Scuola di Scienze Matematiche, Fisiche e Naturali
Corso di Laurea Magistrale in Informatica

Tesi di Laurea Magistrale

PROGETTAZIONE DI UNO SMART CONTRACT A SUPPORTO DEL PROTOCOLLO DI FAIR EXCHANGE DI VERIOSS, UNA PIATTAFORMA BUG BOUNTY BASATA SULLA BLOCKCHAIN

DESIGN OF A SMART CONTRACT TO SUPPORT THE FAIR EXCHANGE PROTOCOL OF VERIOSS, A BLOCKCHAIN-BASED BUG-BOUNTY PLATFORM

FRANCESCO MUCCI

Relatore: Prof. *Rosario Pugliese*
Correlatori: Prof. *Gabriele Costa*, Dott. *Letterio Galletta*

Anno Accademico 2023-2024

A <Nome>,
<frase di dedica>.

"Le vent se lève!... il faut tenter de vivre!"
— Paul Valéry, Le Cimetière marin, 1920 [1].

"The best theory is inspired by practice
and the best practice is inspired by theory."
— Donald E. Knuth, Theory and practice, 1991 [2].

PREFAZIONE

Durante l'anno accademico 2019-2020 ho collaborato con l'unità di ricerca SySMA della Scuola IMT Alti Studi Lucca in qualità di beneficiario della borsa di ricerca **VeriOSS smart contract development** (finanziata con i fondi del progetto PAI 2018 "*VeriOSS: a security-by-smart contract verification framework for Open Source Software*" - P0137). L'obiettivo della borsa era quello di progettare e sviluppare smart contract Solidity a supporto del protocollo di fair exchange di VeriOSS, una piattaforma per la bug bounty basata sulla blockchain. Il lavoro di tesi svolto prosegue e conclude quanto iniziato durante la suddetta collaborazione di ricerca.

Tutto il materiale prodotto per questo lavoro di tesi è accessibile attraverso diverse repository pubbliche su GitHub; in particolare:

- i file LaTeX associati a questo documento si trovano in `github.com/FrancescoMucci/VeriOSS-thesis`;
- il codice implementato è disponibile in `github.com/FrancescoMucci/VeriOSS-challenge-reward`;
- infine, i diagrammi di sequenza, di stato e di classe sono raccolti in `github.com/FrancescoMucci/VeriOSS-diagrams`.

Questa tesi è stato realizzata utilizzando come base un template che ho sviluppato a partire da quello fornito dal Corso di Laurea Magistrale in Informatica dell'Università degli Studi di Firenze. Tale template è pubblicamente accessibile nella seguente repository GitHub: `github.com/FrancescoMucci/LaTeX-thesis-template-cs-unifi`.

Per individuare e correggere involontarie somiglianze o citazioni non adeguate, è stato utilizzato *Turnitin*, il software antiplagio messo a disposizione dall'Università degli Studi di Firenze.

INDICE

ACRONIMI

CVD Crowdsourced Vulnerability Discovery

BBP Bug Bounty Program

VDP Vulnerability Disclosure Program

BI Bounty Issuer

BH Bounty Hunter

ELENCO DELLE FIGURE

ELENCO DELLE TABELLE

ELENCO DEI CODICI

RIFERIMENTI BIBLIOGRAFICI PER ARGOMENTO

INTRODUZIONE ALL'APPENDICE

In questa appendice, riservata unicamente alla bozza della tesi, vengono presentati i riferimenti bibliografici consultati, organizzati in base all'argomento e alla tipologia di documento.

VERIOSS

Articoli scientifici di Costa et al.

- VeriOSS: Using the Blockchain to Foster Bug Bounty Programs [3];
- Verifying a Blockchain-Based Remote Debugging Protocol for Bug Bounty [4].

PIATTAFORME BUG BOUNTY

Tesi di dottorato di Walshe e articoli scientifici di Walshe et al.

- Supporting Data-driven Software Development Life-cycles with Bug Bounty Programmes [5];
- Organisational Perspectives [6];
- Current State of Bug Bounty Programmes and Platforms [7];
- An Empirical Study of Bug Bounty Programs [8];
- Coordinated Vulnerability Disclosure programme effectiveness: Issues and recommendations [9].

Articoli scientifici di Akgul et al.

- Bug Hunters' Perspectives on the Challenges and Benefits of the Bug Bounty Ecosystem [10];
- The Hackers' Viewpoint: Exploring Challenges and Benefits of Bug-Bounty Programs [11].

Altri articoli scientifici

- Bug Bounty Programs for Cybersecurity: Practices, Issues, and Recommendations [12];
- Web Science Challenges in Researching Bug Bounties [13].

PIATTAFORME BUG BOUNTY BASATE SULLE BLOCKCHAIN

Articoli scientifici di Hoffman et al. su Bountychain

- Decentralized Security Bounty Management on Blockchain and IPFS [14];
- Bountychain: Toward Decentralizing a Bug Bounty Program with Blockchain and IPFS [15].

Articoli scientifici di Badash et al. su BBBB Framework

- Blockchain-Based Bug Bounty Framework [16].

Articoli scientific di Lisi et al. su ARD

- Automated Responsible Disclosure of Security Vulnerabilities [17].

PROTOCOLLI DI FAIR EXCHANGE

Articoli scientifici seminali

- Optimistic Protocols for Multi-Party Fair Exchange [18];
- Fair Exchange with a Semi-trusted Third Party [19];
- Optimistic Fair Exchange of Digital Signatures [20];
- Secure Group Barter: Multi-party Fair Exchange with Semi-trusted Neutral Parties [21].

Revisioni sistematiche

- A Review of Fair Exchange Protocols [22];
- A Survey on Optimistic Fair Exchange Protocol and its Variants [23];
- Fair Exchange Protocol in Electronic Transactions Revisited [24].

PROTOCOLLI DI FAIR EXCHANGE BASATI SULLA BLOCKCHAIN

Articoli scientifici su FairSwap

- FairSwap: How To Fairly Exchange Digital Goods [25];
- Privacy-preserving FairSwap: Fairness and privacy interplay [26].

Articoli scientifici su OptiSwap

- OptiSwap: Fast Optimistic Fair Exchange [27];
- Privacy-enhanced OptiSwap [28].

Articoli scientifici su cost fairness

- Cost Fairness for Blockchain-Based Two-Party Exchange Protocols [29];
- Formalizing Cost Fairness for Two-Party Exchange Protocols using Game Theory and Applications to Blockchain [30];
- Formalizing Cost Fairness for Two-Party Exchange Protocols using Game Theory and Applications to Blockchain (Extended Version) [31].

Articoli scientifici su protocolli che usano zero-knowledge proof

- FileBounty: Fair Data Exchange [32];
- Contingent Payments from Two-party Signing and Verification for Abelian Groups [33].

Altri articoli scientifici

- FairTrade: Efficient Atomic Exchange-based Fair Exchange Protocol for Digital Data Trading [34].

PROOF OF KNOWLEDGE

Monografie

- Proofs, Arguments, and Zero-Knowledge [35].

Capitoli di libri

- Sigma Protocols and Efficient Zero-Knowledge [36];
- Identification and signatures from Sigma protocols [37];
- Proving properties in zero-knowledge [38];
- A Survey on Zero-Knowledge Proofs [39].

Articoli scientifici seminali

- The Knowledge Complexity of Interactive Proof-Systems [40].

Altri articoli scientifici

- Do You Need a Zero Knowledge Proof? [41];
- A Survey on Zero Knowledge Range Proofs and Applications [42].

PROOF OF KNOWLEDGE PER LA BLOCKCHAIN

Revisioni sistematiche

- Overview of Zero-Knowledge Proof and Its Applications in Blockchain [43];
- Non-Interactive Zero-Knowledge for Blockchain: A Survey [44];
- A Survey on Zero-Knowledge Proof in Blockchain [45].

FONDAMENTI DI BLOCKCHAIN, ETHEREUM E SOLIDITY

Libri generici sulla blockchain

- Handbook on Blockchain [46];
- Blockchain Essentials - Core Concepts and Implementations [47].

Libri specifici su Ethereum e sviluppo di smart contracts Solidity

- Mastering Ethereum: Building Smart Contracts and DApps [48];
- Ethereum Smart Contract Development in Solidity [49];
- Blockchain and Ethereum Smart Contract Solution Development - Dapp Programming with Solidity [50];
- Solidity Programming Essentials: A guide to building smart contracts and tokens using the widely used Solidity language [51].

Documentazione di Ethereum e Solidity

- Ethereum Development Documentation [52];
- Solidity Documentation - Release 0.8.18 [53].

White e yellow paper

- Bitcoin: A Peer-to-peer Electronic Cash System [54];
- Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform [55];
- Ethereum: A Secure Decentralised Generalised Transaction Ledger [56].

Lavori seminali

- Pricing via Processing or Combatting Junk Mail [57];
- Smart Contracts [58];
- Formalizing and Securing Relationships on Public Networks [59];
- b-money [60];
- Karma: A Secure Economic Framework for Peer-to-peer Resource Sharing [61];
- RPOW - Reusable Proofs of Work [62];
- Bit Gold [63].

ARCHITETTURA E SVILUPPO DI APPLICAZIONI BLOCKCHAIN-BASED

Libro e articoli scientifici di Xu et al.

- Architecture for Blockchain Applications [64];
- A Pattern Collection for Blockchain-based Applications [65];
- Applying Design Patterns in Smart Contracts [66];
- A Taxonomy of Blockchain-Based Systems for Architecture Design [67].

Tesi di dottorato di Wöhrer e articoli scientifici di Wöhrer et al.

- Engineering Blockchain-Based Applications in the Context of the Ethereum Ecosystem [68];
- Design Patterns for Smart Contracts in the Ethereum Ecosystem [69];
- Smart Contracts: Security Patterns in the Ethereum Ecosystem and Solidity[70];
- Architectural Design Decisions for Blockchain-Based Applications [71];
- Architecture Design of Blockchain-Based Applications [72].

Articoli scientifici di Marchesi et al.

- Design Patterns for Gas Optimization in Ethereum [73];
- ABCDE - Agile BlockChain Dapp Engineering [74];
- An Agile Software Engineering Method to Design Blockchain Applications [75].

Altri articoli scientifici - architettura

- Do you Need a Blockchain? [76].

Altri articoli scientifici - revisioni sistematiche

- A Systematic Literature Review of Blockchain and Smart Contract Development: Techniques, tools, and open challenges [77];
- A Comprehensive Survey on Smart Contract Construction and Execution: Paradigms, Tools, and Systems [78];
- Ethereum Smart Contract Analysis Tools: A Systematic Review [79].

Altri articoli scientifici - design pattern

- Challenges and Common Solutions in Smart Contract Development [80];
- Some Blockchain Design Patterns for Overcoming Immutability, Chain-Boundedness, and Gas Fees [81];
- Towards Saving Money in Using Smart Contracts [82].

Altri articoli scientifici - gas cost

- Computing Exact Worst-Case Gas Consumption for Smart Contracts [83];
- Profiling Gas Consumption in Solidity Smart Contracts [84];
- Reduction in Gas Cost for Blockchain Enabled Smart Contract [85].

ORACOLI BLOCKCHAIN

Introduzione agli oracoli blockchain

- A Study of Blockchain Oracles [86];

Design pattern per oracoli blockchain

- Blockchain Patterns [87];
- Foundational Oracle Patterns: Connecting Blockchain to the Off-Chain World [88];
- Off-chain Data Fetching Architecture for Ethereum Smart Contract [89].

