WHEN INTRUSION DETECTION MEETS BLOCKCHAIN TECHNOLOGY: A REVIEW

Seminar Report

submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

In

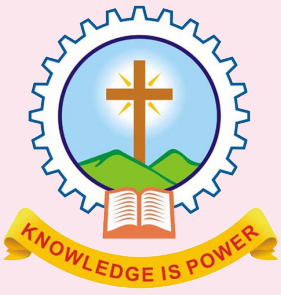
COMPUTER SCIENCE AND ENGINEERING

of

APJ Abdul Kalam Technological University

submitted by

ALKA SUSAN SLEEBA



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MAR ATHANASIUS COLLEGE OF ENGINEERING
KOTHAMANGALAM

WHEN INTRUSION DETECTION MEETS BLOCKCHAIN TECHNOLOGY: A REVIEW

Seminar Report

submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

of

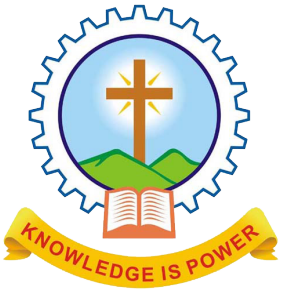
APJ Abdul Kalam Technological University

submitted by

ALKA SUSAN SLEEBA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MAR ATHANASIUS COLLEGE OF ENGINEERING
KOTHAMANGALAM

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MAR ATHANASIUS COLLEGE OF ENGINEERING
KOTHAMANGALAM



CERTIFICATE

This is to certify that the report entitled **When Intrusion Detection Meets Blockchain Technology : A Review** *submitted by* Ms. **ALKA SUSAN SLEEBA,** Reg. No. **MAC15CS009** *towards partial fulfillment of the requirement for the award of Degree of Bachelor of Technology in Computer science and Engineering Engineering from APJ Abdul Kalam Technological University for December 2018 is a bonafide record of the seminar carried out by her under our supervision and guidance.*

.....	.....	.....
Prof. Joby George	**Prof. Neethu Subash**	**Dr. Surekha Mariam Varghese**
Faculty Guide	***Faculty Guide***	***Head of the Department***

Date: Dept. Seal

ACKNOWLEDGEMENT

First and foremost, I sincerely thank the God Almighty for his grace for the successful and timely completion of the seminar.

I express my sincere gratitude and thanks to Dr. Solly George, Principal and Dr. Surekha Mariam Varghese, Head Of the Department for providing the necessary facilities and their encouragement and support.

I owe special thanks to the staff-in-charge Prof. Joby George , Prof. Neethu Subash and Prof. Joby Anu Mathew for their corrections, suggestions and sincere efforts to co-ordinate the seminar under a tight schedule.

I express my sincere thanks to staff members in the Department of Computer Science and Engineering who have taken sincere efforts in helping me to conduct this seminar.

Finally, I would like to acknowledge the heartfelt efforts, comments, criticisms, co-operation and tremendous support given to me by my dear friends during the preparation of the seminar and also during the presentation without whose support this work would have been all the more difficult to accomplish.

ABSTRACT

With the purpose of identifying cyber threats and possible incidents, intrusion detection systems (IDSs) are widely deployed in various computer networks. In order to enhance the detection capability of a single IDS collaborative intrusion detection networks (or collaborative IDSs) have been developed which allow IDS nodes to exchange data with each other. However, data and trust management still remain two challenges for current detection architectures which may degrade the effectiveness of such detection systems. In recent years, blockchain technology has shown its adaptability in many fields, such as supply chain management, international payment, interbanking, and so on. A blockchain can protect the integrity of data and ensure process transparency hence it has the potential to be applied to intrusion detection domain. A generic architecture for the incorporation of blockchains into the field of CIDSs is proposed. In particular, more on intrusion detection and blockchain, along with the applicability of blockchain to intrusion detection are introduced, and open challenges in this direction are identified.

Contents

List of Figures

List of Abbreviation

CIDN	Collaborative intrusion detection Network
HIDS	Host based intrusion detection System
IDS	Intrusion detection system
NIDS	Network based intrusion Detection System
SHA	Secure hash algorithm
TTP	Trusted third party

Introduction

Currently, cyber-attacks have become even more complicated and advanced. To help detect intrusions in a timely manner, IDSs are being widely implemented in different types of networks (e.g., education and financial organizations). Based on the deployed location [1] , an IDS can be categorized into host-based IDS (HIDS) and network-based IDS (NIDS). The former mainly monitors the characteristics of a local system and the system events in a host for malicious activities. The latter by contrast monitors network traffic and analyzes network protocols and traffic payloads for suspicious events.

