

UNIVERSITY OF GÖTTINGEN

**Master's Thesis**

**Securing the Authcoin Protocol Using  
Security Risk-oriented Patterns**

Author: Benjamin Leiding

First Examiner: Dieter Hogrefe

Second Examiner: Alexander Horst Norta

March 16, 2017



# *Abstract*

Designing and developing new security and authentication protocols in the field of computer science is a challenging task. Design flaws and missing specifications as well as security and privacy issues of such protocols pose risks for its users. Formal methods, such as Colored Petri Nets, are utilized for the design, development and analysis of such new protocols in order to detect flaws and mitigate identified security risks.

In this thesis, the Authcoin protocol is formalized using Colored Petri Nets in order to detect and eliminate eventual design flaws, missing specifications as well as security and privacy issues. Furthermore, a risk and threat analysis based on the ISSRM domain model is performed on the formal CPN models of the protocol. Subsequently, the identified risks are mitigated by applying security risk-oriented patterns to the formal model of the Authcoin protocol. Security risk-oriented patterns are a means to mitigate common security and privacy risks in processes by applying thoroughly tested and proven best-practice solutions. The goal of this thesis is to reduce the risks and vulnerabilities of the Authcoin protocol using the techniques and approaches mentioned above. In addition, we share the lessons learned during the novel application of security risk-oriented patterns to Colored Petri Nets and evaluate the resulting CPN models using state space analyses.

# *Acknowledgements*

I want to appreciate and thank all the persons that contributed towards the successful completion of this thesis. First of all, I would like to express my gratitude to Alexander Norta who did a tremendous job in supervising this thesis. Alex patiently guided me through this thesis, provided countless ideas and never hesitated to help or provide feedback no matter whether it was day or night. It was a real pleasure to create this thesis under his guidance.

Many thanks to Raimundas Matulevičius for his patience, guidance and feedback throughout the course of this master's thesis and his expertise on security risk-oriented patterns. Furthermore, I would like to thank Dieter Hogrefe as supervisor and first examiner of this thesis for his support and help. I am also deeply indebted to Clemens H. Cap who initiated this cross-border supervised thesis, provided valuable feedback and established the foundations for this work through careful and patient guidance during my time at the University of Rostock.

Most importantly, I have to thank Natascha Jakob who supported me all along the way, not only during the creation process of this thesis, but also over the period of many years. I would like to express my deepest gratitude to Natascha for her love, support and encouragement in times of doubt.

Last but not least, I thank my parents for their support, patience and help during all those years.

# Contents

<b>Abstract</b>	<b>i</b>
<b>Acknowledgements</b>	<b>ii</b>
<b>List of Abbreviations</b>	<b>vi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Existing Body of Knowledge . . . . .	1
1.1.1 Authcoin . . . . .	2
1.1.2 Formal Methods for Protocol Formalization . . . . .	2
1.1.3 Security Risk-oriented Patterns . . . . .	3
1.1.4 Contribution - Detecting a Gap . . . . .	4
1.2 Research Methodology and Research Questions . . . . .	4
1.2.1 Design Science Research - Theory . . . . .	5
1.2.2 Design Science Research - Practice . . . . .	5
1.2.3 Research Questions . . . . .	8
1.3 Thesis Structure . . . . .	9
<b>2 Presuppositions</b>	<b>10</b>
2.1 Authcoin . . . . .	10
2.1.1 Overview . . . . .	10
2.1.2 General Validation and Authentication . . . . .	11
2.1.3 Storing Information and Smart Contracts . . . . .	12
2.1.4 Challenges . . . . .	13
2.1.5 Validation and Authentication Requests - VARs . . . . .	14
2.2 Security Risk-oriented Patterns . . . . .	15
2.2.1 Security Patterns . . . . .	15
2.2.2 Security Risk-oriented Patterns . . . . .	16
2.2.3 Existing Security Risk-oriented Patterns . . . . .	17
<b>3 Formal Specification of Authcoin Using Coloured Petri Nets</b>	<b>18</b>
3.1 Introduction . . . . .	18
3.2 Modeling Strategy . . . . .	19
3.2.1 Goal Model . . . . .	19
3.2.2 Behavior Interface Model . . . . .	20
3.3 Protocol Formalization Using CPNs . . . . .	21

3.3.1	Coloured Petri Nets - CPNs	22
3.3.2	Modeling Protocols Using CPNs	22
3.3.3	CPN Modules	23
3.4	Mapping the Agent-oriented Model to CPN Models	23
3.5	Protocol Semantics	25
3.6	Refined CPN Models	34
3.6.1	Top-Level Modules	34
3.6.2	Sub-Modules of “KeyGenerationEstablishBinding”	37
3.6.3	Sub-Modules of “V&A-Processing”	38
3.6.4	Sub-Modules of “Mining”	46
3.6.5	Sub-Modules of “Revocations”	48
3.7	Discussion	49
3.8	Conclusion	53
<b>4</b>	<b>Risk and Threat Analysis of the Authcoin Protocol</b>	<b>55</b>
4.1	Introduction	55
4.2	Information Systems Security Risk Management (ISSRM) Domain Model	56
4.3	Identification of Assets	57
4.3.1	Systems and Processes	58
4.3.2	Identification of Exchanged Data Objects	59
4.4	Risk and Threat Analysis of Identified Assets	64
4.4.1	Risk and Threat Analysis - Posting to Blockchain	64
4.4.2	Risk and Threat Analysis - Send/Receive	66
4.4.3	Risk and Threat Analysis - DDoS	67
4.5	Risk Treatment	68
4.5.1	Risk Treatment - Posting to Blockchain and Send/Receive	69
4.5.2	Risk Treatment - DDoS	69
4.6	Discussion	70
4.7	Conclusion	70
<b>5</b>	<b>Application of Security Risk-oriented Patterns on the Authcoin Protocol</b>	<b>72</b>
5.1	Introduction	72
5.2	Application of Existing Security Risk-oriented Patterns	73
5.2.1	Identification of Applicable Security Risk-oriented Patterns	73
5.2.2	Identify Occurrences of Security Risk-oriented Patterns	75
5.3	Implementation of Chosen Security Risk-oriented Pattern	76
5.3.1	Updated Goal Model	77
5.3.2	Updated Behavior Interface Model	77
5.3.3	Updated Protocol Semantics	81
5.3.4	Updated CPN Models	83
5.4	Lessons Learned from the Implementation of Security Risk-oriented Patterns in CPN	95
5.5	Discussion	96
5.6	Conclusion	97
<b>6</b>	<b>Evaluation</b>	<b>98</b>
6.1	Introduction	98

6.2	Evaluation of Authcoin's CPN Models Without Security Risk-oriented Patterns . . . . .	99
6.3	Evaluation of Authcoin's CPN Models With Security Risk-oriented Patterns	101
6.4	Related Work . . . . .	103
6.5	Discussion . . . . .	103
6.6	Conclusion . . . . .	104
<b>7</b>	<b>Conclusion and Future Work</b>	<b>106</b>
7.1	Conclusion . . . . .	106
7.2	Answering the Research Questions . . . . .	107
7.2.1	RQ-1 - How to formalize the Authcoin protocol? . . . . .	107
7.2.2	RQ-2 - How to analyze security threats of the Authcoin protocol? . . . . .	107
7.2.3	RQ-3 - How to apply security risk-oriented patterns to the Authcoin protocol? . . . . .	108
7.3	Limitations . . . . .	108
7.4	Future Work . . . . .	109
<b>A</b>	<b>Goal Model</b>	<b>111</b>
<b>B</b>	<b>Behavioral Interfaces of Activities</b>	<b>114</b>
B.1	Behavioral Interfaces - Key Generation and Establish Binding . . . . .	114
B.2	Behavioral Interfaces - V&A Processing . . . . .	115
B.2.1	Behavioral Interfaces for Subgoal "Formal Validation" . . . . .	116
B.2.2	Behavioral Interfaces for Subgoal "Validation and Authentication" . . . . .	117
B.3	Behavioral Interfaces - Revocations . . . . .	121
B.3.1	Behavioral Interfaces - Signature Revocation . . . . .	121
B.3.2	Behavioral Interfaces - EIR Revocation . . . . .	122
B.4	Behavioral Interfaces - Mining . . . . .	122
B.4.1	Behavioral Interfaces for Subgoal "Symbolic Mining" . . . . .	123
B.4.2	Behavioral Interfaces for Subgoal "Process VAR" . . . . .	123
<b>C</b>	<b>Files</b>	<b>125</b>
C.1	CPN Models . . . . .	125
C.2	State Space Analyses . . . . .	125
C.3	Online Sources . . . . .	125
	<b>Bibliography</b>	<b>126</b>

# List of Abbreviations

<b>PGP</b> .....	Pretty Good Privacy - An encryption program
<b>WoT</b> .....	Web of Trust
<b>PKI</b> .....	Public Key Infrastructure
<b>CA</b> .....	Certificate Authority
<b>CPN</b> .....	Coloured Petri Net
<b>DSR</b> .....	Design Science Research
<b>IS</b> .....	Information System
<b>RQ</b> .....	Research Question
<b>V&amp;A</b> .....	Validation and Authentication
<b>VAR</b> .....	Validation and Authentication Request
<b>SRP</b> .....	Security Risk-oriented Pattern
<b>AOM</b> .....	Agent-Oriented Modeling
<b>EIR</b> .....	Entity Identity Record
<b>CR</b> .....	Challenge Record
<b>RR</b> .....	Response Record
<b>SR</b> .....	Signature Record
<b>UTC</b> .....	Coordinated Universal Time
<b>INT</b> .....	Integer
<b>VAE</b> .....	V&A Entry
<b>ISSRM</b> .....	Information Systems Security Risk Management



<b>UML</b>	.....	Unified Modeling Language
<b>BPMN</b>	.....	Business Process Model and Notation
<b>MITM</b>	.....	Man-in-the-Middle
<b>DDoS</b>	.....	Distributed Denial of Service
<b>SCC</b>	.....	Strongly Connected Component



# Chapter 1

## Introduction

The design and development of new security protocols in the field of computer science is a challenging task [1][2][3][4]. Design flaws and missing specifications as well as security [5] and privacy issues of such protocols pose a risk for its users. In context of computer science, security refers to the protection of different entities against threats [6], whereas privacy aims to protect information related to the involved entities [7]. In a best case, issues of a security protocol are inconvenient to users who rely on it, but in a worst case design flaws and errors are fatal. An example for the first case is a broken encryption of a wireless network [8]. On the other hand a broken security protocol that grants an attacker access to sensible parts of nuclear power plants is a serious threat [9].

Different formal methods, such as Petri nets [10],  $\pi$ -calculus [11] and communicating sequential processes [12], are utilized for the design, development and analysis of new protocols [13][14][15]. In this thesis, the Authcoin protocol [16] is formalized using formal methods in order to detect and eliminate eventual design flaws, missing specifications as well as security and privacy issues.

In addition, security risk-oriented patterns [17][18] are applied to the formal model of the protocol. A pattern is a tested, commonly accepted, generic solutions for a recurring and common problem [19]. Security risk-oriented patterns are a means to mitigate common security and privacy risks in processes by applying thoroughly tested and proven best-practice solutions. The goal of this thesis is to reduce the risks and vulnerabilities of the Authcoin protocol using the techniques and approaches mentioned above.

### 1.1 Existing Body of Knowledge

The following sections provide a short and general overview on the Authcoin protocol (Section 1.1.1), formal methods for protocol verification (Section 1.1.2) and security

risk-oriented patterns (Section 1.1.3). The final part of this section (Section 1.1.4) deals with the detected research gap that provides the foundation for this thesis.

### 1.1.1 Authcoin

Authcoin is an alternative approach to the commonly used public key infrastructures such as central authorities and the PGP Web of Trust (WoT). It combines a challenge response-based validation and authentication process for domains, certificates, email accounts and public keys, with the advantages of a blockchain-based storage system. The blockchain technology provides a publicly available, transparent and fault tolerant mechanism for storing data in a distributed and decentralized manner [20]. Users can setup challenges for other entities and ask them to fulfill these challenges. Either the entity fails to do so or is able to successfully complete the challenge and create a corresponding response. The chosen challenge depends on the required level of security and the given threat level of the involved entities. Even though Leiding et al. [16] outline a public/private key based solution, it is also possible to abstract from this assumption and use other identifiers, e.g. biometric identifiers.

Systems for distributing and managing public keys are referred to as public key infrastructures (PKIs). The hierarchical trust model for certificate authentication, commonly used by certificate authorities (CAs) and web browsers, relies on hierarchically structured central authorities [21], whereas the PGP<sup>1</sup> Web of Trust uses a decentralized approach [22]. Instead of relying on central CAs, each user acts as an authority itself and ensures a number of bindings between (third-party) users and their public keys. CAs as well as the PGP WoT have several flaws and disadvantages, but a more detailed discussion on that is out of scope of this work. Authcoin is a flexible alternative to these two concepts and utilizes the blockchain technology, resulting in a decentralized system that is also fault tolerant, difficult to manipulate, transparent and provides partially automated validation and bidirectional authentication [16].

### 1.1.2 Formal Methods for Protocol Formalization

“A formal method is a mathematically-based technique used in computer science to describe properties of hardware and/or software systems. It provides a framework within which large complex systems may be specified, developed, and verified in a systematic rather than ad-hoc manner. A method is formal if it has a sound mathematical basis, typically given by a formal specification language” [23]. Over many decades, several approaches and concepts for formal methods have been introduced. Among them, without

---

<sup>1</sup>In the following, PGP will be used as a synonym for all PGP compatible software.

claiming completeness: Statecharts [24], abstract state machines (ASMs) [25], Petri nets [10], the Calculus of Communicating Systems [26] and  $\pi$ -calculus [11], as well as timed automata [27] and communicating sequential processes [12].

In this thesis, the Authcoin protocol is modeled using formal methods based on Coloured Petri Nets (CPNs) [28], a special type of Petri nets. “A CPN is a graphical oriented language for the design, specification, simulation and verification of systems. It is in particular well-suited for systems that comprise a number of processes that communicate and synchronize. Typical examples of application areas are communication protocols, distributed systems, automated production systems, or work flow analysis” [29]. A further advantage of CPNs is, that it “allows for the semantically deterministic design of system structures and also behavior that is verifiable for correctness and performance tests with tool support” [29]. CPN models are regularly used for the modeling and verification of security protocols as well as authentication protocols (e.g. [13][30][31][32][33][34][35]). The resulting formal representation of the Authcoin protocol is used as a specification for a future implementation of the protocol and for further validation and evaluation.

### 1.1.3 Security Risk-oriented Patterns

Patterns are a commonly used technique in the process of software development. Many problems in different software projects tend to be similar and occur on a regular base. Therefore, it is not necessary to design a new solution for each occurrence of a problem and instead, it is possible to rely on thoroughly tested high-quality solutions that are called patterns. The latter are independent from a specific implementation, or technology in order to make them applicable for broader sets of problems. Furthermore, it is also possible to structure patterns in a hierarchical way, using smaller patterns to solve sub-problems of a larger problem using a larger pattern [19].

Security patterns [36] are a specific sub-category of general patterns and aim to solve security related issues. Schumacher [37] defines security patterns as “a particular recurring security problem that arises in specific contexts and presents a well-proven generic scheme for its solution”. A security pattern-system “is a collection of security patterns, together with guidelines for their implementation, combination and practical use in security engineering” [37].

Ahmed and Matulevičius [18] introduce a set of five security risk-oriented patterns and apply them to business processes. Their patterns are based on understanding security risks that arise within business processes. To mitigate the risks, the patterns recommend secure solutions. The five security risk-oriented patterns listed by Ahmed and Matulevičius provide a first insight on this new concept of security patterns that is still

a topic of research. Based on Uzunov and Fernandez [38] collection of security-threat patterns, Samarütel [39] further extends the library of security risk-oriented patterns.

While Ahmed et al. initially propose to apply security risk-oriented patterns to business processes, this thesis applies the existing security patterns to formal CPN models of the Authcoin protocol that originate from the formalization process described in Section 1.1.2. The goal is to eliminate common security and privacy issues by applying existing patterns.

#### 1.1.4 Contribution - Detecting a Gap

This thesis aims to produce two main contributions. First, providing a complete and correct formal specification of the Authcoin protocol that guides future implementations. Completeness guarantees that no specifications of the protocol are missing and correctness ensures that if the protocol performs any action at all, it is in accord with its specifications [40].

Second, as mentioned in Section 1.1.3, security risk-oriented patterns are intended to be applied on business processes. In this thesis, security risk-oriented patterns are applied to the formal CPN models of Authcoin.

## 1.2 Research Methodology and Research Questions

The natural choice for this thesis is to follow the approach of the design science research methodology [41]. “The design science paradigm seeks to extend the boundaries of human and organizational capabilities by creating new and innovative artifacts” [41]. Hevner et al. [41] define artifacts as constructs, models, methods and utility. According to the authors, a construct might be a vocabulary, or symbols and models refer to abstractions and representations. Algorithms and practices represent the methods and the utility can be an implementation, or a prototype system.

The following Section 1.2.1 introduces the theoretical background of the design science research methodology. In Section 1.2.2, the discussed research methodology is applied to this thesis. Finally, Section 1.2.3 defines the main research question as well as the subquestions of this work.

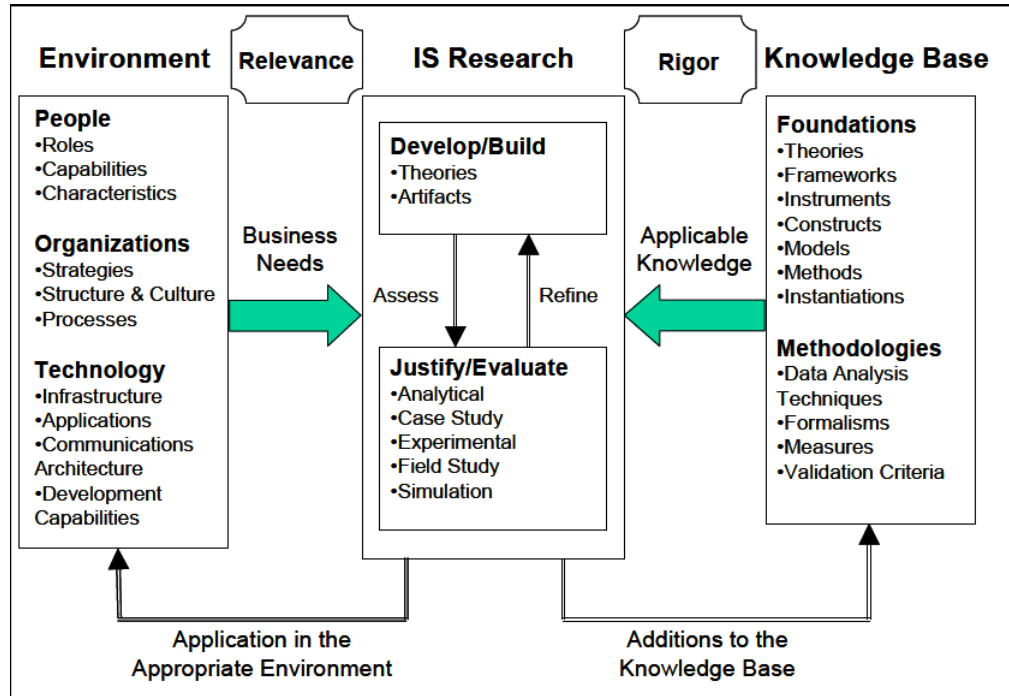


FIGURE 1.1: Design Science Research - Framework (Source: [41])

### 1.2.1 Design Science Research - Theory

Hevner et al. [41] describe a process on how to conduct design science research (DSR) in the context of information systems (IS) research. In order to create new artifacts, Hevner et al. propose an information system research framework as depicted in Figure 1.1. In context of the Hevner et al. DSR framework, artifacts have to address a relevant need of the defined environment. The environment is represented by people, organizations, businesses and their technologies. The knowledge base of the DSR framework provides foundations in the form of existing research as well as methodologies, such as guidelines for the evaluation and justification of new artifacts. The methodologies also ensure the rigor of IS research. The resulting artifacts contribute to the knowledge base and are applied to the environment that provided the initial need.

### 1.2.2 Design Science Research - Practice

Besides the DSR framework depicted in Figure 1.1, Hevner et al. also provide DSR guidelines that assist researchers in the practical process of conducting DSR. Table 1.1 illustrates the seven DSR guidelines that are adhered to in this thesis. The following sections describe the application of the abstract guidelines to the context of this thesis.

Guideline	Description
Guideline 1: Design as an Artifact	Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.
Guideline 2: Problem Relevance	The objective of design-science research is to develop technology-based solutions to important and relevant business problems.
Guideline 3: Design Evaluation	The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.
Guideline 4: Research Contributions	Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.
Guideline 5: Research Rigor	Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.
Guideline 6: Design as a Search Process	The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.
Guideline 7: Communication of Research	Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.

TABLE 1.1: Design Science Research - Guidelines (Source: [41])

### 1.2.2.1 Design as an Artifact

As described in Section 1.1.4, the created artifact aims to close an existing gap in the knowledge base by formalizing the Authcoin protocol using CPNs and application of security risk-oriented patterns to the CPN models of the protocol.

### 1.2.2.2 Problem Relevance

The problems addressed in this thesis are relevant in various respects. First, the Authcoin protocol aims to provide an alternative validation and authentication concept that solves several issues of existing approaches, such as the susceptibility of the PGP WoT to sybil attacks [16]. Therefore, a formal specification and verification of the protocol using CPN models is an important step towards the final implementation and contributes to the existing knowledge base.

Furthermore, applying security risk-oriented patterns to the formal CPN models increases the protocol's security and provides guidelines for other security protocols on how to apply the same techniques and methodologies. Moreover, security risk-oriented patterns are applied to CPN models for the first time.



### 1.2.2.3 Design Evaluation

Hevner et al. provide a listing of design evaluation methods as presented in Table 1.2. The evaluation of the created artifacts of this work is performed using multiple methods. An analytical evaluation is performed on the developed CPN models of the Authcoin protocol using CPN-Tools<sup>2</sup> and state space analyses. The evaluation is performed before and after applying the security risk-oriented patterns and both results are compared. An implementation and real-world evaluation of Authcoin is out of scope for this thesis. Therefore, we limit this work to artificial evaluation methods only.

1. Observational	Case Study: Study artifact in depth in business environment
	Field Study: Monitor use of artifact in multiple projects
2. Analytical	Static Analysis: Examine structure of artifact for static qualities (e.g., complexity)
	Architecture Analysis: Study fit of artifact into technical IS architecture
	Optimization: Demonstrate inherent optimal properties of artifact or provide optimality bounds on artifact behavior
	Dynamic Analysis: Study artifact in use for dynamic qualities (e.g., performance)
3. Experimental	Controlled Experiment: Study artifact in controlled environment for qualities (e.g., usability)
	Simulation – Execute artifact with artificial data
4. Testing	Functional (Black Box) Testing: Execute artifact interfaces to discover failures and identify defects
	Structural (White Box) Testing: Perform coverage testing of some metric (e.g., execution paths) in the artifact implementation
5. Descriptive	Informed Argument: Use information from the knowledge base (e.g., relevant research) to build a convincing argument for the artifact's utility
	Scenarios: Construct detailed scenarios around the artifact to demonstrate its utility

TABLE 1.2: Design Science Research - Evaluation Methods (Source: [41])

As illustrated in Figure 1.1, the knowledge base contains methodologies that provide guidance for the evaluation and justification of new artifacts. We perform a case-study evaluation focusing on the adaption of security risk-oriented patterns to the formal CPN models. Case-study-based evaluations map to an evaluation methodology based on validation criteria as represented in the right pillar of Figure 1.1. Based on Shenton [42], the trustworthiness of these evaluations is ensured by adhering to the criteria of credibility, transferability, dependability and confirmability. Credibility refers to the internal validity of the findings, whereas transferability ensures the applicability of the findings to other situations. In context of the evaluation of qualitative research, dependability

<sup>2</sup><http://cpntools.org/>

confirms that the processes within the study are reported in detail, thereby enabling future researchers to repeat the work. According to Shenton, confirmability ensures “as far as possible, that the works findings are the result of the experiences and ideas of the informants, rather than the characteristics and preferences of the researcher”.

#### **1.2.2.4 Research Contribution**

This thesis makes two contributions to the knowledge base. First, formalizing the Authcoin protocol using CPN models that are used as a specification for a future implementation. Second, providing a process for applying security risk-oriented patterns to CPN models.

#### **1.2.2.5 Research Rigor**

In the making of this thesis, CPN models and CPN-Tools [43] are used to formalize the Authcoin protocol, afterwards security risk-oriented patterns are applied to the formal models. CPN models [28] as well as security risk-oriented patterns [17][18] fulfill the requirements of rigorous methods.

#### **1.2.2.6 Design as a Search Process**

Based on the Leiding et al. Authcoin paper [16], a formal CPN model is created using CPN-Tools. Security issues of the resulting formal models are identified using a risk and threat analysis and afterwards minimized by the application of security risk-oriented patterns. Subsequently, the results are evaluated by means of the described DSR evaluation methods from Section 1.2.2.3.

#### **1.2.2.7 Communication of Research**

We publish the results of this thesis in an academic environment as well as to make them accessible to involved or interested businesses.

### **1.2.3 Research Questions**

Based on the previous sections, the main research question (RQ) is phrased as follows:  
**How to secure the Authcoin protocol by employing formal techniques combined with applying security risk-oriented patterns?**

In order to answer this question in a more structured and comprehensive way, the main research question is divided into three subquestions:

- **RQ-1: How to formalize the Authcoin protocol?**
- **RQ-2: How to analyze security threats of the Authcoin protocol?**
- **RQ-3: How to apply security risk-oriented patterns on the Authcoin protocol?**

RQ-1 is answered by formalizing the Authcoin protocol using Colored Petri Nets and development of the corresponding CPN models using CPN Tools. As part of RQ-2, an analysis of the Authcoin protocol determines the potential risks that threaten the protocol. Based on the results of RQ-1 and the risk-analysis, security risk-oriented patterns are applied to the formal CPN models as part of RQ-3.

### 1.3 Thesis Structure

The rest of the thesis is structured as follows. Chapter 2 provides an overview on the Authcoin protocol before discussing the formalization process, followed by an introduction to security risk-oriented patterns. The formal CPN models of the Authcoin protocol are developed in Chapter 3 using CPN Tools. Afterwards in Chapter 4, a risk and threat analysis is performed in order to identify the risks that threaten the Authcoin protocol. In Chapter 5, based on the results of the previous chapters, the existing security risk-oriented patterns are reviewed and appropriated ones are selected in order to mitigate the identified risks. In Chapter 6, an analytical evaluation is performed on the developed CPN models of the Authcoin protocol. Furthermore, case-study-based evaluations are performed on the adaption of security risk-oriented pattern to CPN models. Chapter 7 concludes the thesis and provides an outlook on future work.

## Chapter 2

# Presuppositions

The following chapter introduces and explains fundamental approaches and concepts of this thesis. First, Section 2.1 gives a high-level introduction to Authcoin itself and provides the reader with a basic understanding of the protocol. Second, Section 2.2 describes and introduces security risk-oriented patterns in more detail.

### 2.1 Authcoin

Based on the introduction of the Authcoin protocol from Section 1.1.1, the following sections provide a high level overview on Authcoin, its work-flow and basic functionalities in order to give the reader a basic understanding of the protocol and its purpose. The section is based on the original Authcoin paper [16] that also provides further information as well as more detailed explanation.

Section 2.1.1 provides an overview of the protocol's work-flow. The following Section 2.1.2 deals with the general validation and authentication process of Authcoin and Section 2.1.3 explains the storage concept as well as smart contracts. Information on challenges are provided in Section 2.1.4, whereas Section 2.1.5 deals with automated validation and authentication requests.

#### 2.1.1 Overview

A general overview of the protocol's work-flow is illustrated in Figure 2.1 and consists of four main steps. In the first step, a new key pair is created. Step 2 establishes a binding between the generated key and the owning entity. Afterwards, the public key is posted to the blockchain together with information used to establish the binding. Step 3 deals with the formal validation, followed by the validation and authentication (V&A)

procedure in step 4. Finally, all information related to the V&A process are also posted to the blockchain.

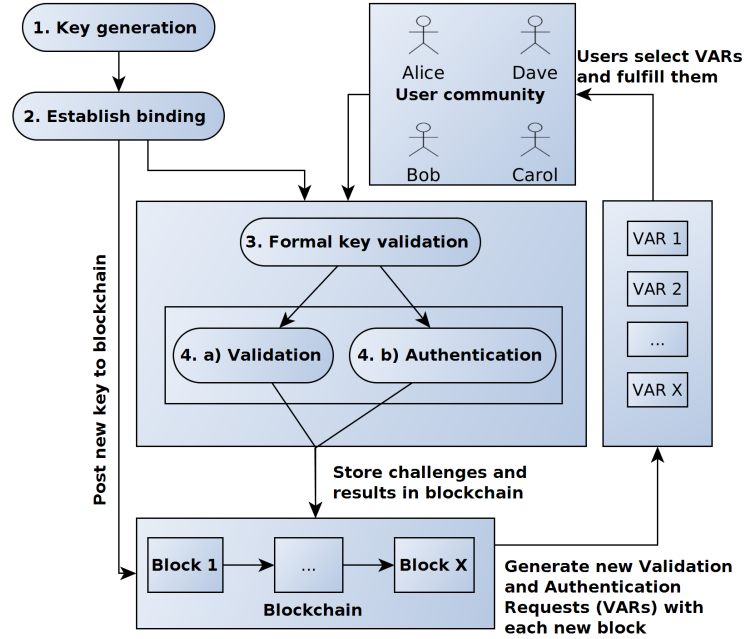


FIGURE 2.1: Abstract overview on the work-flow of the Authcoin protocol (Source: [16])

More detailed information on validation and authentication requests as per Figure 2.1, are provided later in Section 2.1.5.

### 2.1.2 General Validation and Authentication

The general validation and authentication process starts with an automated formal key validation. This process guarantees that the key is well formed and adheres to the syntax rules specified for the key, has a sufficient key length and that the key is still valid and has not been revoked yet [16]. Other properties can be checked as well depending on the use case. Properties of PGP keys used for email encryption or decryption have other properties than domain certificates for websites. Afterwards the actual V&A procedures begins as depicted in Figure 2.2.

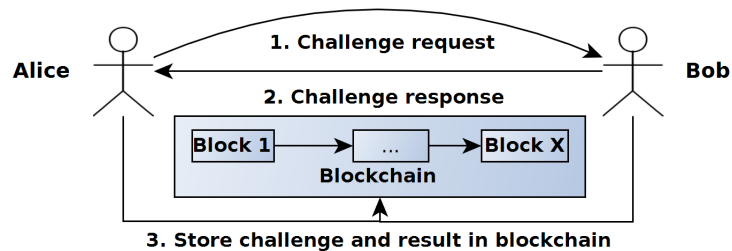


FIGURE 2.2: General validation and authentication procedure (Source: [16])

The protocol distinguishes between validation and authentication. In the context of Authcoin, validation aims to prove the following three facts:

1. A specific entity has access to a certain account, e.g email, that is under validation.
2. A certain entity has access to a specified private and public key.
3. The specified key pair corresponds to the tested account.

Authcoin's authentication procedure continues the validation by verifying the identity and aims to confirm that the alleged owner of the key and the actual owner of the key are equal.

### 2.1.3 Storing Information and Smart Contracts

Authcoin uses an Ethereum-powered [44][45] blockchain-based storage system. The blockchain technology gained popularity with the inception of the peer-to-peer payment system Bitcoin [20]. A blockchain is a public distributed ledger which records transaction events without the need for a trusted central authority. The blockchain consists of an unlimited number of blocks that are chained together in a chronological order. Each block consists of transactions that contain the actual data to be stored in a blockchain. During the mining process, miners collect valid transactions and agree on a global consensus on which transactions are part of a new block that is added to the blockchain. Each block depends on its predecessor block and therefore tampering and manipulating a block requires a recalculation of all successor blocks.

The advantage of Ethereum and similar systems is that they do not only provide a blockchain-based storage but also incorporate Turing-complete programming languages on the protocol-layer on top of a blockchain in order to realize smart contract capabilities. Smart contracts are, "orchestration and choreography protocols that facilitate, verify and enact with computing means a negotiated agreement between consenting parties" [46]. Participating entities establish binding agreements and deploy applications on the blockchain using such smart contracts. In the context of Ethereum, a contract is an instance of an application that consists of its program code, storage and an account balance [47]. "The contract's code is executed whenever it receives a message, either from a user or from another contract. While executing its code, the contract may read from or write to its storage file. A contract can also receive money into its account balance, and send money from its account balance to other contracts or users. The code of a contract determines how it behaves" [47]. The smart contracts are powered by "gas", Ethereum's internal fuel and each computational step of the code execution requires a certain amount of gas and therefore prevents accidental or hostile infinite

loops [44]. Contracts are implemented in one of Ethereum’s high-level languages, e.g. Serpent, afterwards compiled into bytecode and posted to the blockchain.

#### 2.1.4 Challenges

Authcoin’s security and reliability heavily depends on the chosen challenges and the challenge design. The chosen type of a challenge depends on the use case, security requirements and the user’s threat level. For example, the threat level and security requirements for a financial transaction are much higher compared to the simple login to an arbitrary social network. Therefore, different challenges are necessary. The protocol itself does not provide any default challenges and leaves it to the implementation and the users to create their own customized challenges according to their needs. For this thesis, only a basic understanding for challenges is required and more detailed information can be found in the original Authcoin paper [16].