Confronto tra oracoli blockchain

- Trustworthy Blockchain Oracles: Review, Comparison, and Open Research Challenges [90];
- From Trust to Truth: Advancements in Mitigating the Blockchain Oracle Problem [91];
- Connect API with Blockchain: A Survey on Blockchain Oracle Implementation [92].

Provable (Oraclize)

- Provable Documentation [93].

Chainlink

- Chainlink Docs [94];
- Chainlink 2.0: Next Steps in the Evolution of Decentralized Oracle Networks [95];
- Chainlink Off-chain Reporting Protocol [96].

OFF-CHAIN DATA STORAGES

Confronto tra on-chain e off-chain data storages

- An Overview of Blockchain Scalability for Storage [97];
- Performance Comparison of On-Chain and Off-Chain Data Storage Model Using Blockchain Technology [98].

Confronto tra diverse soluzioni per off-chain data storage

- Cost and Performance Analysis on Decentralized File Systems for Blockchain-Based Applications: State-of-the-Art Report [99];
- Blockchain-Based Distributed File System Security and Privacy: A Systematic Mapping Study [100].

Documentazione e articoli scientifici ufficiali di IPFS

- IPFS Documentation [101];
- IPFS - Content Addressed, Versioned, P2P File System [102];
- Design and evaluation of IPFS: a storage layer for the decentralized web [103].

Altri articoli scientifici su IPFS

- Toward Decentralized Cloud Storage With IPFS: Opportunities, Challenges, and Future Considerations [104];
- IPFS: An Off-Chain Storage Solution for Blockchain [105].

FONDAMENTI DI VERIFICA FORMALE

Libri di testo

- Handbook of Model Checking [106];
- Handbook of Satisfiability [107];
- Logic: Reference Book for Computer Scientists [108].

Nozioni di base

- Software Verification [109];
- Predicate Abstraction for Program Verification [110];
- Control Flow Analysis [111];
- Propositional SAT Solving [112];
- Sentential Logic (SL) [113];
- On Sentences Which are True of Direct Unions of Algebras [114].

Model checking

- Model Checking [115];
- 2^5 Years of Model Checking [116].

Satisfiability Modulo Theories (SMT)

- Satisfiability Modulo Theories [117];
- Satisfiability Modulo Theories [118];
- A Survey of Satisfiability Modulo Theory [119];
- A Tutorial on Satisfiability Modulo Theories [120].

Bounded Model Checking (BMC)

- SAT-Based Model Checking [121];
- Bounded Model Checking [122].

Lavori seminali su BMC

- Bounded model checking using satisfiability solving [123];
- SMT-Based Bounded Model Checking for Embedded ANSI-C Software [124].

Verifica di programmi e clausole di Horn

- Program Verification with Constrained Horn Clauses [125];
- Horn Clause Solvers for Program Verification [126];
- Analysis and Transformation of Constrained Horn Clauses for Program Verification [127];

Lavori seminali su Horn SAT

- Linear-time Algorithms for Testing the Satisfiability of Propositional Horn Formulae [128];
- Algorithms for Testing the Satisfiability of Propositional Formulae [129].

VERIFICA FORMALE DI SMART CONTRACT

Revisioni sistematiche

- Formal Verification of Smart Contracts [130];
- A Survey of Smart Contract Formal Specification and Verification [131];
- Formal Methods for the Verification of Smart Contracts: A Review [132];
- Formally Verifying a Real World Smart Contract [133].

Documentazione e articoli scientifici su SMTChecker di Solidity

- Solidity Documentation - SMTChecker and Formal Verification [134]
- A Solicitous Approach to Smart Contract Verification [135];
- Accurate Smart Contract Verification Through Direct Modelling [136];
- SMT-Based Verification of Solidity Smart Contracts [137];
- SolCMC: Solidity Compiler's Model Checker [138].

DEBUGGING

Revisioni sistematiche

- Debugging: a Review of the Literature from an Educational Perspective [139];
- A Systematic Review on Program Debugging Techniques [140].

Remote debugging

- Mercury: Properties and Design of a Remote Debugging Solution using Reflection [141];
- Remote Debugging for Containerized Applications in Edge Computing Environments [142].

Reverse debugging

- A Review of Reverse Debugging [143];
- Implementation of Live Reverse Debugging in LLDB [144].

WEAKEST PRECONDITION CALCULUS

Articoli scientifici seminali

- Guarded Commands, Nondeterminacy and Formal Derivation of Programs [145].

Libri di testo

- A Discipline of Programming [146];
- The Science of Programming [147];
- Predicate Calculus and Program Semantics [148].

Altri articoli scientifici

- The Weakest Precondition Calculus: Recursion and Duality [149].

SYMBOLIC EXECUTION

Revisioni sistematiche

- A Survey of Symbolic Execution Techniques [150];
- Advances in Symbolic Execution [151];
- Symbolic Execution and Recent Applications to Worst-Case Execution, Load Testing, and Security Analysis [152].

Revisioni di tools

- Benchmarking the Capability of Symbolic Execution Tools with Logic Bombs [153];
- Concolic Execution on Small-Size Binaries: Challenges and Empirical Study [154];
- Systematic Comparison of Symbolic Execution Systems: Intermediate Representation and its Generation [155].

Altri articoli scientifici

- Symbolic Execution Formally Explained [156].

SYMBOLIC EXECUTION CON ANGR

Documentazione

- angr: The angr Project [157].

Articoli scientifici

- SOK: (State of) The Art of War: Offensive Techniques in Binary Analysis [158];

- Driller: Augmenting Fuzzing Through Selective Symbolic Execution [159];
- Firmalice - Automatic Detection of Authentication Bypass Vulnerabilities in Binary Firmware [160].

Altri articoli scientifici

- Teaching with angr: A Symbolic Execution Curriculum and CTF [161];
- Tutorial: An Overview of Malware Detection and Evasion Techniques [162].

BACKWARD SYMBOLIC EXECUTION

Backward symbolic execution via weakest precondition calculus

- Snugglebug: a Powerful Approach to Weakest Preconditions [163];
- Handling Heap Data Structures in Backward Symbolic Execution [164];
- Higher-order Demand-driven Symbolic Evaluation [165];
- Backward Symbolic Execution with Loop Folding [166];
- Generation of the Weakest Preconditions of Programs with Dynamic Memory in Symbolic Execution [167].

NOZIONI BASE DI COMPUTER SCIENCE

Software quality and security

- Code Quality [168];
- Securing Software [169].

1

INTRODUZIONE

1.1 CONTESTO

1.2 PROBLEMA AFFRONTATO

1.3 STATO DELL'ARTE

1.4 DOMANDE DI RICERCA

1.5 APPROCCIO USATO

1.6 CONTRIBUTI ORIGINALI

1.7 STRUTTURA DELLA TESI

- Capitolo 2 - PRELIMINARI:
- Capitolo 3 - APPROCCIO:
- Capitolo 4 - VALUTAZIONE:
- Capitolo 5 - DISCUSSIONE:
- Capitolo 6 - LAVORI CORRELATI:
- Capitolo 7 - CONCLUSIONI:

La tesi include, oltre ai capitoli, anche le seguenti appendici:

- Appendice A - CODICI SORGENTE ADDIZIONALI:

2

PRELIMINARI

2.1 INTRODUZIONE AL CAPITOLO

2.2 NOZIONI PRELIMINARI

2.2.1 *Programmi bug bounty*

Definizioni relative ai programmi bug bounty

Definizione 2.1 (*Bug*)**.** Il termine *bug* identifica un qualunque difetto presente in un software che può alterarne il comportamento rispetto a quello atteso [13, 168].

Definizione 2.2 (*Vulnerabilità*)**.** Una *vulnerabilità* indica un *bug* che può essere sfruttato per portare alla compromissione di uno degli attributi di sicurezza del software (confidenzialità, integrità e disponibilità) [13, 169].

Definizione 2.3 (*Crowdsourced Vulnerability Discovery*)**.** Un *Crowdsourced Vulnerability Discovery (CVD)* è un approccio adottabile da un'organizzazione per identificare *vulnerabilità* presenti nei propri prodotti software tramite il supporto della comunità formata da esperti, ricercatori e appassionati di sicurezza informatica, esterni rispetto all'organizzazione stessa [6, 9].

Definizione 2.4 (*Bug Bounty Program*)**.** Un *Bug Bounty Program (BBP)* è una tipologia di *CVD* in cui vengono offerte ricompense monetarie in cambio dei report relativi alle *vulnerabilità* identificate [6, 9, 11].

Definizione 2.5 (*Vulnerability Disclosure Program*)**.** Un *CVD* in cui non vengono offerte ricompense monetarie viene normalmente chiamato *Vulnerability Disclosure Program (VDP)* [6, 9, 11].

Terminologia adottata per i programmi bug bounty

Definizione 2.6 (*Bounty Issuer*)**.** Con il termine *Bounty Issuer (BI)* indichiamo l'organizzazione (e.g. azienda, università o ente) che, attraverso un *BBP*, offre ricompense per le *vulnerabilità* presenti nei propri prodotti software [3].

Definizione 2.7 (*Bounty Hunter*)**.** Con l'espressione *Bounty Hunter (BH)* denotiamo un esperto, un ricercatore o un'appassionato di sicurezza informatica, indipendente rispetto al *BI*, che cerca *vulnerabilità* nel software designato da quest'ultimo [3].

Definizione 2.8 (*Bug Bounty*)**.** Con *bug bounty* indichiamo una descrizione, pubblicata dal *BI*, che specifica quali applicazioni software sono testabili, quali tipologie di *vulnerabilità* sono considerabili valide, le ricompense proposte e altri vincoli legali che il *BH* dovrà rispettare [15, 14]. In sostanza una *bug bounty* specifica le linee guida, spesso identificate con il termine di *safe harbor* o *rules of engagement*, che permettono al *BH* di ricercare *vulnerabilità* in modo del tutto legale [9, 8, 7].

Definizione 2.9 (*Bounty Reward*)**.** *Bounty reward* è il termine con cui denotiamo la ricompensa monetaria offerta dal *BI* [15]; normalmente, il valore di tale ricompensa dipenderà dalla tipologia e dalla criticità della *vulnerabilità* rilevata [3].

Definizione 2.10 (*Bug Report*)**.** Con l'espressione *bug report* indichiamo una descrizione dettagliata, prodotta dal *BH*, per descrivere una *vulnerabilità* identificata [15].

Interazione nei programmi bug bounty

Descriviamo brevemente l'interazione tra il *BI* e il *BH* [15]:

1. il *BI* rende pubblica la *bug bounty*;
2. quando il *BH* identifica una *vulnerabilità* che ritiene valida, prepara un *bug report* e lo invia al *BI*;
3. il *bug report* viene esaminato dal *BI* al fine di stabilire la validità della segnalazione; se questa viene confermata, il *BH* riceve la *bounty reward* per il suo contributo.

Classificazione dei programmi bug bounty

Possiamo classificare i *BBP* in base al fatto che l'interazione tra il *BI* e il *BH* avvenga direttamente o tramite da un intermediario:

- *Internal BBP*: sono *BBP* gestiti direttamente dal *BI* [15]; in tal caso la comunicazione tra il *BI* ed il *BH* avviene in modo diretto.
- *Bug Bounty Platform*: sono piattaforme che fungono da mediatore tra il *BI* ed il *BH* [15]; esempi di queste piattaforme sono HackerOne, Intigriti, Bugcrowd, Synack e Yogosha [7, 8].