Moreover, an IDS can be generally classified into two types based on the detection approaches: signature-based [2] IDS and anomaly-based IDS. The signature-based detection identifies an attack by comparing its stored signatures against observed system or network events for potential incidents. A signature (or rule) is a kind of pattern describing a known attack or exploit. The anomaly-based detection [3] finds a suspicious activity by identifying significant deviations between its pre-built normal profile and the observed events. A normal profile is often created by monitoring the characteristics of typical activity over a period of time, which can represent the normal behavior related to users, network connections and applications [4]. An alarm could be generated if an abnormal scenario is identified.

Such detection systems have proven their capability of protecting the networks they are deployed in against cyber threats. However, with the increasing number and complexity of intrusions, a single or isolated IDS turns to be ineffective in many scenarios, i.e., can be bypassed by advanced attacks [5]. Without timely detection of cyber-attacks, the whole

network is vulnerable to various damages, even the paralysis of the entire network. To enhance the detection capability of an IDS, collaborative intrusion detection systems/networks (CIDSs/CIDNs) have been designed, allowing IDS nodes to collect and exchange required information with each other. For example, by collecting traffic characteristics from different detection sensors, a central server is more sensitive to network anomalies than a single IDS.

Collaborative intrusion detection frameworks are widely adopted and deployed in various organizations due to the enhanced detection performance, but two major issues still remain: data sharing and trust computation. Firstly, data sharing is a big challenge for collaborative detection, as not all parties want to share their information explicitly.

For example, anomaly detection often employs machine learning techniques to build normal profiles, in which a classifier requires a large number of training items. Due to privacy concerns, some organizations are not willing to share their data, making the detection performance hard to optimize. Secondly, insider attacks are one big challenge for collaborative detection, which can greatly degrade the network security . Thus, how to effectively evaluate the trustworthiness of an IDS node is a challenge in a distributed and collaborative environment. For instance, with many collaborating parties, it is not an easy task to effectively measure their reputation levels.

In the literature, a central server is often used as a trusted point to help manage data sharing and trust computation for IDSs among collaborating parties, even though this server can become the weakest point for network security. To address the above challenges, new technologies are needed in the area of intrusion detection. In recent years, blockchain technology received much attention from both academia and industry due to its innovation, which allows mutually mistrusting parties to exchange financial data without the need of a trusted third party. This is the exactly desirable property for collaborative detection, which opens a chance to solve the problems regarding data sharing and trust management.

Contributions: A blockchain can be treated as a continuously growing list of records, called blocks. Each block is linked to the previous block using a cryptographic hash. Blockchains are usually managed by a peer-to-peer network, offering a transparent and integrity protected data storage (i.e., be inherently resistant to modification of the data). More

specifically, the recorded data in any given block cannot be altered retroactively without the alteration of all subsequent blocks. In such case, an attacker has to control the majority of network nodes for a successful modification, which is not realistic in terms of current network size. Blockchain technology has been initially applied to several domains like international payment , healthcare, energy , etc.

Related works

2.1 Intrusion detection

Intrusion detection describes the process of monitoring network or system events for any sign of possible incidents[7] . An IDS is an application to realize the process of intrusion detection. Basically, an IDS can provide two main functions:

- Information Recording: An IDS can monitor the target objects and record information locally. Then, the collected data can be sent to other facilities for analysis, like a central event management system.

- Alert Generation: The main task of an IDS is to generate alerts (alarms) to inform security administrators of important identified anomalies. False alarm rates are an important measurement to decide whether an IDS is effective or not.

As mentioned, an IDS can be generally classified into HIDS and NIDS, whereas such classification can be more specifc according to the deployed locations like wireless based IDS, which identifies malicious activities through monitoring wireless network packets and protocols. In practice, an IDS product often combines these two types of detection[5], as they can complement each other and provide a more thorough protection. Fig. 2.1 depicts the deployment of both HIDS and NIDS in a network environment.