##### 2.1.4.1 Basic Challenges

The design of a challenge depends on the available information regarding the involved entities and available communication channels. Leiding et al. [16] outline three types of V&A challenges. An example for each is presented and security as well as privacy implications are discussed. Below we present two examples.

The following example, based on Figure 2.2, illustrates a validation procedure in context of Authcoin: User Alice sends a challenge, e.g. “this is a challenge”, encrypted with Bob’s public key, to Bob’s email account. Bob is asked to fulfill the challenge, sign the response with his private key and send it back to Alice. Alice checks the results and in case that the process finishes successfully, deduces the following three facts from the challenge and response: 1.) Bob has access to the email account - account validation. 2.) Bob has access to the public and private key - key validation. 3.) The key pair corresponds to the tested email account. The validation requests and results of the validation processes are stored as part of the blockchain. Both, Alice and Bob, independently post the challenge and response to the blockchain.

The second example concerns the authentication process of the previous example. Assuming the case that Alice and Bob once went to the same school, but lost contact. After a successful bidirectional validation, Alice wants to verify Bob’s identity in order to ensure that the email account and the public key belongs to Bob and not to somebody else. Therefore, she sends a challenge to Bob and asks him to send her a picture of himself holding a copy of the current issue of a specific newspaper. In this simple example, it is plausible that Alice somehow knows what Bob looks like and therefore is

able to decide whether the person on the image is Bob or not. Since Bob also wants to make sure that he is really communicating with Alice, he verifies Alice in the same manner, resulting in a bidirectional verification of both participants.

An important difference between Authcoin's V&A process and earlier solutions, e.g. the PGP WoT, is that the V&A process is performed in a bidirectional manner. Alice sends a challenge to Bob and receives a matching response. In parallel, Bob does the same when receiving Alice's challenge in order to also verify Alice's mail account. As a result, each user is validated on a regular base. Therefore, it is more difficult for malicious users to introduce fake keys into the system and maintain malicious keys. A major advantage of Authcoin's bidirectional validation processes is that they can be performed in an automated manner. As a result, each domain, certificate or account can be validated on a regular basis resulting in an improved overall security of the network.

It is important to keep in mind that users cannot solely rely on successfully passed challenges. They have to analyze and examine the challenges used during the V&A process of a key and decide on their own if these challenges provide a suitable level of reliability and security or not.

#### **2.1.4.2 Adaptable Level of Required Security**

The customizable nature of Authcoin's challenges is a key feature of the protocol. Depending on the challenge design, Authcoin can accommodate the need for different levels of security. There are use cases where a simple validation procedure is sufficient. In other scenarios, it is necessary to combine different challenges based on different identifiers in order to provide a maximum level of security and reliability. Other scenarios lie in between these two extrema. Even though the initial design of Authcoin's V&A is based on public and private keys, there is no necessity to be limited to this approach. It is also possible to utilize alternative identifiers such as biometric identifiers. This includes fingerprints, retina or iris recognition, voices and faces, or DNA. Biometric identifiers are either used to derive a new key pair based on the identifier [48] or are included into the new key pairs as an additional identification information.

#### **2.1.5 Validation and Authentication Requests - VARs**

A further feature of Authcoin are validation and authentication requests (VARs). As illustrated in Figure 2.1, VARs are automatically and randomly generated during the mining of a new block of the blockchain. An automated VAR expresses the desire of the system to validate and/or authenticate a randomly chosen entity inside the system. The



number of generated requests depends on the number of existing entities in the chain and the time between new blocks. The more entities, the more requests are generated. The shorter the time between two consecutive blocks the less generated VARs. The main idea behind this feature is to break into collectives of sybil or malicious entities in the system “by accident” and expose them as such in case they fail the V&A process. Identifying one malicious entity leads to questioning all other entities that claim to have successfully validated and/or authenticated the malicious one and therefore identifying them either also as sybil nodes or at least as unreliable verifiers. Due to VARs and the bidirectionality of authentications, it also increases the number of V&As for each key, resulting in higher probabilities of detecting malicious users.

## 2.2 Security Risk-oriented Patterns

A design pattern is defined as a “conceptually formulated knowledge that is technology independent. A design pattern for software architecture [49] describes a particular recurring design problem that arises in specific design contexts and presents a well-proven generic scheme for its solution” [50]. Furthermore, patterns can be structured in a hierarchical way, using smaller patterns to solve sub-problems of a larger problem using a larger patterns [19].

The following section provide an introductory overview on security patterns itself (Section 2.2.1) and security risk-oriented patterns (Section 2.2.2). Afterwards, Section 2.2.3 describes the existing security risk-oriented patterns.

### 2.2.1 Security Patterns

Security patterns [36] are a specific sub-category of general patterns and aim to solve security related issues. Schumacher [37] defines security patterns as “a particular recurring security problem that arises in specific contexts and presents a well-proven generic scheme for its solution”. A relevant aspect of security patterns, besides their design and implementation, is the classification and organization of security patterns. The growing number of existing security patterns makes it difficult for pattern users to find the most appropriate solution for his/her problem. A classification of patterns into small, correlated sets helps users finding a pattern that solves their specific problems more easily. Different approaches for the classifications and categorizations of security pattern exist [51][52][53][54], but a discussion on them is out of scope of this work and not required for the further understanding.

### 2.2.2 Security Risk-oriented Patterns

Ahmed et al. [17][18] introduce five security risk-oriented patterns and apply them to business processes. Their patterns are based on understanding security risks that arise within business processes and mitigate the risks using patterns with recommended and secure solutions. Business analysts have often only a limited expertise in security engineering. Using security risk-oriented patterns, they can design secure business processes either on their own or in cooperation with security analysts. Ahmed and Matulevičius [18] describe a seven step process for business and security analysts on how to integrate security requirements to business processes.

- 1. Identify assets and security objectives:** In the first step, business and security analysts cooperate in order to identify the business assets and the corresponding security objectives in terms of confidentiality, integrity, and availability. The result is an annotated business process model that contains the identified security objectives.
- 2. Analyze security risks:** The security analyst identifies security risks and compiles a list of these risks that are defined by threats, vulnerabilities and the corresponding risk impact.
- 3. Elicit security requirements:** As part of this step, “the business analyst takes risk-treatment decisions to reduce, avoid, retain or transfer the identified risks. Security decisions are refined to the security requirements to mitigate the identified risks” [18].
- 4. Annotate business process model with security requirements:** In order to avoid an interference between the original work-flow described by the business processes and the security requirements, the security analyst annotates the business process model with the security requirements.
- 5. Feed back business models annotated with security requirements:** The business analyst receives the annotated business process model, the security requirements and the security risk-oriented business process model - representing a comprehensive security solution for the work-flow.
- 6. Present security solutions & 7. Rationalize security solution:** The presented security solution is examined with regard to the resulting costs and risk reductions of an implementation. Other relevant factors are the complexity of the security implementation and potential performance issues. Based on these information, the security analyst defines a prioritized set of requirements that should be implemented in the work-flow to increase its security.

While Ahmed et al. [17][18] propose to apply security risk-oriented patterns to business processes, this thesis applies the existing security patterns to the formal CPN models of the Authcoin protocol that originate from the formalization process. Therefore, the

process described above is used as a guideline for the application of security risk-oriented patterns to the CPN models later in Chapter 4 and Chapter 5.

### 2.2.3 Existing Security Risk-oriented Patterns

The five security risk-oriented patterns listed by Ahmed et al. provide a first insight on this new concept of security patterns that is still a topic of research. In the following, the existing library of security risk-oriented patterns (SRPs) is introduced based on [18]:

**SRP 1:** The pattern secures the data transmission between business entities with focus on preventing the loss of data, confidentiality and its integrity. SRP 1 proposes to make the data unreadable before transmitting, calculate checksum values and utilize transmission mediums that cannot be intercepted.

**SRP 2:** The pattern prevents attackers from injecting malicious data into business processes and proposes to define data structures and formats for incoming data.

**SRP 3:** SRP 3 ensures the availability of the business services by mitigating denial of service attacks by “filtering and classifying of incoming requests, detecting abnormal requests, and discarding the attacking ones” [18].

**SRP 4:** This pattern proposes the application of multilevel access rights to the retrieval interface in order to prevent unauthorized access to data and missing system logs on the access history to specific data.

**SRP 5:** In order to ensure the confidentiality of data, SRP 5 proposes to store data in an encrypted and secure manner.

## Chapter 3

# Formal Specification of Authcoin Using Coloured Petri Nets

*The following chapter deals with the process of formalizing the Authcoin protocol using Colored Petri Nets, resulting in a sound and complete CPN model that provides the foundation for later sections of this thesis. An Agent-Oriented Modeling methodology is used to create goal models with corresponding behavioral interfaces. Afterwards, these models are used to derive the Authcoin CPN models. Furthermore, the modeling strategy as well as the required protocol semantics are explained in detail.*

### 3.1 Introduction

The objective of Chapter 3 is to answer the research question RQ-1 - How to formalize the Authcoin protocol? - as outlined earlier in Chapter 1. In order to answer RQ-1 in a more structured and comprehensive way, it is divided into three subquestions:

- **RQ-1.1: What is the top-level of the CPN model?**
- **RQ-1.2: What protocol semantics are required?**
- **RQ-1.3: What are the refining CPN models?**

Each subquestion is answered independently in a separate section. The first part of this chapter focuses on the theoretical background of the designated modeling strategy (Section 3.2). Afterwards, Section 3.3 introduces Colored Petri Nets in detail. Section 3.4 answers RQ-1.1 and illustrates the process of mapping the existing descriptions and entities of the protocol to the corresponding elements of a CPN model. Subsequently,






Symbol	Meaning
	Goal
	Quality goal
	Role
	Relationship between goals
	Relationship between goals and quality goals

TABLE 3.1: AOM goal model notation (Source: [62])

Section 3.5 deals with RQ-1.2 and details the necessary protocol semantics. The refined CPN models of RQ-1.3 are depicted in Section 3.6. Finally, the resulting CPN models are discussed in Section 3.7 followed by the conclusion of this chapter in Section 3.8.

## 3.2 Modeling Strategy

Modeling the Authcoin protocol using CPN requires an appropriate modeling strategy, mapping the existing descriptions and entities of the protocol to the corresponding elements of a CPN model. Authcoin organizes and defines the exchange of validation and authentication information between different entities that are modeled as agents. In software engineering, various agent-oriented approaches exist, such as: Tropos [55], Gaia [56], Prometheus [57], MASB [58][59] and MaSE [60]. In [61], Mahunnah et al. introduce mapping heuristics from agent models to CPN models based on Sterling's and Taveter's [62] methodology for Agent-Oriented Modeling (AOM). The following section introduces the two AOM model types, i.e. the goal model (Section 3.2.1) and the behavioral interface model (Section 3.2.2), necessary to represent the Authcoin protocol.

### 3.2.1 Goal Model

The purpose of an AOM goal model is to capture the functional requirements of a system in the form of goals, as well as non-functional requirements and roles of involved entities. Non-functional requirements represent quality goals of the system [62]. Figure 3.1 illustrates the notation of the following goal model. A goal is represented in form of a parallelogram, quality goals in the form of clouds and sticky men represent roles. As

illustrated in Figure 3.1, the functional requirements of the goal model are structured in a tree-like hierarchy with the overall objective of the system at the top. The main goal is decomposed into further multi-layered sub-goals until the lowest atomic sub-goal is reached. The main objective of Authcoin, and therefore also the main goal of the corresponding goal model, is to provide a secure and reliable validation and authentication protocol. The main goal is further divided into multi-layered sub-goals: Key generation and establish a binding, validation and authentication processing, mining and revocations. The three quality goals “secure”, “correct” and “reliable” are attached to the overall main goal of the goal model.

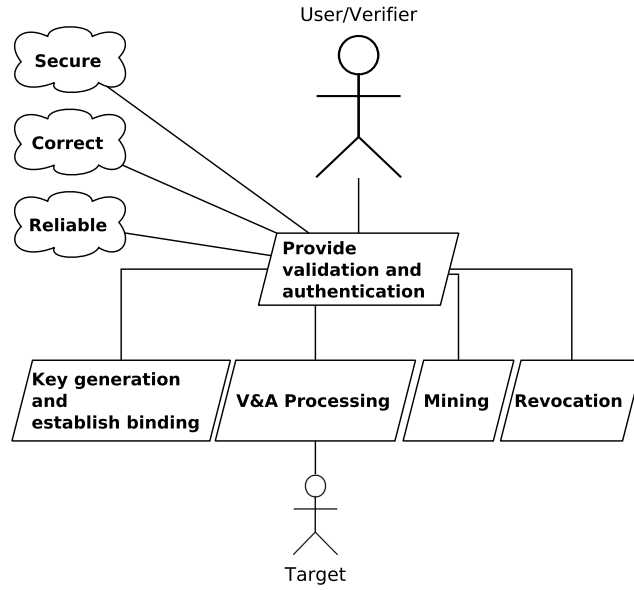


FIGURE 3.1: Authcoin - Top level goal model

The further refined sub-goals are listed in detail in Appendix A.

### 3.2.2 Behavior Interface Model

The behavior model refines the developed goal model for specific agents and activities. “A behaviour model in AOM has two parts: an agent behaviour model coupled with a behaviour interface model [62]. The former describes the rule-based behavior of an agent, while the latter focuses on identifying activities with associated triggers, preconditions and postconditions” [61]. Table 3.2 presents the behavior interface model of the goals depicted in the top level goal model of Figure 3.1. Each activity is listed with its corresponding trigger, eventual preconditions and its postconditions. An execution of an activity is either triggered by an event, or by a precondition after the occurrence of an event [61]. The “Key generation and establish binding”-activity in Table 3.2 is triggered by the user who wants to create a new key pair. To do so, the precondition

has to be fulfilled. After the activity's execution, the user receives the resulting key pair and the corresponding EIR, that contains all identity related information of an entity, is posted to the blockchain. The remaining behavior interface models originating from the goal models are listed in Appendix B.

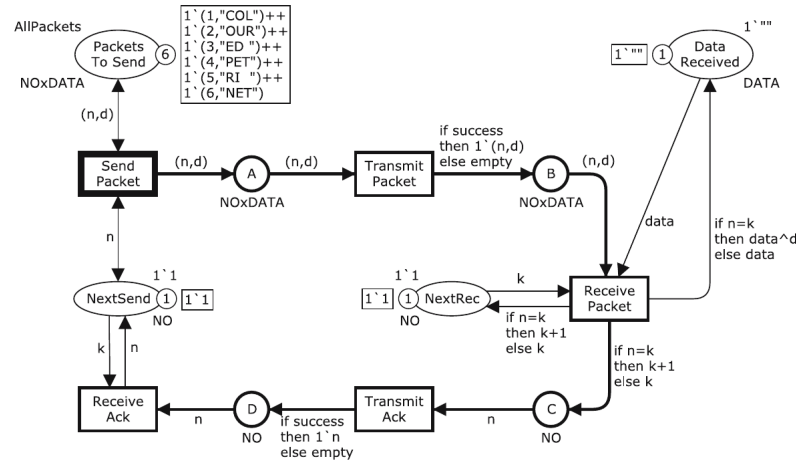
Activity	Trigger	Precondition	Postcondition
Key generation and establish binding	User wants to create a new key pair	Identifier list, key expiration date, key type, key length	Key pair, EIR on blockchain, EIR
V&A Processing	Received EIRs for V&A	Verifier EIR, target EIR	V&A results on blockchain or failure message
Mining	Received input for blockchain	Input transactions	CR, RR and SR on blockchain and VARs or failure message
Revocation	User wants to revoke an EIR or a SR	KeyPair, EIR, SR, CR, RR, VARs	Revoked EIR or SR and updated information on blockchain

TABLE 3.2: Behavioral interfaces of activities for Authcoin

Mapping the behavior interface model that originated from the goal model results in the formal CPN model of the Authcoin protocol. A detailed explanation on this process is outlined in Section 3.4 after an introduction to protocol formalization using CPNs itself in the following Section 3.3.

### 3.3 Protocol Formalization Using CPNs

As suggested earlier in Section 1.1.2, this thesis relies on Coloured Petri Nets (CPNs) as a formal method for modeling, validation and verification of the Authcoin protocol based on the goal models and behavior interfaces from Section 3.2. “A CPN is a graphical oriented language for the design, specification, simulation and verification of systems. It is in particular well-suited for systems that consist of a number of processes which communicate and synchronize. Typical examples of application areas are communication protocols, distributed systems, automated production systems, or work flow analysis” [29].



In the following sections, CPNs are introduced and explained in more detail (Section 3.3.1) as well as the basics of CPN modeling (Section 3.3.2 and Section 3.3.3). The process of deriving the CPN models from the agent-oriented models is outlined later in Section 3.4. Further information on the analysis and evaluation of CPN models are provided in Chapter 6.

### 3.3.1 Coloured Petri Nets - CPNs

“Coloured Petri Nets (CPNs) are a high-level-form of ordinary Petri nets” [34]. The different token colors represent different types of data. “A CPN model of a system describes the states of the system and events (transitions) that can cause the system to change state. By making simulations of the CPN model, it is possible to investigate different scenarios and explore behaviours of the system” [28]. CPNs are represented as a graph or an algebraic structure. Algebraic CPNs represent a system as a grammar language in an algebraic form whereas graph CPN models are represented using a directed bipartite graph that consists of places, transitions, arcs and tokens [28][30][34][43].

### 3.3.2 Modeling Protocols Using CPNs

As illustrated in Figure 3.2, places are represented as circles and transitions as rectangular boxes. The directed arcs connect places and transitions. Arcs can only occur between places and transitions or transitions and places. Other combinations are not allowed. Depending on the orientation of the directed arc, it can either act as an input or an output to a transition or place. The state of a modeled system is represented by its places. Each place can be marked with one or more tokens of whom each has a



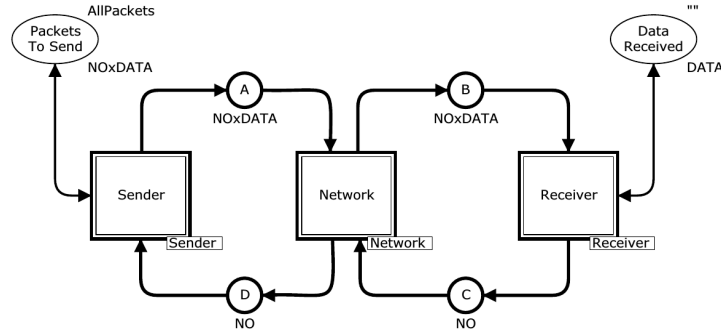


FIGURE 3.3: Top-level module of a hierarchical protocol model (Source: [43])

data value attached to it that is referred to as the token color. The number of tokens and the token colors in the individual places represent the state of the system. When a transition is activated (“fires”), it removes token(s) from the connected input place(s) and adds it to the the output place(s). Each place has an initial marking that specifies the initial number and types of tokens at a specific place, while the current marking denotes the distribution of tokens at all places at any given time [28][43].

Besides the mentioned places, transitions, arcs and tokens, Figure 3.2 also contains several inscriptions in the CPN ML programming language. Inscriptions are used to further specify the data types and operations of the modeled system.

### 3.3.3 CPN Modules

In order to model more complex systems, CPN models are organized in a hierarchical structure consisting of modules. Based on Figure 3.2, a modularized CPN model is created and is represented in Figure 3.3. The modularized CPN model implements the sender, the receiver and the network as modules. Figure 3.4 illustrates the detailed sender module of Figure 3.3. The sender module contains the places and transitions known from the original model in Figure 3.2, but in addition also contains input and output ports. A module exchanges tokens with its environment via input and output ports [28].

## 3.4 Mapping the Agent-oriented Model to CPN Models

The following section answers RQ-1.1 - What is the top-level of the CPN model? - and explains the process of mapping the goal model and behavior interfaces from previous sections 3.2.1 and 3.2.2 to Colored Petri Nets resulting in the top-level CPN model. Table 3.3 illustrates the mapping of an AOM model to CPN models.

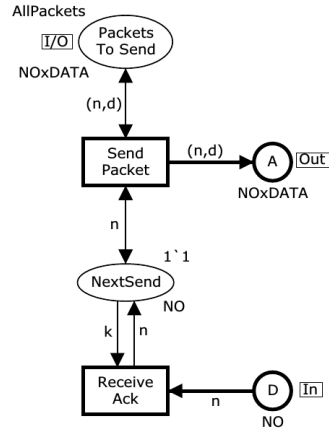


FIGURE 3.4: Sender module of Figure 3.3 (Source: [43])

Notation	Name
$\rightarrow$	Connecting Arc
	Sub-Goal or Activity
	Trigger or Precondition
	Postcondition
	Goal

TABLE 3.3: Notation for mapping AOM to CPN (Source: [61])

Directed arcs connect places and transitions representing the protocol execution through activities. Transitions represent simple activities or sub-goals performed by involved entities. CPN modules, depicted in form of double-boarded rectangles, illustrate goals derived from the goal model. As mentioned in Section 3.3.3, CPN modules can be broken down into smaller sub-parts of the overall model mapping to the same relation between goals and sub-goals in AOM. Places with outgoing arcs either act as triggers or represent a precondition, whereas places with incoming arcs represent postconditions of a given activity in AOM [61]. “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 module. This way, a CPN model represents a hierarchical structure of the goal model in AOM” [61].

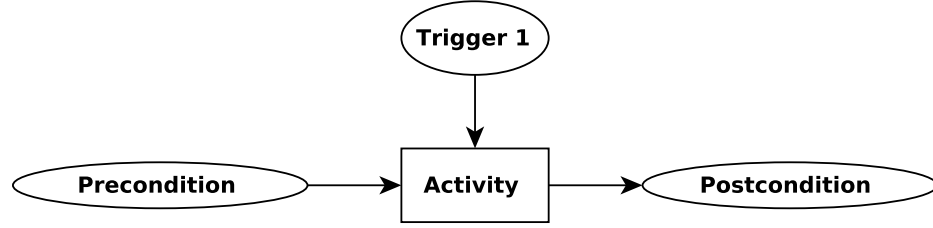


FIGURE 3.5: Mapping a behavior interface model to a CPN model

As shown in Section 3.2.2, the behavior interface model represents each identified activity of the protocol illustrated in the goal model, consisting of triggers, preconditions and postconditions for each activity. Figure 3.5 presents a CPN model consisting of a trigger for an activity, guarded by a precondition and resulting in a postcondition.

Finally, Figure 3.6 shows the complete and formalized top-level CPN model of Authcoin derived from AOM and implemented using CPN-Tools. Further depictions and the refined implementation of all modules and sub-modules are available in Section 3.6.

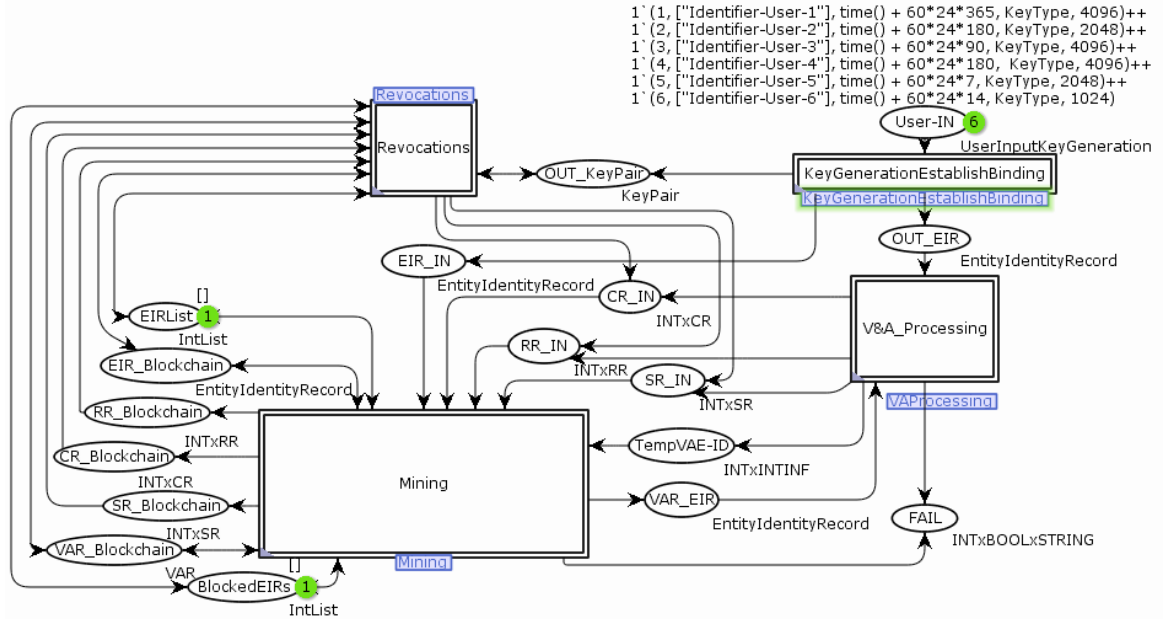


FIGURE 3.6: Authcoin - Top level CPN model

### 3.5 Protocol Semantics

This sections answers RQ-1.2 - What protocol semantics are required? - and shows the full set of CPN token color sets, names and acronyms used in the process of implementing the CPN models of the Authcoin protocol. These data sets are presented in Table 3.4, Table 3.5 and Table 3.6 for the top-level of the Authcoin CPN model. The tables first

row specifies the module of the first occurrence of a certain token, name or acronym. The second row specifies the name, followed by a short description in row number three. The last row provides information on the data type. More detailed refinements for further sub-modules of the top-level can be found in the subsequent tables of this section.

Module	Token color	Description	Type
Top-Level	UserInputKeyGeneration	User input for creating a new key	(Integer, IdentifierList, Large integer, String, Integer)
Top-Level	IdentifierList	List of identifiers for entity	[Identifier]
Top-Level	Identifier	Identifier for entity	String
Top-Level	KeyPair	Simple key pair	(PublicKey, PrivateKey)
Top-Level	PublicKey	Public key	(KeyFingerprint, Key, ExpirationDateUTC, KeyType, KeyLength)
Top-Level	PrivateKey	Private key	String
Top-Level	KeyFingerprint	Fingerprint of key	String
Top-Level	Key	Key itself	String
Top-Level	ExpirationDateUTC	Key's expiration date	Large integer
Top-Level	KeyType	Type of generated key (e.g. RSA)	String
Top-Level	KeyLength	Length of key	Integer

TABLE 3.4: Acronyms, names and description of token colors of Authcoin's top-level module - Part 1.

Module	Token color	Description	Type
Top-Level	EntityIdentityRecord	Contains all relevant information about an entity	(EIR_ID, Timestamp, PublicKey, Identifiers, Revoked)
Top-Level	EIR_ID	ID variable	Integer
Top-Level	Timestamp	Timestamp	Large integer
Top-Level	Identifiers	List of identifiers for an EIR	IdentifierList
Top-Level	Revoked	Has the EIR been revoked?	Boolean
Top-Level	IntList	List of integers	[Integer]
Top-Level	INTxCR	Holds a V&A-ID and a ChallengeRecord	(Integer, ChallengeRecord)
Top-Level	ChallengeRecord	Contains all information about a V&A challenge	(CR_ID, VAE_ID, Timestamp, ChallengeType, Challenge, VerifierEIR_ID, VerificationTargetEIR_ID)
Top-Level	CR_ID	ID variable	Integer
Top-Level	ChallengeType	Information on challenge type	String
Top-Level	Challenge	Description of the challenge	String
Top-Level	VerifierEIR, VerificationTargetEIR	EIR-IDs	Integer
Top-Level	INTxRR	Holds a V&A-ID and a ResponseRecord	(Integer, ResponseRecord)
Top-Level	ResponseRecord	Contains all information regarding a V&A response	(RR_ID, VAE_ID, Timestamp, CorrespondCR_ID, Response)
Top-Level	RR_ID, VAE_ID, CorrespondCR_ID	ID variables	Integer
Top-Level	Response	Response to corresponding challenge	String

TABLE 3.5: Acronyms, names and description of token colors of Authcoin's top-level module - Part 2.

Module	Token color	Description	Type
Top-Level	INTxSR	Holds a V&A-ID and a SignatureRecord	(Integer, SignatureRecord)
Top-Level	SignatureRecord	Contains all information regarding a V&A signature	(SR_ID, VAE_ID, Timestamp, ResponseRR_ID, ExpirationDate, Revoked, SuccessfulVA)
Top-Level	SR_ID, VAE_ID, ResponseRR_ID	ID variables	Integer
Top-Level	SuccessfulVA	States whether the V&A finished successfully or not	Boolean
Top-Level	VAR	Validation and authentication request	(VAR_ID, CreationDate, LastUpdated, VerifierEIR_ID, TargetEIR_ID, Status, VAE_ID)
Top-Level	VAR_ID, VAE_ID, TargetEIR_ID, VerifierEIR_ID	ID variables	Integer
Top-Level	CreationDate, LastUpdated	Timestamps	Large integer
Top-Level	Status	Current status of VAR (e.g. “pending”)	String

TABLE 3.6: Acronyms, names and description of token colors of Auhtcoin’s top-level module - Part 3.

Module	Token color	Description	Type
KeyGeneration- EstablishBinding	INTxOpenSSHInput	Holds a V&A-ID and an OpenSSHInput	(Integer, OpenSSH-Input)
KeyGeneration- EstablishBinding	OpenSSHInput	Placeholder input generated by OpenSSH or similar software	(Large integer, String, Integer)
KeyGeneration- EstablishBinding	INTxKeyPair	Holds a V&A-ID and a key pair	(Integer, KeyPair)
KeyGeneration- EstablishBinding	EIR	Entity identity record	EntityIdentityRecord
KeyGeneration- EstablishBinding	ID	ID	Integer
KeyGeneration- EstablishBinding	iList	Identifier list	IdentifierList

TABLE 3.7: Acronyms, names and description of token colors for the “KeyGeneration-EstablishBinding” module.

Module	Token color	Description	Type
V&A- Processing	Target	EIR of target	EntityIdentityRecord
V&A- Processing	Verifier	EIR of verifier	EntityIdentityRecord
V&A- Processing	VAE_ID_Counter	Counter variable for VAEs	Integer
V&A- Processing	VAE	V&A Entry	(INT, EntityIdentityRecord, EntityIdentityRecord)

TABLE 3.8: Acronyms, names and description of token colors for the “V&amp;A-Processing” module.

Module	Token color	Description	Type
CreateNew-KeyPair	OpenSSHKeyMaterial	Symbolic placeholder for key material provided by OpenSSH or similar software	(String, String, String)

TABLE 3.9: Acronyms, names and description of token colors for the “CreateNewKeyPair” module.

Module	Token color	Description	Type
EstablishBinding	n	Counter variable	Integer
EstablishBinding	pubKey	Public key	PublicKey
EstablishBinding	privKey	Private key	PrivateKey
EstablishBinding	ID1, ID2	ID variables	Integer

TABLE 3.10: Acronyms, names and description of token colors for the “EstablishBinding” module.

Module	Token color	Description	Type
FormalValidation	FVid	ID variable	Integer
FormalValidation	b	Boolean variable	Boolean
FormalValidation	KeyWellFormed	Placeholder for OpenSSH (or similar software) regarding the correct syntax of the provided key	Boolean
FormalValidation	extractedRevocation-Info	Info whether the key has been revoked yet	Boolean
FormalValidation	extractedKeyLength	Info about the key length	Integer
FormalValidation	extractedKey-ExpirationDateUTC	Info regarding the key’s expiration date	Large integer

TABLE 3.11: Acronyms, names and description of token colors for the “FormalValidation” module.



Module	Token color	Description	Type
V&A	CRforVerifier	CR for verifier	ChallengeRecord
V&A	CRforTarget	CR for target	ChallengeRecord
V&A	CRs	CR	ChallengeRecord

TABLE 3.12: Acronyms, names and description of token colors for the “V&amp;A” module.

Module	Token color	Description	Type
CreateSendResponse	RR	Response record	ResponseRecord

TABLE 3.13: Acronyms, names and description of token colors for the “Create-SendResponse” module.

Module	Token color	Description	Type
PostCRsAndRRs-ToBlockchain	TarID, VerID, IncID, CleanID	ID variables	Integer
PostCRsAndRRs-ToBlockchain	CleanUpDeadline	Timer	INTINF
PostCRsAndRRs-ToBlockchain	RRforVerifier	RR for verifier	ResponseRecord
PostCRsAndRRs-ToBlockchain	RRforTarget	RR for target	ResponseRecord

TABLE 3.14: Acronyms, names and description of token colors for the “PostCRsAndRRsToBlockchain” module.

Module	Token color	Description	Type
CreateChallenge-ForTarget	CR_ID_Counter	Counter variable	Integer
CreateChallenge-ForTarget	ChallengeTypeV	Further information on type of challenge	String
CreateChallenge-ForTarget	ChallengeVerifier	Challenge of verifier for target	String

TABLE 3.15: Acronyms, names and description of token colors for the “CreateChallengeForTarget” module.

Module	Token color	Description	Type
CreateChallenge- ForVerifier	ChallengeTarget	Challenge of target for verifier	String
CreateChallenge- ForVerifier	TargetCR_IDs	ID variable	Integer

TABLE 3.16: Acronyms, names and description of token colors for the “CreateChallengeForVerifier” module.

Module	Token color	Description	Type
CreateResponse	RR_ID_CounterV	Counter variable	Integer
CreateResponse	AcceptsChallenge	Does user process the challenge or not?	Boolean
CreateResponse	PerformChallenge	Answer to “Ac- ceptsChallenge”	Boolean
CreateResponse	CR_IDs	ID variable	Integer

TABLE 3.17: Acronyms, names and description of token colors for the “CreateResponse” module.