Storia dei programmi bug bounty

Alla fine del 1995, l'azienda statunitense Netscape fu la prima a implementare un *BBP* al fine di individuare eventuali difetti software presenti nel web browser Netscape Navigator; tuttavia, bisognerà attendere i primi anni duemila affinché l'uso di questi programmi cominci a diffondersi maggiormente [15]. Ad oggi, le aziende di piccole o medie dimensioni che desiderano avviare un *BBP* si affidano a una delle varie *bug bounty platform* esistenti, mentre solo le aziende di grandi dimensioni – come Google, Meta o Microsoft – possono permettersi di gestire dei programmi interni [7].

Vantaggi dei programmi bug bounty

L'adozione dei *BBP* offre alla comunità degli hacker una via legale per la segnalazione delle *vulnerabilità*, disincentivando così l'utilizzo del mercato nero [13, 8, 7] e amplificando la probabilità che vengano scoperti *bug* che altrimenti sarebbero rimasti nascosti agli occhi degli eventuali team di cybersecurity a disposizione delle aziende [8].

Limitazioni dei programmi bug bounty

Una delle limitazioni più importanti degli *internal BBP* è il fatto che, una volta che il *BI* ha ricevuto il *bug report*, questi è fortemente incentivano a considerare i bug segnalati come poco critici o non validi al fine di ridurre al minimo la ricompensa per il *BH* [3, 10]; questa situazione rende il mercato inefficiente per i *BH* che risultano, di conseguenza, portati a cercare altre vie per la vendita dei bug [3].

Una risposta a questo problema arriva dalle *bug bounty platform* che, in quanto mediatori imparziali, dovrebbero riuscire ad ottenere ricompense più adeguate per i *BH* [3, 10].

Tuttavia anche le piattaforme bug bounty presentano delle criticità [16]:

1. Costi aggiuntivi: la maggior parte di queste piattaforme tassa in modo non trascurabile le ricompense elargite ai *BH*.
2. Poca trasparenza: le piattaforme spesso non delineano chiaramente il meccanismo di valutazione delle *vulnerabilità* segnalate; questa mancanza di trasparenza può risultare in un trattamento preferenziale verso i *BI*, che sono loro clienti.
3. Rischi di sicurezza: l'introduzione di un intermediario nell'interazione tra *BI* e il *BH* aumenta il rischio di divulgazione non autorizzata di informazioni sensibili.

Approccio di divulgazione delle vulnerabilità

I *BBP* possono anche essere distinti in base all'approccio adottato per la divulgazione della *vulnerabilità* identificate dai *BH*; in particolare, esistono tre principali approcci di divulgazione [17]:

- *Full Vendor Disclosure*: la vulnerabilità viene comunicata esclusivamente al *BI*, consentendogli di correggere il problema prima di qualsiasi altra divulgazione.
- *Full Public Disclosure*: la vulnerabilità viene resa pubblica immediatamente dopo la sua identificazione, senza concedere al *BI* alcun tempo per la correzione.
- *Responsible Disclosure*: la vulnerabilità è inizialmente comunicata esclusivamente a una terza parte fidata, incaricata di verificarne la validità; se questa viene confermata, il *bug report* viene inoltrato al *BI*, al quale viene dato un periodo di tempo prestabilito per correggere il problema nel proprio sistema software prima della divulgazione pubblica.

Riferimenti bibliografici aggiuntivi

Per eventuali approfondimenti sui *BBP*, oltre ai lavori già citati nel testo, si rimanda ai seguenti documenti:

- Walshe (2023), "Supporting Data-driven Software Development Lifecycles with Bug Bounty Programmes" [5];
- Malladi et al. (2020), "Bug Bounty Programs for Cybersecurity: Practices, Issues, and Recommendations" [12];
- Fryer e Simperl (2017), "Web Science Challenges in Researching Bug Bounties" [13].

2.2.2 *Protocollli di fair exchange*

2.2.3 *Blockchain*

2.3 LAVORI PRECEDENTI

2.4 METODI E TECNICHE UTILIZZATE

2.5 TECNOLOGIE UTILIZZATE

2.6 RIASSUNTO DEL CAPITOLO E CONCLUSIONI

3

APPROCCIO

3.1 INTRODUZIONE AL CAPITOLO

3.2 SPECIFICA DEI REQUISITI

3.2.1 *Requisiti funzionali*

3.2.2 *Requisiti non funzionali*

3.3 ARCHITETTURA DEL SISTEMA

3.3.1 *Design architetturale del sistema*

3.3.2 *Componente 1 del sistema*

Responsabilità

Interfacce

Dettagli algoritmici

Comportamento dinamico

3.3.3 *Componente n del sistema*

Responsabilità

Interfacce

Dettagli algoritmici

Comportamento dinamico

3.3.4 *Considerazioni sulle scelte architetturali*

3.4 RIASSUNTO DEL CAPITOLO E CONCLUSIONI

4

VALUTAZIONE

4.1 INTRODUZIONE AL CAPITOLO

4.2 IMPLEMENTAZIONE

4.2.1 *Implementazione componente 1*

4.2.2 *Implementazione componente n*

4.2.3 *Sfide implementative e soluzioni*

4.3 TEST

4.3.1 *Test d'unità*

Test componente 1

Test componente n

4.3.2 *Test d'integrazione*

Test integrazione componenti 1 e 2

Test integrazione componenti n-1 e n

4.3.3 *Test end-to-end*

4.4 QUALITÀ DEI TEST

4.4.1 *Test coverage*

4.4.2 *Mutation testing*

4.5 RISULTATI

4.6 RIASSUNTO DEL CAPITOLO E CONCLUSIONI

5

DISCUSSIONE

5.1 INTRODUZIONE AL CAPITOLO

5.2 OBIETTIVI RAGGIUNTI

5.3 DEBOLEZZE E LIMITAZIONI

5.4 QUESTIONI IRRISOLTE

5.5 NUOVE DOMANDE EMERSE

5.6 APPROCCI ALTERNATIVI

5.7 IMPATTO SCIENTIFICO E PRATICO DEI RISULTATI

5.8 RIASSUNTO DEL CAPITOLO E CONCLUSIONI

6

LAVORI CORRELATI

6.1 INTRODUZIONE AL CAPITOLO

6.2 PANORAMICA SULLO STATO DELL'ARTE

6.3 LAVORI DEBOLMENTE CORRELATI

6.3.1 *Lavoro debolmente correlato 1*

Idea principale

Punti di forza

Limitazioni e difetti

Influenza sul nostro lavoro

6.3.2 *Lavoro debolmente correlato 2*

Idea principale

Punti di forza

Limitazioni e difetti

Influenza sul nostro lavoro

6.4 LAVORI STRETTAMENTE CORRELATI

6.4.1 *Lavoro strettamente correlato 1*

Idea principale

Punti di forza

Limitazioni e difetti

Influenza sul nostro lavoro

6.4.2 *Lavoro strettamente correlato 2*

Idea principale

Punti di forza

Limitazioni e difetti

Influenza sul nostro lavoro

6.5 TENDENZE IDENTIFICATE

6.6 LACUNE NELLA LETTERATURA E NOSTRO CONTRIBUTO

6.7 RIASSUNTO DEL CAPITOLO E CONCLUSIONI

7

CONCLUSIONI

7.1 RIASSUNTO DELLA TESI

7.2 SVILUPPI FUTURI

A

CODICI SORGENTE ADDIZIONALI

A.1 INTRODUZIONE ALL'APPENDICE

A.2 CODICE ADDIZIONALE 1

A.3 CODICE ADDIZIONALE 2

A.4 CODICE ADDIZIONALE 3

BIBLIOGRAFIA

[1] Paul Valéry: *Il cimitero marino*. Interlinea edizioni, 2016, ISBN 9788868570880. Pubblicato per la prima volta nel 1920 con il titolo *Le Cimetière marin*. (Cited on page iii.)

[2] Donald E. Knuth: *Theory and Practice*. Theoretical Computer Science (Elsevier), vol. 90 (no. 1): pp. 1–15, novembre 1991. https://doi.org/10.1016/0304-3975(91)90295-D. (Cited on page iv.)

[3] Andrea Canidio, Gabriele Costa e Letterio Galletta: *VeriOSS: Using the Blockchain to Foster Bug Bounty Programs*. Nel *2nd International Conference on Blockchain Economics, Security and Protocols (Tokenomics 2020)*, volume 82 della serie *Open Access Series in Informatics (OASIcs)*, pagine 6:1–14. Schloss Dagstuhl, Leibniz-Zentrum für Informatik, Germania, febbraio 2021. https://doi.org/10.4230/OASIcs.Tokenomics.2020.6. (Cited on pages xiii, 3, 4, and 5.)

[4] Pierpaolo Degano, Letterio Galletta e Selene Gerali: *Verifying a Blockchain-Based Remote Debugging Protocol for Bug Bounty*. Nel *Protocols, Strands, and Logic*, volume 13066 della serie *Lecture Notes in Computer Science*, pagine 124–138. Springer, novembre 2021. https://doi.org/10.1007/978-3-030-91631-2_7. (Cited on page xiii.)

[5] Thomas J. Walshe: *Supporting Data-driven Software Development Life-cycles with Bug Bounty Programmes*. Tesi di dottorato, University of Oxford, Wolfson College, Inghilterra, giugno 2023. https://ora.ox.ac.uk/objects/uuid:4a828bbb-8ff4-4cac-9e09-5699b30c6d52. (Cited on pages xiii and 6.)

[6] Thomas J. Walshe: *Organisational Perspectives*. Nel *Supporting data-driven software development life-cycles with bug bounty programmes*, tesi di dottorato, cap. 2, pagine 8–61. University of Oxford, Wolfson College, Inghilterra, giugno 2023. https://ora.ox.ac.uk/objects/uuid:4a828bbb-8ff4-4cac-9e09-5699b30c6d52. (Cited on pages xiii and 2.)

[7] Thomas J. Walshe: *Current State of Bug Bounty Programmes and Platforms*. Nel *Supporting data-driven software development life-cycles with bug bounty programmes*, tesi di dottorato, cap. 3, pagine 62–99. University of Oxford, Wolfson College, Inghilterra, giugno 2023. https://ora.ox.ac.uk/objects/uuid:4a828bbb-8ff4-4cac-9e09-5699b30c6d52. (Cited on pages xiii, 3, and 4.)

[8] Thomas J. Walshe e Andrew C. Simpson: *An Empirical Study of Bug Bounty Programs*. Nel *2020 IEEE 2nd International Workshop on Intelligent Bug Fixing (IBF)*, pagine 35–44. Institute of Electrical and Electronics Engineers (IEEE), febbraio 2020. https://doi.org/10.1109/IBF50092.2020.9034828. (Cited on pages xiii, 3, and 4.)

[9] Thomas J. Walshe e Andrew C. Simpson: *Coordinated Vulnerability Disclosure programme effectiveness: Issues and recommendations*. Computers & Security (Elsevier), vol. 123: pp. 102936:1–14, dicembre 2022. https://doi.org/10.1016/j.cose.2022.102936. (Cited on pages xiii, 2, and 3.)

[10] Omer Akgul, Taha Eghtesad, Amit Elazari, Omprakash Gnawali, Jens Grossklags, Michelle L. Mazurek, Daniel Votipka e Aron Laszka: *Bug Hunters' Perspectives on the Challenges and Benefits of the Bug Bounty Ecosystem*. Nel *Proceedings of the 32nd USENIX Conference on Security Symposium (SEC '23)*, pagine 2275–2291. USENIX Association, agosto 2023. https://usenix.org/conference/usenixsecurity23/presentation/akgul. (Cited on pages xiv, 4, and 5.)

