulation (Updated CPN Model) Report—Part–2*

https://github.com/xenomorph1096/2022_BlockVoke-Appendix-Files/blob/main/2022-09-20_BlockVoke-Journal-Paper--MDPI-Cryptography/ss-v6-SRP-p2 (accessed 29 November 2022).

*Appendix C.5. Partitioned CPN Models*

https://github.com/xenomorph1096/2022_BlockVoke-Appendix-Files/blob/main/2022-09-20_BlockVoke-Journal-Paper--MDPI-Cryptography/BlockVoke-v06-p1.cpn (accessed 29 November 2022).
https://github.com/xenomorph1096/2022_BlockVoke-Appendix-Files/blob/main/2022-09-20_BlockVoke-Journal-Paper--MDPI-Cryptography/BlockVoke-v06-p2.cpn (accessed 29 November 2022).

*Appendix C.6. (Updated) Partitioned CPN Models*

https://github.com/xenomorph1096/2022_BlockVoke-Appendix-Files/blob/main/2022-09-20_BlockVoke-Journal-Paper--MDPI-Cryptography/BlockVoke-v06-p1.cpn (accessed 29 November 2022).
https://github.com/xenomorph1096/2022_BlockVoke-Appendix-Files/blob/main/2022-09-20_BlockVoke-Journal-Paper--MDPI-Cryptography/BlockVoke-v06-p2.cpn (accessed 29 November 2022).

## Appendix D. Proof–of–Concept Implementation Test Results

https://github.com/xenomorph1096/2022_BlockVoke-Appendix-Files/blob/main/2022-09-20_BlockVoke-Journal-Paper--MDPI-Cryptography/TEST_PoC.csv (accessed 29 November 2022).