Module	Token color	Description	Type
CreateSignature- FromRR	SignatureLifespan	Lifespan of signature	INTINF
CreateSignature- FromRR	succB	User satisfied with response?	Boolean
CreateSignature- FromRR	SR_ID_CounterV	Counter variable	Integer
CreateSignature- FromRR	CR_IDs	ID variable	Integer
CreateSignature- FromRR	SR	Signature record	SignatureRecord

TABLE 3.18: Acronyms, names and description of token colors for the “CreateSignatureFromRR” module.

Module	Token color	Description	Type
ProcessVAR	VAR1	Validation and au- thentication request	VAR

TABLE 3.19: Acronyms, names and description of token colors for the “ProcessVAR” module.

Module	Token color	Description	Type
SymbolicMining	EIRcounter	Counter variable	Integer
SymbolicMining	EIRList	List of EIR-IDs	Integer list

TABLE 3.20: Acronyms, names and description of token colors for the “SymbolicMining” module.

Module	Token color	Description	Type
VARCreation	NextTargetID	ID of next VAR target	Integer
VARCreation	EIR_IDList	List of EIR-IDs	Integer list
VARCreation	n,k	Temporary variables	Integer

TABLE 3.21: Acronyms, names and description of token colors for the “VARCreation” module.

Module	Token color	Description	Type
FinishVAR	VARstatus	Status of corresponding VAR	String
FinishVAR	FailReason	Information about failing of VAR	String

TABLE 3.22: Acronyms, names and description of token colors for the “FinishVAR” module.

Module	Token color	Description	Type
EIRRevocation	c	Placeholder for OpenPGP interaction	Boolean
EIRRevocation	RevocationCertificate	Placeholder revocation certificate	String

TABLE 3.23: Acronyms, names and description of token colors for the “EIRRevocation” module.

Module	Token color	Description	Type
SignatureRevocation	SRID, CRID, RRID	ID variables	Integer

TABLE 3.24: Acronyms, names and description of token colors for the “SignatureRevocation” module.

### 3.6 Refined CPN Models

The following section answers RQ-1.3 - What are the refining CPN models? - and presents the further refined sub-modules of the top-level CPN model presented in Section 3.4. The section is structured hierarchically. First, in Section 3.6.1 the refined sub-modules of the top-level models are depicted. Subsequently, Section 3.6.2, Section 3.6.3, Section 3.6.4 and Section 3.6.5 illustrate further refinements of these sub-modules down to the bottom-level.

The source file of the presented CPN models is available in Appendix C.

#### 3.6.1 Top-Level Modules

This section presents the four sub-modules of the top-level CPN model presented earlier in Figure 3.6. First, Figure 3.7 illustrates the process of generating a new key pair and establishing a binding between a new key pair and the owning entity. The user provides a list of identifiers, an expiration date, a key type and key length as input. The input is processed and results in a new key pair and an EIR (EntityIdentityRecord) that is posted to the blockchain. An EIR contains all identity related information of an entity.

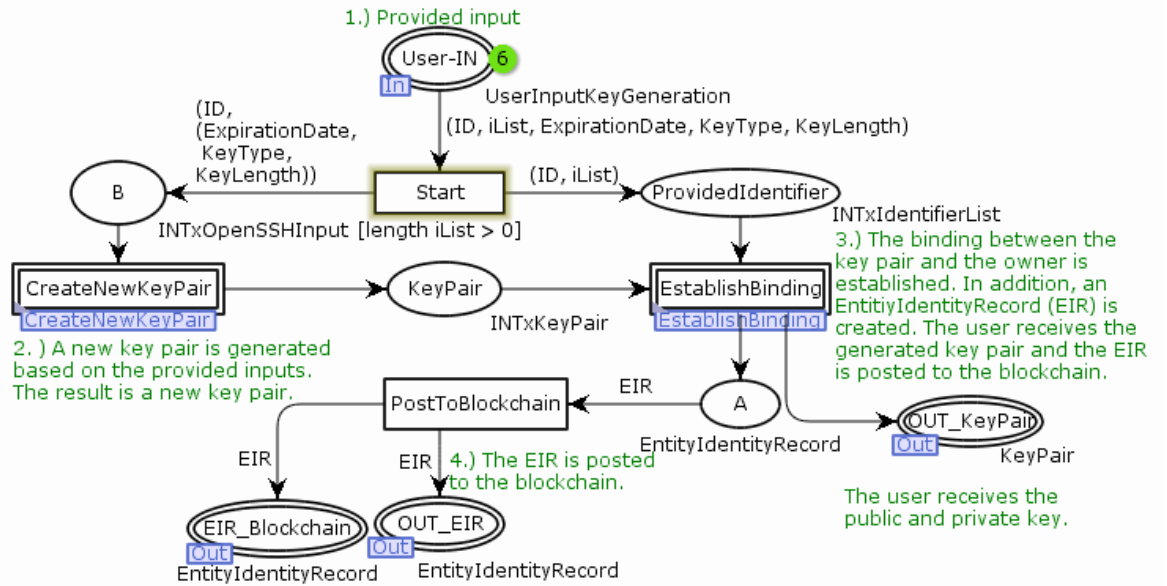


FIGURE 3.7: CPN model of the “KeyGenerationEstablishBinding” module

Figure 3.8 illustrates the “V&A-processing” module in more detail. A set of EIRs is provided as an input, one for the target and one for the verifier. Both EIRs are further processed to create a VAE (V&A Entry) that consists of an ID for this specific V&A process and the target as well as the verifier EIR. The VAE is then further processed in

the “FormalValidation” module (more details in Figure 3.13). If the formal validation is successful, the VAE is passed to the V&A module. If it fails, the V&A processing is canceled and marked as “failed”. In the V&A module (more details in Figure 3.14), the actual validation and authentication is executed, which either fails or succeeds.

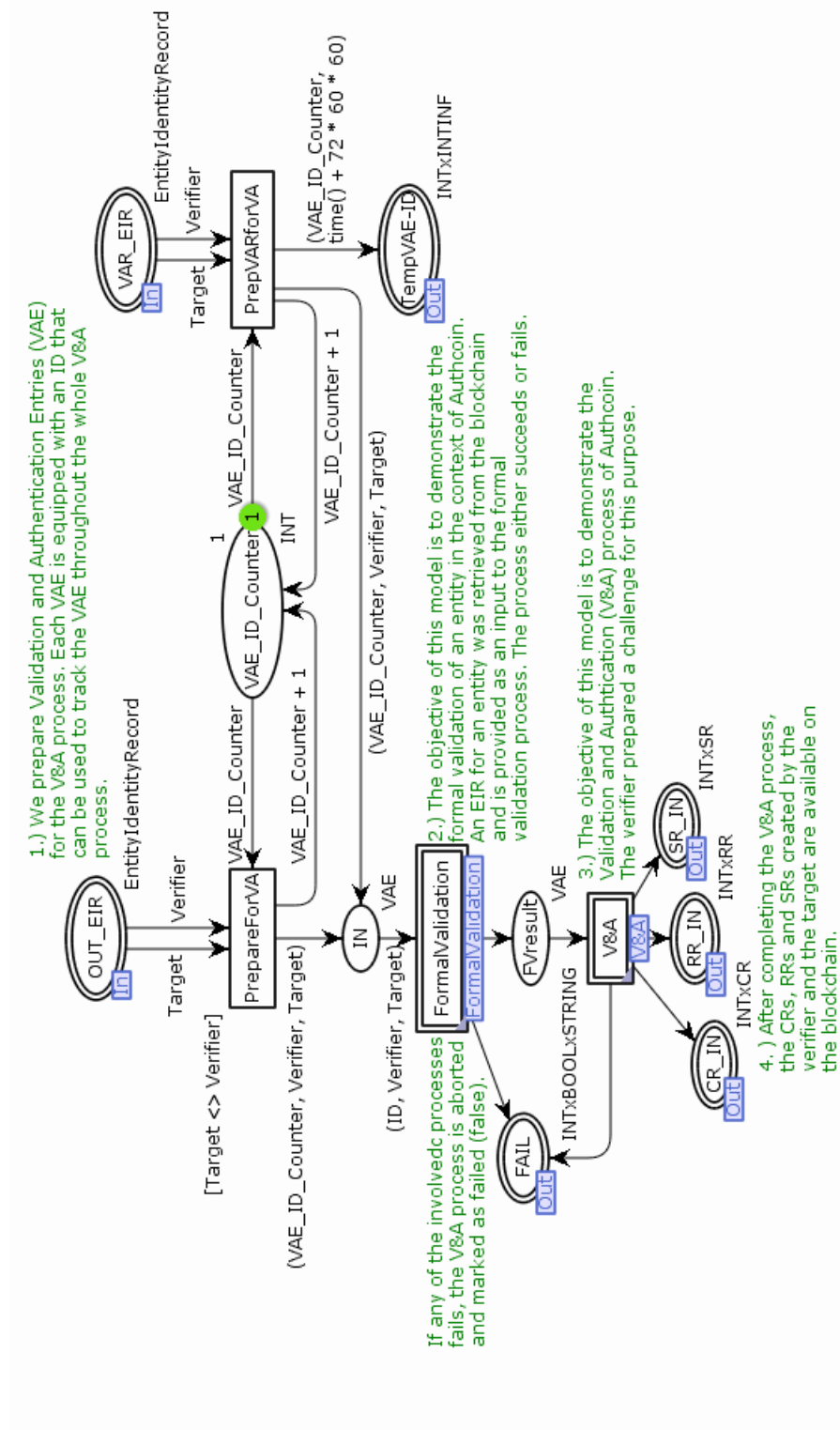


FIGURE 3.8: CPN model of the “V&A-Processing” module

Depending on whether the V&A processing finished successfully, the corresponding information (CRs, RRs and SRs) are posted to the blockchain as illustrated in Figure 3.9.

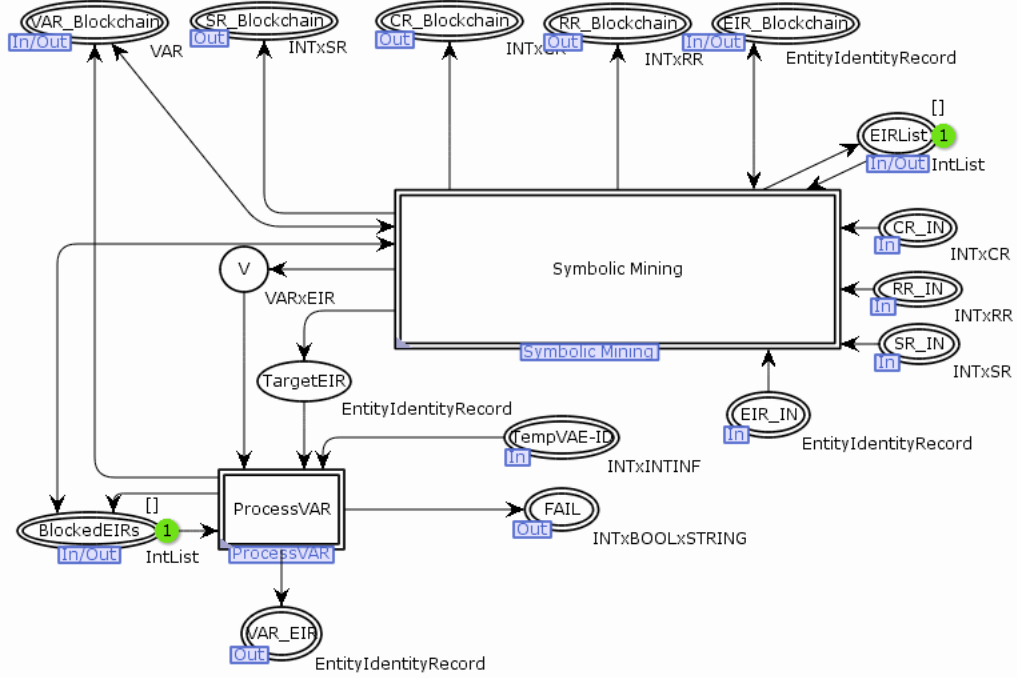


FIGURE 3.9: CPN model of the "Mining" module

The last sub-module of the top-level CPN model is the "Revocation" module as shown in Figure 3.10. It is possible to revoke either signatures in form of SRs or to revoke EIRs. An EIR can be revoked if it is no longer required or not trustworthy anymore. SRs can be revoked in case that the signing entity has to remove the expressed trust relationship. The updated information is posted back to the blockchain.

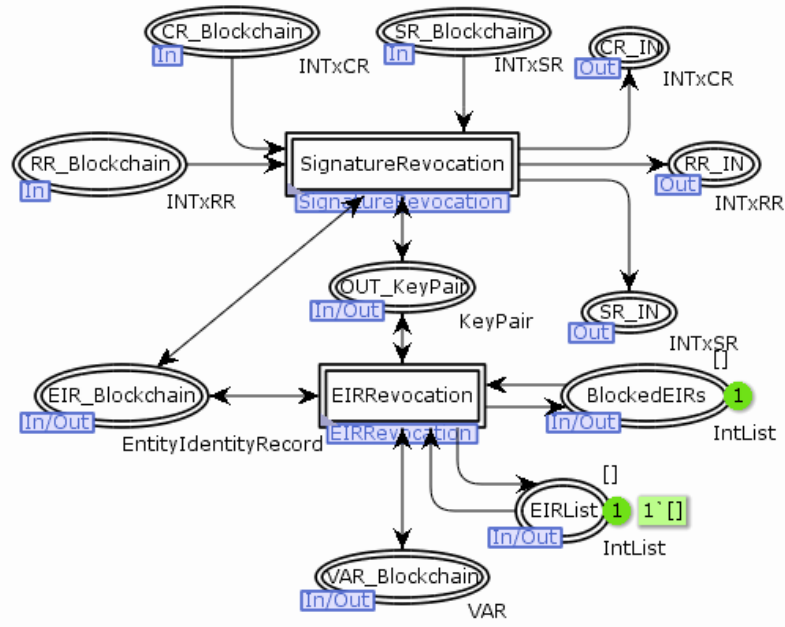


FIGURE 3.10: CPN model of the “Revocations” module

### 3.6.2 Sub-Modules of “KeyGenerationEstablishBinding”

The Figures 3.11 and 3.12 provide a more detailed depiction of the sub-modules of the “KeyGenerationEstablishBinding” module from Figure 3.7. In Figure 3.11, the input provided by the user is used to create a new public and private key using OpenSSH which is only modeled in a symbolic way. Depending on the use case, it is also possible to use similar software such as OpenPGP.

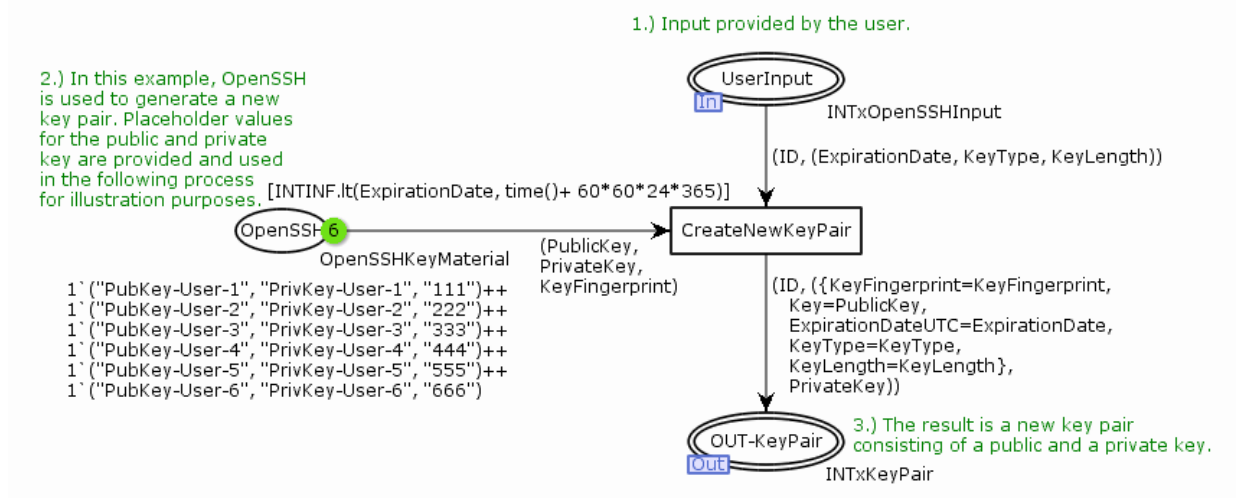


FIGURE 3.11: CPN model of the “CreateNewKeyPair” module

In Figure 3.12, a binding between the generated key pair and the provided identifiers is established. The result is an EIR that is posted to the blockchain. The private key stays with the user and is never made public.

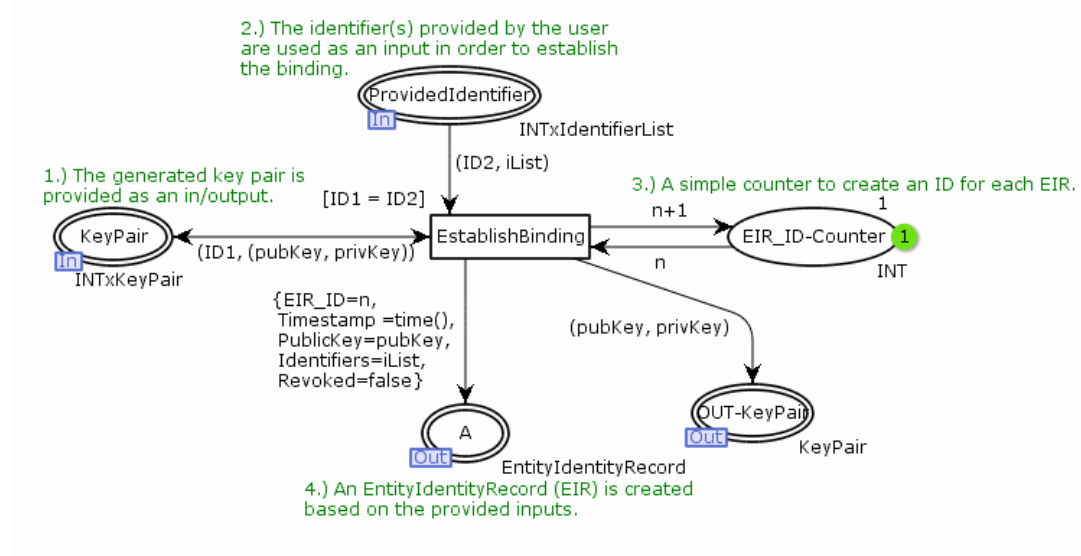


FIGURE 3.12: CPN model of the “EstablishBinding” module

### 3.6.3 Sub-Modules of “V&A-Processing”

The Figures 3.13 and 3.14 refine the CPN model of the “V&A-Processing” module presented in Figure 3.8. First, in Figure 3.13 Authcoin’s formal validation procedure is executed for both EIRs. It is verified if the public keys of each EIR are well formed, have a sufficient key length and have not been expired or revoked yet. If one test fails, the V&A processing fails. Otherwise the VAE is further processed in the “V&A” module depicted in Figure 3.14.



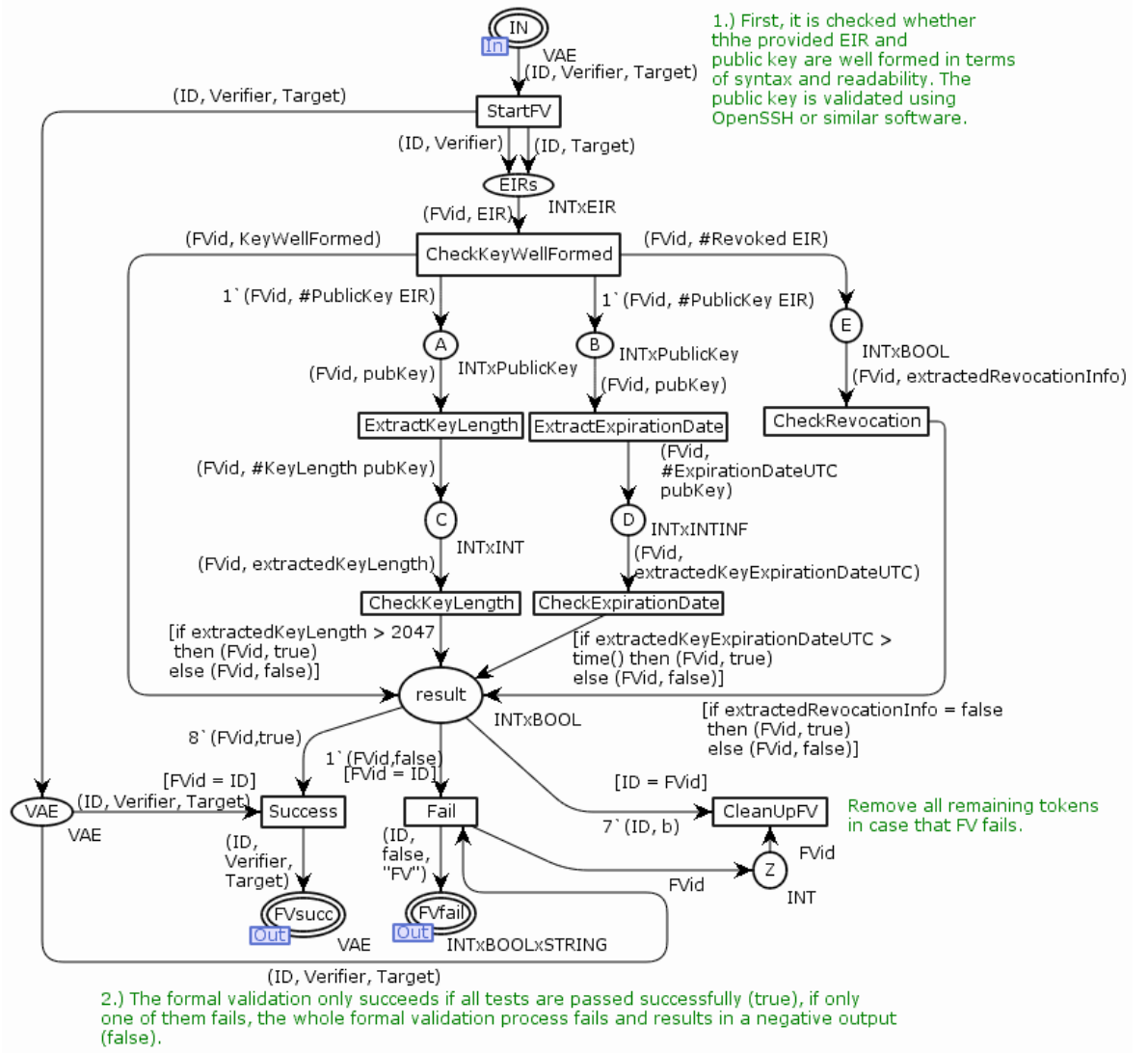


FIGURE 3.13: CPN model of the “FormalValidation” module

During the V&A, the verifier and target exchange challenges (CR - ChallengeRecord), with each other and create the corresponding responses (RR - ResponseRecord) as detailed earlier in Section 2.1.2 and Section 2.1.4. Both entities evaluate the received responses and create corresponding signatures (SR - SignatureRecord) depending on whether they are satisfied with the received response or not. All information are posted to the blockchain. If any of these steps fails, the whole V&A process stops and the specific VAE is marked as failed.

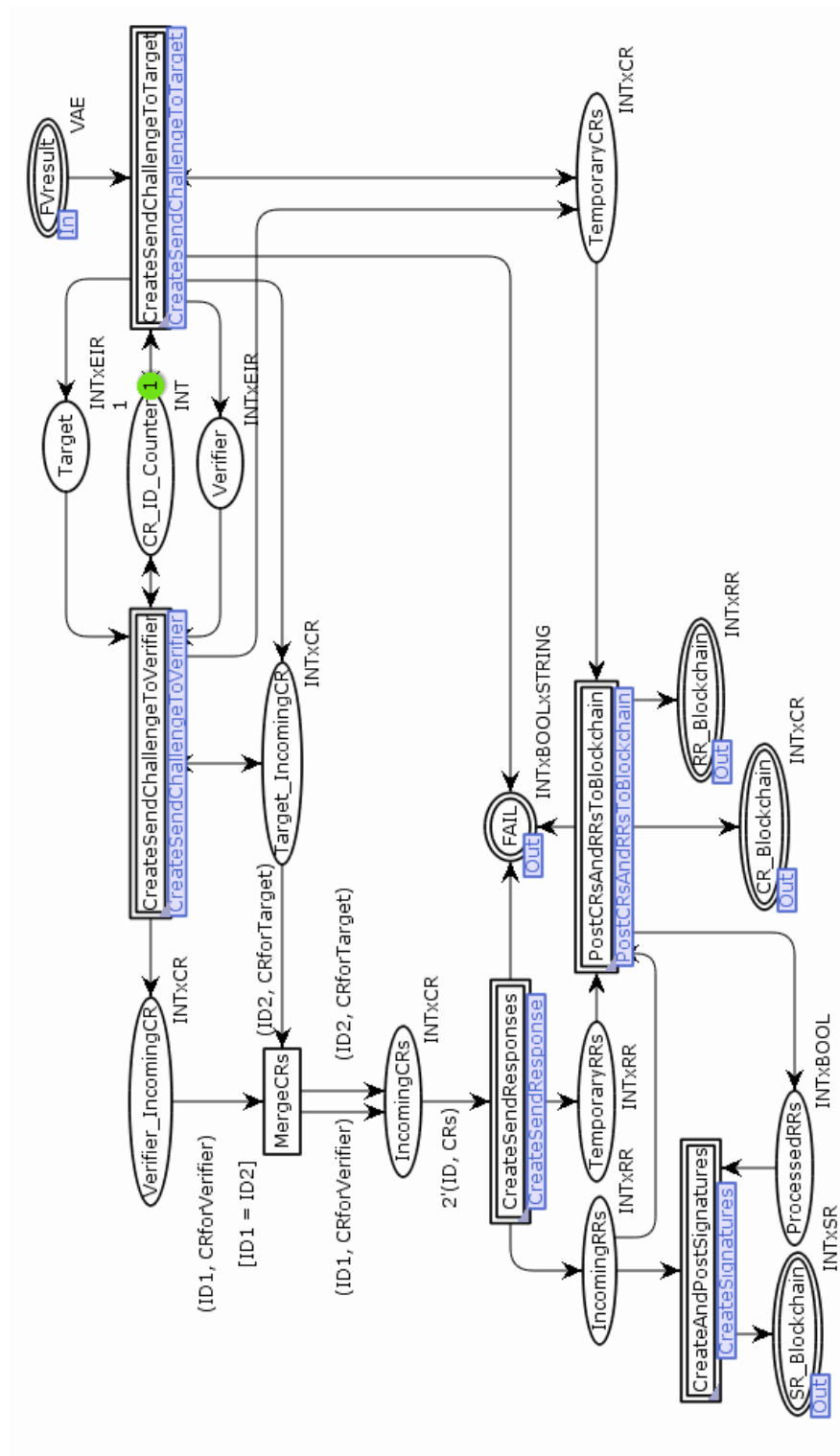


FIGURE 3.14: CPN model of the “V&A” module

### 3.6.3.1 Sub-Modules of “V&A”

The following figures provide detailed depictions of the sub-modules of the “V&A” module of Figure 3.14. Figure 3.15 refines the “CreateSendChallengeToTarget” module. The

verifier, who initiated the V&A process, creates a challenge in form of a CR, for the target and afterwards sends the same record to the target. Figure 3.16 depicts the same procedure where the target creates a challenge for the verifier.

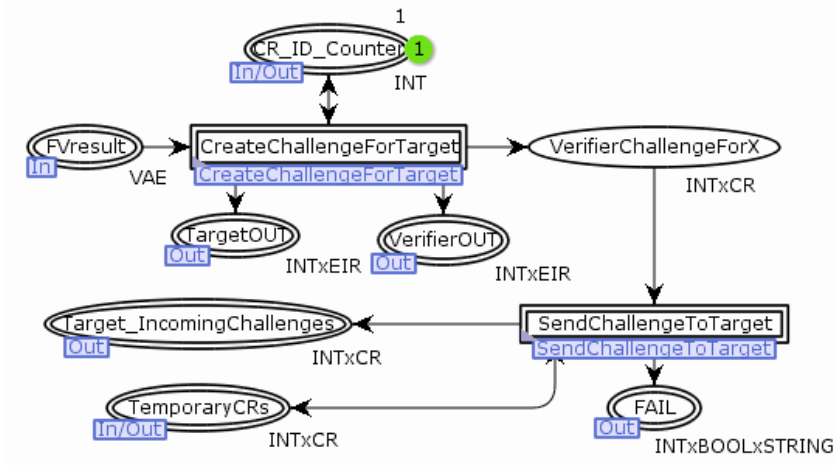


FIGURE 3.15: CPN model of the “CreateSendChallengeToTarget” module

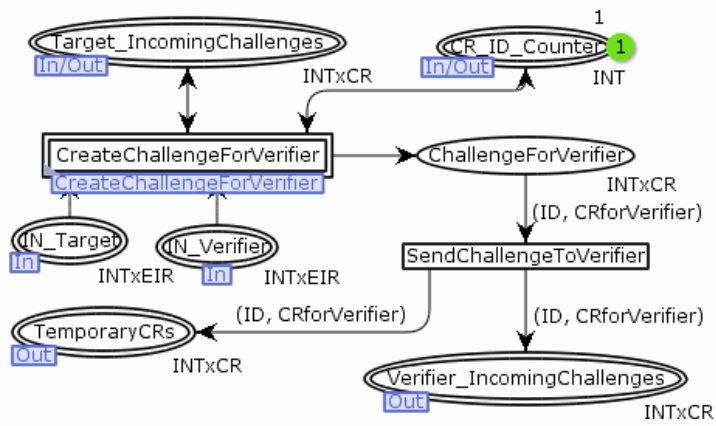


FIGURE 3.16: CPN model of the “CreateSendChallengeToVerifier” module

Based on the created CRs, Figure 3.17 presents the process of creating the corresponding responses of the verifier and the target. Depending on the outcome of these responses, either a positive or negative SR is created and posted to the blockchain as illustrated in Figure 3.18.

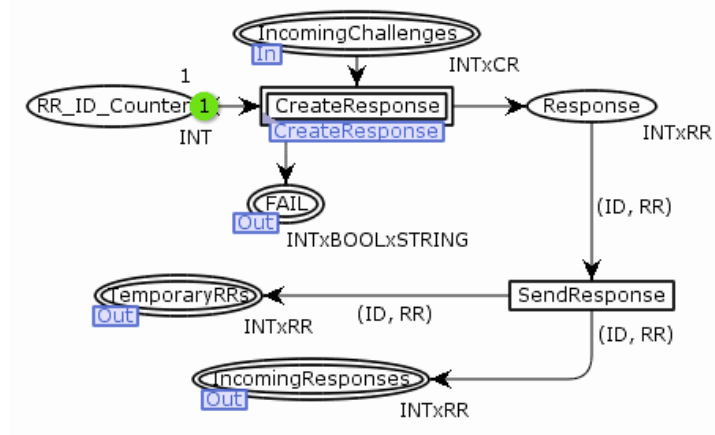


FIGURE 3.17: CPN model of the “CreateSendResponses” module

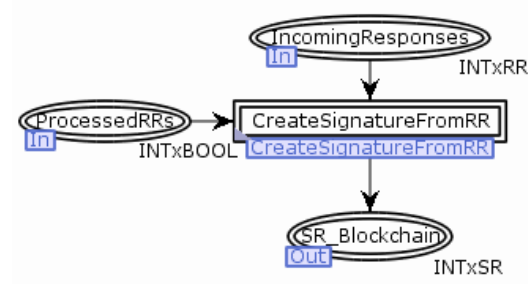


FIGURE 3.18: CPN model of the “CreateSignatures” module

Finally, Figure 3.19 shows the process of posting CRs and RRs to the blockchain. When a CR is posted to the blockchain, a 48 hour timer is triggered. The timer ensures that both involved parties submit the corresponding RRs within 48 hours and post them to the blockchain, otherwise the V&A process fails and terminates.

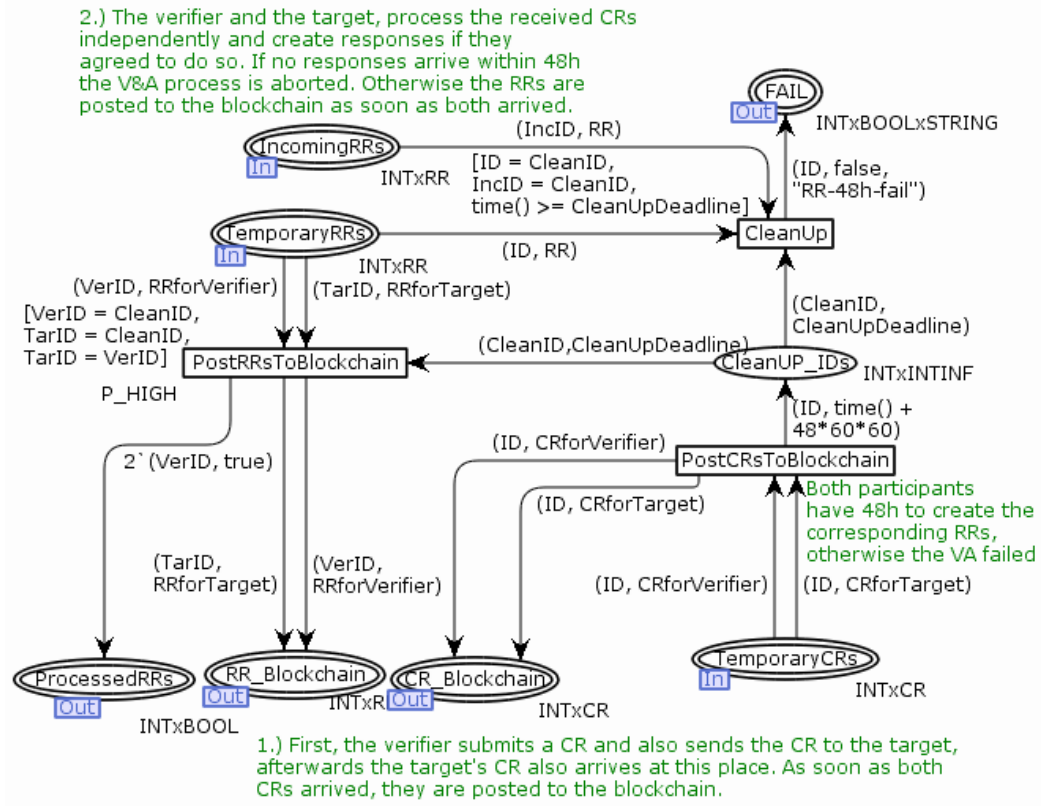


FIGURE 3.19: CPN model of the “PostCRsAndRRsToBlockchain” module

### 3.6.3.2 CreateSendChallengeToTarget

The following Figures 3.20 and 3.21 provide a more detailed view on the verifier's process of creating and sending a challenge to the target as depicted previously in Figure 3.15. In our CPN model of the “CreateChallengeForTarget” module, we use a placeholder challenge in form of string for the ease of modeling. In reality, users can create a customized challenge as explained in Section 2.1.4.