[11] Omer Akgul, Taha Eghtesad, Amit Elazari, Omprakash Gnawali, Jens Grossklags, Daniel Votipka e Aron Laszka: *The Hackers' Viewpoint: Exploring Challenges and Benefits of Bug-Bounty Programs*. Nel *Proceedings of the 6th Workshop on Security Information Workers (WSIW '20)*. Leibniz University Hannover, Germania, novembre 2020. https://wsiw2020.sec.uni-hannover.de/downloads/WSIW2020-The%20Hackers%20Viewpoint.pdf. (Cited on pages xiv and 2.)

[12] Suresh S. Malladi e Hemang C. Subramanian: *Bug Bounty Programs for Cybersecurity: Practices, Issues, and Recommendations*. IEEE Software (Institute of Electrical and Electronics Engineers), vol. 37 (no.

1): pp. 31–39, gennaio-febbario 2020. https://doi.org/10.1109/MS.2018.2880508. (Cited on pages xiv and 6.)

[13] Huw Fryer e Elena Simperl: *Web Science Challenges in Researching Bug Bounties*. Nel *Proceedings of the 2017 ACM on Web Science Conference (WebSci '17)*, pagina 273–277. Association for Computing Machinery (ACM), giugno 2017. https://doi.org/10.1145/3091478.3091517. (Cited on pages xiv, 2, 4, and 6.)

[14] Alex Hoffman, Eric Becerril-Blas, Kevin Moreno e Yoohwan Kim: *Decentralized Security Bounty Management on Blockchain and IPFS*. Nel *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*, pagine 241–247. Institute of Electrical and Electronics Engineers (IEEE), gennaio 2020. https://doi.org/10.1109/CCWC47524.2020.9031109. (Cited on pages xiv and 3.)

[15] Alex Hoffman, Phillipe Austria, Chol Hyun Park e Yoohwan Kim: *Bountychain: Toward Decentralizing a Bug Bounty Program with Blockchain and IPFS*. International Journal of Networked and Distributed Computing (Atlantis Press), vol. 9: pp. 86–93, luglio 2021. https://doi.org/10.2991/ijndc.k.210527.001. (Cited on pages xiv, 3, and 4.)

[16] Lital Badash, Nachiket Tapas, Asaf Nadler, Francesco Longo e Asaf Shabtai: *Blockchain-Based Bug Bounty Framework*. Nel *Proceedings of the 36th Annual ACM Symposium on Applied Computing (SAC '21)*, pagine 239–248. Association for Computing Machinery (ACM), marzo 2021. https://doi.org/10.1145/3412841.3441906. (Cited on pages xiv and 5.)

[17] Andrea Lisi, Prateeti Mukherjee, Laura De Santis, Lei Wu, Dmitrij Lagutin e Yki Kortesniemi: *Automated Responsible Disclosure of Security Vulnerabilities*. IEEE Access (Institute of Electrical and Electronics Engineers), vol. 10: pp. 10472–10489, settembre 2022. https://doi.org/10.1109/ACCESS.2021.3126401. (Cited on pages xiv and 5.)

[18] N. Asokan, Matthias Schunter e Michael Waidner: *Optimistic Protocols for Multi-Party Fair Exchange*. Report di ricerca, IBM Research Division, novembre 1996. https://schunter.org/bibliography/AsSW2_96FairMPX.IBMrep.pdf. (Cited on page xv.)

[19] Matthew K. Franklin e Michael K. Reiter: *Fair Exchange with a Semi-Trusted Third Party*. Nel *Proceedings of the 4th ACM Conference on Computer and Communications Security (CCS '97)*, pagine 1–5. Association for Computing Machinery (ACM), aprile 1997. https://doi.org/10.1145/266420.266424. (Cited on page xv.)

[20] N. Asokan, Victor Shoup e Michael Waidner: *Optimistic Fair Exchange of Digital Signatures*. Nel *Advances in Cryptology (EUROCRYPT'98)*, volume 1403 della serie *Lecture Notes in Computer Science*, pagine 591–606. Springer, maggio 1998. https://doi.org/10.1007/BFb0054156. (Cited on page xv.)

[21] Matt Franklin e Gene Tsudik: *Secure Group Barter: Multi-party Fair Exchange with Semi-trusted Neutral Parties*. Nel *Financial Cryptography (FC '98)*, volume 1465 della serie *Lecture Notes in Computer Science*, pagine 90–102. Springer, maggio 1998. https://doi.org/10.1007/BFb0055475. (Cited on page xv.)

[22] Abdullah AlOtaibi e Hamza Aldabbas: *A Review of Fair Exchange Protocols*. International Journal of Computer Networks & Communications (AIRCC), vol. 4 (no. 4): pp. 20:1–13, luglio 2012. https://doi.org/10.5121/ijcnc.2012.4420. (Cited on page xv.)

[23] Jia Ch'ng Loh, Swee Huay Heng e Syh Yuan Tan: *Fair Exchange Protocol in Electronic Transactions Revisited*. Nel *2017 5th International Conference on Information and Communication Technology (ICoIC7)*, pagine 21:1–6. Institute of Electrical and Electronics Engineers (IEEE), maggio 2017. https://doi.org/10.1109/ICoICT.2017.8074660. (Cited on page xv.)

[24] Surakarn Duangphasuk, Pruegsa Duangphasuk e Chalee Thammarat: *Fair Exchange Protocol in Electronic Transactions Revisited*. Nel *2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pagine 331–334. Institute of Electrical and Electronics Engineers (IEEE), giugno 2020. https://doi.org/10.1109/ECTI-CON49241.2020.9158264. (Cited on page xv.)

[25] Stefan Dziembowski, Lisa Eckey e Sebastian Faust: *FairSwap: How To Fairly Exchange Digital Goods*. Nel *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS '18)*, pagine 967–984. Association for Computing Machinery (ACM),

ottobre 2018. https://doi.org/10.1145/3243734.3243857. (Cited on page xv.)

[26] Sepideh Avizheh, Preston Haffey e Reihaneh Safavi-Naini: *Privacy-preserving FairSwap: Fairness and privacy interplay*. Proceedings on Privacy Enhancing Technologies (De Gruyter Open), vol. 2022 (no. 1): pp. 417–439, gennaio 2022. https://doi.org/10.2478/popets-2022-0021. (Cited on page xv.)

[27] Lisa Eckey, Sebastian Faust e Benjamin Schlosser: *OptiSwap: Fast Optimistic Fair Exchange*. Nel *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security (ASIA CCS '20)*, pagine 543–557. Association for Computing Machinery (ACM), ottobre 2020. https://doi.org/10.1145/3320269.3384749. (Cited on page xv.)

[28] Sepideh Avizheh, Preston Haffey e Reihaneh Safavi-Naini: *Privacy-enhanced OptiSwap*. Nel *Proceedings of the 2021 on Cloud Computing Security Workshop (CCSW '21)*, pagine 39–57. Association for Computing Machinery (ACM), novembre 2021. https://doi.org/10.1145/3474123.3486756. (Cited on page xv.)

[29] Matthias Lohr, Benjamin Schlosser, Jan Jürjens e Steffen Staab: *Cost Fairness for Blockchain-Based Two-Party Exchange Protocols*. Nel *2020 IEEE International Conference on Blockchain (Blockchain)*, pagine 428–435. Institute of Electrical and Electronics Engineers (IEEE), novembre 2020. https://doi.org/10.1109/Blockchain50366.2020.00062. (Cited on page xvi.)

[30] Matthias Lohr, Kenneth Skiba, Marco Konersmann, Jan Jürjens e Steffen Staab: *Formalizing Cost Fairness for Two-Party Exchange Protocols using Game Theory and Applications to Blockchain*. Nel *2022 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pagine 25:1–5. Institute of Electrical and Electronics Engineers (IEEE), maggio 2022. https://doi.org/10.1109/ICBC54727.2022.9805522. (Cited on page xvi.)

[31] Matthias Lohr, Kenneth Skiba, Marco Konersmann, Jan Jürjens e Steffen Staab: *Formalizing Cost Fairness for Two-Party Exchange Protocols using Game Theory and Applications to Blockchain (Extended Version)*. Computing Research Repository: Distributed, Parallel, and Cluster Computing (arXiv), marzo 2022. https://doi.org/10.48550/arXiv.2203.05925. (Cited on page xvi.)

[32] Simon Janin, Kaihua Qin, Akaki Mamageishvili e Arthur Gervais: *FileBounty: Fair Data Exchange*. Nel *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pagine 357–366. Institute of Electrical and Electronics Engineers (IEEE), settembre 2020. https://doi.org/10.1109/EuroSPW51379.2020.00056. (Cited on page xvi.)

[33] Sergiu Bursuc e Sjouke Mauw: *Contingent payments from two-party signing and verification for abelian groups*. Nel *2022 IEEE 35th Computer Security Foundations Symposium (CSF)*, pagine 195–210. Institute of Electrical and Electronics Engineers (IEEE), agosto 2022. https://doi.org/10.1109/CSF54842.2022.9919654. (Cited on page xvi.)

[34] Changhao Chenli, Wenyi Tang e Taeho Jung: *FairTrade: Efficient Atomic Exchange-based Fair Exchange Protocol for Digital Data Trading*. Nel *2021 IEEE International Conference on Blockchain (Blockchain)*, pagine 38–46. Institute of Electrical and Electronics Engineers (IEEE), dicembre 2021. https://doi.org/10.1109/Blockchain53845.2021.00017. (Cited on page xvi.)

[35] Justin Thaler: *Proofs, Arguments, and Zero-Knowledge*. Foundations and Trends in Privacy and Security (Now), vol. 4 (no. 2-4): pp. 117–660, dicembre 2022. https://doi.org/10.1561/3300000030. (Cited on page xvi.)

[36] Carmit Hazay e Yehuda Lindell: *Sigma Protocols and Efficient Zero-Knowledge*. Nel *Efficient Secure Two-Party Protocols: Techniques and Constructions*, capitolo 6, pagine 147–175. Springer, ottobre 2010. https://doi.org/10.1007/978-3-642-14303-8_6. (Cited on page xvi.)

[37] Dan Boneh e Victor Shoup: *Identification and signatures from Sigma protocols*. Nel *A Graduate Course in Applied Cryptography*, capitolo 19, pagine 755–822. Pubblicato su https://toc.cryptobook.us/, versione 0.6, gennaio 2023. https://crypto.stanford.edu/~dabo/cryptobook/BonehShoup_0_6.pdf. (Cited on page xvi.)

[38] Dan Boneh e Victor Shoup: *Proving properties in zero-knowledge*. Nel *A Graduate Course in Applied Cryptography*, capitolo 20, pagine 823–854. Pubblicato su https://toc.cryptobook.us/, versione 0.6, gennaio 2023. https://crypto.stanford.edu/~dabo/cryptobook/BonehShoup_0_6.pdf. (Cited on page xvi.)

[39] Feng Li e Bruce McMillin: *A Survey on Zero-Knowledge Proofs*. Nel *Advances in Computers*, volume 94, capitolo 2, pagine 25–69. Elsevier, luglio 2014. https://doi.org/10.1016/B978-0-12-800161-5.00002-5. (Cited on page xvi.)

[40] Shafi Goldwasser, Silvio Micali e Charles Rackoff: *The Knowledge Complexity of Interactive Proof-Systems*. Nel *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing (STOC '85)*, pagine 291–304. Association for Computing Machinery (ACM), dicembre 1985. https://doi.org/10.1145/22145.22178. (Cited on page xvii.)