## References

1. Bugzilla. Bugzilla #1619179—Let's Encrypt: Incomplete Revocation for CAA Rechecking Bug. 2020. Available online: https://bugzilla.mozilla.org/show_bug.cgi?id=1619179#c7 (accessed on 20 September 2022).
2. Jacob Hoffman-Andrews. Let's Encrypt—29 February 2020 CAA Rechecking Bug. 2020. Available online: https://community.letsencrypt.org/t/2020-02-29-caa-rechecking-bug/114591 (accessed on 20 September 2022).
3. JamesLE. Let's Encrypt – Revoking Certain Certificates on 4 March 2020. Available online: https://community.letsencrypt.org/t/revoking-certain-certificates-on-march-4/114864 (accessed on 20 September 2022).
4. Cohn-Gordon, K.; Cremers, C.; Dowling, B.; Garratt, L.; Stebila, D. A Formal Security Analysis of the Signal Messaging Protocol. *J. Cryptol.* **2020**, *33*, 1914–1983. [CrossRef]
5. Kulik, T.; Dongol, B.; Larsen, P.G.; Macedo, H.D.; Schneider, S.; Tran-Jørgensen, P.W.; Woodcock, J. A Survey of Practical Formal Methods for Security. *Form. Asp. Comput.* **2022**, *34*, 1–39. [CrossRef]
6. Jensen, K.; Kristensen, L.M.; Wells, L. Coloured Petri Nets and CPN Tools for Modelling and Validation of Concurrent Systems. *Int. J. Softw. Tools Technol. Transf.* **2007**, *9*, 213–254. [CrossRef]
7. Dubois, E.; Heymans, P.; Mayer, N.; Matulevičius, R. A Systematic Approach to Define the Domain of Information System Security Risk Management. In *Intentional Perspectives on Information Systems Engineering*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 289–306.
8. Matulevičius, R. *Fundamentals of Secure System Modelling*; Springer International Publishing: Berlin/Heidelberg, Germany, 2017.
9. Garba, A.; Bochem, A.; Leiding, B. BlockVoke – Fast, Blockchain-Based Certificate Revocation for PKIs and the Web of Trust. In Proceedings of the International Conference on Information Security, Bali, Indonesia, 16–18 December 2020; pp. 315–333.
10. Sujatanagarjuna, A.; Bochem, A.; Leiding, B. Formalizing the Blockchain-Based BlockVoke Protocol for Fast Certificate Revocation Using Colored Petri Nets. *Information* **2021**, *12*, 277. [CrossRef]
11. Barnes, R.; Hoffman-Andrews, J.; McCarney, D.; Kasten, J. *Automatic Certificate Management Environment (ACME)*; RFC 8555; RFC: Nanjapuram, India, 2019.
12. Aas, J.; Barnes, R.; Case, B.; Durumeric, Z.; Eckersley, P.; Flores-López, A.; Halderman, J.A.; Hoffman-Andrews, J.; Kasten, J.; Rescorla, E.; et al. Let's Encrypt: An Automated Certificate Authority to Encrypt the Entire Web. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. Association for Computing Machinery, London, UK, 11–15 November 2019; pp. 2473–2487.
13. Smith, T.; Dickinson, L.; Seamons, K. Let's Revoke: Scalable Global Certificate Revocation. In Proceedings of the 27th Annual Network and Distributed System Security Symposium (NDSS 2020), Diego, CA, USA, 23–26 February 2020.
14. Cooper, D.; Santesson, S.; Farrell, S.; Boeyen, S.; Housley, R.; Polk, W. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. IETF RFC5280, May 2008. Available online: https://datatracker.ietf.org/doc/html/rfc5280 (accessed on 20 September 2022).
15. Duo, W.; Xin, H.; Xiaofeng, M. Formal Analysis of Smart Contract Based on Colored Petri Nets. *IEEE Intell. Syst.* **2020**, *35*, 19–30. [CrossRef]
16. Rahman, M.S.; Khalil, I.; Bouras, A. Formalizing Dynamic Behaviors of Smart Contract Workflow in Smart Healthcare Supply Chain. In Proceedings of the International Conference on Security and Privacy in Communication Systems, Washington, DC, USA, 21–23 October 2020; pp. 391–402.
17. Liu, Z.; Liu, J. Formal Verification of Blockchain Smart Contract Based on Colored Petri Net Models. In Proceedings of the 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), Milwaukee, WI, USA, 15–19 July 2019; Volume 2, pp. 555–560.
18. Leiding, B.; Norta, A. Mapping Requirements Specifications Into a Formalized Blockchain-Enabled Authentication Protocol for Secured Personal Identity Assurance. In Proceedings of the 4th International Conference on Future Data and Security Engineering—FDSE 2017, Ho Chi Minh City, Vietnam, 29 November–1 December 2017; pp. 181–196.
19. Norta, A.; Matulevičius, R.; Leiding, B. Safeguarding a Formalized Blockchain-Enabled Identity-Authentication Protocol by Applying Security Risk-Oriented Patterns. *Comput. Secur.* **2019**, *86*, 253–269. [CrossRef]
20. Leiding, B.; Cap, C.H.; Mundt, T.; Rashidibajgan, S. Authcoin: Validation and Authentication in Decentralized Networks. In Proceedings of the 10th Mediterranean Conference on Information Systems—MCIS 2016, Paphos, Cyprus, 4–6 September 2016.
21. Jensen, K. Coloured Petri Nets. In Proceedings of the Discrete Event Systems: A New Challenge for Intelligent Control Systems, IEE Colloquium on IET, London, UK, 4 June 1993; pp. 1–5.
22. Sterling, L.; Taveter, K. *The Art of Agent-oriented Modeling*; MIT Press: Cambridge, MA, USA, 2009.
23. Mahunnah, M.; Norta, A.; Ma, L.; Taveter, K. Heuristics for Designing and Evaluating Socio–Technical Agent–Oriented Behaviour Models with Coloured Petri Nets. In Proceedings of the 38th International Computer Software and Applications Conference Workshops, Washington, DC, USA, 21–25 July 2014; pp. 438–443.