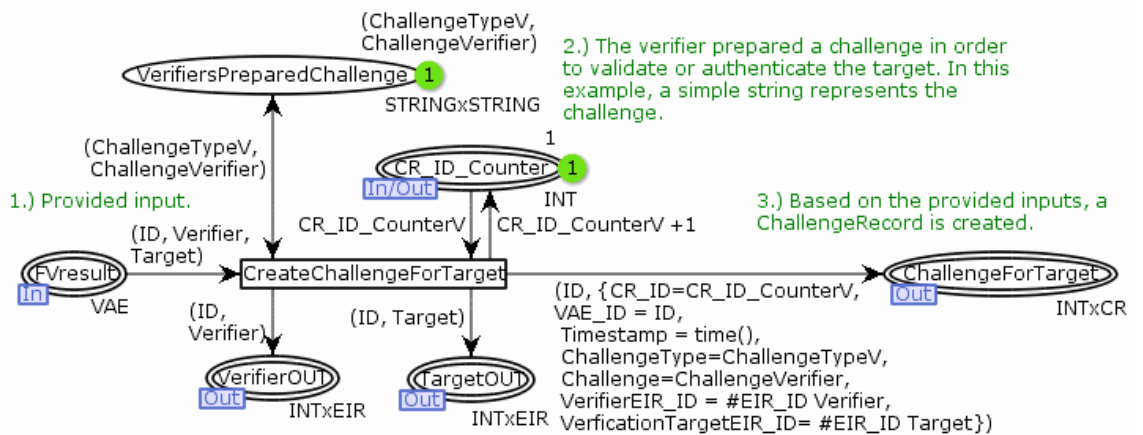


FIGURE 3.20: CPN model of the “CreateChallengeForTarget” module

The process of sending the CR is presented in Figure 3.21. The CR is forwarded to the target and triggers a timer. If the target refuses to respond to the challenge or fails to create a corresponding challenge for the verifier, then the whole V&A process fails and terminates.

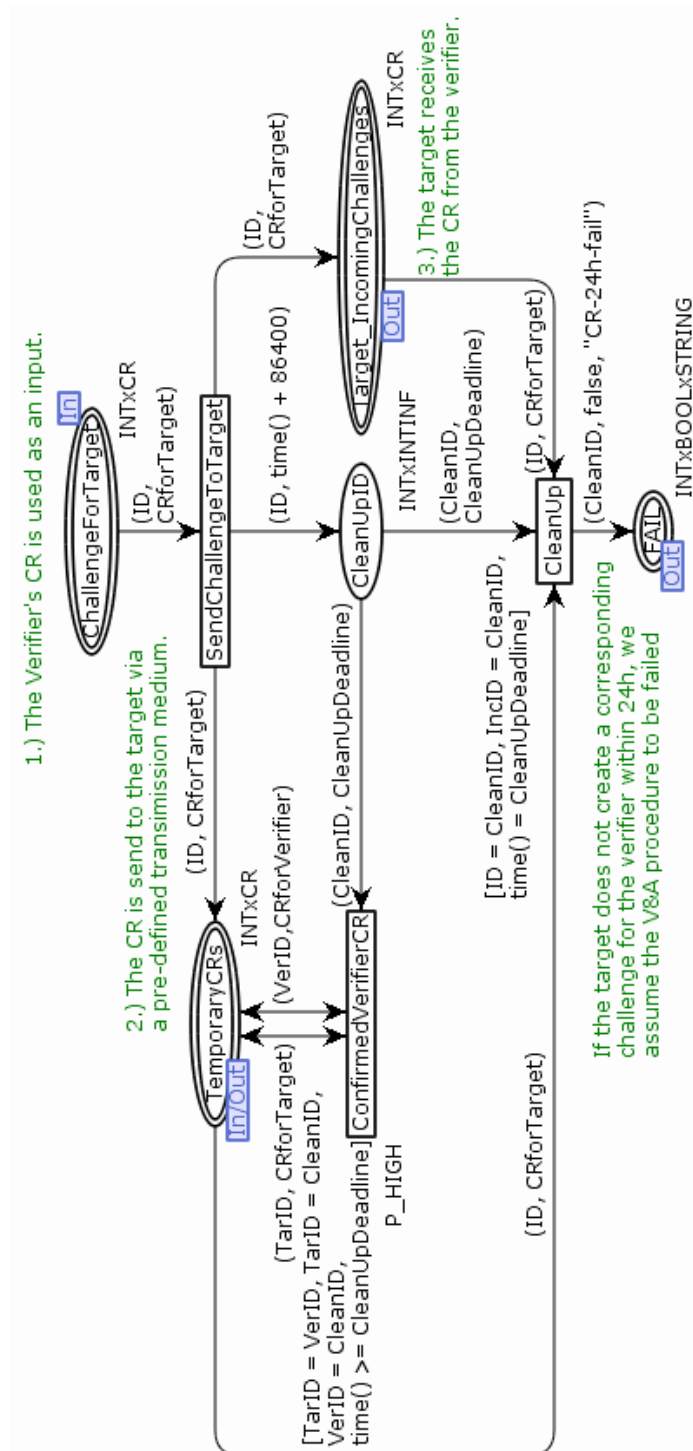


FIGURE 3.21: CPN model of the “SendChallengeToTarget” module

### 3.6.3.3 CreateSendChallengeToVerifier

The following Figure 3.22 illustrates a similar process of creating a challenge as presented in Figure 3.20, but for the verifier instead of the target.

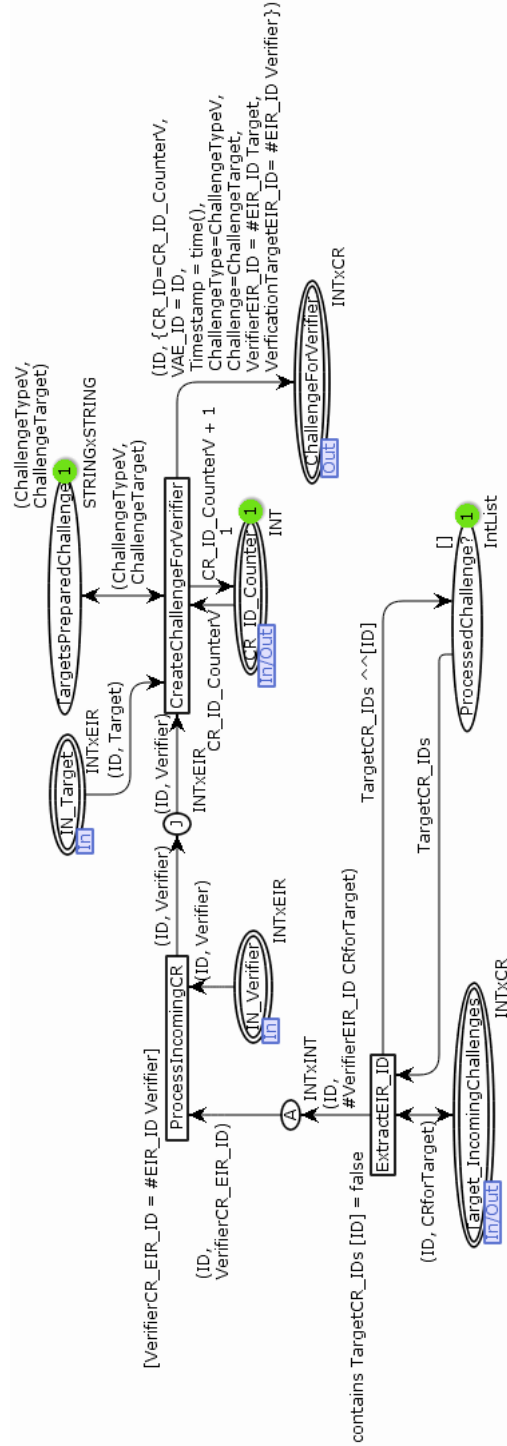


FIGURE 3.22: CPN model of the “CreateChallengeForVerifier” module

### 3.6.3.4 CreateSendResponses

As soon as the verifier and the target received their challenges, the incoming CRs are processed and corresponding RRs are created for each of the CRs. This procedure is depicted in Figure 3.23 and starts with the processing of the incoming CR and extracting the challenge. Afterwards, the involved user decides whether or whether not he/she accepts to fulfill the challenge. In some cases it is possible that a user decides that it is not possible to fulfill a given challenge and therefore refuses to proceed. In this case the V&A process fails and terminates no matter how the other involved entity proceeds with its own CR. If the entities decide to fulfill their challenges, corresponding RRs with the responses are created and send to the challenger.

### 3.6.3.5 CreateAndPostSignatures

The last sub-module of the V&A module is shown in Figure 3.24 and illustrates the process of creating a positive or negative SR for an incoming response to a challenge. The user decides if he/she is satisfied with provided response and specifies the lifespan of the created signature. Afterwards, the resulting SR is posted to the blockchain.

## 3.6.4 Sub-Modules of “Mining”

The four CPN models presented in Figure 3.25, 3.26, 3.27 and 3.28 refine the “Mining” module. First, in Figure 3.25 the mining of a new blockchain block is modeled in a symbolic way. In our CPN model, every time a new EIR, CR, RR or SR is posted to the blockchain, a new block is mined followed by the creation of a new VAR which is also posted to the chain. The creation of a new VAR, illustrated in Figure 3.26, is triggered when a new block is added to the blockchain. The role and functionality of a VAR was explained earlier in Section 2.1.5 as well as in [16], but the presented CPN model is different in two aspects. First, in the model users do not chose VARs on their own, instead a user is randomly chosen. Second, the model is limited to process only two VARs in order to avoid an endless loop during the simulation. The “ProcessVAR” module describes the processing steps of a VAR chosen by a user. First, the status of the VAR is updated in order to avoid multiple processing of the same VAR. Afterwards, the EIR of the VAR’s target is retrieved. In combination with the verifier’s EIR, the V&A process of the “V&A-processing” module is triggered and executed. The results of the V&A are processed in the “FinishVAR” module as depicted in Figure 3.28. The results of the V&A process are used to update the pending VAR and change the status of the VAR to “finished”. The updates are posted to the blockchain.



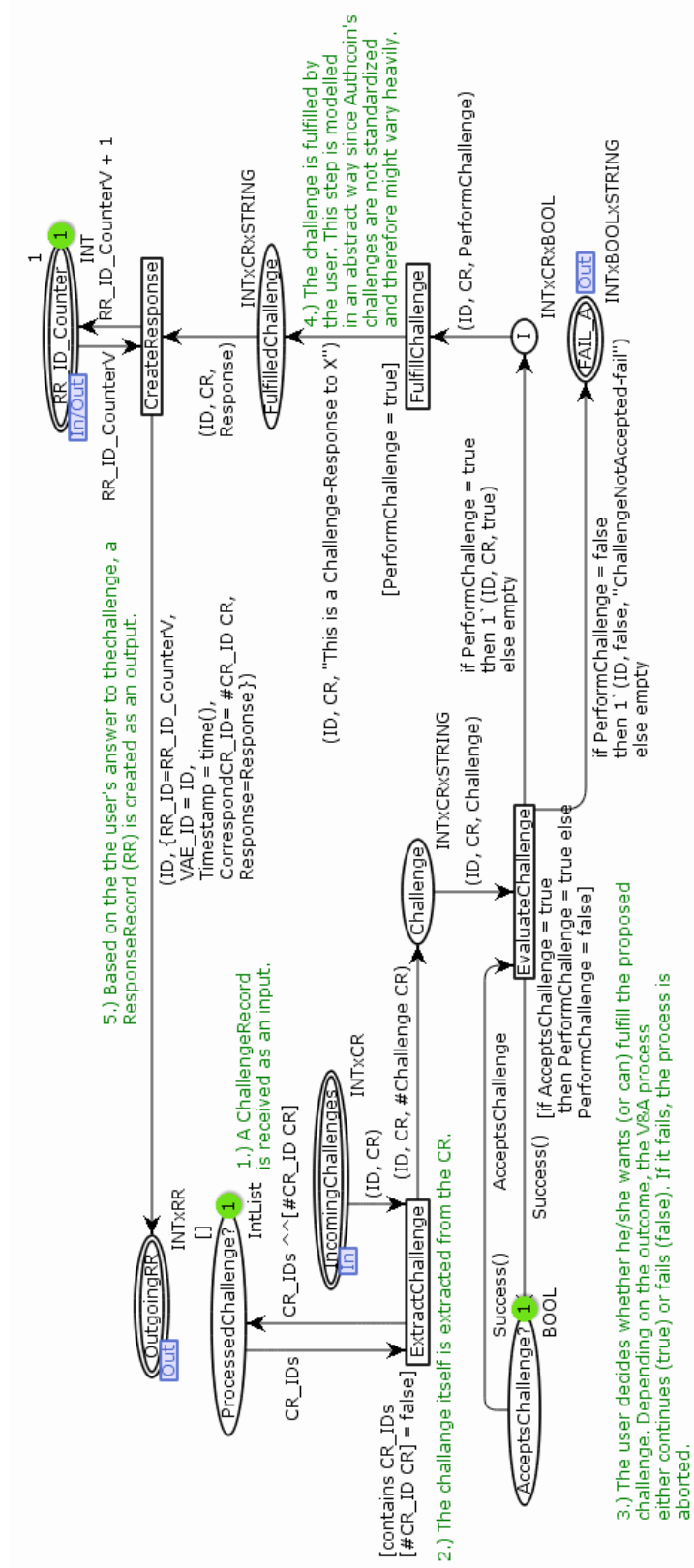


FIGURE 3.23: CPN model of the “CreateResponse” module

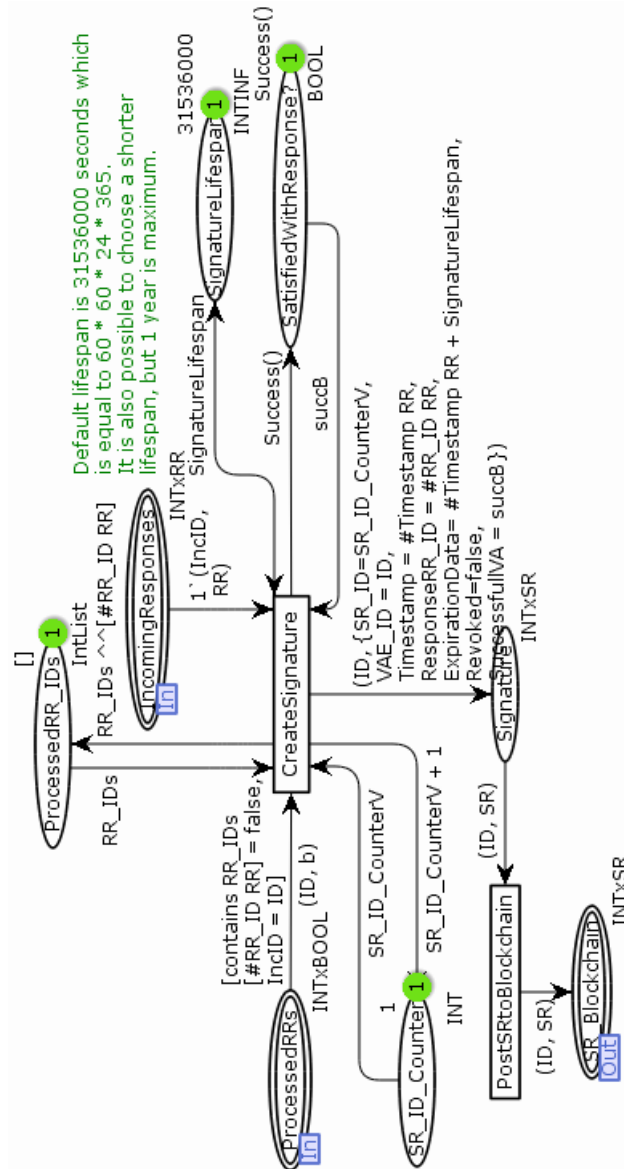


FIGURE 3.24: CPN model of the “CreateSignaturesFromRR” module

### 3.6.5 Sub-Modules of “Revocations”

The “Revocation” module in Figure 3.10 contains two sub-modules. One for the revocation of EIRs and a second one for the revocation of SRs. Figures 3.29 and 3.30 illustrate the two sub-modules. An EIR is revoked by generating a revocation certificate for the contained public key and revoking the key. Afterwards, the EIR is updated accordingly and the updates are pushed to the blockchain. In addition, all open or pending VARs for the revoked EIR are closed.

Revoking a SR works similarly. Based on the provided inputs, the SR is revoked and the updated version is posted to the blockchain.

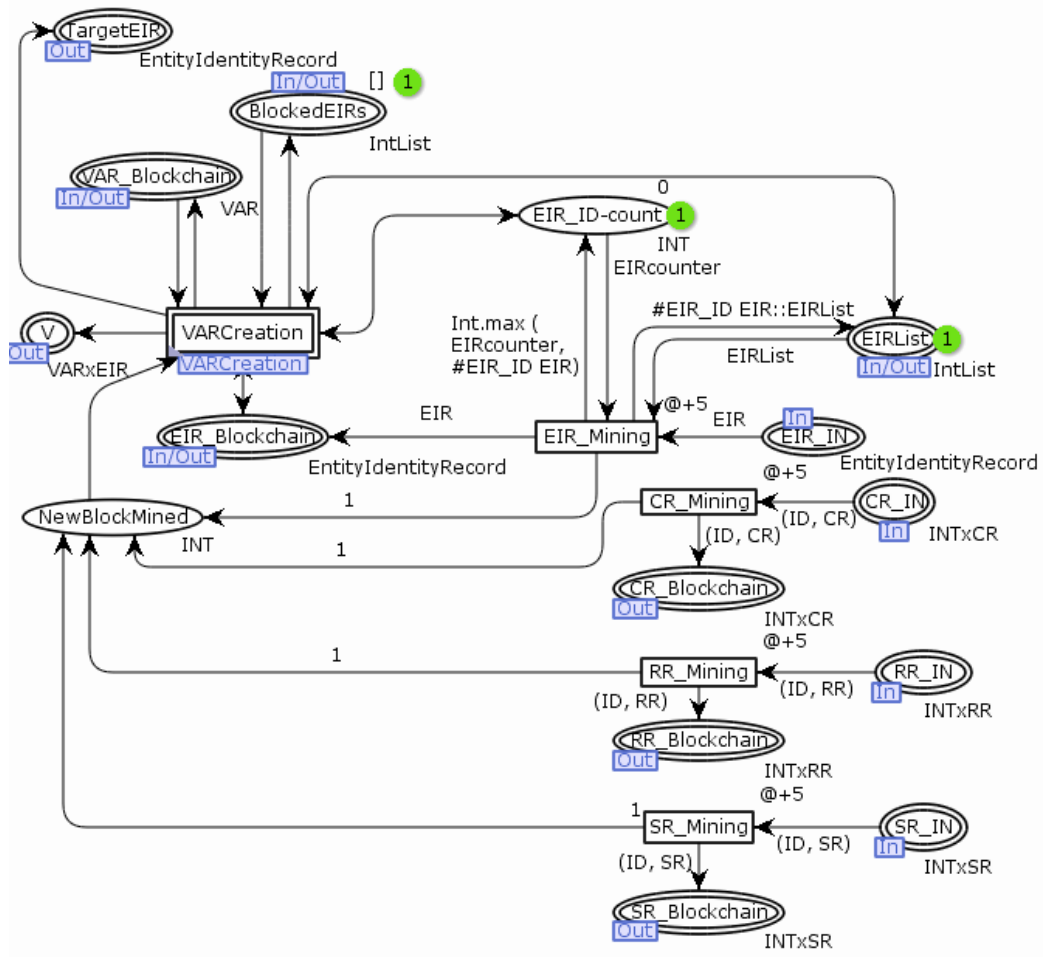


FIGURE 3.25: CPN model of the “SymbolicMining” module

### 3.7 Discussion

As shown in Section 3.4 and Section 3.6, we implement the high-level descriptions of the Authcoin protocol as described in [16], by mapping the conceptual AOM models onto a formal CPN representation. Nevertheless, the derived CPN models have certain limitations. First, the blockchain mining process is implemented in a symbolic way since Authcoin can be deployed on top of different blockchain systems with varying mining-concepts. Furthermore, the process of mining a new block does not affect the protocol itself as long as it guarantees that a transaction posted to the blockchain is mined in a given timespan. It is only relevant for Authcoin that with each new block a defined number of VARs is generated. Therefore, we implement the mining process in a symbolic way as illustrated in Section 3.6.4. In addition, we artificially limited the number of processed VARs to two, in order to avoid an infinite loop of VAR processing. Furthermore, the CPN models do not contain limitations regarding which user can fulfill a VAR, or not, as described in [16].

Due to the socio-technical nature of the Authcoin protocol, certain aspects of the model

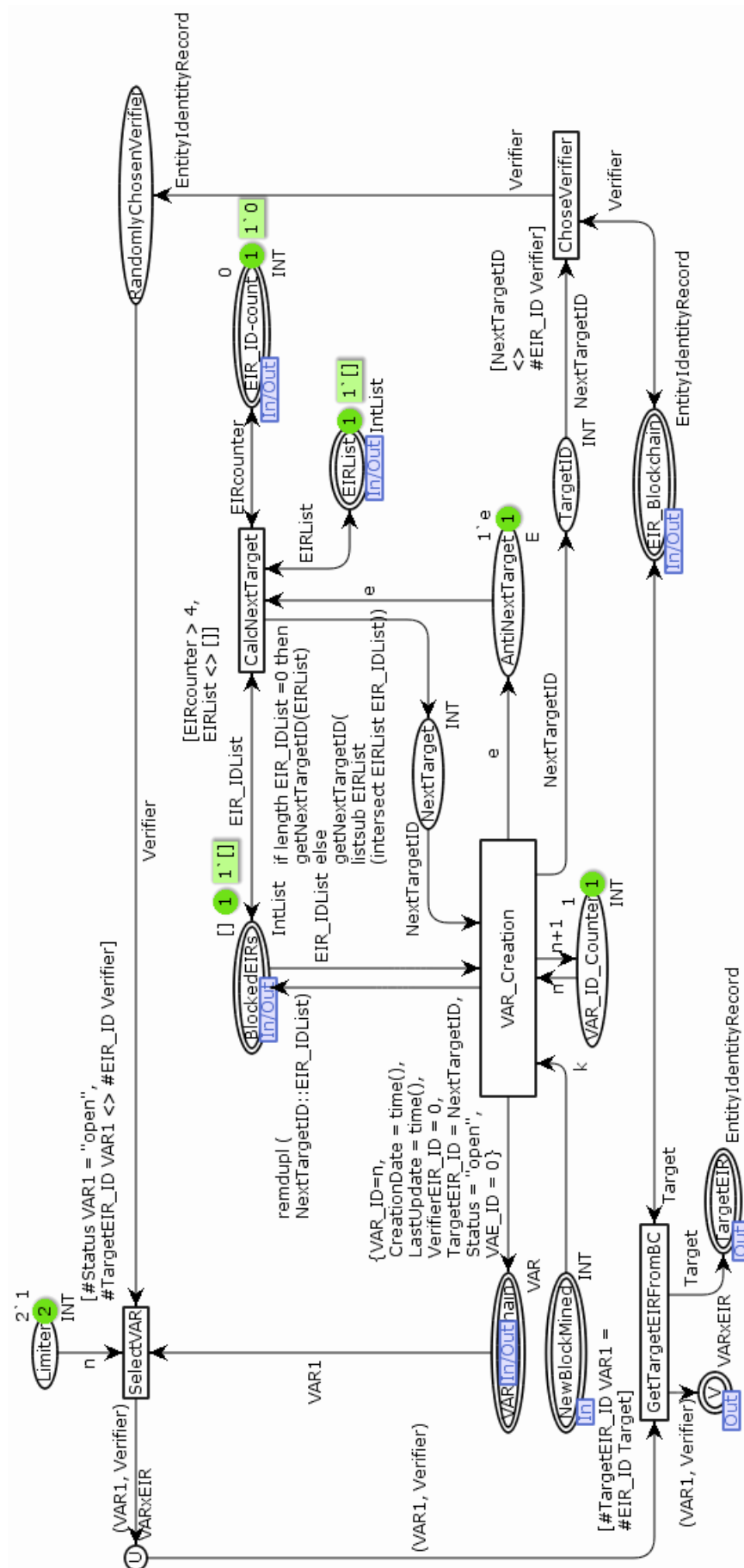


FIGURE 3.26: CPN model of the “VARCreation” module

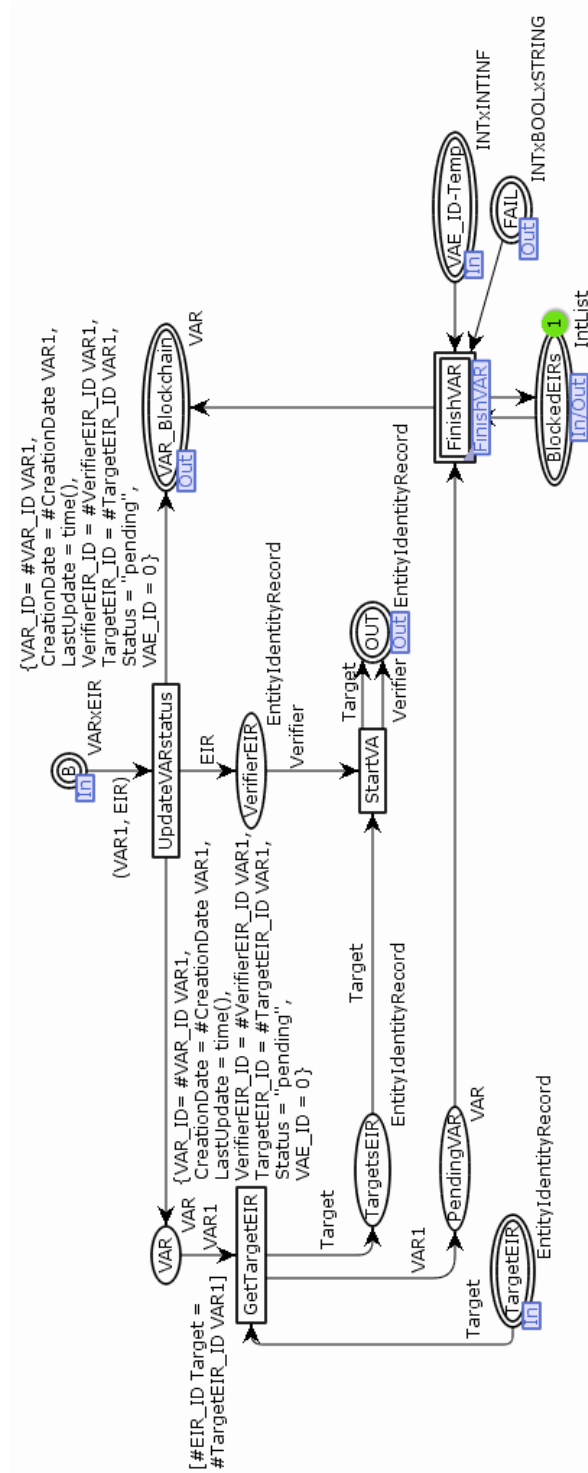


FIGURE 3.27: CPN model of the "ProcessVAR" module

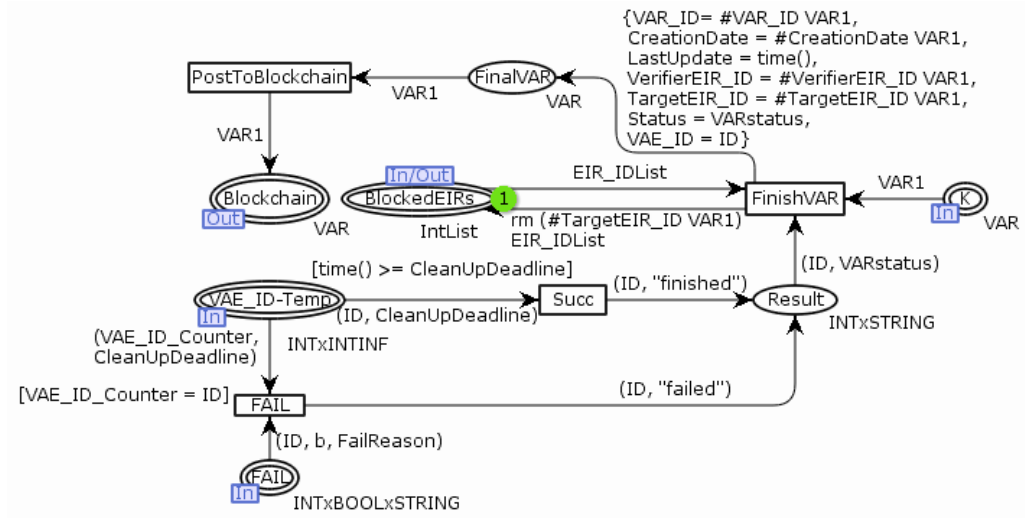


FIGURE 3.28: CPN model of the "FinishVAR" module

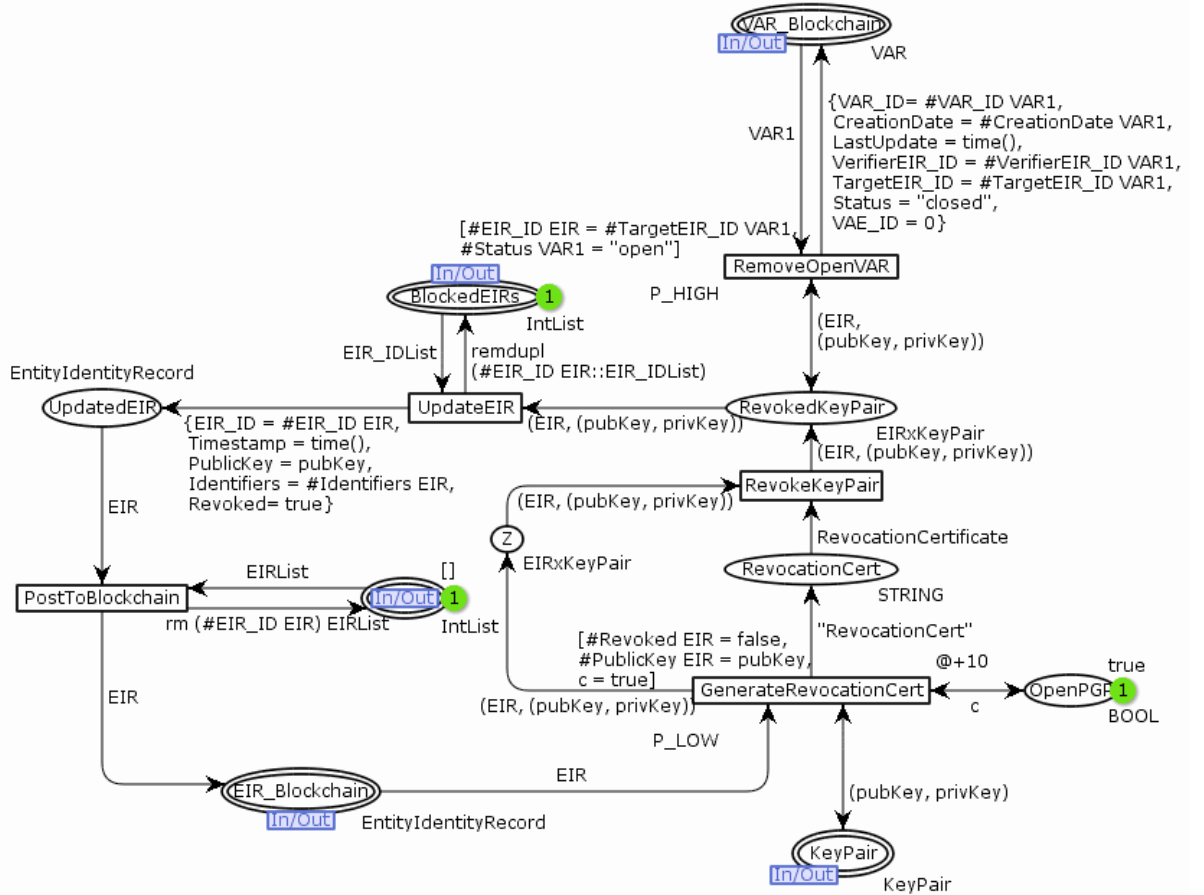


FIGURE 3.29: CPN model of the "EIRRevocation" module



Moreover, the top-level model as well as the further refined sub-modules are illustrated and described in depth.