[41] Jens Ernstberger, Stefanos Chaliasos, Liyi Zhou, Philipp Jovanovic e Arthur Gervais: *Do You Need a Zero Knowledge Proof?* Cryptology ePrint Archive, paper 2024/050, gennaio 2024. https://eprint.iacr.org/2024/050. (Cited on page xvii.)

[42] Eduardo Morais, Tommy Koens, Cees van Wijk e Aleksei Koren: *A survey on zero knowledge range proofs and applications*. SN Applied Sciences (Springer), vol. 1 (no. 8): pp. 5:1–17, luglio 2019. https://doi.org/10.1007/s42452-019-0989-z. (Cited on page xvii.)

[43] Yu Zhou, Zeming Wei, Shansi Ma e Hua Tang: *Overview of Zero-Knowledge Proof and Its Applications in Blockchain*. Nel *Blockchain Technology and Application - 5th CCF China Blockchain Conference*, volume 1736 della serie *Communications in Computer and Information Science*, pagine 60–82. Springer, dicembre 2022. https://doi.org/10.1007/978-981-19-8877-6_5. (Cited on page xvii.)

[44] Juha Partala, Tri Hong Nguyen e Susanna Pirttikangas: *Non-Interactive Zero-Knowledge for Blockchain: A Survey*. IEEE Access (Institute of Electrical and Electronics Engineers), vol. 8: pp. 227945–227961, dicembre 2020. https://doi.org/10.1109/ACCESS.2020.3046025. (Cited on page xvii.)

[45] Xiaoqiang Sun, F. Richard Yu, Peng Zhang, Zhiwei Sun, Weixin Xie e Xiang Peng: *A Survey on Zero-Knowledge Proof in Blockchain*. IEEE Network (Institute of Electrical and Electronics Engineers), vol. 35 (no. 4): pp. 198–205, agosto 2021. https://doi.org/10.1109/MNET.011.2000473. (Cited on page xvii.)

[46] Duc A. Tran, My T. Thai e Bhaskar Krishnamachari: *Handbook on Blockchain*. Springer Optimization and Its Applications. Springer,

prima edizione, novembre 2022, ISBN 9783031075353. https://doi.org/10.1007/978-3-031-07535-3. (Cited on page xvii.)

[47] Ramchandra Sharad Mangrulkar e Pallavi Vijay Chavan: *Blockchain Essentials - Core Concepts and Implementations*. Apress, gennaio 2024, ISBN 9781484299753. https://doi.org/10.1007/978-1-4842-9975-3. (Cited on page xvii.)

[48] Andreas M. Antonopoulos e Gavin Wood: *Mastering Ethereum: Building Smart Contracts and DApps*. O'Reilly, prima edizione, dicembre 2018, ISBN 9781491971949. https://github.com/ethereumbook/ethereumbook. (Cited on page xvii.)

[49] Gavin Zheng, Longxiang Gao, Liqun Huang e Jian Guan: *Ethereum Smart Contract Development in Solidity*. Springer, prima edizione, agosto 2020, ISBN 9789811562181. https://doi.org/10.1007/978-981-15-6218-1. (Cited on page xvii.)

[50] Weijia Zhang e Tej Anand: *Blockchain and Ethereum Smart Contract Solution Development - Dapp Programming with Solidity*. Apress, prima edizione, agosto 2022, ISBN 9781484281635. https://doi.org/10.1007/978-1-4842-8164-2. (Cited on page xvii.)

[51] Ritesh Modi: *Solidity Programming Essentials: A guide to building smart contracts and tokens using the widely used Solidity language*. Packt, seconda edizione, giugno 2022, ISBN 9781803231181. https://packtpub.com/en-us/product/solidity-programming-essentials-9781803231181. (Cited on page xvii.)

[52] Ethereum Foundation: *Ethereum Development Documentation*. Documentazione online. https://ethereum.org/developers/docs, consultato in data 15 gennaio 2024. (Cited on page xviii.)

[53] The Solidity Authors: *Solidity Documentation - Release 0.8.18*, febbraio 2023. https://docs.soliditylang.org/_/downloads/en/v0.8.18/pdf/. (Cited on page xviii.)

[54] Satoshi Nakamoto: *Bitcoin: A Peer-to-Peer Electronic Cash System*, 2008. https://bitcoin.org/bitcoin.pdf. (Cited on page xviii.)

[55] Vitalik Buterin: *Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform*, dicembre 2014.

https://ethereum.org/content/whitepaper/whitepaper-pdf/Ethereum_Whitepaper_-_Buterin_2014.pdf. (Cited on page xviii.)

[56] Gavin Wood: *Ethereum: A Secure Decentralised Generalised Transaction Ledger (Paris Vesion 2f36cfo)*, gennaio 2024 (la prima versione è stata pubblicata nel 2014). https://ethereum.github.io/yellowpaper/paper.pdf, al link precedente è reperibile l'ultima versione disponibile. (Cited on page xviii.)

[57] Cynthia Dwork e Moni Naor: *Pricing via Processing or Combatting Junk Mail*. Nel *Advances in Cryptology - CRYPTO '92 (12th Annual International Cryptology Conference)*, volume 740 della serie *Lecture Notes in Computer Science*, pagine 139–147. Springer, ottobre 1993. https://doi.org/10.1007/3-540-48071-4_10. (Cited on page xviii.)

[58] Nick Szabo: *Smart Contracts*, 1994. https://nakamotoinstitute.org/smart-contracts/. (Cited on page xviii.)

[59] Nick Szabo: *Formalizing and Securing Relationships on Public Networks*. First monday (First Monday Editorial Group), vol. 2 (no. 9), settembre 1997. https://doi.org/10.5210/fm.v2i9.548. (Cited on page xviii.)

[60] Wei Dai: *b-money*, novembre 1997. https://nakamotoinstitute.org/b-money/. (Cited on page xviii.)

[61] Vivek Vishnumurthy, Sangeeth Chandrakumar e Emin Gun Sirer: *KARMA : A Secure Economic Framework for Peer-to-Peer Resource Sharing*. Nel *Workshop on Economics of Peer-to-peer Systems*. University of California, Berkeley, School of Information, giugno 2003. https://groups.ischool.berkeley.edu/archive/p2pecon/papers/s5-vishnumurthy.pdf. (Cited on page xviii.)

[62] Hal Finney: *RPOW - Reusable Proofs of Work*, agosto 2004. https://nakamotoinstitute.org/finney/rpow/index.html. (Cited on page xviii.)

[63] Nick Szabo: *Bit Gold*, dicembre 2005. https://nakamotoinstitute.org/bit-gold/. (Cited on page xviii.)

[64] Xiwei Xu, Ingo Weber e Mark Staples: *Architecture for Blockchain Applications*. Springer, prima edizione, mar-

zo 2019, ISBN 9783030030353. https://doi.org/10.1007/978-3-030-03035-3. (Cited on page xix.)

[65] Xiwei Xu, Cesare Pautasso, Liming Zhu, Qinghua Lu e Ingo Weber: *A Pattern Collection for Blockchain-based Applications*. Nel *Proceedings of the 23rd European Conference on Pattern Languages of Programs (EuroPLoP '18)*, pagine 3:1–20. Association for Computing Machinery (ACM), luglio 2018. https://doi.org/10.1145/3282308.3282312. (Cited on page xix.)

[66] Yue Liu, Qinghua Lu, Xiwei Xu, Liming Zhu e Haonan Yao: *Applying Design Patterns in Smart Contracts*. Nel *Blockchain – ICBC 2018*, volume 10974 della serie *Lecture Notes in Computer Science*, pagine 92–106. Springer, giugno 2018. https://doi.org/10.1007/978-3-319-94478-4_7. (Cited on page xix.)

[67] Xiwei Xu, Ingo Weber, Mark Staples, Liming Zhu, Jan Bosch, Len Bass, Cesare Pautasso e Paul Rimba: *A Taxonomy of Blockchain-Based Systems for Architecture Design*. Nel *2017 IEEE International Conference on Software Architecture (ICSA)*, pagine 243–252. Institute of Electrical and Electronics Engineers (IEEE), aprile 2017. https://doi.org/10.1109/ICSA.2017.33. (Cited on page xix.)

[68] Maximilian Wöhrer: *Engineering Blockchain-Based Applications in the Context of the Ethereum Ecosystem*. Tesi di dottorato, Universität Wien, Faculty of Computer Science, Austria, settembre 2022. http://eprints.cs.univie.ac.at/7485/. (Cited on page xix.)

[69] Maximilian Wöhrer e Uwe Zdun: *Design Patterns for Smart Contracts in the Ethereum Ecosystem*. Nel *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pagine 1513–1520. Institute of Electrical and Electronics Engineers (IEEE), luglio 2018. https://doi.org/10.1109/Cybermatics_2018.2018.00255. (Cited on page xix.)

[70] Maximilian Wöhrer e Uwe Zdun: *Smart Contracts: Security Patterns in the Ethereum Ecosystem and Solidity*. Nel *2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE)*, pagine 2–8. Institute of Electrical and Electronics Engineers (IEEE), marzo 2018. https://doi.org/10.1109/IWBOSE.2018.8327565. (Cited on page xix.)

[71] Maximilian Wöhrer e Uwe Zdun: *Architectural Design Decisions for Blockchain-Based Applications*. Nel *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pagine 1–5. Institute of Electrical and Electronics Engineers (IEEE), maggio 2021. https://doi.org/10.1109/ICBC51069.2021.9461109. (Cited on page xix.)

[72] Maximilian Wöhrer, Uwe Zdun e Stefanie Rinderle-Ma: *Architecture Design of Blockchain-Based Applications*. Nel *2021 3rd Conference on Blockchain Research and Applications for Innovative Networks and Services (BRAINS)*, pagine 173–180. Institute of Electrical and Electronics Engineers (IEEE), settembre 2021. https://doi.org/10.1109/BRAINS52497.2021.9569813. (Cited on page xix.)

[73] Lodovica Marchesi, Michele Marchesi, Giuseppe Destefanis, Giulio Barabino e Danilo Tigano: *Design Patterns for Gas Optimization in Ethereum*. Nel *2020 IEEE International Workshop on Blockchain Oriented Software Engineering (IWBOSE)*, pagine 9–15. Institute of Electrical and Electronics Engineers (IEEE), febbraio 2020. https://doi.org/10.1109/IWBOSE50093.2020.9050163. (Cited on page xix.)

[74] Lodovica Marchesi, Michele Marchesi e Roberto Tonelli: *ABCDE–agile block chain DApp engineering*. Blockchain: Research and Applications (Elsevier), vol. 1 (no. 1): pp. 100002:1–18, dicembre 2020. https://doi.org/10.1016/j.bcra.2020.100002. (Cited on page xix.)

[75] Michele Marchesi, Lodovica Marchesi e Roberto Tonelli: *An Agile Software Engineering Method to Design Blockchain Applications*. Nel *Proceedings of the 14th Central and Eastern European Software Engineering Conference Russia (CEE-SECR '18)*, pagine 3:1–8. Association for Computing Machinery (ACM), ottobre 2018. https://doi.org/10.1145/3290621.3290627. (Cited on page xix.)

[76] Karl Wüst e Arthur Gervais: *Do you Need a Blockchain?* Nel *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, pagine 45–54. Institute of Electrical and Electronics Engineers (IEEE), giugno 2018. https://doi.org/10.1109/CVCBT.2018.00011. (Cited on page xx.)