24. Ahmed, N.; Matulevičius, R. Securing Business Process Using Security Risk-oriented Patterns. *Comput. Stand. Interfaces* **2014**, *36*, 723–733. [CrossRef]
25. Ahmed, N.; Matulevičius, R. Presentation and Validation of Method for Security Requirements Elicitation from Business Processes. In Proceedings of the Information Systems Engineering in Complex Environments, Selected extended papers from CAiSE Forum 2014, Thessaloniki, Greece, 16–20 June 2014.
26. Mayer, N. Model-based Management of Information System Security Risk. Ph.D. Thesis, University of Namur, Namur, Belgium, 2009.
27. Yoder, J.; Barcalow, J. Architectural Patterns for Enabling Application Security. *Urbana* **1998**, *51*, 61801.
28. Schumacher, M. *Security Eengineering With Patterns: Origins, Theoretical Models, And New Applications*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2003; Volume 2754.
29. Milner, R.; Parrow, J.; Walker, D. A Calculus of Mobile Processes, I. *Inf. Comput.* **1992**, *100*, 1–40. [CrossRef]
30. Hoare, C.A.R. Communicating Sequential Processes. In *The Origin of Concurrent Programming*; Springer: Berlin/Heidelberg, Germany, 1978; pp. 413–443.
31. Jensen, K.; Kristensen, L.M. *Coloured Petri Nets: Modelling and Validation of Concurrent Systems*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2009.
32. Bochem, A.; Leiding, B. Rechained: Sybil-Resistant Distributed Identities for the Internet of Things and Mobile Ad Hoc Networks. *Sensors* **2021**, *21*, 3257. [CrossRef] [PubMed]
33. Basyouni, A.; Tavares, S. New Approach to Cryptographic Protocol Analysis Using Coloured Petri Nets. In Proceedings of the Electrical and Computer Engineering, 1997. Engineering Innovation: Voyage of Discovery, St. John's, NF, Canada, 25–28 May 1997; Volume 1, pp. 334–337.
34. Dresp, W. Security Analysis of the Secure Authentication Protocol by Means of Coloured Petri Nets. In Proceedings of the IFIP International Conference on Communications and Multimedia Security, Salzburg, Austria, 19–21 September 2005; pp. 230–239.
35. Vanek, T.; Rohlik, M. Model of DoS Rresistant Broadcast Authentication Protocol in Colored Petri Net Environment. In Proceedings of the IWSSIP 2010 Proceedings, Rio de Janeiro, Brazil, 17–19 June 2010; pp. 264–267.
36. Xu, Y.; Xie, X. Modeling and Analysis of Security Protocols Using Colored Petri Nets. *JCP* **2011**, *6*, 19–27. [CrossRef]
37. Pinna, A.; Tonelli, R. On the use of Petri Nets in Smart Contracts Modeling, Generation and Verification. In Proceedings of the 2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), Honolulu, HI, USA, 15–18 March 2022; pp. 1207–1211.
38. Al-Azzoni, I.; Down, D.; Khedri, R. Modeling and Verification of Cryptographic Protocols Using Coloured petri Nets. *Nord. J. Comput.* **2005**, *12*, 200–228.
39. Sornkhom, P.; Permpoontanalarp, Y. Security Analysis of Micali's Fair Contract Signing Protocol by Using Coloured Petri Nets: Multi-session Case. In Proceedings of the Parallel & Distributed Processing, Rome, Italy, 23–29 May 2009; pp. 1–8.
40. Yoshioka, N.; Washizaki, H.; Maruyama, K. A Survey on Security Patterns. *Prog. Inform.* **2008**, *5*, 35–47. [CrossRef]
41. Samarütel, S.; Matulevičius, R.; Norta, A.; Nõukas, R. Securing Airline-turnaround Processes Using Security Risk-oriented Patterns. In Proceedings of the IFIP Working Conference on The Practice of Enterprise Modeling, Skövde, Sweden, 8–10 November 2016; pp. 209–224.
42. Matulevičius, R.; Norta, A.; Udokwu, C.; Nõukas, R. Security Risk Management in the Aviation Turnaround Sector. In Proceedings of the International Conference on Future Data and Security Engineering, Can Tho City, Vietnam, 23–25 November 2016; pp. 119–140.
43. Ahmed, N.; Matulevičius, R.; Khan, N.H. Eliciting Security Requirements for Business Processes using Patterns. In Proceedings of the 9th International Workshop on Security in Information Systems, Bordeaux, France, 15 March 2016.
44. Liu, Y.; Tome, W.; Zhang, L.; Choffnes, D.; Levin, D.; Maggs, B.; Mislove, A.; Schulman, A.; Wilson, C. An End-to-End Measurement of Certificate Revocation in the Web's PKI. In Proceedings of the 2015 Internet Measurement Conference, Tokyo, Japan, 28–30 October 2015; pp. 183–196.
45. Basin, D.; Cremers, C.; Kim, T.H.J.; Perrig, A.; Sasse, R.; Szalachowski, P. Design, analysis, and implementation of ARPKI: An attack-resilient public-key infrastructure. *IEEE Trans. Dependable Secur. Comput.* **2016**, *15*, 393–408. [CrossRef]