Authcoin is formalized using Colored Petri Nets and an agent-oriented modeling methodology used for creating the goal models as well as the corresponding behavior interface models. Based on these models, the top-level CPN model as well as the refined CPN models are derived and implemented using CPN-Tools. The top-level CPN model consists of four sub-modules that are further refined and described. Furthermore, the required protocol semantics are provided in detail by defining the necessary token color sets representing the used data structures. Moreover, we acknowledge and discuss the limitations of the CPN models resulting either from simplification of operations or the socio-technical nature of the Authcoin protocol. That includes e.g. the symbolic mining of new blockchain blocks or the simulated user-input at certain points of the protocol execution.

For future work, we plan to replace the symbolic modeling and mining of the blockchain with a complete CPN model of a specific blockchain implementation for better integration of the VAR creation process into the Authcoin CPN model.



## Chapter 4

# Risk and Threat Analysis of the Authcoin Protocol

*In the following chapter a risk and threat analysis based on the ISSRM domain model is performed on the Authcoin protocol. The assets of Authcoin are identified and for each asset potential threat agents, attack methods, threats, vulnerabilities, events, impacts and risks are analyzed. Afterwards, security requirements and security controls necessary to mitigate the identified risks are provided.*

### 4.1 Introduction

The objection of Chapter 4 is to answer the research question RQ-2 - How to analyze security threats of the Authcoin protocol? - as outlined earlier in Chapter 1. In order to answer RQ-2 in a more structured and comprehensive way, it is divided into three subquestions:

- **RQ-2.1:** Which assets and data object are involved in the Authcoin protocol?
- **RQ-2.2:** Which risks threaten the Authcoin protocol?
- **RQ-2.3:** What are the necessary security requirements and security controls?

Each subquestion is answered independently in a separate section. The first part of this chapter focuses on the introduction of security risk management system used in

information systems as described in Section 4.2. Afterwards, Section 4.3 answers RQ-2.1 and focuses on the identification of assets of the Authcoin protocol. Subsequently, Section 4.4 deals with RQ-2.2 and describes the risk and threat analysis performed on the identified assets of the previous section. Section 4.5 answers RQ-2.3 and outlines the security requirements and controls that are necessary to mitigate the identified risks. Finally, Section 4.6 discusses the findings of this chapter followed by a conclusion in Section 4.7.

## 4.2 Information Systems Security Risk Management (ISSRM) Domain Model

In order to determine what kinds of security issues threaten the Authcoin protocol, it is necessary to identify the underlying risks. Identifying such risks is part of software risk management, “a discipline whose objectives are to identify, address and eliminate software risk items before they become threats to successful software operation or major sources of expensive software rework” [63].

According to Dubois et al. [64], there are over 200 risk management methods and frameworks, including e.g. CRAMM [65], OCTAVE [66] or CORAS [67]. Such risk management methods propose guidelines “that help to identify vulnerable assets, determine security objectives, and assess risks as well as define and implement security requirements to treat the risks” [64]. Closely related to the first and second phase of the multi-step OCTAVE framework is the ISSRM domain model, that provides guideline on the identification of risks, analysis of risks and the design of security requirements for managing and mitigating the risks.

The ISSRM domain model is based on an analysis of security and risk management standards, methods and frameworks and provides a methodology for analyzing, evaluating and quantifying security risks [68][64]. A representation of the ISSRM domain model is provided in Figure 4.1 and illustrates the key concepts of the model. The three key concepts are: Asset-related concepts, risk-related concepts and risk treatment-related concepts.

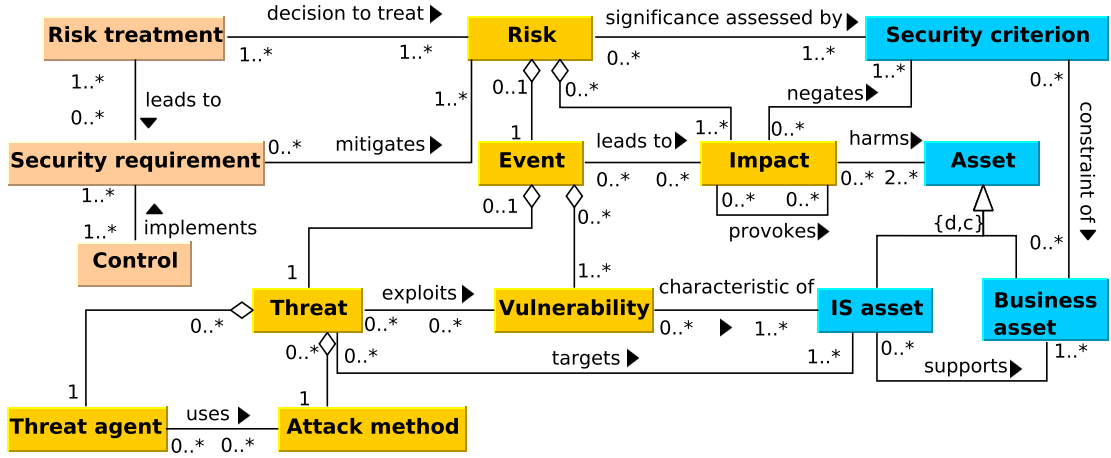


FIGURE 4.1: ISSRM domain model (Adapted from: [64])

Asset-related concepts describe which important assets have to be protected and the criteria that guarantee the assets security. Assets can be immaterial as well as material. Risk-related concepts deal with risks itself and immediate components. “A risk is composed of an event that leads to one or more negative impacts. An impact negates one or more security criterion and as a result harms the assets. An event is a consequence of a threat exploiting a vulnerability in the system. A threat is a potential attack, carried out by a threat agent using an attack method” [39]. Risk treatment-related concepts cover the decisions, security requirements and controls for treating identified security risks by avoiding, reducing, transferring or retaining the risk. Security requirements refine a security risk treatment that mitigates a risk and security controls implement the security requirements [64].

### 4.3 Identification of Assets

In order to identify the risks that threaten Authcoin’s security, it is necessary to identify the relevant assets of the protocol and its users. Therefore, the goal of this section is to answer RQ-2.1: Which assets and data object are involved in the Authcoin protocol? Such assets and data object include involved systems and processes as well as information exchanged during the protocol execution between different entities and processes. To do so, we identify the relevant systems and processes in Section 4.3.1. Afterwards the exchanged information that need to be secured are identified in Section 4.3.2.

### 4.3.1 Systems and Processes

Based on the CPN models of Authcoin presented in Chapter 3, we identify the following systems and processes.

**Systems:**

- Messaging system
- Blockchain
- User devices and operational software
- Underlying communication network

**Processes:**

- Send - Send information to other entities (e.g. CRs, RRs, etc.)
- Receive - Receive information from other entities (e.g. CRs, RRs, etc.)
- Post - Post information to the blockchain

Some of the identified assets, e.g. blockchain, user devices and the underlying communication network pose a security risk but cannot be mitigated in context of the Authcoin protocol. For example, it is possible that a user's device, operation system or other software (OpenSSH/OpenPGP for key generation process, etc.) have been compromised and therefore the integrity, confidentiality and security of the protocol execution cannot be guaranteed anymore. The same applies for the blockchain and the underlying communication network. The security implications of a compromised communication network for the V&A process have already been outlined in the Authcoin paper [16] and are out of scope of this thesis.

The remaining system that is left is the messaging system. The messaging system refers to the protocol(s) used for exchanging information between different entities, e.g. verifier and target. The specific protocol used for this kind of communication depends on the use case and scenario in which Authcoin is deployed. This asset is a worthwhile target for an attacker since a manipulation of transmitted data records can lead to manipulated outcomes of the V&A process.

We identify three processes as important assets. The send and receive processes as part of the messaging system has already been described earlier. A further important asset is the process of posting information to the blockchain that transfers information produced during the key generation process, the V&A process or the mining process to the blockchain.

### 4.3.2 Identification of Exchanged Data Objects

In addition, we identify the following business assets based on the CPN models of Authcoin. During the key generation process and the V&A process of the protocol specific data objects are generated as well as during the mining process. This includes the following data objects.

**Business assets:**

- EIRs
- CRs
- RRs
- SRs
- VARs

The asset identification tables presented in the following sections identify IS and business assets that support Authcoin processes. For each of the identified assets, a process description is provided. Furthermore, the required security criteria for each asset are listed as well.

#### 4.3.2.1 ChallengeRecord Asset

A CR is generated as part of each V&A process. As described in Section 3.6.3.2, verifier and target create challenges for each other in order to validate and/or authenticate the other party. The challenge itself and further information are stored in a ChallengeRecord and transmitted to the users. In order to ensure a correct validation or authentication, it is important to prevent any manipulation of the transmitted CR, either by an attacker or one of the involved entities. Manipulating the challenge or related information undermines the reliability of the V&A results. Furthermore, CRs are involved in the revocation process as described in Section 3.6.5 and are important for third-party entities that wish to track the results of a V&A process in order to assess its trustworthiness. Therefore, a secure and tamper-proof retrieval and posting of CRs from and to the blockchain is important.

<b>Business assets</b>	ChallengeRecord (CR)
<b>IS assets</b>	Send/Receive
<b>Process description</b>	<p>“Send CR” process description:</p> <ul style="list-style-type: none"> <li>• Verifier creates a CR as part of V&amp;A process.</li> <li>• CR is send to the target via a pre-defined communication channel that depends on the use case.</li> <li>• CR is transmitted.</li> </ul> <p>“Receive CR” process description:</p> <ul style="list-style-type: none"> <li>• User participates in V&amp;A process as a target.</li> <li>• Verifiers transmits created CR to the target.</li> <li>• Target receives the CR and processes it.</li> </ul>
<b>Security criteria</b>	Integrity and availability of the CR

TABLE 4.1: ChallengeRecord asset identification

Table 4.1 outlines the involved business and IS assets as well as the process description and the required security criteria of the described process.

#### 4.3.2.2 ResponseRecord Asset

A RR is generated as part of the V&A process. As described in Section 3.6.3.4, the verifier and the target create responses to the corresponding CRs and transmit the RR to the other party. A RR contains the response itself and related information. In order to ensure a correct validation or authentication, it is important to prevent any manipulation of the transmitted RR, either by an attacker or one of the involved entities. Manipulating the response or related information undermines the reliability of the V&A results. Furthermore, RRs are involved in the revocation process as described in Section 3.6.5 and are important for third-party entities that wish to track the results of a V&A process in order to assess its trustworthiness. Therefore, a secure and tamper-proof retrieval and posting of RRs from and to the blockchain is important.

<b>Business assets</b>	ResponseRecord (RR)
<b>IS assets</b>	Send/Receive
<b>Process description</b>	<p>“Send RR” process description:</p> <ul style="list-style-type: none"> <li>• User creates a RR to a corresponding CR</li> <li>• RR is send to the target via a pre-defined communication channel that depends on the use case</li> <li>• RR is transmitted</li> </ul> <p>“Receive RR” process description:</p> <ul style="list-style-type: none"> <li>• User participates in V&amp;A process.</li> <li>• Other user transmits a RR to the user.</li> <li>• User receives the RR and processes it</li> </ul>
<b>Security criteria</b>	Integrity and availability of the RR

TABLE 4.2: ResponseRecord asset identification

Table 4.2 outlines the involved business and IS assets as well as the process description and the required security criteria of the described process.

#### 4.3.2.3 EntityIdentityRecord Asset

EIRs are created during the key generation process as described in Section 3.6.2 and retrieved for further processing at multiple occasions as part of the V&A process as explained in Section 3.6.3. Furthermore, EIRs are updated during the revocation process presented in Section 3.6.5. They contain all information that link an entity to a certain identity and the corresponding public key. In order to ensure a correct validation or authentication, it is important to prevent any manipulation of the EIR, either by an attacker or one of the involved entities. Manipulating an EIR undermines the reliability of the Authcoin protocol itself and results in unreliable results of the V&A process.

<b>Business assets</b>	EntityIdentityRecord (EIR)
<b>IS assets</b>	Post to the blockchain
<b>Process description</b>	Posting an EIR to the blockchain: <ul style="list-style-type: none"> <li>• User creates a new EIR as part of the "Key generation process" or updates an existing EIR.</li> <li>• EIR is pushed to the blockchain as part of a transaction and mined into a new block.</li> </ul>
<b>Security criteria</b>	Integrity and availability of the EIR

TABLE 4.3: EntityIdentityRecord asset identification

Table 4.3 outlines the involved business and IS assets as well as the process description and the required security criteria of the described process.

#### 4.3.2.4 SignatureRecord Asset

A SR is generated as part of the V&A process. As described in Section 3.6.3.5, the verifier and the target create SRs based on the received responses from the corresponding entities. Depending on the outcome of the V&A process, either a positive or negative SR is created and posted to the blockchain. In order to ensure a correct validation or authentication, it is important to prevent any manipulation of the transmitted SR, either by an attacker or one of the involved entities. Manipulating a SR by e.g. changing a negative outcome into a positive outcome poses a risk to the reliability and security of the protocol.

Furthermore, SRs can be revoked as described in Section 3.6.5 in case the signing entity lost confidence in the established trust relationship. In this scenario, an updated SR is posted to the blockchain. Moreover, SRs are important for third-party entities that wish to track the results of a V&A process in order to assess its trustworthiness.



<b>Business assets</b>	SignatureRecord (SR)
<b>IS assets</b>	Post to the blockchain
<b>Process description</b>	Posting SR to the blockchain: <ul style="list-style-type: none"> <li>• User creates a new SR as part of the "CreateSignature" process or revokes an existing SR.</li> <li>• SR is pushed to the blockchain as part of a transaction and mined into a new block.</li> </ul>
<b>Security criteria</b>	Integrity and availability of the SR

TABLE 4.4: SignatureRecord asset identification

Table 4.4 outlines the involved business and IS assets as well as the process description and the required security criteria of the described process.

#### 4.3.2.5 ValidationAuthenticationRecord Asset

VARs are generated during the mining process of the blockchain as described in Section 3.6.4. Users can select an available VAR if they wish to fulfill the request and retrieve it from the blockchain. Afterwards, the VAR status is updated, then the VAR is further processed and finally finished by adding the V&A results to the VAR. The updated VAR is posted to blockchain at multiple occasions of the described procedure. Preventing any kind of manipulation is necessary to ensure a reliable validation and authentication of entities.

<b>Business assets</b>	ValidationAuthenticationRecord (VAR)
<b>IS assets</b>	Post to the blockchain
<b>Process description</b>	Posting an updated VAR to the blockchain: <ul style="list-style-type: none"> <li>• User decides to fulfill a specific VAR.</li> <li>• User process VAR - e.g. updates status and/or adds results of corresponding V&amp;A process.</li> <li>• User pushes updated VAR to the blockchain as part of a transaction.</li> </ul>
<b>Security criteria</b>	Integrity and availability of the VAR

TABLE 4.5: VAR asset identification

Table 4.5 outlines the involved business and IS assets as well as the process description and the required security criteria of the described process.

In this section, we identified the business as well as IS assets of the Authcoin protocol. We outlined a process description for each asset and defined the required security criteria. The following Section 4.4 provides a risk and threat analysis of the identified assets.

## 4.4 Risk and Threat Analysis of Identified Assets

The goal of this section is to analyze the security risks of the assets identified in the previous Section 4.3. The analyses performed in this section provide an answer to RQ-2.2: Which risks threaten the Authcoin protocol?

For each of the identified assets a risk and threat analysis is performed by considering the following risk components: threat agents, attack methods, threats, vulnerabilities, events, impacts and risks. The resulting analyses are presented in Table 4.6, Table 4.7 and Table 4.8.

### 4.4.1 Risk and Threat Analysis - Posting to Blockchain

The risk and threat analysis for the process of posting data records to the blockchain starts with identifying potential attackers in form of threat agents. As shown in Table 4.6, the threat agent is further described by his/her motivation, resources and expertise. In case of the process of posting information to the blockchain, an outside attacker

who is able to intercept the data records posted to the blockchain by a genuine user, can manipulate the transmitted information and undermine the trustworthiness and reliability of the protocol. The attack method process described in Table 4.6 either takes place directly at the users local home network or at later stages. After an attack, the manipulated data records are no longer useful for a reliable and correct V&A process.

<b>Threat Agent</b>	Outside attacker (MITM)  <u>Motivation:</u> Undermine trustworthiness and reliability of the protocol <u>Resources:</u> Intercept information posted from genuine user to the blockchain <u>Expertise:</u> Intercept and manipulate transmitted data records
<b>Attack Method</b>	<ul style="list-style-type: none"> <li>• Outside attacker intercepts data records (EIR, CR, RR, SR or VAR) that have been created and posted to the blockchain by a genuine user.</li> <li>• Outside attacker manipulates data records.</li> <li>• Outside attacker forwards manipulated data records to intended receiver (blockchain miners).</li> </ul>
<b>Threat</b>	Outside attacker manipulates data records.
<b>Vulnerability</b>	Data records transmitted during the process of posting information to the blockchain can be manipulated.
<b>Event</b>	Outside attacker manipulates transmitted data records and forwards the false records to be posted to the blockchain due to a lack of integrity checks of transmitted data records.
<b>Impact</b>	<ul style="list-style-type: none"> <li>• Data records with false information available on the blockchain.</li> <li>• Loss of integrity of transmitted data records.</li> </ul>
<b>Risk</b>	Outside attacker manipulates data records that should be posted to the blockchain which leads to loss of data integrity and false information available on the blockchain.

TABLE 4.6: Risk and threat analysis for posting EIRs, CRs, RRs, SRs and updated VARs to the blockchain.

#### 4.4.2 Risk and Threat Analysis - Send/Receive

The risk and threat analysis for the process of sending and receiving data records during the V&A process is similar to the risk and threat analysis of posting data records to the blockchain as described in the previous Section 4.4.1. The threat agent, the attack methods, threat, vulnerability, event, impact and risk are the same but focus on a different communication channel. Instead of posting information to the blockchain, two genuine users exchange V&A process related information via a side channel which's type depends on the use case. Compromising the data record exchange over this side channel results in the same risks as for the process of posting information to the blockchain.

<b>Threat Agent</b>	<p>Outside attacker (MITM)</p> <p><u>Motivation:</u> Undermine trustworthiness and reliability of the protocol</p> <p><u>Resources:</u> Intercept user traffic</p> <p><u>Expertise:</u> Intercept and manipulate transmitted data records</p>
<b>Attack Method</b>	<ul style="list-style-type: none"> <li>• Outside attacker intercepts transmitted (send/receive) CRs or RRs created during the V&amp;A process.</li> <li>• Outside attacker manipulates CR or RR content.</li> <li>• Outside attacker forwards manipulated CR/RR to intended receiver.</li> </ul>
<b>Threat</b>	Outside attacker manipulates data records.
<b>Vulnerability</b>	Data records transmitted during the process of sending and receiving can be manipulated.
<b>Event</b>	Outside attacker manipulates transmitted data records and forwards the false records.
<b>Impact</b>	<ul style="list-style-type: none"> <li>• Data records with false information transmitted to users.</li> <li>• Loss of integrity of transmitted data records.</li> </ul>
<b>Risk</b>	Outside attacker manipulates transmitted data records which leads to loss of data integrity and false information transmitted to the users.

TABLE 4.7: Risk and threat analysis for sending and receiving CRs and RRs.

#### 4.4.3 Risk and Threat Analysis - DDoS

In this attack scenario an outside attacker aims to disrupt the services provided by Authcoin using a DDoS (Distributed Denial of Service) attack. Such an attack focuses either on a specific local user who is targeted by the attacker or on global parts of the system that effect all users, such as the underlying blockchain that is utilized by Authcoin. As a result of such an attack, it is no longer possible to proceed with pending V&As that might time out depending on how long the attack continues. Furthermore, it is not possible to post any new information to blockchain, either for the locally targeted user or all users. For some users who downloaded the whole blockchain to their local devices, it is still possible to perform lookups of information for past events. Users who only run light clients cannot perform any lookups during the attack.

<b>Threat Agent</b>	Outside attacker  <u>Motivation:</u> Disrupt Authcoin services <u>Resources:</u> DDoS network with sufficient power <u>Expertise:</u> Running DDoS attacks
<b>Attack Method</b>	<ul style="list-style-type: none"> <li>• Outside attacker performs a DDoS attack on network infrastructures relevant for user communication - either for a specific local user or on a global scale.</li> <li>• Users are no longer able to exchange data records (CRs, RRs) or access Authcoin information on the blockchain.</li> </ul>
<b>Threat</b>	Outside attacker performs a DDoS attack on the network infrastructure.
<b>Vulnerability</b>	Network infrastructure can be overloaded by an outside attacker.
<b>Event</b>	Outside attacker is able to perform a DDoS attack on the network infrastructure.

<b>Impact</b>	<ul style="list-style-type: none"> <li>• Pending V&amp;As might time out.</li> <li>• Users are not able to perform any new or pending V&amp;A procedures.</li> <li>• No information lookup on the global blockchain.</li> <li>• General unavailability of Authcoin services.</li> </ul>
<b>Risk</b>	Outside attacker performs a DDoS attack on the local or global network infrastructure used by Authcoin's users resulting in a general unavailability of the service. Furthermore pending V&As might time out and users cannot execute any operation of the protocol anymore.

TABLE 4.8: Risk and threat analysis of a DDoS attack.

We performed a threat and risk analysis for the assets identified in Section 4.3 based on the following risk components: threat agents, attack methods, threats, vulnerabilities, events, impacts and risks. The analyses are presented in Table 4.6, Table 4.7 and Table 4.8 resulting in the identification of three risks. First, the risk of a manipulation of data records of the process of posting information to the blockchain. Second, the risk of a manipulation of data records during the process of sending/receiving information during the V&A process. Third, the risk of a DDoS attack that disrupts the services provided by the Authcoin protocol.

## 4.5 Risk Treatment

The goal of this section is to apply security controls that manage the security risks identified in the previous Section 4.4 and subsequently answering RQ-2.3: What are the necessary security requirements and security controls?

Each of the identified security risks is managed by applying security requirements and controls. The security requirements and controls mitigate the identified risks either by eliminating them or reducing the resulting effects. The security requirements and controls for the risks are presented in Table 4.9 and Table 4.10.

### 4.5.1 Risk Treatment - Posting to Blockchain and Send/Receive

Risk 1 enables an outside attacker to manipulate data records that should be posted to the blockchain which subsequently leads to a loss of integrity of the transmitted data records and false information available on the blockchain. As proposed in Table 4.9, the security requirement necessary to mitigate the risk is to implement integrity checks in form of signed hash sums to avoid undetected manipulation of data records posted to the blockchain.

<b>Risk treatment</b>	Risk reduction
Security requirement	Integrity checks of submitted records
Controls	Signed hashes

TABLE 4.9: Treatment of risk 1: “Posting to blockchain” and risk 2: “Send/Receive”

Risk 2 is closely related to risk 1 and enables an outside attacker to manipulate transmitted data records that are exchanged during the V&A process between involved entities. The manipulation of the records leads to a loss of integrity of transmitted data records and false information being transmitted to the users. Risk 2 is mitigated in the same way as risk 1 by implementing integrity checks in form of signed hash sums to avoid undetected manipulation of exchanged data records as illustrated in Table 4.9.

### 4.5.2 Risk Treatment - DDoS

Risk 3 enables an outside attacker to performs a DDoS attack on the local or global network infrastructure used by Authcoin’s users resulting in a general unavailability of the service or disruption of pending protocol executions. As presented in Table 4.10, this risk is mitigated by service decentralization, load distribution and load balancing.

<b>Risk treatment</b>	Risk reduction
Security requirement	Mitigate service disruption
Controls	Decentralization, load distribution and balancing

TABLE 4.10: Treatment of risk 3: DDoS

## 4.6 Discussion

The main objection of Chapter 4 is to perform a risk and threat analysis of the Authcoin protocol based on the ISSRM domain model. We identified three risks and determined corresponding security requirements and security controls to mitigate the risks. Nevertheless, it does not guarantee that there are no additional undetected risks and security flaws in the protocol. Application of additional risk analysis methods and professional penetration testing might uncover further risks and security flaws. Due to time limitations, it was not possible to perform these steps as part of this thesis. Moreover, the results of the risk analysis have been performed by one of the authors of the Authcoin protocol which might result in a biased perception of the protocol's security.

Finally, it is also important to keep in mind that we limited the scope of this risk analysis and excluded certain aspects. Some of the identified assets, e.g. blockchain, user devices and the underlying communication network pose security risks that cannot be mitigated in context of the Authcoin protocol. For example, it is possible that a user's device, operation system or other software have been compromised and therefore the integrity, confidentiality and security of the protocol execution cannot be guaranteed anymore. The same applies for the blockchain and the underlying communication network. The security implications of a compromised communication network for the V&A process have already been outlined in the Authcoin paper [16].

## 4.7 Conclusion

In Chapter 4 we perform a risk and threat analysis of the Authcoin protocol based on the ISSRM domain model. We determine the assets of the protocol and identify potential threat agents, attack methods, threats, vulnerabilities, events, impacts and risks for each of the assets. Afterwards, security requirements and security controls necessary to mitigate the identified risks are provided.

As the first step of the risk and threat analysis, we identify the processes of sending and receiving data records as well as the process of posting data records to the blockchain as assets of Authcoin. Furthermore, EIRs, CRs, RRs, SRs and VARs are identified as relevant data objects exchanged during the protocol execution. For each of the identified assets, a risk and threat analysis is performed by considering the following risk components: threat agents, attack methods, threats, vulnerabilities, events, impacts and risks. As a result, we identify three risks that threaten the protocol. First, the risk of data record manipulation during the process of posting information to the blockchain. Second, the risk of data record manipulation during the process of exchanging V&A related information during the protocol execution. Third, the risk of a local or global



DDoS attack on Authcoin's users or infrastructure. Afterwards, security requirements and security controls necessary to mitigate the three identified risks are provided. In addition, we acknowledge and discuss the limitations of the performed risk and threat analysis resulting from the used ISSRM domain model, time limitations or biases of the author of this thesis with regards to the protocol analysis.

For future work, we plan to incorporate feedback of external risk analysts as well as penetration testers.

## Chapter 5

# Application of Security Risk-oriented Patterns on the Authcoin Protocol

*The following chapter deals with the application of existing security risk-oriented pattern on the Authcoin protocol. The existing patterns are reviewed and appropriated patterns are selected for implementation in order to mitigate the security risks identified in Chapter 4. To do so, the occurrences of the selected pattern are identified, followed by an implementation and integration of the pattern in the CPN models of Authcoin. The updated CPN models are presented and described. Finally, information and issues of the novel process of applying security risk-oriented patterns to CPN models are provided.*

### 5.1 Introduction

The objection of Chapter 5 is to answer the research question RQ-3 - How to apply security risk-oriented patterns to the Authcoin protocol? - as outlined earlier in Chapter 1. In order to answer RQ-3 in a more structured and comprehensive way, it is divided into three subquestions:

- **RQ-3.1:** Which existing security risk-oriented patterns are applicable?
- **RQ-3.2:** What modification of the existing CPN models is required to implement the chosen security risk-oriented patterns?
- **RQ-3.3:** Which problems arise in the process of applying the existing security risk-oriented pattern?

Each subquestion is answered independently in a separate section. The first part of this chapter (Section 5.2) focuses on the identification of applicable existing security risk-oriented patterns and how to identify the occurrences of selected patterns, thereby answering RQ-3.1. The subsequent Section 5.3 deals with RQ-3.2 and the implementation and integration of the chosen security risk-oriented patterns in the existing CPN models of Authcoin. Afterwards, Section 5.4 answers RQ-3.3 by discussing characteristics of the process of applying security risk-oriented patterns to CPN models. Finally, the application of security risk-oriented patterns is discussed in Section 5.5 followed by the conclusion in Section 5.6.

## 5.2 Application of Existing Security Risk-oriented Patterns

The goal of this section is to review the existing security risk-oriented patterns and identify those which are suitable to mitigate risks identified during the risk and threat analysis of the Authcoin protocol in Chapter 4. Based on these findings, this section provides an answer to RQ-3.1: Which existing security risk-oriented patterns are applicable?

First in Section 5.2.1, the existing security risk-oriented patterns are reviewed and discussed with regards to their applicability to the CPN models in order to mitigate risks. Afterwards, in Section 5.2.2 we identify the occurrences of the chosen pattern in the CPN model of Authcoin before implementing the pattern in Section 5.3.

### 5.2.1 Identification of Applicable Security Risk-oriented Patterns

In Section 2.2.3, we presented the five security risk-oriented patterns (SRPs) introduced by Ahmed et al. [18]. In the following, we review each of the five patterns and discuss their applicability to the CPN models with regards to the identified risk that threaten the protocol from Chapter 4.

#### 5.2.1.1 SRP 1

*SRP 1 secures the data transmission between business entities with focus on preventing the loss of data, confidentiality and its integrity. The pattern proposes to make the data unreadable before transmitting, calculate checksum values and utilize transmission mediums that cannot be intercepted.*

SRP 1 is suitable to mitigate issues posed by risk 1 and risk 2 as identified in Chapter 4. Making the data unreadable before the transmission is not relevant in case of Authcoin since all data is available on a public blockchain anyway. However, ensuring the integrity by checksum calculation prevents an outside attacker, as described for risk 1 and risk 2 from undetected manipulation of transmitted data records. Therefore, SRP 1 is implemented in form of hashing data records before the transmission and securing the hash with the signature of the sender based on the senders key pair. The receiver can first verify the signature and afterwards verify the provided hash.

#### 5.2.1.2 SRP 2

*SRP 2 prevents attackers from injecting malicious data into business processes and proposes to define data structures and formats for incoming data.*

SRP 2 does not correspond to any of the identified risks from Chapter 4. In addition, the pattern has already been implemented in the provided CPN models of the Authcoin protocol by defining the protocol semantics. The protocol semantics are defined in Section 3.5 in form of token colors that represent the recommended data structures of SRP 2. As a result, it is not possible to inject malicious data even though it is still possible to inject false information into the data records.

#### 5.2.1.3 SRP 3

*SRP 3 ensures the availability of the business services by mitigating denial of service attacks by “filtering and classifying of incoming requests, detecting abnormal requests, and discarding the attacking ones” [18].*

Ahmed et al. [18] propose SRP 3 to ensure the availability of business services in case of DDoS attacks. The proposed mitigation strategies might be applicable for a vast variety of business services and systems that are controlled by an authority but cannot be applied to Authcoin. The mitigation strategies do not prevent DDoS attacks on a single local user, since neither filtering nor classifying or similar strategies defend a single user or a small group of user against nowadays DDoS attacks [69]. In case of DDoS attacks on the miners of a blockchain the same reasoning applies in case that only a small group of miners exist, e.g. for smaller blockchain projects. If that is not the case and a large group of miners participates in the mining of new blocks, such as for Bitcoin, then a targeted attack on all users is unlikely to be successful since an eliminated miner or a group of miners, e.g. a whole mining pool, will be compensated by the remaining miners. In other words, the decentralized nature of blockchain architectures already constitutes

a DDoS mitigation strategy given a sufficient number of participating miners. As a result, it is not possible to implement SRP 3 in context of Authcoin. Instead, it is recommended to either utilize an existing blockchain system with a sufficient number of miners or ensure an own system that attracts enough miners themselves. As argued in [16], both options are possible in context of Authcoin.

#### **5.2.1.4 SRP 4**

*SRP 4 proposes the application of multilevel access rights to the retrieval interface in order to prevent unauthorized access to data and missing system logs on the access history to specific data.*

SRP 4 does not correspond to any of the identified risks and is therefore not considered to be implemented.

#### **5.2.1.5 SRP 5**

*SRP 5 ensures the confidentiality of data and proposes to store data in an encrypted and secure manner.*

SRP 5 does not correspond to any of the identified risks from Chapter 4. In addition, Authcoin is designed to be transparent and therefore almost all data are stored in a transparent manner on a publicly available blockchain. The only exception is the private key of a user that is stored on the user's device(s).

Based on the review and discussion of the existing security risk-oriented patterns, we decided to implement SRP 1 to mitigate risk 1 and risk 2 in order to prevent manipulation of data records during the process of posting them to the blockchain or during the process of exchanging data records between users.

### **5.2.2 Identify Occurrences of Security Risk-oriented Patterns**

Before implementing the chosen security risk-oriented pattern SRP 1, it is important to identify the occurrences of the pattern in the CPN models. As stated by Ahmed et al., this “is a manual activity, that potentially requires a good comprehension of the modelled domain and problem” [17]. In case of the CPN models of Authcoin, it is necessary to identify all occurrences of the process of posting data records to the blockchain as well as the occurrences of the process of exchanging data records between users. This task is performed based on the CPN models provided in Chapter 3. As a

result of this process, we identified occurrences of SRP 1 in the following modules of the CPN models:

- “EstablishBinding” module
- “FormalValidation” module
- “CreateSendChallengeToVerifier” module
- “CreateSendResponses” module
- “CreateChallengeForTarget” module
- “CreateChallengeForVerifier” module
- “CreateResponse” module
- “CreateSignaturesFromRR” module
- “VARCreation” module
- “ProcessVAR” module
- “FinishVAR” module
- “EIRRevocation” module
- “SignatureRevocation” module

It is important to keep in mind that the occurrences of a pattern in a specific module or sub-module also require changes in depending modules that have to be updated as well.

### 5.3 Implementation of Chosen Security Risk-oriented Pattern

The following section presents the implementation of SRP 1 in context of the CPN models of Authcoin. The same AOM methodology as in Chapter 3 is used to implement the pattern and thereby answering RQ-3.2: What modification of the existing CPN models is required to implement the chosen security risk-oriented pattern?