[77] Anna Vacca, Andrea Di Sorbo, Corrado A. Visaggio e Gerardo Canfora: *A systematic literature review of blockchain and smart contract development: Techniques, tools, and open challenges*. Journal of

Systems and Software (Elsevier), vol. 174: pp. 110891:1–19, aprile 2021. https://doi.org/10.1016/j.jss.2020.110891. (Cited on page xx.)

[78] Bin Hu, Zongyang Zhang, Jianwei Liu, Yizhong Liu, Jiayuan Yin, Rongxing Lu e Xiaodong Lin: *A comprehensive survey on smart contract construction and execution: paradigms, tools, and systems*. Patterns (Cell Press), vol. 2 (no. 2): pp. 5:1–51, febbraio 2021. https://doi.org/10.1016/j.patter.2020.100179. (Cited on page xx.)

[79] Satpal Singh Kushwaha, Sandeep Joshi, Dilbag Singh, Manjit Kaur e Heung No Lee: *Ethereum Smart Contract Analysis Tools: A Systematic Review*. IEEE Access (Institute of Electrical and Electronics Engineers), vol. 10 (no.): pp. 57037–57062, aprile 2022. https://doi.org/10.1109/ACCESS.2022.3169902. (Cited on page xx.)

[80] Niclas Kannengießer, Sebastian Lins, Christian Sander, Klaus Winter, Hellmuth Frey e Ali Sunyaev: *Challenges and Common Solutions in Smart Contract Development*. IEEE Transactions on Software Engineering (Institute of Electrical and Electronics Engineers), vol. 48 (no. 11): pp. 4291–4318, novembre 2022. https://doi.org/10.1109/TSE.2021.3116808. (Cited on page xx.)

[81] Valerio Mandarino, Giuseppe Pappalardo e Emiliano Tramontana: *Some Blockchain Design Patterns for Overcoming Immutability, Chain-Boundedness, and Gas Fees*. Nel *2022 3rd Asia Conference on Computers and Communications (ACCC)*, pagine 65–71. Institute of Electrical and Electronics Engineers (IEEE), dicembre 2022. https://doi.org/10.1109/ACCC58361.2022.00018. (Cited on page xx.)

[82] Ting Chen, Zihao Li, Hao Zhou, Jiachi Chen, Xiapu Luo, Xiaoqi Li e Xiaosong Zhang: *Towards Saving Money in Using Smart Contracts*. Nel *Proceedings of the 40th International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER '18)*, pagine 81–84. Association for Computing Machinery (ACM), maggio 2018. https://doi.org/10.1145/3183399.3183420. (Cited on page xx.)

[83] Matteo Marescotti, Martin Blicha, Antti E. J. Hyvärinen, Sepideh Asadi e Natasha Sharygina: *Computing Exact Worst-Case Gas Consumption for Smart Contracts*. Nel *Leveraging Applications of Formal Methods, Verification and Validation: Industrial Practice (ISoLA 2018)*,

volume 11247 della serie *Lecture Notes in Computer Science*, pagine 450–465. Springer, novembre 2018. https://doi.org/10.1007/978-3-030-03427-6_33. (Cited on page xx.)

[84] Andrea Di Sorbo, Sonia Laudanna, Anna Vacca, Corrado A. Visaggio e Gerardo Canfora: *Profiling gas consumption in solidity smart contracts*. Journal of Systems and Software (Elsevier), vol. 186: pp. 111193:1–17, aprile 2022. https://doi.org/10.1016/j.jss.2021.111193. (Cited on page xx.)

[85] Nitima Masla, Vaibhav Vyas, Jyoti Gautam, Rabindra Nath Shaw e Ankush Ghosh: *Reduction in Gas Cost for Blockchain Enabled Smart Contract*. Nel *2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON)*, pagine 951:1–6. Institute of Electrical and Electronics Engineers (IEEE), settembre 2021. https://doi.org/10.1109/GUCON50781.2021.9573701. (Cited on page xx.)

[86] Abdeljalil Beniiche: *A Study of Blockchain Oracles*. Computing Research Repository: Cryptography and Security (arXiv), luglio 2020. https://doi.org/10.48550/arXiv.2004.07140. (Cited on page xxi.)

[87] Xiwei Xu, Ingo Weber e Mark Staples: *Blockchain Patterns*. Nel *Architecture for Blockchain Applications*, capitolo 7, pagine 113–148. Springer, prima edizione, marzo 2019. https://doi.org/10.1007/978-3-030-03035-3_7. (Cited on page xxi.)

[88] Roman Mühlberger, Stefan Bachhofner, Eduardo Castelló Ferrer, Claudio Di Ciccio, Ingo Weber, Maximilian Wöhrer e Uwe Zdun: *Foundational Oracle Patterns: Connecting Blockchain to the Off-Chain World*. Nel *Business Process Management: Blockchain and Robotic Process Automation Forum (BPM '20)*, volume 393 della serie *Lecture Notes in Business Information Processing*, pagine 35–51. Springer, settembre 2020. https://doi.org/10.1007/978-3-030-58779-6_3. (Cited on page xxi.)

[89] Xiaolong Liu, Riqing Chen, Yu Wen Chen e Shyan Ming Yuan: *Off-chain Data Fetching Architecture for Ethereum Smart Contract*. Nel *2018 International Conference on Cloud Computing, Big Data and Blockchain (ICCBB)*, pagine 15:1–4. Institute of Electrical and Electronics Engineers (IEEE), novembre 2018. https://doi.org/10.1109/ICCBB.2018.8756348. (Cited on page xxi.)

[90] Hamda Al-Breiki, Muhammad Habib Ur Rehman, Khaled Salah e Davor Svetinovic: *Trustworthy Blockchain Oracles: Review, Comparison, and Open Research Challenges*. IEEE Access (Institute of Electrical and Electronics Engineers), vol. 8 : pp. 85675–85685, maggio 2020. https://doi.org/10.1109/ACCESS.2020.2992698. (Cited on page xxi.)

[91] Ammar Hassan, Imran Makhdoom, Waseem Iqbal, Awais Ahmad e Asad Raza: *From trust to truth: Advancements in mitigating the Blockchain Oracle problem*. Journal of Network and Computer Applications (Elsevier), vol. 217: pp. 103672:1–17, agosto 2023. https://doi.org/10.1016/j.jnca.2023.103672. (Cited on page xxi.)

[92] Amirmohammad Pasdar, Young Choon Lee e Zhongli Dong: *Connect API with Blockchain: A Survey on Blockchain Oracle Implementation*. ACM Computing Surveys (Association for Computing Machinery), vol. 55 (no. 10): pp. 208:1–39, febbraio 2023. https://doi.org/10.1145/3567582. (Cited on page xxi.)

[93] Provable Things: *Provable Documentation*. Documentazione online. https://docs.provable.xyz, consultato in data 13 febbraio 2024. (Cited on page xxi.)

[94] Chainlink Foundation: *Chainlink Docs*. Documentazione online. https://docs.chain.link, consultato in data 13 febbraio 2024. (Cited on page xxi.)

[95] Lorenz Breidenbach, Christian Cachin, Benedict Chan, Alex Coventry, Steve Ellis, Ari Juels, Farinaz Koushanfar, Andrew Miller, Brendan Magauran, Daniel Moroz, Sergey Nazarov, Alexandru Topliceanu, Florian Tramèr e Fan Zhang: *Chainlink 2.0: Next Steps in the Evolution of Decentralized Oracle Networks*, aprile 2021. https://research.chain.link/whitepaper-v2.pdf. (Cited on page xxi.)

[96] Lorenz Breidenbach, Christian Cachin, Alex Coventry, Ari Juels e Andrew Miller: *Chainlink Off-chain Reporting Protocol*, febbraio 2021. https://research.chain.link/ocr.pdf. (Cited on page xxi.)

[97] Fanshu Gong, Lanju Kong, Yuxuan Lu, Jin Qian e Xinping Min: *An Overview of Blockchain Scalability for Storage*. Nel *2023 26th International Conference on Computer Supported Cooperative Work in*

Design (CSCWD), pagine 516–521. Institute of Electrical and Electronics Engineers (IEEE), maggio 2023. https://doi.org/10.1109/CSCWD57460.2023.10152720. (Cited on page xxii.)

[98] E. Sweetline Priya e R. Priya: *Performance Comparison of On-Chain and Off-Chain Data Storage Model Using Blockchain Technology*. Nel *Evolution in Computational Intelligence - Proceedings of the 11th International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA 2023)*, volume 370 della serie *Smart Innovation, Systems and Technologies*, pagine 499–511. Springer, novembre 2023. https://doi.org/10.1007/978-981-99-6702-5_41. (Cited on page xxii.)

[99] Aisyah Ismail, Mark Toohey, Young Choon Lee, Zhongli Dong e Albert Y. Zomaya: *Cost and Performance Analysis on Decentralized File Systems for Blockchain-Based Applications: State-of-the-Art Report*. Nel *2022 IEEE International Conference on Blockchain (Blockchain)*, pagine 230–237. Institute of Electrical and Electronics Engineers (IEEE), agosto 2022. https://doi.org/10.1109/Blockchain55522.2022.00039. (Cited on page xxii.)

[100] Zulwaqar Zain Mohtar, Mohd Yazid Idris e Farhan Mohamed: *Blockchain-Based Distributed File System Security and Privacy: A Systematic Mapping Study*. Nel *2022 4th International Conference on Smart Sensors and Application (ICSSA)*, pagine 64–69. Institute of Electrical and Electronics Engineers (IEEE), luglio 2022. https://doi.org/10.1109/ICSSA54161.2022.9870967. (Cited on page xxii.)

[101] Protocol Labs: *IPFS Documentation*. Documentazione online. https://docs.ipfs.tech/, consultato in data 9 febbrario 2024. (Cited on page xxii.)

[102] Juan Benet: *IPFS - Content Addressed, Versioned, P2P File System*. Computing Research Repository: Networking and Internet Architecture (arXiv), luglio 2014. https://doi.org/10.48550/arXiv.1407.3561. (Cited on page xxii.)

[103] Dennis Trautwein, Aravindh Raman, Gareth Tyson, Ignacio Castro, Will Scott, Moritz Schubotz, Bela Gipp e Yiannis Psaras: *Design and Evaluation of IPFS: a Storage Layer for the Decentralized Web*. Nel

Proceedings of the ACM SIGCOMM 2022 Conference, pagine 739–752. Association for Computing Machinery (ACM), agosto 2022. https://doi.org/10.1145/3544216.3544232. (Cited on page xxii.)

[104] Trinh Viet Doan, Yiannis Psaras, Jörg Ott e Vaibhav Bajpai: *Toward Decentralized Cloud Storage With IPFS: Opportunities, Challenges, and Future Considerations*. IEEE Internet Computing (Institute of Electrical and Electronics Engineers), vol. 26 (no. 6): pp. 7–15, novembre 2022. https://doi.org/10.1109/MIC.2022.3209804. (Cited on page xxii.)

[105] Manpreet Kaur, Shikha Gupta, Deepak Kumar, Maria Simona Raboaca, S. B. Goyal e Chaman Verma: *IPFS: An Off-Chain Storage Solution for Blockchain*. Nel *Proceedings of International Conference on Recent Innovations in Computing (ICRIC 2022)*, volume 1001 della serie *Lecture Notes in Electrical Engineering*, pagine 513–525. Springer, maggio 2023. https://doi.org/10.1007/978-981-19-9876-8_39. (Cited on page xxii.)

[106] Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith e Roderick Bloem: *Handbook of Model Checking*. Springer, prima edizione, maggio 2018, ISBN 9783319105741. https://doi.org/10.1007/978-3-319-10575-8. (Cited on page xxiii.)