First, in Section 5.3.1 and Section 5.3.2, the updated goal model and the updated behavioral interface model are presented, followed by the updated protocol semantics in Section 5.3.3. Finally, Section 5.3.4 presents updated CPN models resulting from the implementation and integration of SRP 1 in CPN.

### 5.3.1 Updated Goal Model

After identifying the occurrences of SRP 1 in Section 5.2.2, the next step is to update Authcoin’s goal model from Chapter 3 accordingly. Since not all goals and sub-goals of the existing goal model are influenced by the implementation of SRP 1, we only present the parts of the goal model that have been adapted. As illustrated in Figure 5.1, four integrity-check-related sub-goals have been added to the updated “V&A Processing” sub-goal. “IntegrityCheck” is a new sub-goal of “Formal validation”, “CheckIntegrity” a new sub-goal of “CreateChallengeForVerifier”, “CheckIntegrity” a new sub-goal of “CreateSendChallengeToVerifier” and “CheckIntegrity” a new sub-goal of “CreateSendResponse”. The rest of the original goal model remains untouched.

### 5.3.2 Updated Behavior Interface Model

The behavioral interface model introduced in Section 3.2.2 and displayed in Appendix B is updated as well. The following Tables 5.1 - 5.8 present the resulting updates based on the corresponding behavioral interface model of Appendix B. In order to avoid redundancy, only new or updated entries are listed in the following tables.

Activity	Trigger	Precondition	Postcondition
Check key well formed	Started formal validation	VAE-ID, EIR	Key well formed result, public key, revocation information
Integrity check	Key is well formed	VAE-ID, EIR	VAE-ID, result of integrity check based on hash sum

TABLE 5.1: Updated behavioral interfaces of activities for the subgoal “Formal validation” - based on Table B.3

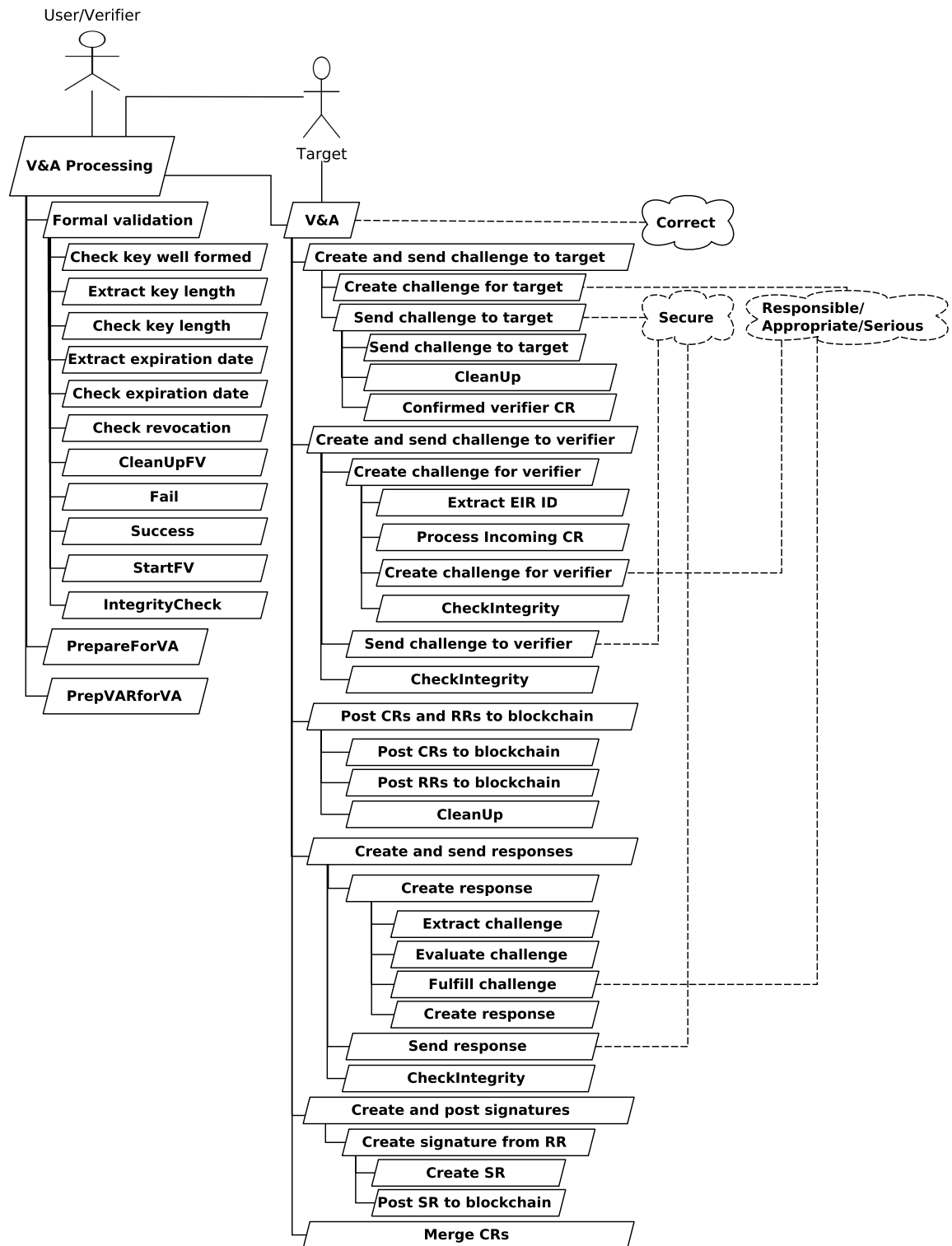


FIGURE 5.1: Updated goal model for sub-goal "V&A Processing"



<b>Activity</b>	<b>Trigger</b>	<b>Precondition</b>	<b>Postcondition</b>
Create and send challenge to verifier	Received challenge from verifier	VAE-ID, verifier EIR, target EIR, incoming CR	Verifier and target CRs as well as EIRs or failure message
Create and send responses	Received incoming CRs	VAE-ID, verifier and target CR as well as EIRs	VAE-ID, target and verifier RRs as well as EIRs or failure message
Create and post signatures	Received incoming RRs, CRs and RRs posted to blockchain	VAE-ID, verifier and target RRs, verifier and target EIRs	SRs available on blockchain, validation and authentication finished

TABLE 5.2: Updated behavioral interfaces of activities for the subgoal “Validation and authentication” - based on Table B.4

<b>Activity</b>	<b>Trigger</b>	<b>Precondition</b>	<b>Postcondition</b>
Create challenge for verifier	Target received challenge from verifier	VAE-ID, target CR, verifier and target EIR	VAE-ID, verifier CR, target challenge, verifier and target EIR
Check integrity	Original verifier receives CR from target	VAE-ID, verifier CR, target CR, verifier and target EIR	VAE-ID, verifier CR, target CR, verifier and target EIR or failure message

TABLE 5.3: Updated behavioral interfaces of activities for the subgoal “Create and send challenge to verifier” - based on Table B.8

Activity	Trigger	Precondition	Postcondition
Check integrity	Received CR from verifier	VAE-ID, CR for target, verifier EIR	VAE-ID, target CR, verifier EIR
Extract EIR-ID	CR passed integrity check	VAE-ID, CR for target, verifier EIR	VAE-ID, target CR, verifier EIR-ID, verifier EIR
Process incoming CR	Extracted verifier EIR-ID	VAE-ID, verifier EIR, verifier EIR-ID	VAE-ID, Verifier EIR
Create challenge for verifier	Processed incoming CR	VAE-ID, verifier and target EIRs, challenge for verifier	VAE-ID, CR for verifier, verifier and target EIRs

TABLE 5.4: Updated behavioral interfaces of activities for the subgoal “Create challenge for verifier” - based on Table B.9

Activity	Trigger	Precondition	Postcondition
Create response	Verifier and target CRs received	VAE-ID, verifier and target CRs as well as EIRs	VAE-ID and RR and EIRs or failure message
Send response	Created response for CR	VAE-ID, RR	VAE-ID,RR
Check integrity	User receives a RR	VAE-ID, RR, verifier and target EIR	VAE-ID,RR, verifier and target EIR or failure message

TABLE 5.5: Updated behavioral interfaces of activities for the subgoal “Create and send response” - based on Table B.10

Activity	Trigger	Precondition	Postcondition
Evaluate challenge	Extracted challenge from CR	VAE-ID, CR, challenge, challenge evaluation, verifier and target EIRs	VAE-ID, CR and evaluation result, verifier and target EIRs or failure message
Create response	User fulfilled challenge	VAE-ID, CR, fulfilled challenge, verifier and target EIRs	VAE-ID, RR, verifier and target EIRs

TABLE 5.6: Updated behavioral interfaces of activities for the subgoal “Create response” - based on Table B.11

Activity	Trigger	Precondition	Postcondition
Create signature from RR	Received RRs	VAE-ID, RRs, verifier and target EIRs	SRs available on blockchain

TABLE 5.7: Updated behavioral interfaces of activities for the subgoal “Create and post signature” - based on Table B.13

Activity	Trigger	Precondition	Postcondition
Create Signature	Received a RR	VAE-ID, RR, signature lifespan, response evaluation, verifier and target EIRs	VAE-ID, SR

TABLE 5.8: Updated behavioral interfaces of activities for the subgoal “Create signatures from RR” - based on Table B.14

### 5.3.3 Updated Protocol Semantics

Section 3.3 introduced the protocol semantics of the Authcoin CPN models. As a result of implementing SRP 1 and integrating the pattern into the existing CPN models, the semantics were updated. Table 5.9 presents the list of updated token colors. Attribute fields for hashes of the data records and signatures from the creator of the data records are now part of all EIRs, CRs, RRs, SRs and VARs. The first signature of a VAR is produced by the miner who mined the new block containing the VAR. Moreover, ResponseRecords received additional attribute fields that specify the receiver and sender of the RR based on the EIR-ID.

Token color	Description	Type
EntityIdentityRecord	Contains all relevant information about an entity	(EIR_ID, Timestamp, PublicKey, Identifiers, Revoked, hashEIR, EIRsig)
ChallengeRecord	Contains all information about a V&A challenge	(CR_ID, VAE_ID, Timestamp, ChallengeType, Challenge, VerifierEIR_ID, VerificationTargetEIR_ID, hashCR, CRsig)
ResponseRecord	Contains all information regarding a V&A response	(RR_ID, VAE_ID, Timestamp, CorrespondCR_ID, Response, hashRR, RRs sig, RRreceiver, RRSender)

SignatureRecord	Contains all information regarding a V&A signature	(SR_ID, VAE_ID, Timestamp, ResponseRR_ID, ExpirationDate, Revoked, SuccessfulVA, hashSR, SRsig)
VAR	Validation and authentication request	(VAR_ID, CreationDate, LastUpdated, VerifierEIR_ID, TargetEIR_ID, Status, VAE_ID, hashVAR, VARsig)

TABLE 5.9: Updated acronyms, names and description of token colors.

Furthermore a general hash function for calculating hashes of strings as well as specific hash functions for calculating the hash of EIRs, CRs, RRs, SRs and VARs is introduced. Listing 5.1 presents a simple hash function for mapping a string to an integer. In order to reduce the overhead of implementing a full grown hash function, a simplified demonstration example is used based on basic modulo arithmetic. When implementing Authcoin, a stronger hash function has to be chosen, preferably not SHA-1 [70] or similar weak or even weaker hash functions.

---

(\*Based on: <http://homepages.inf.ed.ac.uk/mfourman/teaching/mlCourse/notes/sml-arrays.html>\*)

```

val hashtablesize = 98764327 (* a prime*)
fun combine [] = 0
| combine (h :: t) = (ord h + 7 * combine t) mod hashtablesize
fun hash s = combine (explode s) ;

```

---

LISTING 5.1: SML code for the general hash function.

In addition to the general hash function, we implement specific hash functions for each of our data records that utilize the general hash function from Listing 5.1. Listing 5.2 illustrates the EIR hash function, Listing 5.3 the CR hash function and Listing 5.4 the RR hash function. Specific hash functions for SRs and VARs are not required and the corresponding hashes are calculate as part of arc inscriptions.

---

```

fun hashEIR(EIR : EntityIdentityRecord) =hash (
Int.toString(#EIR_ID EIR) ^
IntInf.toString(#Timestamp EIR) ^
#KeyFingerprint (#PublicKey EIR) ^
#Key (#PublicKey EIR) ^
Bool.toString(#Revoked EIR));

```

---

LISTING 5.2: SML code for the EIR hash function.

---

```

fun hashCR(CR : ChallengeRecord) =hash (
Int.toString(#CR_ID CR) ^
Int.toString(#VAE_ID CR) ^
IntInf.toString(#Timestamp CR) ^
#ChallengeType CR ^
#Challenge CR ^
Int.toString(#VerifierEIR_ID CR) ^
Int.toString(#VerificationTargetEIR_ID CR));

```

---

LISTING 5.3: SML code for the CR hash function.

---

```

fun hashRR(RR : ResponseRecord) =hash (
Int.toString(#RR_ID RR) ^
Int.toString(#VAE_ID RR) ^
IntInf.toString(#Timestamp RR) ^
Int.toString(#CorrespondCR_ID RR) ^
(#Response RR) ^
Int.toString(#RRreceiver RR) ^
Int.toString(#RRSender RR));

```

---

LISTING 5.4: SML code for the RR hash function.

### 5.3.4 Updated CPN Models

After updating the goal model, the behavioral interface model and the protocol semantics, the last step is to update the corresponding CPN models of Authcoin. The following Figures 5.2 - 5.16 illustrate the updated CPN models based on the original models from Chapter 3. In the following, only the modified CPN models are listed, all remaining modules and sub-modules remain the same as presented in Section 3.6. The source file of the presented CPN models is available in Appendix C.

#### 5.3.4.1 Updated Sub-Modules of “KeyGenerationEstablishBinding”

The “KeyGenerationEstablishBinding” module in Figure 5.2 now contains a hash calculation for the created EIR whereas the rest of the module remains untouched.

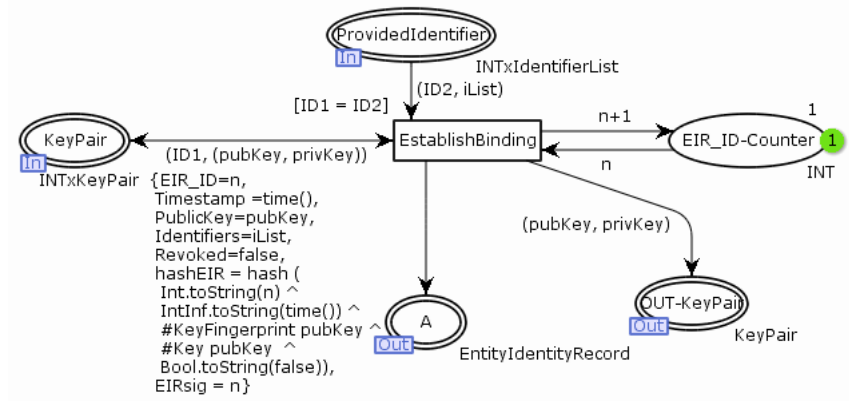


FIGURE 5.2: Updated CPN model of the “EstablishBinding” module

#### 5.3.4.2 Updated Sub-Modules of “V&A-Processing”

The results of integrating an integrity check into the “FormalValidation” module is illustrated in Figure 5.3. As a consequence, the number of required tokens for the “Success” and “Fail” transitions have been updated. Furthermore, minor changes in the “V&A” module result in a new place “VT\_EIRs” in order to accommodate EIRs required for creating and verifying the created signatures on data records.

#### 5.3.4.3 Updated Sub-Modules of “V&A”

Figures 5.5, 5.6 and 5.7 present changes of the “CreateSendChallengeToVerifier” module, the “CreateSendResponses” module and the “CreateSignatures” module resulting from implementing an integrity check for incoming challenges in form of CRs and incoming responses in form of RRs. If an integrity check fails, caused by a mismatching checksum or a wrong signature, the V&A process is aborted and a failure message generated.

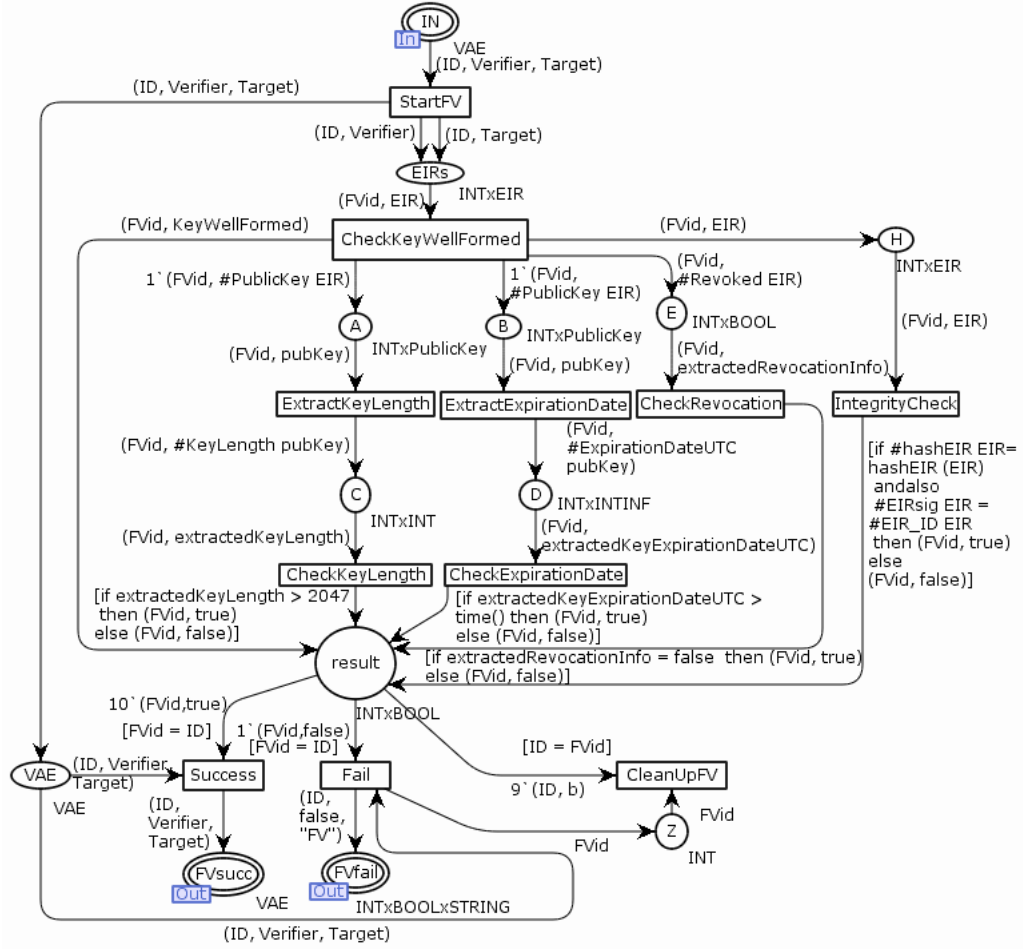


FIGURE 5.3: Updated CPN model of the “FormalValidation” module

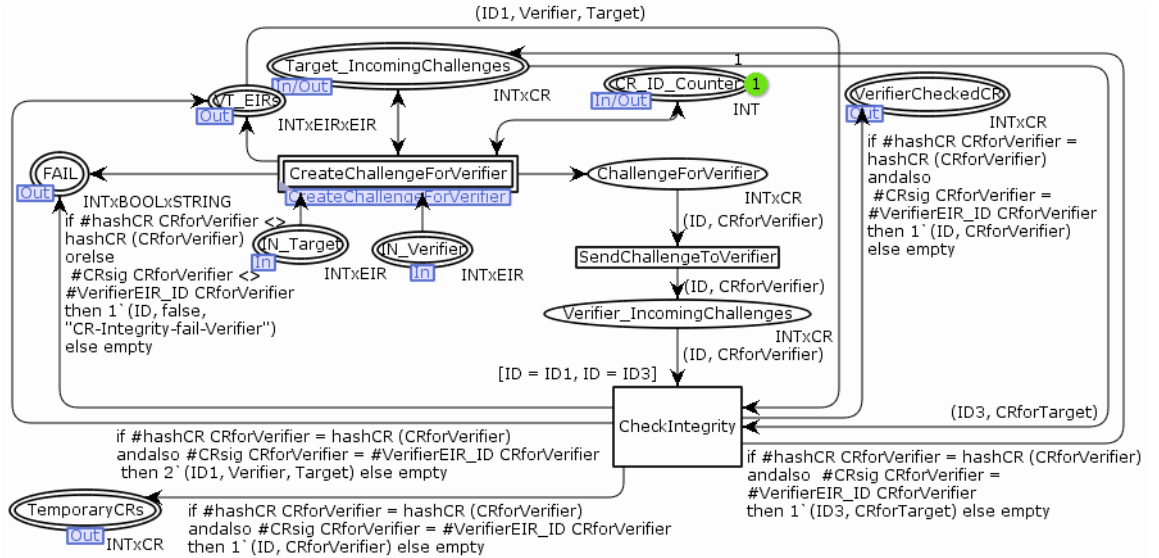


FIGURE 5.5: Updated CPN model of the “CreateSendChallengeToVerifier” module

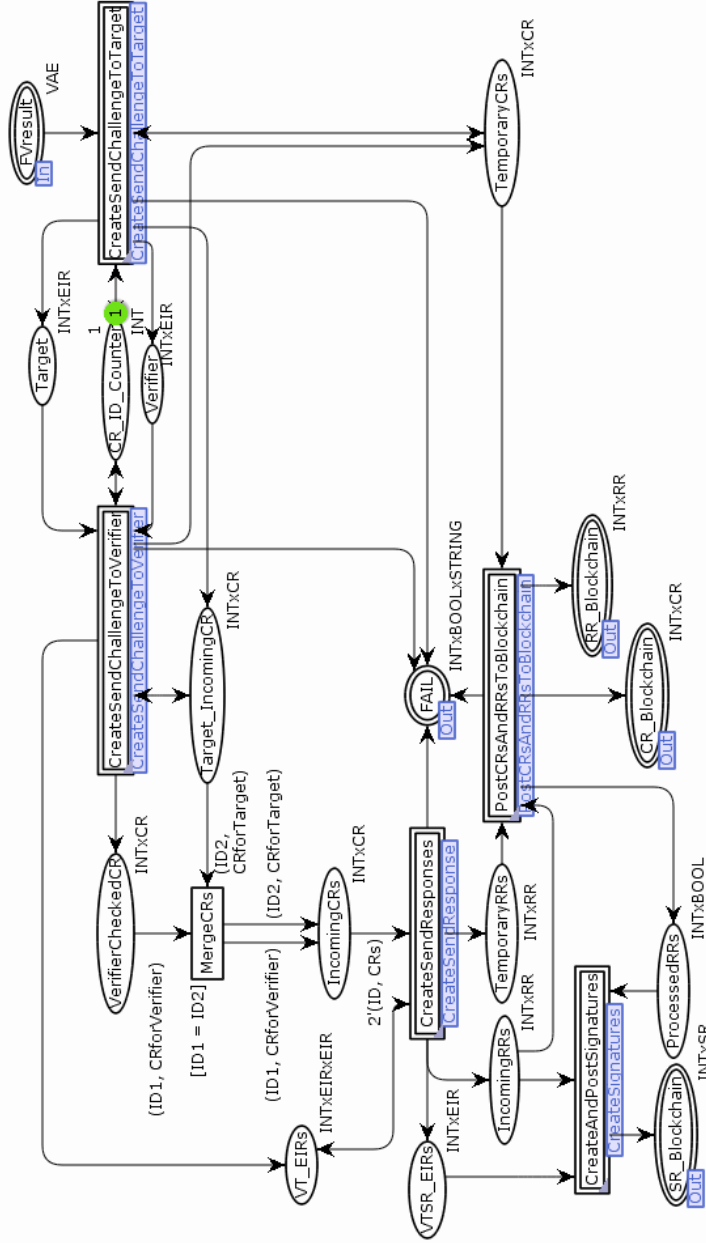


FIGURE 5.4: Updated CPN model of the “V&amp;A” module

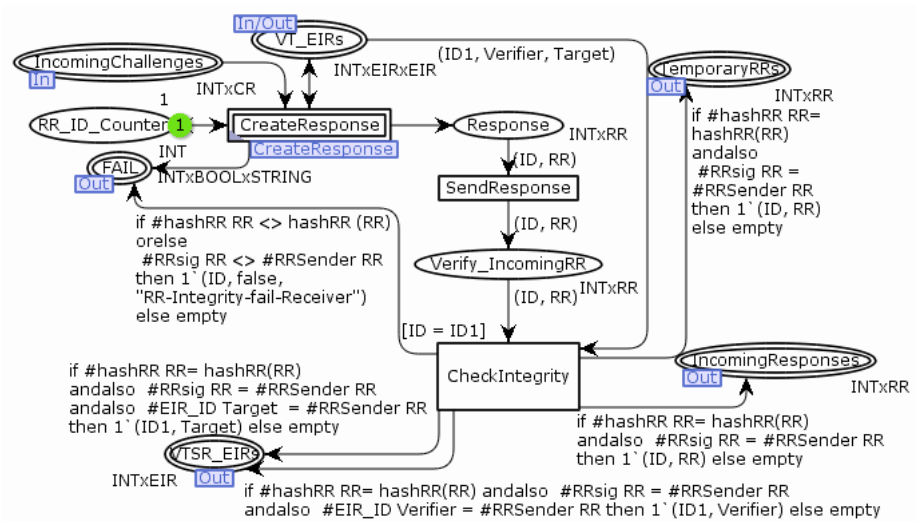


FIGURE 5.6: Updated CPN model of the “CreateSendResponses” module



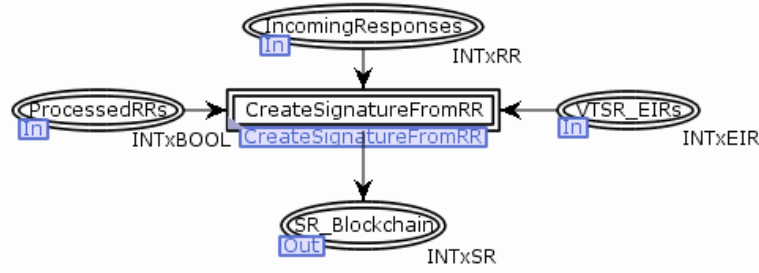


FIGURE 5.7: Updated CPN model of the “CreateSignatures” module

#### 5.3.4.4 Updated Sub-Modules of “CreateSendChallengeToTarget”

The “CreateChallengeForTarget” module in Figure 5.8 is modified in order to calculate a hash for the created CR whereas the rest of the module remains untouched.

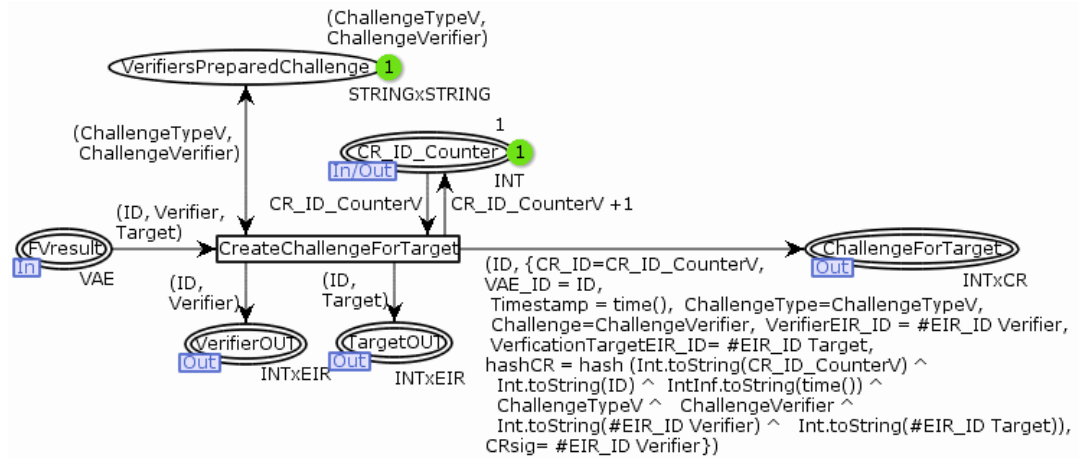


FIGURE 5.8: Updated CPN model of the “CreateChallengeForTarget” module

#### 5.3.4.5 Updated Sub-Modules of “CreateSendChallengeToVerifier”

Similar to the “CreateSendChallengeToVerifier” module in Figure 5.5, an integrity check for incoming challenges in form of CRs is integrated into the “CreateChallengeForVerifier” module illustrated in Figure 5.9.



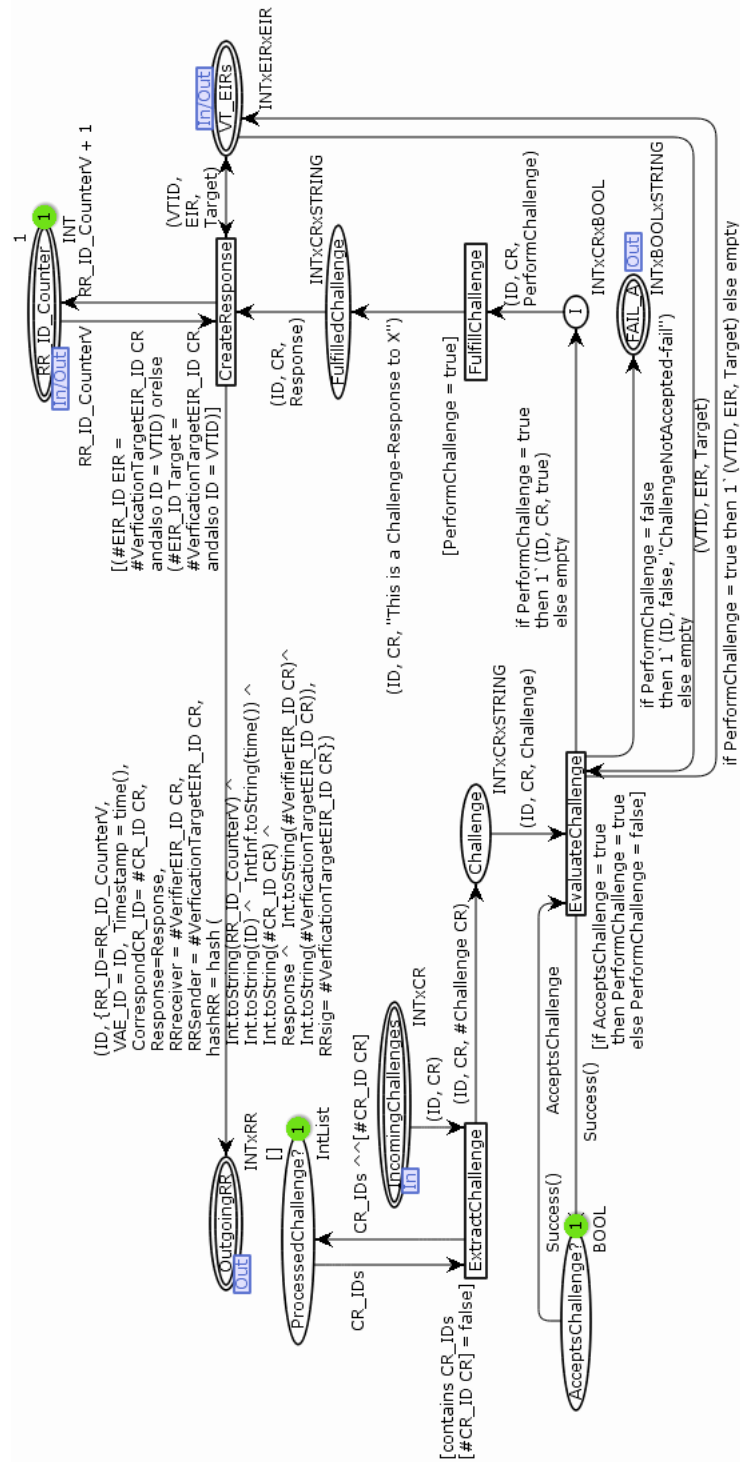


FIGURE 5.10: Updated CPN model of the “CreateResponse” module

#### 5.3.4.7 Updated Sub-Modules of “CreateAndPostSignatures”

Similar to the “CreateResponse” module above, the “CreateSignaturesFromRR” module now calculates a hash-based checksum for each created SR as illustrated in Figure 5.11.

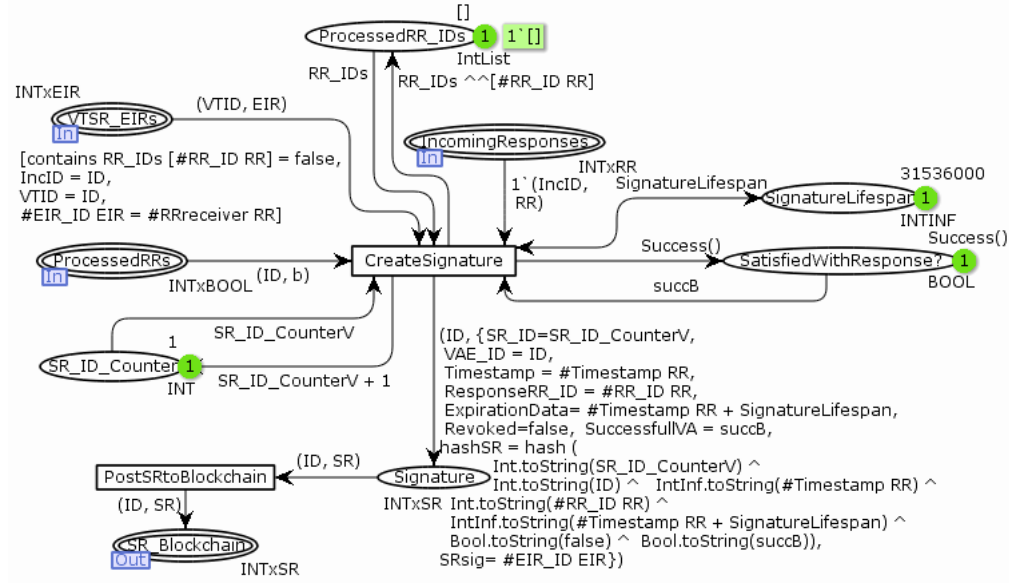


FIGURE 5.11: Updated CPN model of the "CreateSignaturesFromRR" module

#### 5.3.4.8 Updated Sub-Modules of "Mining"

The "VARCreation" module in Figure 5.12, the "ProcessVAR" module in Figure 5.13 and the "FinishVAR" module in Figure 5.14 all calculate hashes for the created and modified VARs whereas the rest of the modules remain untouched.