[107] Armin Biere, Marijn Heule, Hans van Maaren e Toby Walsh: *Handbook of Satisfiability*, volume 336 della serie *Frontiers in Artificial Intelligence and Applications*. IOS Press, seconda edizione, aprile 2021, ISBN 9781643681603. https://doi.org/10.3233/FAIA336. (Cited on page xxiii.)

[108] Lech T. Polkowski: *Logic: Reference Book for Computer Scientists*, volume 245 della serie *Intelligent Systems Reference Library*. Springer, seconda edizione, ottobre 2023, ISBN 9783031420337. https://doi.org/10.1007/978-3-031-42034-4. (Cited on page xxiii.)

[109] Daniel Kroening: *Software Verification*. Nel *Handbook of Satisfiability*, volume 336 della serie *Frontiers in Artificial Intelligence and Applications*, capitolo 20, pagine 791–818. IOS Press, seconda edizione, aprile 2021. https://doi.org/10.3233/FAIA201004. (Cited on page xxiii.)

[110] Ranjit Jhala, Andreas Podelski e Andrey Rybalchenko: *Predicate Abstraction for Program Verification*. Nel *Handbook of Model Checking*,

capitolo 15, pagine 447–491. Springer, prima edizione, maggio 2018. https://doi.org/10.1007/978-3-319-10575-8_15. (Cited on page xxiii.)

[111] Frances E. Allen: *Control Flow Analysis*. ACM SIGPLAN Notices (Association for Computing Machinery), vol. 5 (no. 7): pp. 1–19, luglio 1970. https://doi.org/10.1145/390013.808479. (Cited on page xxiii.)

[112] Joao Marques-Silva e Sharad Malik: *Propositional SAT Solving*. Nel *Handbook of Model Checking*, capitolo 9, pagine 247–275. Springer, prima edizione, aprile 2018. https://doi.org/10.1007/978-3-319-10575-8_9. (Cited on page xxiii.)

[113] Lech T. Polkowski: *Sentential Logic (SL)*. Nel *Logic: Reference Book for Computer Scientists*, volume 245 della serie *Intelligent Systems Reference Library*, capitolo 3, pagine 61–110. Springer, seconda edizione, ottobre 2023. https://doi.org/10.1007/978-3-031-42034-4_2. (Cited on page xxiii.)

[114] Alfred Horn: *On Sentences Which are True of Direct Unions of Algebras*. The Journal of Symbolic Logic (Cambridge University Press), vol. 16 (no. 1): pp. 14–21, marzo 1951. https://doi.org/10.2307/2268661. (Cited on page xxiii.)

[115] Edmund M. Clarke: *Model checking*. Nel *Foundations of Software Technology and Theoretical Computer Science (FSTTCS 1997)*, volume 1346 della serie *Lecture Notes in Computer Science*, pagine 54–56. Springer, ottobre 1997. https://doi.org/10.1007/BFb0058022. (Cited on page xxiii.)

[116] Edmund M. Clarke e Qinsi Wang: 2^5 *Years of Model Checking*. Nel *Perspectives of System Informatics*, volume 8974 della serie *Lecture Notes in Computer Science*, pagine 26–40. Springer, giugno 2014. https://doi.org/10.1007/978-3-662-46823-4_2. (Cited on page xxiii.)

[117] Clark W. Barrett, Roberto Sebastiani, Sanjit A. Seshia e Cesare Tinelli: *Satisfiability Modulo Theories*. Nel *Handbook of Satisfiability*, volume 336 della serie *Frontiers in Artificial Intelligence and Applications*, capitolo 33, pagine 1267–1329. IOS Press, seconda edizione, aprile 2021. https://doi.org/10.3233/FAIA201017. (Cited on page xxiii.)

[118] Clark Barrett e Cesare Tinelli: *Satisfiability Modulo Theories*. Nel *Handbook of Model Checking*, capitolo 11, pagine 305–343. Springer, prima edizione, aprile 2018. https://doi.org/10.1007/978-3-319-10575-8_11. (Cited on page xxiii.)

[119] David Monniaux: *A Survey of Satisfiability Modulo Theory*. Nel *Computer Algebra in Scientific Computing (CASC 2016)*, volume 9890 della serie *Lecture Notes in Computer Science*, pagine 401–425. Springer, settembre 2016. https://doi.org/10.1007/978-3-319-45641-6_26. (Cited on page xxiii.)

[120] Leonardo de Moura, Bruno Dutertre e Natarajan Shankar: *A Tutorial on Satisfiability Modulo Theories*. Nel *Computer Aided Verification (CAV 2007)*, volume 4590 della serie *Lecture Notes in Computer Science*, pagine 20–36. Springer, luglio 2007. https://doi.org/10.1007/978-3-540-73368-3_5. (Cited on page xxiii.)

[121] Armin Biere e Daniel Kröning: *SAT-Based Model Checking*. Nel *Handbook of Model Checking*, capitolo 10, pagine 277–303. Springer, prima edizione, aprile 2018. https://doi.org/10.1007/978-3-319-10575-8_10. (Cited on page xxiv.)

[122] Armin Biere: *Bounded Model Checking*. Nel *Handbook of Satisfiability*, volume 336 della serie *Frontiers in Artificial Intelligence and Applications*, capitolo 18, pagine 739–764. IOS Press, seconda edizione, aprile 2021. https://doi.org/10.3233/FAIA201002. (Cited on page xxiv.)

[123] Edmund Clarke, Armin Biere, Richard Raimi e Yunshan Zhu: *Bounded Model Checking Using Satisfiability Solving*. Formal methods in system design (Springer), vol. 19 (no. 1): pp. 7–34, luglio 2001. https://doi.org/10.1023/A:1011276507260. (Cited on page xxiv.)

[124] Lucas Cordeiro, Bernd Fischer e Joao Marques-Silva: *SMT-Based Bounded Model Checking for Embedded ANSI-C Software*. Nel *2009 IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pagine 137–148. Institute of Electrical and Electronics Engineers (IEEE), novembre 2009. https://doi.org/10.1109/ASE.2009.63. (Cited on page xxiv.)

[125] Arie Gurfinkel: *Program Verification with Constrained Horn Clauses (Invited Paper)*. Nel *Computer Aided Verification (CAV 2022)*,

volume 13371 della serie *Lecture Notes in Computer Science*, pagine 19–29. Springer, agosto 2022. https://doi.org/10.1007/978-3-031-13185-1_2. (Cited on page xxiv.)

[126] Nikolaj Bjørner, Arie Gurfinkel, Ken McMillan e Andrey Rybalchenko: *Horn Clause Solvers for Program Verification*. Nel *Fields of Logic and Computation II*, volume 9300 della serie *Lecture Notes in Computer Science*, pagine 24–51. Springer, settembre 2015. https://doi.org/10.1007/978-3-319-23534-9_2. (Cited on page xxiv.)

[127] Emanuele De Angelis, Fabio Fioravanti, John P. Gallagher, Manuel V. Hermenegildo, Alberto Pettorossi e Maurizio Proietti: *Analysis and Transformation of Constrained Horn Clauses for Program Verification*. Theory and Practice of Logic Programming (Cambridge University Press), vol. 22 (no. 6): pp. 974–1042, novembre 2021. https://doi.org/10.1017/S1471068421000211. (Cited on page xxiv.)

[128] William F. Dowling e Jean H. Gallier: *Linear-time Algorithms for Testing the Satisfiability of Propositional Horn Formulae*. The Journal of Logic Programming (Elsevier), vol. 1 (no. 3): pp. 267–284, ottobre 1984. https://doi.org/10.1016/0743-1066(84)90014-1. (Cited on page xxiv.)

[129] Giorgio Gallo e Giampaolo Urbani: *Algorithms for Testing the Satisfiability of Propositional Formulae*. The Journal of Logic Programming (Elsevier), vol. 7 (no. 1): pp. 45–61, luglio 1989. https://doi.org/10.1016/0743-1066(89)90009-5. (Cited on page xxiv.)

[130] Ethereum Foundation: *Formal Verification of Smart Contracts*. Parte della *Ethereum Development Documentation*. https://ethereum.org/en/developers/docs/smart-contracts/formal-verification, consultato in data 15 gennaio 2024. (Cited on page xxv.)

[131] Palina Tolmach, Yi Li, Shang Wei Lin, Yang Liu e Zengxiang Li: *A Survey of Smart Contract Formal Specification and Verification*. ACM Computing Surveys (Association for Computing Machinery), vol. 54 (no. 7): pp. 148:1–38, luglio 2021. https://doi.org/10.1145/3464421. (Cited on page xxv.)

[132] Moez Krichen, Mariam Lahami e Qasem Abu Al–Haija: *Formal Methods for the Verification of Smart Contracts: A Review*. Nel *2022 15th International Conference on Security of Information and Networks (SIN)*, pagine 37–44. Institute of Electrical and Electronics Engineers

(IEEE), novembre 2022. https://doi.org/10.1109/SIN56466.2022.9970534. (Cited on page xxv.)

[133] Alexandre Mota, Fei Yang e Cristiano Teixeira: *Formally Verifying a Real World Smart Contract*. Computing Research Repository: Software Engineering (arXiv), luglio 2023. https://doi.org/10.48550/arXiv.2307.02325. (Cited on page xxv.)

[134] The Solidity Authors: *SMTChecker and Formal Verification*. Nel *Solidity Documentation - Release 0.8.18*, sezione 3.30, pagine 282–296. febbraio 2023. https://docs.soliditylang.org/_/downloads/en/v0.8.18/pdf/. (Cited on page xxv.)

[135] Rodrigo Otoni, Matteo Marescotti, Leonardo Alt, Patrick Eugster, Antti Hyvärinen e Natasha Sharygina: *A Solicitous Approach to Smart Contract Verification*. ACM Transactions on Privacy and Security (Association for Computing Machinery), vol. 26 (no. 2): pp. 15:1–28, marzo 2023. https://doi.org/10.1145/3564699. (Cited on page xxv.)

[136] Matteo Marescotti, Rodrigo Otoni, Leonardo Alt, Patrick Eugster, Antti E. J. Hyvärinen e Natasha Sharygina: *Accurate Smart Contract Verification Through Direct Modelling*. Nel *Leveraging Applications of Formal Methods, Verification and Validation: Applications (ISoLA 2020)*, volume 12478 della serie *Lecture Notes in Computer Science*, pagine 178–194. Springer, ottobre 2020. https://doi.org/10.1007/978-3-030-61467-6_12. (Cited on page xxv.)

[137] Leonardo Alt e Christian Reitwiessner: *SMT-Based Verification of Solidity Smart Contracts*. Nel *Leveraging Applications of Formal Methods, Verification and Validation: Industrial Practice (ISoLA 2018)*, volume 11247 della serie *Lecture Notes in Computer Science*, pagine 376–388. Springer, ottobre 2018. https://doi.org/10.1007/978-3-030-03427-6_28. (Cited on page xxv.)

[138] Leonardo Alt, Martin Blicha, Antti E. J. Hyvärinen e Natasha Sharygina: *SolCMC: Solidity Compiler's Model Checker*. Nel *Computer Aided Verification (CAV 2022)*, volume 13371 della serie *Lecture Notes in Computer Science*, pagine 325–338. Springer, agosto 2022. https://doi.org/10.1007/978-3-031-13185-1_16. (Cited on page xxv.)

[139] Renée McCauley, Sue Fitzgerald, Gary Lewandowski, Laurie Murphy, Beth Simon, Lynda Thomas e Carol Zander: *Debugging: a review of the literature from an educational perspective*. Computer Science Education (Routledge), vol. 18 (no. 2): pp. 67–92, giugno 2008. https://doi.org/10.1080/08993400802114581. (Cited on page xxv.)

[140] Debolina Ghosh e Jagannath Singh: *A Systematic Review on Program Debugging Techniques*. Nel *Smart Computing Paradigms: New Progresses and Challenges - Proceedings of ICACNI 2018*, volume 767 della serie *Advances in Intelligent Systems and Computing*, pagine 193–199. Springer, dicembre 2019. https://doi.org/10.1007/978-981-13-9680-9_16. (Cited on page xxv.)

[141] Nick Papoulias, Noury Bouraqadi, Luc Fabresse, Stéphane Ducasse e Marcus Denker: *Mercury: Properties and Design of a Remote Debugging Solution using Reflection*. Journal of Object Technology (AITO), vol. 14 (no. 2): pp. 1:1–36, maggio 2015. https://doi.org/10.5381/jot.2015.14.2.a1. (Cited on page xxvi.)

[142] Muhammet Oguz Ozcan, Fatih Odaci e Ismail Ari: *Remote Debugging for Containerized Applications in Edge Computing Environments*. Nel *2019 IEEE International Conference on Edge Computing (EDGE)*, pagine 30–32. Institute of Electrical and Electronics Engineers (IEEE), luglio 2019. https://doi.org/10.1109/EDGE.2019.00021. (Cited on page xxvi.)

[143] Jakob Engblom: *A review of reverse debugging*. Nel *Proceedings of the 2012 System, Software, SoC and Silicon Debug Conference*, pagine 5:1–6. Institute of Electrical and Electronics Engineers (IEEE), settembre 2012. https://ieeexplore.ieee.org/abstract/document/6338149. (Cited on page xxvi.)

[144] Anthony Savidis e Vangelis Tsiatsianas: *Implementation of Live Reverse Debugging in LLDB*. Computing Research Repository: Software Engineering (arXiv), agosto 2021. https://doi.org/10.48550/arXiv.2105.12819. (Cited on page xxvi.)

[145] Edsger W. Dijkstra: *Guarded Commands, Nondeterminacy and Formal Derivation of Programs*. Communications of the ACM (Association for Computing Machinery), vol. 18 (no. 8): pp. 453–457, agosto 1975. https://doi.org/10.1145/360933.360975. (Cited on page xxvi.)

[146] Edsger W. Dijkstra: *A Discipline of Programming*. Series in Automatic Computation. Prentice Hall, prima edizione, 1976, ISBN 97801321587l8. https://worldcat.org/oclc/01958445. (Cited on page xxvi.)

[147] David Gries: *The Science of Programming*. Monographs in Computer Science. Springer, prima edizione, febbraio 1987, ISBN 9780387964805. https://doi.org/10.1007/978-1-4612-5983-1. (Cited on page xxvi.)

[148] Edsger W. Dijkstra e Carel S. Scholten: *Predicate Calculus and Program Semantics*. Monographs in Computer Science. Springer, prima edizione, 1990, ISBN 9781461232285. https://doi.org/10.1007/978-1-4612-3228-5. (Cited on page xxvi.)

[149] Marcello M. Bonsangue e Joost N. Kok: *The Weakest Precondition Calculus: Recursion and Duality*. Formal Aspects of Computing (Springer), vol. 6 (no. 1): pp. 788–800, novembre 1994. https://doi.org/10.1007/BF01213603. (Cited on page xxvi.)

[150] Roberto Baldoni, Emilio Coppa, Daniele Cono D'elia, Camil Demetrescu e Irene Finocchi: *A Survey of Symbolic Execution Techniques*. ACM Computing Surveys (Association for Computing Machinery), vol. 51 (no. 3): pp. 50:1–39, maggio 2019. https://doi.org/10.1145/3182657. (Cited on page xxvii.)

[151] Guowei Yang, Antonio Filieri, Mateus Borges, Donato Clun e Junye Wen: *Advances in Symbolic Execution*. Volume 113 della serie *Advances in Computers*, capitolo 5, pagine 225–287. Elsevier, prima edizione, gennaio 2019. https://doi.org/10.1016/bs.adcom.2018.10.002. (Cited on page xxvii.)

[152] Corina S. Păsăreanu, Rody Kersten, Kasper Luckow e Quoc Sang Phan: *Symbolic Execution and Recent Applications to Worst-Case Execution, Load Testing, and Security Analysis*. Volume 113 della serie *Advances in Computers*, capitolo 6, pagine 289–314. Elsevier, prima edizione, gennaio 2019. https://doi.org/10.1016/bs.adcom.2018.10.004. (Cited on page xxvii.)

[153] Hui Xu, Zirui Zhao, Yangfan Zhou e Michael R. Lyu: *Benchmarking the Capability of Symbolic Execution Tools with Logic Bombs*. IEEE Transactions on Dependable and Secure Computing (Institute of Electrical and Electronics Engineers), vol. 17 (no. 6): pp. 1243–1256,

novembre-dicembre 2020. https://doi.org/10.1109/TDSC.2018.2866469. (Cited on page xxvii.)

[154] Hui Xu, Yangfan Zhou, Yu Kang e Michael R. Lyu: *Concolic Execution on Small-Size Binaries: Challenges and Empirical Study*. Nel *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pagine 181–188. Institute of Electrical and Electronics Engineers (IEEE), giugno 2017. https://doi.org/10.1109/DSN.2017.11. (Cited on page xxvii.)

[155] Sebastian Poeplau e Aurélien Francillon: *Systematic Comparison of Symbolic Execution Systems: Intermediate Representation and its Generation*. Nel *Proceedings of the 35th Annual Computer Security Applications Conference (ACSAC '19)*, pagine 163–176. Association for Computing Machinery (ACM), dicembre 2019. https://doi.org/10.1145/3359789.3359796. (Cited on page xxvii.)

[156] Frank S. de Boer e Marcello Bonsangue: *Symbolic execution formally explained*. Formal Aspects of Computing (Springer), vol. 33 (no. 4): pp. 617–636, agosto 2021. https://doi.org/10.1007/s00165-020-00527-y. (Cited on page xxvii.)

[157] The angr Project contributors: *angr: The angr Project (v9.2.90)*, febbraio 2024. https://docs.angr.io/_/downloads/en/v9.2.90/pdf/. (Cited on page xxvii.)

[158] Yan Shoshitaishvili, Ruoyu Wang, Christopher Salls, Nick Stephens, Mario Polino, Andrew Dutcher, John Grosen, Siji Feng, Christophe Hauser, Christopher Kruegel e Giovanni Vigna: *SOK: (State of) The Art of War: Offensive Techniques in Binary Analysis*. Nel *2016 IEEE Symposium on Security and Privacy (SP)*, pagine 138–157. Institute of Electrical and Electronics Engineers (IEEE), maggio 2016. https://doi.org/10.1109/SP.2016.17. (Cited on page xxvii.)

[159] Nick Stephens, Jessie Grosen, Christopher Salls, Audrey Dutcher, Ruoyu Wang, Jacopo Corbetta, Yan Shoshitaishvili, Christopher Kruegel e Giovanni Vigna: *Driller: Augmenting Fuzzing Through Selective Symbolic Execution*. Nel *Network and Distributed System Security (NDSS) Symposium 2016*. Internet Society (ISOC), febbraio 2016. https://doi.org/10.14722/NDSS.2016.23368. (Cited on page xxviii.)

[160] Yan Shoshitaishvili, Ruoyu Wang, Christophe Hauser, Christopher Kruegel e Giovanni Vigna: *Firmalice - Automatic Detection of Authentication Bypass Vulnerabilities in Binary Firmware*. Nel *Network and Distributed System Security (NDSS) Symposium 2015*. Internet Society (ISOC), febbraio 2015. https://doi.org/10.14722/NDSS.2015.23294. (Cited on page xxviii.)

[161] Jacob Springer e Wu chang Feng: *Teaching with angr: A Symbolic Execution Curriculum and CTF*. Nel *2018 USENIX Workshop on Advances in Security Education (ASE 18)*. USENIX Association, agosto 2018. https://usenix.org/conference/ase18/presentation/springer. (Cited on page xxviii.)

[162] Fabrizio Biondi, Thomas Given-Wilson, Axel Legay, Cassius Puodzius e Jean Quilbeuf: *Tutorial: An Overview of Malware Detection and Evasion Techniques*. Nel *Leveraging Applications of Formal Methods, Verification and Validation: Modeling (ISoLA 2018)*, volume 11244 della serie *Lecture Notes in Computer Science*, pagine 565–586. Springer, ottobre 2018. https://doi.org/10.1007/978-3-030-03418-4_34. (Cited on page xxviii.)

[163] Satish Chandra, Stephen J. Fink e Manu Sridharan: *Snugglebug: A Powerful Approach To Weakest Preconditions*. Nel *Proceedings of the 30th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '09)*, pagine 363–374. Association for Computing Machinery (ACM), giugno 2009. https://doi.org/10.1145/1542476.1542517. (Cited on page xxviii.)

[164] Robert Husák, Jan Kofroň e Filip Zavoral: *Handling Heap Data Structures in Backward Symbolic Execution*. Nel *Formal Methods - FM 2019 International Workshops*, volume 12233 della serie *Lecture Notes in Computer Science*, pagine 537–556. Springer, agosto 2020. https://doi.org/10.1007/978-3-030-54997-8_33. (Cited on page xxviii.)

[165] Zachary Palmer, Theodore Park, Scott Smith e Shiwei Weng: *Higher-Order Demand-Driven Symbolic Evaluation*. Proceedings of the ACM on Programming Languages (Association for Computing Machinery), vol. 4 (no. ICFP): pp. 102:1–28, agosto 2020. https://doi.org/10.1145/3408984. (Cited on page xxviii.)

[166] Marek Chalupa e Jan Strejček: *Backward Symbolic Execution with Loop Folding*. Nel *Static Analysis - 28th International Symposium (SAS*

2021), volume 12913 della serie *Lecture Notes in Computer Science*, pagine 49–76. Springer, ottobre 2021. https://doi.org/10.1007/978-3-030-88806-0_3. (Cited on page xxviii.)

[167] Alexander V. Misonizhnik, Yury O. Kostyukov, Mikhail P. Kostitsyn, Dmitry A. Mordvinov e Dmitry V. Koznov: *Generation of the weakest preconditions of programs with dynamic memory in symbolic execution*. Scientific and Technical Journal of Information Technologies, Mechanics and Optics (ITMO Univeristy), vol. 22 (no. 5): pp. 982–991, settembre-ottobre 2022. https://doi.org/10.17586/2226-1494-2022-22-5-982-991, tradotto dal russo con DeepL. (Cited on page xxviii.)

[168] Lorenzo Bettini: *Code Quality*. Nel *Test-Driven Development, Build Automation, Continuous Integration with Java, Eclipse and friends*, capitolo 15, pagine 504–524. Leanpub, febbraio 2021. https://leanpub.com/tdd-buildautomation-ci. (Cited on pages xxviii and 2.)

[169] Dimitris Mitropoulos: *Securing Software*. Nel *Encyclopedia of Computer Science and Technology*, volume 2, pagine 678–687. CRC Press, seconda edizione, ottobre 2017. https://books.google.it/books?vid=ISBN9781482208214. (Cited on pages xxviii and 2.)

INDICE ANALITICO

RINGRAZIAMENTI

<Ringraziamento 1>.

<Ringraziamento 2>.

<Ringraziamento 3>.

<Ringraziamento 4>.

<Ringraziamento 5>.