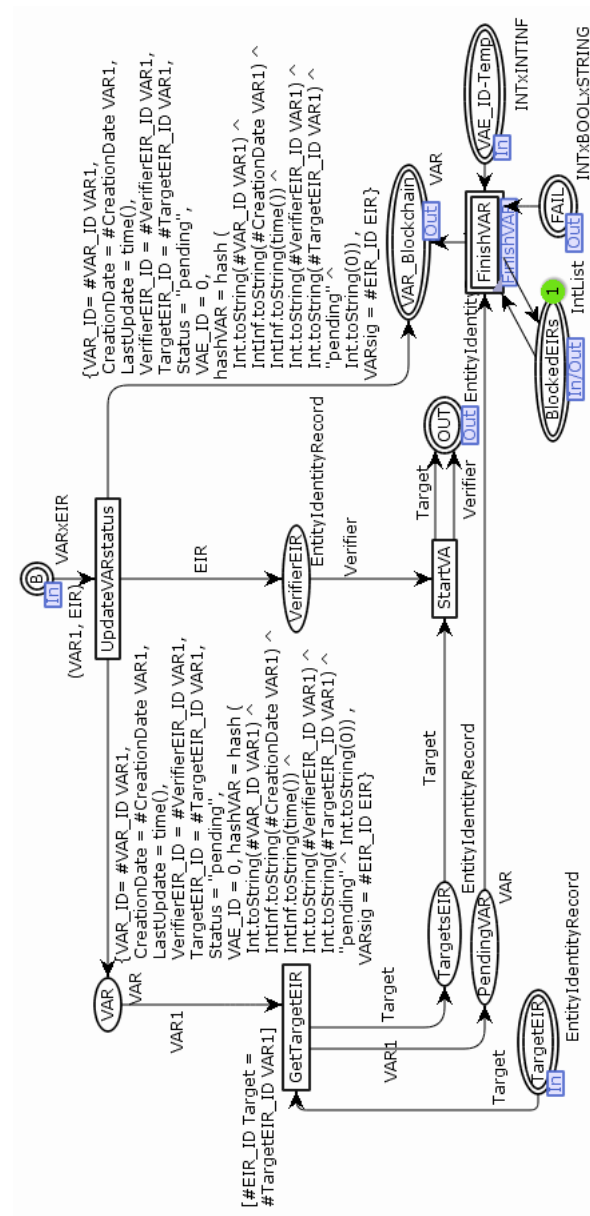


FIGURE 5.13: Updated CPN model of the “ProcessVAR” module

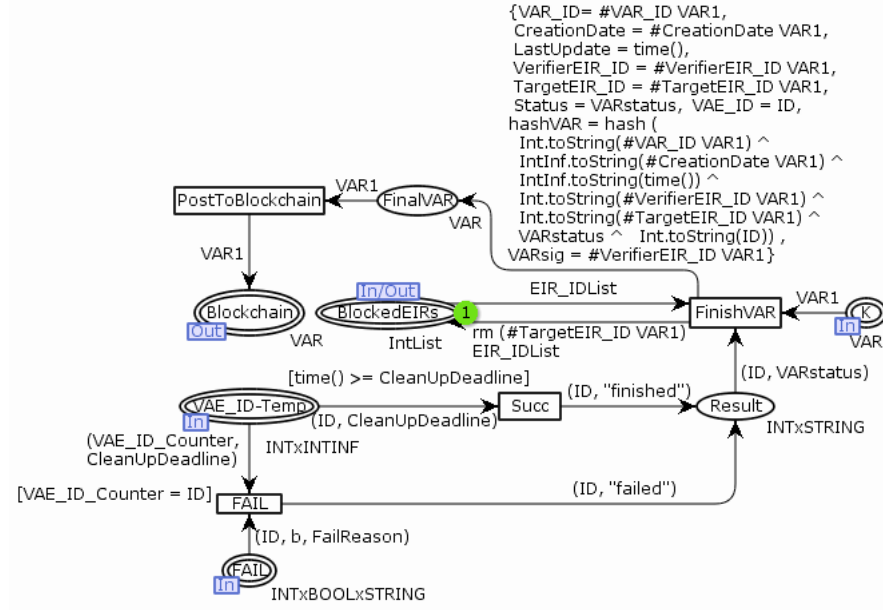


FIGURE 5.14: Updated CPN model of the "FinishVAR" module

#### 5.3.4.9 Updated Sub-Modules of "Revocations"

Finally, the "EIRrevocation" module as well as the "SignatureRevocation" module also received updates in order to calculate checksums of updated and revoked EIRs and SRs.







out of scope of this thesis, but a worthwhile goal of future work. In the context of the CPN models of Authcoin, this particular risk is not an issue due to the limited size of the CPN models.

A CPN-Tools specific limitation arises from the limited functional programming interface. As mentioned in Section 5.3.3, only a simple generic implementation of a hash function is used to integrate SRP 1 into the existing CPN models. Besides the fact that a simple hash function is sufficient for the demonstration of implementing SRP 1, a further reason is that implementing complex hash functions in functional languages such as SML is a tedious, time consuming and error-prone task. Providing access to reviewed default implementation and libraries of major programming languages reduces the time for integrating certain patterns dramatically.

A more general limitation is related to the boundaries of the modeled system. Most models have dependencies on external systems that have only been modeled in a symbolic manner, such as the mining process or the public/private key system of the Authcoin protocol, or not at all. As a result, an implementation of security risk-oriented patterns with dependencies to such external systems is also limited to a symbolic manner. The symbolic signature on the hashes of data records is an example for this issue. Due to the symbolic implementation of the public and private keys, only a symbolic signature consisting of the user's EIR-ID has been implemented.

Besides the issues and problems mentioned above, there are some general differences of the application of security risk-oriented patterns in context of this thesis and previous applications. As already mentioned, so far security risk-oriented patterns have never been applied to CPN models before. Apart from that, the previous use cases and applications focus on systems with no existing security requirements at all whereas Authcoin already implemented basic security features [71][72][73]. Furthermore, the previous use cases target existing systems that are already in operation whereas in this thesis, the security risk-oriented patterns are applied during the design phase which makes the implementation of mitigation strategies less complicated.

## 5.5 Discussion

As a result of implementing and integrating the security risk-oriented pattern based on the same AOM methodology as in Chapter 3, the same limitations as stated in the discussion in Section 3.7 apply to the resulting CPN models of this chapter. However, certain limitations are specific results of the implementation of SRP 1. First, as mentioned before only an exemplary hash function is implemented in order to illustrate the process of hashing data records. As part of later real-world implementation of the

protocol a stronger and more reliable hash function has to be chosen and implemented. Due to the symbolic implementation of public-key cryptography in Chapter 3, only a symbolic signing of hashed data records is implemented. However, for the purpose of demonstrating the implementation in CPN, the provided solution is sufficient.

A further issue and limitation of the process described in this chapter is the manual process of pattern detection. As stated by Ahmed et al. this task, “is a manual activity, that potentially requires a good comprehension of the modelled domain and problem” [17]. Even though the author of this thesis is also a main author of the Authcoin protocol and the corresponding CPN models are not that complex, it still does not guarantee that all pattern occurrences have been detected correctly.

## 5.6 Conclusion

Selected existing security risk-oriented patterns are implemented in order to mitigate security risks identified in Chapter 4. The selected pattern is implemented and integrated in the existing CPN models of Authcoin. The updated CPN models are presented and described. Moreover, further information on the novel process of applying security risk-oriented patterns to CPN models are provided.

As a first step, the existing security risk-oriented patterns are reviewed and afterwards appropriated patterns are selected for implementation in order to mitigate the security risks. The occurrences of the selected pattern in the CPN models of the protocol are identified. In order to implement and integrate the chosen pattern, the Authcoin goal model, the behavioral interface model as well as the protocol semantics are updated accordingly. The resulting updated CPN models of Authcoin are presented and described afterwards. Finally, we discuss further information and issues on the novel process of applying security risk-oriented patterns to CPN models. In addition, we outline differences between the previous application of security risk-oriented patterns to business process and the application to CPN models.

A potential subject of future work is the development of a process for automated occurrence detection of a security risk-oriented pattern in a given system model.

## Chapter 6

# Evaluation

*In the following chapter, state space analyses are performed on the CPN models of the Authcoin protocol, before and after integrating the security risk-oriented pattern as described in Chapter 5. In order to avoid a state space explosion, both models are separated into sub-modules and a full state space is calculated for each. Based on the results of the analyses, we derive certain model properties and explain their implications. Furthermore, we discuss and compare the increased model complexity caused by the integration of the security risk-oriented pattern.*

### 6.1 Introduction

Based on the design science research methodology outlined in Chapter 1, the following chapter focuses on the evaluation of artifacts created in the course of this thesis. The CPN models of Authcoin developed in Chapter 3 as well as the enhanced CPN models from Chapter 5, that were created based on the risk and threat analysis of Chapter 4, are evaluated and compared using state space analyses.

The idea of a state space analysis is to calculate all reachable states and state changes of a given CPN model and represent the results in a directed graph, “where the nodes correspond to the set of reachable markings and the arcs correspond to occurring binding elements” [43]. Based on this graph, it is possible to deduce certain properties of the CPN models and the systems presented by the models. The state space analyses used in this chapter are generated using built-in functionalities of CPN-Tools. In addition to a full state space analysis, a SCC graph (Strongly Connected Component) is calculated based on the directed graph of the state space analysis. The nodes of the SCC graph “are obtained by making a disjoint division of the nodes in the state space such that two state space nodes are in the same SCC if and only if they are mutually reachable, i.e.,

there exists a path in the state space from the first node to the second node and vice versa” [43]. Based on the SCC graph it is possible to deduce further model properties, e.g. if the SCC graph has less nodes than the state space graph, than at least one cycle exists in the the state space graph of the CPN model. Further properties are explained as needed in the Sections 6.2 and 6.3.

The chapter is structured as follows: First, a state space analysis is performed on the CPN models from Chapter 3 in Section 6.2, afterwards the analysis of the CPN models from Chapter 5 is presented in Section 6.3. Section 6.4 focuses on related work. Finally, the results are discussed in Section 6.5 followed by the conclusion in Section 6.6. The source files of the state space analyses are available in Appendix C.

## 6.2 Evaluation of Authcoin’s CPN Models Without Security Risk-oriented Patterns

The first state space analysis is performed on the CPN models of the Authcoin protocol presented in Chapter 3. Since the full computational verification of the whole CPN model is not feasible for this size of models and causes a state space explosion, all parts of the models are tested independently. As listed in Table 6.1, the CPN model is divided into six parts, that are tested with prepared input statements that aim to cover as many execution paths as possible without causing a state space explosion. The “Key-GenerationEstablishBinding” module is depicted in Figure 3.7, the “FormalValidation” module in Figure 3.13, the “V&A” module in Figure 3.14 and the “Revocations” module in Figure 3.10. The “VARcreation” module listed in Table 6.1, refers to a slightly modified version of the “SymbolicMining” module illustrated in Figure 3.25. The number of created VARs is artificially limited to only one in order to guarantee a feasible calculation time of the state space. The “ProcessVAR” module from Table 6.1, consists of the “Mining” module depicted in Figure 3.9 in combination with the “V&A” module of Figure 3.14, minus the “SymbolicMining” module of Figure 3.25. A full state space is calculated for all modules listed above, followed by the calculation of the SCC graph. During these calculations, all other parts of the CPN models have been disabled. Relevant results and derived properties are presented in the following Table 6.1.

Module	Loops	Home markings	Dead markings	Dead transition	Live transition
Key Generation Establish Binding	No	No	Yes*	No	No
Formal Validation	No	No	Yes*	Yes*	No

Validation & Authentication	No	No	Yes*	Yes*	No
VAR Creation	No	No	Yes*	No	No
Process VAR	No	No	Yes*	Yes*	No
Revocations	No	No	Yes*	Yes*	No

TABLE 6.1: Selected state space analysis results for the CPN models of Chapter 3.

\* resulting from intentional disabling of marking paths for the purpose of focusing on specific marking paths under investigation and prevent state space explosion

As presented in Table 6.1, none of the tested modules contains any loops. Thus, there are no infinite occurrences of execution paths which guarantees the termination of each module. It is important to keep in mind that the properties of the separated modules might differ from the properties of the whole CPN model itself, due to the combination and influences of the different components on each other. But due to an unfeasible state space size it is not possible to calculate the properties of the whole system at once. However, it is possible to deduce from the design of the Authcoin protocol that there are loops in the complete model, since the blockchain architecture causes loops when chaining new blocks to the blockchain.

The state space analysis also revealed the absence of any home markings. A home marking is a marking that can be reached from any other reachable marking, meaning that it is impossible to have an occurrence sequence that cannot be extended to reach the home marking. “In other words, we cannot do things which will make it impossible to reach ” [43] the home marking afterwards.

All detected dead markings of the modules are caused either by intentionally disabling certain parts of the CPN models or customized input values that prevent a state space explosion. “A dead marking is a marking in which no binding elements are enabled” [43]. The existence of at least one dead marking for each of the modules guarantees a termination of executable actions at a certain point preventing infinite runtime.

Due to the fact that all modules contain a dead marking, none of them has a live transition. By definition, “a transition is live if from any reachable marking we can always find an occurrence sequence containing the transition” [43], though this is not possible from a dead marking. Therefore, the modules do not contain any live transitions. However, the detected dead transitions are all caused by the intentional disabling of execution paths and prepared input statements. A transition is considered dead if there is no reachable marking that enables the transition. Since all occurrences of dead transitions are artificially enforced, it means that all transitions of all tested modules can be potentially

enabled at a certain point during the protocol execution.

### 6.3 Evaluation of Authcoin's CPN Models With Security Risk-oriented Patterns

The second state space analysis is performed on the CPN models of the Authcoin protocol presented in Chapter 5, resulting from the integration of a security risk-oriented pattern into the CPN models. For this state space analysis, the same module separation into six sub-modules is used as previously in Section 6.2. The only difference for this section is, that certain parts of the original modules have been updated during the integration of the security risk-oriented pattern as outlined in Chapter 5. The modifications of certain sub-modules have already been illustrated and discussed in Section 5.3.4. Again, a full state space is calculated for all modules listed above followed by the calculation of the SCC graph. During these calculations, all other parts of the CPN models have been disabled. The same customized inputs are used as in Section 6.2, only extended by the new attributed fields of some token colors as introduced in Chapter 5. Relevant results and derived properties are presented in the following Table 6.2.

Module	Loops	Home markings	Dead markings	Dead transition	Live transition
Key Generation Establish Binding	No	No	Yes*	No	No
Formal Validation	No	No	Yes*	Yes*	No
Validation & Authentication	No	No	Yes*	Yes*	No
VAR Creation	No	No	Yes*	No	No
Process VAR	No	No	Yes*	Yes*	No
Revocations	No	No	Yes*	Yes*	No

TABLE 6.2: Selected state space analysis results for the CPN models of Chapter 5.

\* resulting from intentional disabling of marking paths for the purpose of focusing on specific marking paths under investigation and prevent state space explosion

The model properties derived from the second state space analysis in Table 6.2 are the same as for the first state space analysis in Table 6.1. Again, none of the modules contains any loops, home marking or live transitions. Furthermore, all dead markings and dead transitions are caused by intentional disabling of execution paths and prepared input

statements. Based on these findings, we deduce that the implementation and integration of the security risk-oriented pattern has not changed relevant model properties causing undesired results or behavior.

Table 6.3 compares the number of nodes and arcs in the directed graph of the state space analysis for both analyses conducted in Section 6.2 and Section 6.3. The number of nodes and arcs of the “KeyGenerationEstablishBinding” module, the “V&A” module, the “VARcreation” module and the “Revocations” module only slightly increased due to the integration of the security risk-oriented pattern. However, the number of nodes and arcs of the graph of the “FormalValidation” module as well as the “ProcessVAR” module increased heavily. In case of the “FormalValidation” module, the number of nodes quadrupled whereas the number of arcs quintupled. The number of nodes and arcs of the “ProcessVAR” module tripled. Even though all modules below have been modified in the process of implementing and integrating the security risk-oriented pattern, only the complexity of the state space of two of them grew significantly due to additional integrity-check-related execution paths. Since most integrity check-related changes of the CPN models are part of the two mentioned modules, it is reasonable that their complexity increased more than the complexity of the other remaining modules.

Module	Nodes*	Arcs*	Nodes**	Arcs**
Key Generation Establish Binding	466	772	466	772
Formal Validation	708	2523	2877	13588
Validation & Au- thentication	308	601	372	830
VAR Creation	355	911	373	971
Process VAR	2502	6288	6067	21884
Revocations	13	12	13	12

TABLE 6.3: Complexity comparisons of the two state space analyses based on number of nodes and arcs.

\* CPN models without security risk-oriented patterns

\*\*CPN models with security risk-oriented patterns

It is important to keep in mind that the number of nodes and arcs listed in Table 6.3 depends on the chosen input parameters. Since we use the same input for the both state space analyses, it is possible to compare them and deduce complexity tendencies for the different modules.



## 6.4 Related Work

Novel security and authentication protocols are proposed on a regular base and analyzed in terms of security and correctness. Several works demonstrated the feasibility of Colored Petri Nets for formalizing and analyzing existing protocols [32][33][74][75]. Vanek and Rohlik [74] demonstrate an implementation of an authentication protocol and show how Coloured Petri Nets can be used to create fully functional models. Xu and Xie [75] demonstrate the usability of Coloured Petri Nets to not only model existing security protocols but also to identify potential attacks through model checking. Many other publications demonstrate the general feasibility of Coloured Petri Nets for protocol analysis and verification, e.g. [34][30][35][32][33]. In other projects, the Alloy<sup>1</sup> framework is used to analyze and verify authentication protocols, e.g. [76][77].

Additional formalisms and approaches have been deployed in the past in order to perform a security analysis and verification of security and authentication protocols [78][79][80][81]. Most recently, a formal security analysis was performed on the underlying protocol of the popular Signal Messenger<sup>2</sup> by Cohn-Gordon et al. [15]. Furthermore, Basin et al. [82] provide an overview of the main applications of model checking in security protocol analysis in their book.

The use of security patterns in software engineering is a common practice, e.g. [36][37][83]. In the context of business processes, the applicability of security risk-oriented patterns introduced by Ahmed and Matulevičius [18] have been demonstrated in different scenarios of the aviation sector, e.g. [71][72].

The security analysis and verification of security and authentication protocols using formal methods as well as the use of security patterns is a common practice based on the related work provided above. However, to the knowledge of the author, the combination of formal verification of protocol behavior in combination with a risk and threat analysis and the implementation and integration of security patterns into the formal models seems to be novel.

## 6.5 Discussion

In Section 6.2 and Section 6.3 we perform state space analyses on both of our CPN models of Authcoin. First, on the CPN models of Authcoin as described in [16]. Afterwards, an analysis is performed on the CPN models containing the integrated security

---

<sup>1</sup><http://alloy.mit.edu/alloy/>

<sup>2</sup><https://whispersystems.org/>

risk-oriented pattern. In both analyses, certain limitations apply. In order to avoid a state space explosion, the CPN models are separated into six modules and each module is tested independently. The properties of those separated modules might differ from the properties of the whole CPN model itself, due to the combination and influences of the different components on each other. Furthermore, the correctness of the separated modules does not guarantee a correctness of the combination of all modules. Nevertheless, verifying the correct execution and behavior of sub-modules strengthens the assumptions that the protocol as a whole works as intended if all sub-modules do so.

Further limitations of the evaluation result from the customized input statements. Even though the input statements are designed to cover as many paths of the state space graph as possible, certain places and transition with minor relevance had to be left out intentionally in order to avoid a state space explosion. For the same reason, the number of iterative executions of the “VARcreation” module has been limited artificially in such a manner, that only one VAR is created.

Finally, we left out several model properties generated in the process of the state space analyses since they are not relevant in the context of our evaluation. Properties such as the upper and lower bounds of places or multi-set bounds depend on the chosen inputs and do not provide any further relevant information regarding the protocols behavior and properties.

## 6.6 Conclusion

We conduct state space analyses of the CPN models of Authcoin, developed earlier in Chapter 3 and Chapter 5. The first CPN model represents the Authcoin protocol as described in [16], whereas the second CPN model extends the initial model by implementing and integrating a security risk-oriented pattern in order to mitigate risks identified in Chapter 4. Both models are separated into sub-modules and a full state space is calculated for each of the modules, followed by an evaluation. Certain model properties are derived from the results of the state space analysis. None of the models contains any loops, home marking or live transitions. Furthermore, all dead marking or dead transitions are caused by the chosen customized inputs or intentional disabling of execution paths. Based on the results we deduce that each of the sub-modules is executed as designed and does not demonstrate any unintended behavior. Furthermore, we discuss and compare the increased model complexity caused by the integration of the security risk-oriented pattern.

However, we also acknowledge and discuss the limitations of our evaluation based on the state space analyses. We performed the analyses on separated sub-modules of the CPN

models and also used customized input statements in order to avoid a state space explosion. Moreover, the properties of the separated modules might differ from the properties of the whole CPN model itself and do not guarantee the correctness of the combination of all modules.

## Chapter 7

# Conclusion and Future Work

The following Chapter 7 concludes this thesis, summarizes the research efforts and answers the research questions outlined in Chapter 1. Section 7.1 provides a general conclusion of the thesis, followed by Section 7.2 that answers each of the defined research questions independently. Afterwards, Section 7.3 describes limitations of the this thesis. Finally, Section 7.4 provides an outlook on future work.

### 7.1 Conclusion

This thesis outlines the process of formalizing the Authcoin protocol using Colored Petri Nets. Based on an agent-oriented modeling methodology, the goal models and behavioral interface models of Authcoin are defined as well as the protocol semantics in form of token colors that represent the used data structures. Afterwards, the CPN model of Authcoin is derived from the AOM models and the defined protocol semantics using CPN-Tools.

Subsequently, a risk and threat analysis of the protocol is performed using the ISSRM domain model in order to identify and analyze security risks that threaten Authcoin. The risks are identified and necessary security requirements and security controls defined that mitigate the identified risks. Following the identification of the risks, the existing security risk-oriented patterns are reviewed and appropriate ones are selected. Finally, the selected pattern is implemented and integrated into the existing CPN models of Authcoin, followed by state space analyses of the CPN models before and after the integration of the security risk-oriented pattern. We discuss and explain implications of model properties derived from the state space analyses and compare changes of the model complexity due to the integration of the security risk-oriented pattern.

The result of this work is a complete CPN model of the Authcoin protocol secured using

security risk-oriented patterns based on a novel process for applying such patterns to CPN models.

## 7.2 Answering the Research Questions

The main research question defined for this thesis is: How to secure the Authcoin protocol by employing formal techniques combined with applying security risk-oriented patterns? As outlined in Chapter 1, we divided this research question into three sub-questions in order to answer it in a more structured way. The following sections conclude the answers to each of the subquestions.

### 7.2.1 RQ-1 - How to formalize the Authcoin protocol?

The Authcoin protocol is formalized using Colored Petri Nets and CPN-Tools. First, an agent-oriented modeling methodology is used to create the goal models as well as the corresponding behavior interface models of the protocol. In addition, the protocol semantics are defined in the form of token colors representing the used data structures. Afterwards, a complete CPN model of Authcoin is created based on the AOM models and the defined protocol semantics using CPN-Tools. The resulting top-level CPN model of Authcoin consists of four sub-modules. The top-level CPN modules as well as the further refined sub-modules are illustrated and described in depth.

### 7.2.2 RQ-2 - How to analyze security threats of the Authcoin protocol?

A risk and threat analysis of the Authcoin protocol is performed using the ISSRM domain model in order to identify and analyze security threats of the protocol. As the first step of the risk and threat analysis, the assets of the protocol are determined followed by an identification of potential threat agents, attack methods, threats, vulnerabilities, events, impacts and risks for each of the assets. We identify the processes of sending and receiving data records as well as the process of posting data records to the blockchain as assets of Authcoin. Furthermore, EIRs, CRs, RRs, SRs and VARs are identified as relevant data objects exchanged during the protocol execution. For each of the identified assets, a risk and threat analysis is performed by considering the following risk components: threat agents, attack methods, threats, vulnerabilities, events, impacts and risks. As a result, we identify three risks that threaten the protocol. First, the risk of data record manipulation during the process of posting information to the blockchain. Second, the risk of data record manipulation during the process of exchanging V&A related

information during the protocol execution. Third, the risk of a local or global DDoS attack on Authcoin’s users or infrastructure. Finally, we provide security requirements and security controls necessary to mitigate the identified risks.

### **7.2.3 RQ-3 - How to apply security risk-oriented patterns to the Authcoin protocol?**

We review the existing security risk-oriented patterns in order to select the appropriate patterns that mitigate the security risks identified during the risk and threat analysis of RQ-2. Afterwards, the occurrences of the selected risk-oriented pattern are manually detected in the CPN models. The Authcoin goal models, the behavioral interface models as well as the protocol semantics are updated in order to implement and integrate the chosen pattern. Afterwards, the resulting updated CPN models are presented and described in detail. Finally, we discuss further information and issues on the novel process of applying security risk-oriented patterns to CPN models. In addition, we outline differences between the previous application of security risk-oriented patterns to business process and the application to CPN models.

## **7.3 Limitations**

Several limitations apply to different parts and aspects of this thesis. First, we decided to simplify certain aspects during the modeling process of the CPN models. The blockchain mining process is implemented in a symbolic way, the number of processed VARs is artificially limited to two and the CPN models do not contain limitations regarding which user can fulfill a VAR or not in contrast to the protocol description in [16]. Furthermore, due to the socio-technical nature of the Authcoin protocol, certain aspects of the model are simplified, such as STRING-based placeholder challenges and randomized variables at different places and transitions in order to simulate decision of external entities.

Additionally, the performed risk and threat analysis does not guarantee that there are no further undetected risks and security flaws in the protocol. Additional risk analysis methods and professional penetration testing might uncover further risks and security flaws. Moreover, the risk and threat analysis has been performed by one of the authors of the Authcoin protocol that might result in a biased perception of the protocol’s security. Finally, it is also important to keep in mind that we limited the scope of the risk analysis and excluded certain aspects. Some of the identified assets, e.g. blockchain, user devices

and the underlying communication network pose security risks that cannot be mitigated in the context of Authcoin.

Further limitation originate from the implementation and integration of the security risk-oriented pattern, e.g. only an exemplary hash function is implemented in order to illustrate the process of hashing data records. In addition, the symbolic implementation of public-key cryptography only allows symbolic signing of hashed data records. More issues and limitation are caused by the manual and complex process of pattern detection which requires a good comprehension of the modeled system. Even though the author of this thesis is also a main author of Authcoin, it still does not guarantee that all pattern occurrences have been detected correctly.

## 7.4 Future Work

Throughout this thesis, we identified several open issues that require further research. Some of them are specifically linked to this thesis, others focus on more general problems related to this work. First, specific limitations of the CPN models of the Authcoin protocol have to be eliminated. This concerns especially the symbolic modeling of the blockchain storage system as well as the symbolic mining procedure. In addition, the elimination of simplifications of the CPN model as discussed in Section 3.7 can also improve the quality of the CPN models. Moreover, a real world implementation of Authcoin is planned based on the provided CPN models.

Furthermore, the risk and threat analysis performed in Chapter 4 is conducted by one of the main authors of Authcoin and therefore, provides only a biased view on the protocol. Incorporating the feedback of external experts and different analysis methods improves the security and reliability of the Authcoin protocol.

A further potential topic of research focuses on the development of a process for automated occurrence detection of a security risk-oriented pattern in a given system model. As demonstrated in Chapter 5, the manual detection of pattern occurrences is a difficult and time consuming task and requires a good comprehension of the modeled domain and problem. Therefore, an automated detection of occurrences, or at least a partially automated support for detections, is desirable.

The identification of additional unknown security risk-oriented patterns is another topic of research. In the process of this thesis, we list the existing security risk-oriented patterns and discuss their applicability to the Authcoin CPN models in order to mitigate identified risks. Future research focuses on answering the questions whether there are

any missing security risk-oriented patterns that mitigate additional risks. Furthermore, it is worth to evaluate the application and integration of these patterns in other contexts.

Finally, even though the five utilized security risk-oriented patterns meet all criteria of a pattern, additional vetting and analysis of them further strengthen their validity and recognition within the pattern community.



# Appendix A

## Goal Model

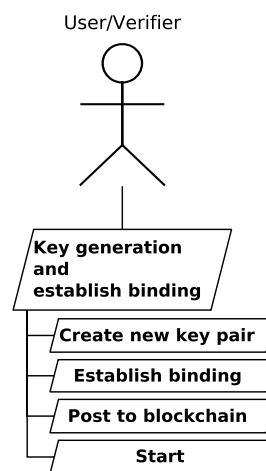


FIGURE A.1: Goal model for sub-goal “Key generation and establish binding”

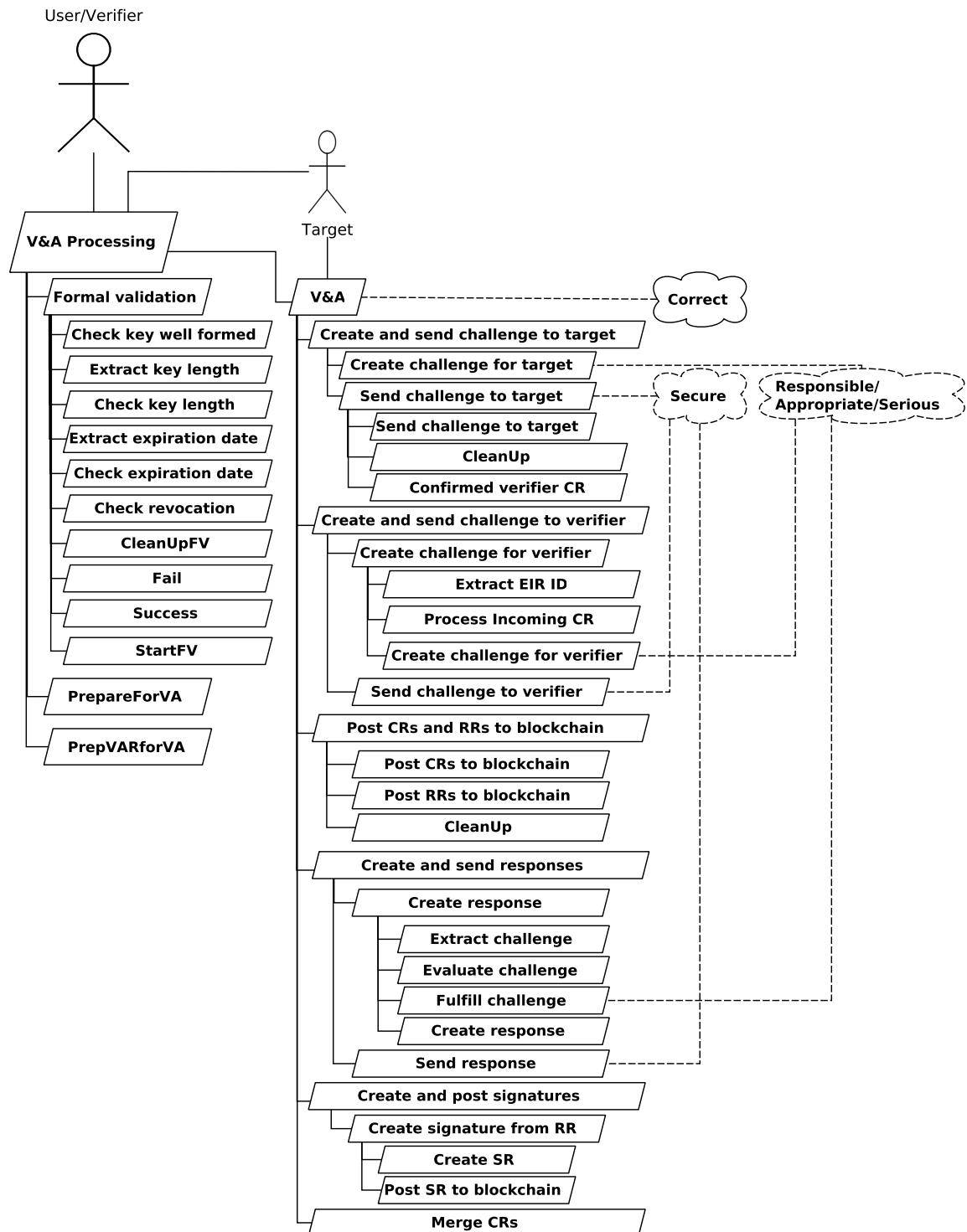


FIGURE A.2: Goal model for sub-goal "V&amp;A Processing"

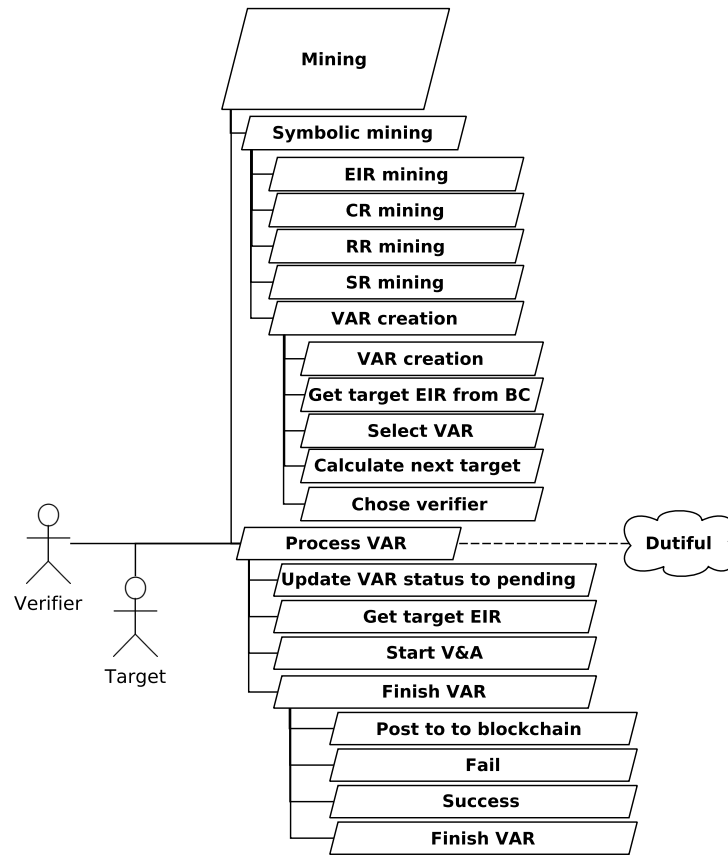


FIGURE A.3: Goal model for sub-goal "Mining"

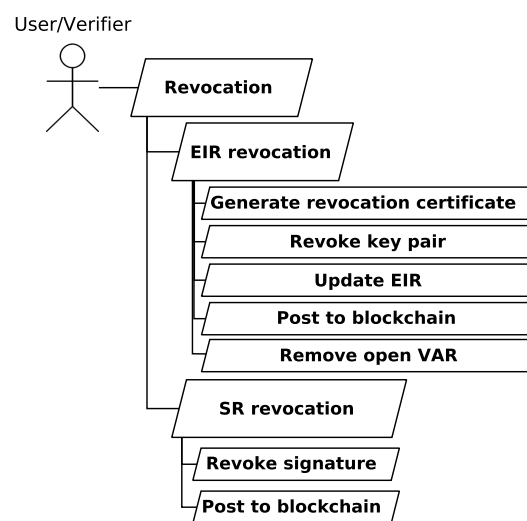


FIGURE A.4: Goal model for sub-goal "Revocation"

## Appendix B

# Behavioral Interfaces of Activities

### B.1 Behavioral Interfaces - Key Generation and Establish Binding

Activity	Trigger	Precondition	Postcondition
Start	User wants to create new key pair	Identifier list, key expiration date, key type, key length	Separated input for succeeding steps
Create new key pair	User started key generation	Key expiration date, key type, key length, OpenPGP/OpenSSH output	Public and private key
Establish binding	User started key generation	Public/Private key and identifier list	Entity identity record (EIR)
Post to blockchain	Generated new EIR	EIR	EIR available on blockchain

TABLE B.1: Behavioral interfaces of activities for the goal “Key generation and establish binding”

## B.2 Behavioral Interfaces - V&A Processing

Activity	Trigger	Precondition	Postcondition
Prepare for VA	Received target and verifier EIR	Target and verifier EIR, ID	VAE
Prep VAR for VA	Received target and verifier EIR	Target and verifier EIR, ID	VAE
Formal validation	Received VAE	VAE	VAE or failure message
Validation and authentication	Received VAE	VAE, successful formal validation	CR, RR and SR on blockchain or failure message

TABLE B.2: Behavioral interfaces of activities for the goal “Validation and authentication”

**B.2.1 Behavioral Interfaces for Subgoal “Formal Validation”**

<b>Activity</b>	<b>Trigger</b>	<b>Precondition</b>	<b>Postcondition</b>
StartFV	Received VAE	VAE	VAE-ID, verifier EIR, target EIR
Check key well formed	Started formal validation	VAE-ID, EIR	Key well formed result, public key, revocation information
Extract key length	Received VAE-ID, public key	VAE-ID, public key	VAE-ID, key length
Check key length	Extracted key length	VAE-ID, key length	VAE-ID, result of key length evaluation
Extract expiration date	Received VAE-ID, public key	VAE-ID, public key	VAE-ID, expiration date
Check expiration date	Extracted expiration date	VAE-ID, expiration date	VAE-ID, result of expiration date evaluation
Check revocation	Received VAE-ID, revocation information	VAE-ID, revocation information	VAE-ID, result of evaluation
Success	Received successful confirmation for all evaluations	VAE-ID, positive evaluation results	VAE-ID, verifier EIR, target EIR
Fail	Received at least one negative evaluation result	VAE-ID, negative evaluation results	VAE-ID, false, failure message
CleanUpFV	Formal validation for VAE-ID failed	VAE-ID, remaining evaluation results	None

TABLE B.3: Behavioral interfaces of activities for the subgoal “Formal validation”

### B.2.2 Behavioral Interfaces for Subgoal “Validation and Authentication”

Activity	Trigger	Precondition	Postcondition
Create and send challenge to target	Passed formal validation	VAE	VAE-ID, verifier EIR, target EIRs and target CR or failure message
Create and send challenge to verifier	Received challenge from verifier	VAE-ID, verifier EIR, target EIR, incoming CR	Verifier and target CRs or failure message
Merge CRs	Received incoming CRs	VAE-ID, verifier and target CR	VAE-ID, verifier and target CR
Create and send responses	Received incoming CRs	VAE-ID, verifier and target CR	VAE-ID, target and verifier RRs or failure message
Post CRs and RRs to blockchain	Received incoming RRs	VAE-ID, verifier and target RRs as well as CRs	Verifier and target CRs and RRs available on blockchain or failure message
Create and post signatures	Received incoming RRs, CRs and RRs posted to blockchain	VAE-ID, verifier and target RRs	SRs available on blockchain, validation and authentication finished

TABLE B.4: Behavioral interfaces of activities for the subgoal “Validation and authentication”

#### B.2.2.1 Behavioral Interfaces for Subgoal “Create and Send Challenge to Target”

Activity	Trigger	Precondition	Postcondition
Create challenge for target	Passed formal validation	VAE	VAE-ID, verifier and target EIRs, verifier challenge
Send challenge to target	Verifier created challenge	VAE-ID, verifier challenge	VAE-ID, Target CR or failure message

TABLE B.5: Behavioral interfaces of activities for the subgoal “Create and send challenge to target”

Activity	Trigger	Precondition	Postcondition
Create challenge for target	Passed formal validation	VAE, CR-ID, Prepared challenge	VAE-ID, verifier and target EIRs, verifier challenge

TABLE B.6: Behavioral interfaces of activities for the subgoal “Create challenge for target”

Activity	Trigger	Precondition	Postcondition
Send challenge to target	Verifier created challenge	VAE-ID, CR for target	VAE-ID, cleanup deadline, target CR
CleanUp	Verifier sent challenge to target	VAE-ID, cleanup deadline, CR for target	Failure message
Confirmed verifier CR	Verifier sent challenge to target and target sent challenge to verifier	VAE-ID, cleanup deadline, target and verifier CRs	Target and verifier CRs

TABLE B.7: Behavioral interfaces of activities for the subgoal “Send challenge to target”

### B.2.2.2 Behavioral Interfaces for Subgoal “Create and Send Challenge to Verifier”

Activity	Trigger	Precondition	Postcondition
Create challenge for verifier	Target received challenge from verifier	VAE-ID, target CR, verifier and target EIR	VAE-ID, verifier CR, target challenge
Send challenge to verifier	Target created challenge for verifier	VAE-ID, CR for verifier	VAE-ID, CR for verifier

TABLE B.8: Behavioral interfaces of activities for the subgoal “Create and send challenge to verifier”



Activity	Trigger	Precondition	Postcondition
Extract EIR-ID	Received CR from verifier	VAE-ID, CR for target	VAE-ID, target CR, verifier EIR-ID
Process incoming CR	Received CR from verifier and extracted verifier EIR-ID	VAE-ID, verifier EIR, verifier EIR-ID	VAE-ID, Verifier EIR
Create challenge for verifier	Processed incoming CR	VAE-ID, verifier and target EIRs, challenge for verifier	VAE-ID, CR for verifier

TABLE B.9: Behavioral interfaces of activities for the subgoal “Create challenge for verifier”

### B.2.2.3 Behavioral Interfaces for Subgoal “Create and Send Responses”

Activity	Trigger	Precondition	Postcondition
Create response	Verifier and target CRs received	VAE-ID, verifier and target CRs	VAE-ID and RR or failure message
Send response	Created response for CR	VAE-ID, RR	VAE-ID,RR

TABLE B.10: Behavioral interfaces of activities for the subgoal “Create and send response”

Activity	Trigger	Precondition	Postcondition
Extract challenge	Received CR	VAE-ID, CR	VAE-ID,CR, extracted challenge
Evaluate challenge	Extracted challenge from CR	VAE-ID, CR, challenge, challenge evaluation	VAE-ID, CR and evaluation result or failure message
Fulfill challenge	Evaluated challenge	VAE-ID, CR, challenge evaluation	VAE-ID, CR, fulfilled challenge
Create response	User fulfilled challenge	VAE-ID, CR, fulfilled challenge	VAE-ID, RR

TABLE B.11: Behavioral interfaces of activities for the subgoal “Create response”

**B.2.2.4 Behavioral Interfaces for Subgoal “Post CRs and RRs to Blockchain”**

Activity	Trigger	Precondition	Postcondition
Post CRs to blockchain	Verifier and target CRs received	VAE-ID, verifier and target CRs	VAE-ID, cleanup deadline and CRs available on blockchain
Post RRs to blockchain	Received RRs for verifier and target CRs	VAE-ID, RRs, CleanUp-ID, cleanup deadline	VAE-ID, RRs processed and RRs available on blockchain
CleanUp	CRs posted to blockchain, waiting for RRs	VAE-ID, cleanup deadline, RR	Failure message

TABLE B.12: Behavioral interfaces of activities for the subgoal “Post CRs and RRs to blockchain”

**B.2.2.5 Behavioral Interfaces for Subgoal “Create and Post Signatures”**

Activity	Trigger	Precondition	Postcondition
Create signature from RR	Received RRs	VAE-ID, RRs	SRs available on blockchain

TABLE B.13: Behavioral interfaces of activities for the subgoal “Create and post signature”

Activity	Trigger	Precondition	Postcondition
Create Signature	Received a RR	VAE-ID, RR, signature lifespan, response evaluation	VAE-ID, SR
Post SR to blockchain	Created SR	VAE-ID, SR	SR available on blockchain

TABLE B.14: Behavioral interfaces of activities for the subgoal “Create signatures from RR”

### B.3 Behavioral Interfaces - Revocations

Activity	Trigger	Precondition	Postcondition
Signature revocation	User wants to revoke a SR	CR, RR, SR, EIR, (VAR) available on blockchain, key pair	Revoked SR
EIR revocation	User wants to revoke an EIR	EIR, (VAR) and key pair	Revoked EIR

TABLE B.15: Behavioral interfaces of activities for the goal “Revocations”

#### B.3.1 Behavioral Interfaces - Signature Revocation

Activity	Trigger	Precondition	Postcondition
Revoke signature	User wants to revoke a SR	CR, RR, SR, EIR, (VAR) available on blockchain, OpenPGP/OpenSSH, Verifier’s key pair	Revoked SR
Post to blockchain	Revoked SR	Revoked SR	Revoked SR available on blockchain

TABLE B.16: Behavioral interfaces of activities for the goal “SR revocation”

### B.3.2 Behavioral Interfaces - EIR Revocation

Activity	Trigger	Precondition	Postcondition
Generate revocation certificate	User wants to revoke an EIR	User's key pair, EIR, OpenPGP/OpenSSH	Revocation certificate
Revoke key pair	User wants to revoke key pair	Revocation certificate, Verifier's key pair, EIR	Revoked key pair
Update EIR	Received revoked key pair and EIR	EIR and corresponding revoked key pair	Updated EIR
Remove corresponding open VAR(s)	Received revoked key pair and EIR	EIR, revoked key pair, corresponding VAR(s)	Updated corresponding VARs to "finished"
Post to blockchain	Updated EIR	EIR	Updated EIR available on blockchain

TABLE B.17: Behavioral interfaces of activities for the goal "EIR revocation"

## B.4 Behavioral Interfaces - Mining

Activity	Trigger	Precondition	Postcondition
Symbolic mining	New transactions posted to blockchain	Input transactions	Transactions included in blockchain, (VARs)
Process VAR	VAR selected	VAR	VAR and corresponding results available on blockchain

TABLE B.18: Behavioral interfaces of activities for the goal "Mining"

#### B.4.1 Behavioral Interfaces for Subgoal “Symbolic Mining”

Activity	Trigger	Precondition	Postcondition
EIR mining	New EIR(s) transaction(s) posted to blockchain	Input transactions, blockchain	EIR(s) included in blockchain
CR mining	New CR(s) transaction(s) posted to blockchain	Input transactions, blockchain	CR(s) included in blockchain
RR mining	New RR(s) transaction(s) posted to blockchain	Input transactions, blockchain	RR(s) included in blockchain
SR mining	New SR(s) transaction(s) posted to blockchain	Input transactions, blockchain	SR(s) included in blockchain
VAR creation	Mining of new block finished	New block, EIR	New VAR

TABLE B.19: Behavioral interfaces of activities for the subgoal “Symbolic mining”

#### B.4.2 Behavioral Interfaces for Subgoal “Process VAR”

Activity	Trigger	Precondition	Postcondition
Update VAR status	User selected VAR	VAR, EIR	VAR and updated VAR on blockchain, EIR
Get target EIR	User started VAR processing	VAR, target EIR	VAR, target EIR
Start V&A (see Table B.2)	-	-	-
Finish VAR	V&A finished	VAE-ID, VAR	Updated VAR on blockchain

TABLE B.20: Behavioral interfaces of activities for the subgoal “Process VAR”

**B.4.2.1 Behavioral Interfaces for Subgoal “Finish VAR”**

<b>Activity</b>	<b>Trigger</b>	<b>Precondition</b>	<b>Postcondition</b>
Post to blockchain	Received final VAR	VAR	VAR available on blockchain
Fail	V&A failed	VAE-ID, cleanup deadline, failure message	VAE-ID, VAR result
Success	V&A successful	VAE-ID, cleanup deadline	VAE-ID, VAR result
Finish VAR	Successful/failed V&A	VAE-ID, VAR result	Final VAR

TABLE B.21: Behavioral interfaces of activities for the subgoal “Finish VAR”

# Appendix C

## Files

### C.1 CPN Models

CPN models without integrated security risk-oriented patterns:

<https://owncloud.gwdg.de/index.php/s/EX0EFxY6go4oj23>

CPN models with integrated security risk-oriented patterns:

<https://owncloud.gwdg.de/index.php/s/KCgl3fjRfUs9F5y>

### C.2 State Space Analyses

State space analysis without integrated security risk-oriented patterns:

<https://owncloud.gwdg.de/index.php/s/wKUcymG14WU7Cgo>

State space analysis with integrated security risk-oriented patterns:

<https://owncloud.gwdg.de/index.php/s/3cKtovp7rZTr0tL>

### C.3 Online Sources

Collection of archived online sources:

<https://owncloud.gwdg.de/index.php/s/VXnZBgRHXlHKuUd>

# Bibliography

- [1] Nancy Cam-Winget, Russ Housley, David Wagner, and Jesse Walker. Security Flaws in 802.11 Data Link Protocols. *Communications of the ACM*, 46(5):35–39, 2003.
- [2] Ulf Carlsen. Cryptographic Protocol Flaws: Know Your Enemy. In *Computer Security Foundations Workshop VII, 1994. CSFW 7. Proceedings*, pages 192–200. IEEE, 1994.
- [3] Serge Vaudenay. Security Flaws Induced by CBC Padding - Applications to SSL, IPSEC, WTLS... In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 534–545. Springer, 2002.
- [4] F. Javier Thayer Fábrega, Jonathan C. Herzog, and Joshua D. Guttman. Strand Spaces: Why is a Security Protocol Correct? In *Security and Privacy, 1998. Proceedings. 1998 IEEE Symposium on*, pages 160–171. IEEE, 1998.
- [5] Algirdas Avizienis, J. C. Laprie, Brian Randell, and Carl Landwehr. Basic Concepts and Taxonomy of Dependable and Secure Computing. *IEEE transactions on dependable and secure computing*, 1(1):11–33, 2004.
- [6] Morrie Gasser. *Building a Secure Computer System*. Van Nostrand Reinhold Company New York, 1988.
- [7] Benjamin Fung, Ke Wang, Rui Chen, and Philip S. Yu. Privacy-preserving Data Publishing: A Survey of Recent Developments. *ACM Computing Surveys (CSUR)*, 42(4):14, 2010.
- [8] Adam Stubblefield, John Ioannidis, and Aviel D. Rubin. A Key Recovery Attack on the 802.11b Wired Equivalent Privacy Protocol (WEP). *ACM transactions on information and system security (TISSEC)*, 7(2):319–332, 2004.
- [9] Chris Brook. Nuclear Power Plant Disrupted by Cyber Attack. <https://threatpost.com/nuclear-power-plant-disrupted-by-cyber-attack/121216/>, 2016. (Accessed March 13, 2017).



- [10] Carl Adam Petri. *Kommunikation mit Automaten*. PhD thesis, Technical University of Darmstadt, 1962.
- [11] Robin Milner, Joachim Parrow, and David Walker. A Calculus of Mobile Processes, I. *Information and computation*, 100(1):1–40, 1992.
- [12] Charles Antony Richard Hoare. Communicating Sequential Processes. In *The origin of concurrent programming*, pages 413–443. Springer, 1978.
- [13] Federico Crazzolari and Glynn Winskel. Events in Security Protocols. In *Proceedings of the 8th ACM conference on Computer and Communications Security*, pages 96–105. ACM, 2001.
- [14] Martín Abadi and Andrew D. Gordon. A Calculus for Cryptographic Protocols: The Spi Calculus. In *Proceedings of the 4th ACM conference on Computer and communications security*, pages 36–47. ACM, 1997.
- [15] Katriel Cohn-Gordon, Cas Cremers, Benjamin Dowling, Luke Garratt, and Douglas Stebila. A Formal Security Analysis of the Signal Messaging Protocol. URL: <https://eprint.iacr.org/2016/1013.pdf>. (Accessed March 13, 2017).
- [16] Benjamin Leiding, Clemens H. Cap, Thomas Mundt, and Samaneh Rashidibajgan. Authcoin: Validation and Authentication in Decentralized Networks. In *The 10th Mediterranean Conference on Information Systems - MCIS 2016*, Cyprus, CY, September 2016.
- [17] Naved Ahmed, Raimundas Matulevičius, and Naiad Hossain Khan. Eliciting Security Requirements for Business Processes Using Patterns. In *Proceedings of the 9th International Workshop on Security in Information Systems (ICEIS 2012)*, pages 49–58, 2012. ISBN 978-989-8565-15-0. doi: 10.5220/0004100200490058.
- [18] Naved Ahmed and Raimundas Matulevičius. Securing Business Processes Using Security Risk-oriented Patterns. *Computer Standards & Interfaces*, 36(4):723–733, 2014.
- [19] Frank Buschmann, Kevin Henney, and Douglas C. Schmidt. *Pattern-oriented Software Architecture - On Patterns and Pattern Languages*, volume 5. John Wiley & Sons, 2007.
- [20] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. URL: <https://bitcoin.org/bitcoin.pdf>, 2008. (Accessed March 13, 2017).
- [21] Radia Perlman. An Overview of PKI Trust Models. *Network, IEEE*, 13(6):38–43, 1999.

- [22] Philip R. Zimmermann. PGP 2.X Manual. <ftp://ftp.pgpi.org/pub/pgp/2.x/doc/pgpdoc1.txt>, 1994. (Accessed March 13, 2017).
- [23] Jeannette M. Wing. A Specifier's Introduction to Formal Methods. *Computer*, 23(9):8–22, 1990.
- [24] David Harel. Statecharts: A Visual Formalism for Complex Systems. *Science of computer programming*, 8(3):231–274, 1987.
- [25] Yuri Gurevich et al. Evolving Algebras 1993: Lipari Guide. *Specification and validation methods*, pages 9–36, 1995.
- [26] Robin Milner. *Communication and Concurrency*, volume 84. Prentice Hall New York etc., 1989.
- [27] Rajeev Alur and David L. Dill. A Theory of Timed Automata. *Theoretical computer science*, 126(2):183–235, 1994.
- [28] Kurt Jensen. Coloured Petri Nets. In *Discrete Event Systems: A New Challenge for Intelligent Control Systems, IEE Colloquium on*, pages 5–1. IET, 1993.
- [29] Alex Norta, Collaborative CINCO, and Interoperable Computing. Safeguarding Trusted eBusiness Transactions of Lifecycles for Cross-Enterprise Collaboration. *University of Helsinki, Department of Computer Science, Helsinki, Finland, Tech. Rep. C-2012-1*, 2012.
- [30] Salah Aly and Khaled Mustafa. Protocol Verification and Analysis Using Colored Petri Nets, 2004.
- [31] A. M. Basyouni and S. E. Tavares. New Approach to Cryptographic Protocol Analysis Using Coloured Petri Nets. In *Electrical and Computer Engineering, 1997. Engineering Innovation: Voyage of Discovery. IEEE 1997 Canadian Conference on*, volume 1, pages 334–337. IEEE, 1997.
- [32] Wiebke Dresp. Security Analysis of the Secure Authentication Protocol by Means of Coloured Petri Nets. In *IFIP International Conference on Communications and Multimedia Security*, pages 230–239. Springer, 2005.
- [33] Panupong Sornkhom and Yongyuth Permpoontanalarp. Security Analysis of Micali's Fair Contract Signing Protocol by Using Coloured Petri Nets. In *2008 Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, 2008.
- [34] Kim Edwards. *Cryptographic Protocol Specification and Analysis Using Coloured Petri Nets and Java*. PhD thesis, Queen's University Kingston, 1999.

- [35] Issam Al-Azzoni, Douglas G. Down, and Ridha Khedri. Modeling and Verification of Cryptographic Protocols Using Coloured Petri Nets and Design/CPN. *Nord. J. Comput.*, 12(3):200–228, 2005.
- [36] Joseph Yoder and Jeffrey Barcalow. Architectural Patterns for Enabling Application Security. *Urbana*, 51:61801, 1998.
- [37] Markus Schumacher. *Security Engineering with Patterns: Origins, Theoretical Models, and New Applications*, volume 2754. Springer, 2003.
- [38] Anton V. Uzunov and Eduardo B. Fernandez. An Extensible Pattern-based Library and Taxonomy of Security Threats for Distributed Systems. *Computer Standards & Interfaces*, 36(4):734–747, 2014.
- [39] Silver Samarütel. Revision of Security Risk-oriented Patterns for Distributed Systems. Master’s thesis, University of Tartu, 2016.
- [40] Gregor Bochmann and Carl Sunshine. Formal Methods in Communication Protocol Design. *IEEE Transactions on Communications*, 28(4):624–631, 1980.
- [41] Alan R. Hevner, Salvatore T. March, Jinsoo Park, and Sudha Ram. Design Science in Information Systems Research. *MIS quarterly*, 28(1):75–105, 2004.
- [42] Andrew K. Shenton. Strategies for Ensuring Trustworthiness in Qualitative Research Projects. *Education for information*, 22(2):63–75, 2004.
- [43] Kurt Jensen, Lars Michael Kristensen, and Lisa Wells. Coloured Petri Nets and CPN Tools for Modelling and Validation of Concurrent Systems. *International Journal on Software Tools for Technology Transfer*, 9(3-4):213–254, 2007.
- [44] Buterin, Vitalik. Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform. URL: <https://github.com/ethereum/wiki/wiki/White-Paper>, 2014. (Accessed March 13, 2017).
- [45] Gavin Wood. Ethereum: A Secure Decentralized Generalised Transaction Ledger. URL: <http://gavwood.com/paper.pdf>, 2014. (Accessed March 13, 2017).
- [46] Patrick Dai, Neil Mahi, Jordan Earls, and Alex Norton. Smart-Contract Value-Transfer Protocols on a Distributed Mobile Application Platform. URL: <https://qtum.org/uploads/files/cf6d69348ca50dd985b60425ccf282f3.pdf>, 2017. (Accessed March 13, 2017).
- [47] Kevin Delmolino, Mitchell Arnett, Ahmed Kosba, Andrew Miller, and Elaine Shi. A Programmers Guide to Ethereum and Serpent. URL: [https://mc2-umd.github.io/ethereumlab/docs/serpent\\_tutorial.pdf](https://mc2-umd.github.io/ethereumlab/docs/serpent_tutorial.pdf), 2015. (Accessed October 11, 2016).

- [48] V. Bjorn. Cryptographic Key Generation Using Biometric Data. URL: <https://www.google.com/patents/US6035398>, March 7 2000. US Patent 6,035,398.
- [49] Frank Buschman, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal, and Frank Buschmann. *Pattern Oriented Software Architecture: A System of Patterns*. Chichester, UK: John Wiley and Sons, 1996.
- [50] Alex Norta, Paul Grefen, and Nanjangud C. Narendra. A Reference Architecture for Managing Dynamic Inter-organizational Business Processes. *Data & Knowledge Engineering*, 91:52–89, 2014.
- [51] Markus Schumacher, Eduardo Fernandez-Buglioni, Duane Hybertson, Frank Buschmann, and Peter Sommerlad. *Security Patterns: Integrating Security and Systems Engineering*. John Wiley & Sons, 2013.
- [52] Munawar Hafiz, Paul Adamczyk, and Ralph E. Johnson. Organizing Security Patterns. *IEEE software*, 24(4):52, 2007.
- [53] Michaela Bunke, Rainer Koschke, and Karsten Sohr. Organizing Security Patterns Related to Security and Pattern Recognition Requirements. *International Journal on Advances in Security*, 5, 2012.
- [54] Anas Motii, Brahim Hamid, Agnès Lanusse, and Jean-Michel Bruel. Guiding the Selection of Security Patterns Based on Security Requirements and Pattern Classification. In *Proceedings of the 20th European Conference on Pattern Languages of Programs*, EuroPLoP '15, pages 10:1–10:17, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3847-9. doi: 10.1145/2855321.2855332.
- [55] Paolo Giorgini, John Mylopoulos, and Roberto Sebastiani. Goal-oriented Requirements Analysis and Reasoning in the Tropos Methodology. *Engineering Applications of Artificial Intelligence*, 18(2):159–171, 2005.
- [56] Michael Wooldridge, Nicholas R. Jennings, and David Kinny. The Gaia Methodology for Agent-oriented Analysis and Design. *Autonomous Agents and multi-agent systems*, 3(3):285–312, 2000.
- [57] Lin Padgham and Michael Winikoff. Prometheus: A Methodology for Developing Intelligent Agents. In *International Workshop on Agent-Oriented Software Engineering*, pages 174–185. Springer, 2002.
- [58] B. Moulin and L. Cloutier. Soft computing. chapter Collaborative Work Based on Multiagent Architectures: A Methodological Perspective, pages 261–296. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1994. ISBN 0-13-146234-2.

- [59] Bernard Moulin and Mario Brassard. A Scenario-based Design Method and an Environment for the Development of Multiagent Systems. In *Australian Workshop on Distributed Artificial Intelligence*, pages 216–232. Springer, 1995.
- [60] Scott A. DeLoach. Analysis and Design using MaSE and agentTool. Technical report, DTIC Document, 2001.
- [61] Msury Mahunnah, Alex Norta, Lixin Ma, and Kuldar Taveter. Heuristics for Designing and Evaluating Socio-technical Agent-Oriented Behaviour Models with Coloured Petri Nets. In *Computer Software and Applications Conference Workshops (COMPSACW), 2014 IEEE 38th International*, pages 438–443. IEEE, 2014.
- [62] Leon Sterling and Kuldar Taveter. *The Art of Agent-oriented Modeling*. MIT Press, 2009.
- [63] Barry Boehm. Software Risk Management. In *European Software Engineering Conference*, pages 1–19. Springer, 1989.
- [64] Éric Dubois, Patrick Heymans, Nicolas Mayer, and Raimundas Matulevičius. A Systematic Approach to Define the Domain of Information System Security Risk Management. In *Intentional Perspectives on Information Systems Engineering*, pages 289–306. Springer, 2010.
- [65] CCTA British. CRAMM (CCTA Risk Analysis and Management Method) - User Guide version 5.0. *Insight Consulting*, 2003.
- [66] Christopher J. Alberts and Audrey J. Dorofee. *OCTAVE Method Implementation Guide: Version 2.0*. Carnegie Mellon University, 2001.
- [67] Folker den Braber, Ida Hogganvik, M. S. Lund, Ketik Stølen, and Fredrik Vraalsen. Model-based Security Analysis in Seven Steps - A Guided Tour to the CORAS Method. *BT Technology Journal*, 25(1):101–117, 2007.
- [68] Nicolas Mayer. *Model-based Management of Information System Security Risk*. PhD thesis, University of Namur, 2009.
- [69] Brian Krebs. Akamai on the Record KrebsOnSecurity Attack. URL: <https://krebsonsecurity.com/2016/11/akamai-on-the-record-krebsonsecurity-attack/>, 2016. (Accessed March 13, 2017).
- [70] Marc Stevens, Elie Bursztein, Pierre Karpman, Ange Albertini, and Yarik Markov. The First Collision for Full SHA-1. URL: <https://shattered.it/static/shattered.pdf>, 2017. (Accessed March 13, 2017).

- [71] Silver Samarütel, Raimundas Matulevičius, Alex Norta, and Rein Nõukas. Securing Airline-Turnaround Processes Using Security Risk-Oriented Patterns. In *IFIP Working Conference on The Practice of Enterprise Modeling*, pages 209–224. Springer, 2016.
- [72] Raimundas Matulevičius, Alex Norta, Chibozur Udokwu, and Rein Nõukas. Security Risk Management in the Aviation Turnaround Sector. In *International Conference on Future Data and Security Engineering*, pages 119–140. Springer, 2016.
- [73] Chibuzor Joseph Udokwu. Analysis of Digital Security Threats in Aviation Sector. Master’s thesis, Tallinn University of Technology, 2017.
- [74] Tomas Vanek and Matej Rohlik. Model of DoS Resistant Broadcast Authentication Protocol in Colored Petri Net Environment. In *Proc. 17th Int. Conf. Systems, Signals and Image Processing, (IWSSIP 2010)*, pages 264–267, 2010.
- [75] Yang Xu and Xiaoyao Xie. Modeling and Analysis of Security Protocols Using Colored Petri Nets. *JCP*, 6(1):19–27, 2011.
- [76] Marino Miculan and Caterina Urban. Formal Analysis of Facebook Connect Single Sign-on Authentication Protocol. In *SOFSEM*, volume 11, pages 22–28, 2011.
- [77] Suhas Pai, Yash Sharma, Sunil Kumar, Radhika M. Pai, and Sanjay Singh. Formal Verification of OAuth 2.0 Using Alloy Framework. In *Communication Systems and Network Technologies (CSNT), 2011 International Conference on*, pages 655–659. IEEE, 2011.
- [78] Alessandro Armando, Roberto Carbone, Luca Compagna, Jorge Cuéllar, Giancarlo Pellegrino, and Alessandro Sorniotti. An Authentication Flaw in Browser-based Single Sign-On Protocols: Impact and Remediations. *Comput. Secur.*, 33:41–58, March 2013. ISSN 0167-4048. doi: 10.1016/j.cose.2012.08.007.
- [79] Hsiu-Lien Yeh, Tien-Ho Chen, Pin-Chuan Liu, Tai-Hoo Kim, and Hsin-Wen Wei. A Secured Authentication Protocol for Wireless Sensor Networks Using Elliptic Curves Cryptography. *Sensors*, 11(5):4767–4779, 2011.
- [80] Younsung Choi, Donghoon Lee, Jiye Kim, Jaewook Jung, Junghyun Nam, and Dongho Won. Security Enhanced User Authentication Protocol for Wireless Sensor Networks Using Elliptic Curves Cryptography. *Sensors*, 14(6):10081–10106, 2014.
- [81] Muhammad Khurram Khan and Khaled Alghathbar. Cryptanalysis and Security Improvements of Two-factor User Authentication in Wireless Sensor Networks. *Sensors*, 10(3):2450–2459, 2010.

- 
- [82] David Basin, Cas Cremers, and Catherine Meadows. Model Checking Security Protocols. *Handbook of Model Checking*, 2011.
  - [83] Nobukazu Yoshioka, Hironori Washizaki, and Katsuhisa Maruyama. A Survey on Security Patterns. *Progress in Informatics*, 5(5):35–47, 2008.

# Declaration of Authorship

I, Benjamin Leiding, declare that this thesis titled “Securing the Authcoin Protocol Using Security Risk-oriented Patterns” and the work presented in it are my own. I confirm that I authored this thesis independently, that I have not used any sources other than the declared sources, and that I have explicitly marked all material which has been quoted either literally or by content from external sources.

Date:

---

Signed:

---