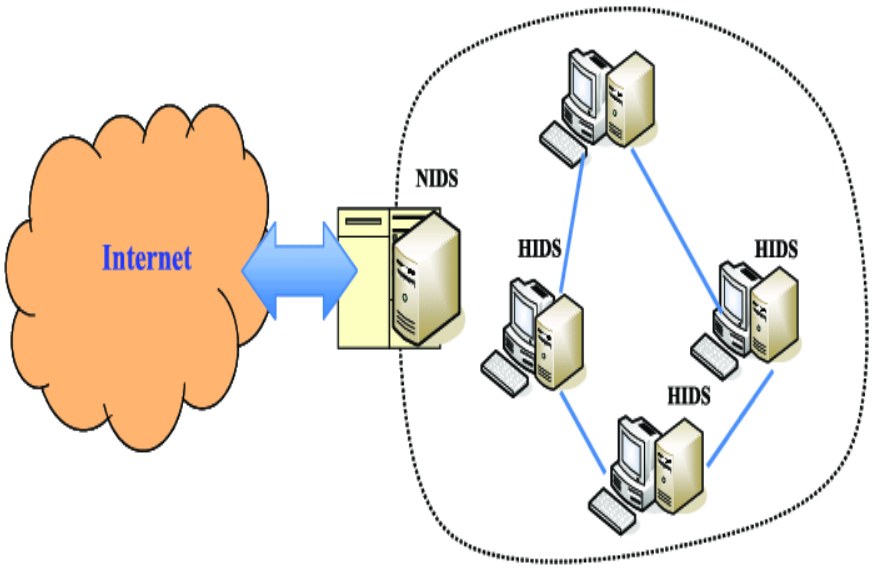


Fig. 2.1: The deployment of HIDS and NIDS in a network environment.

Based on the detection approaches, an IDS can be either a signature-based or an anomaly-based system. The signature based detection method, also called misuse-based detection, is usually effective in detecting known exploits but would be ineffective for unseen threats and the variants of known threats. For instance, given a signature that searches a filename of 'malware.exe', an attacker can write a malicious application named as 'malware1.exe' to easily bypass it.

By contrast, the anomaly-based detection has the capability of detecting unknown threats (or zero-day threats). Such detection firstly establishes a normal profile by monitoring the system or network events for a period of time, and then identifies any behavior that would be significantly different from the established profiles. In literature, various machine learning classifiers have been researched in building a normal profile. In particular, profiles can be either static or dynamic in practical usage . A static profile would not be updated while a dynamic profile would be updated periodically based on the security policies. High false rates are a big limitation for the anomaly-based detection. In addition to the above two basic detection approaches, there exists another detection method, called specification based detection, which identifies deviations between predetermined benign profile and observed events. The benign

profile is different from the normal profile in that the former defines generally accepted events in advance. For example, a benign profile can specify how particular protocols should and should not be used.

Collaborative intrusion detection

Collaborative intrusion detection including CIDNs and CIDSs were developed in practice, with the purpose of enhancing the performance of a single IDS, which may be easily bypassed by advanced or complicated attacks like denial-of-service (DoS) attack. The root cause is that an IDS usually has no information about its protected environments. While the collaborative intrusion detection framework allows various IDS nodes understanding the context by exchanging data and information with each other. Traditional collaborative systems can be classified into the following types.

- Hierarchical collaboration systems like EMERALD and DIDS .
- Subscribe collaboration systems like COSSACK and DOMINO .
- Peer-to-peer (P2P) query-based collaboration systems like Netbait and PIER .

In particular, EMERALD (Event Monitoring Enabling Responses to Anomalous Live Disturbances) was proposed by Porras et al. , which aimed to detect malicious events across a set of abstract layers in a large network. Similarly[7], DIDS (Distributed Intrusion Detection System) was designed by Snapp et al. , which identified anomalies by combining distributed monitoring, data reduction and centralized data analysis. COSSACK was developed by Papadopoulos et al. , which required no human intervention to mitigate DDoS attack intelligently.

DOMINO (Distributed Overlay for Monitoring InterNet Outbreaks) was proposed by Yegneswaran et al. , which could improve detection performance by guiding collaboration among heterogeneous nodes. PIER , as an Internet-scale query engine, could supports massively distributed and continuous queries, and could serve as a building block for a set of information centric applications.

Fig. 2.2 shows the typical architecture of a collaborative intrusion detection network. It is seen that a node, say node A, can exchange required information with each other (e.g., node B, C, and D). A node is usually composed of several components: IDS module, collaboration component, and P2P communication component. More specifically, IDS module can perform the intrusion detection functions including monitoring network traffic and recording events. The collaboration component is responsible for assisting a node to exchange required data with other nodes and conduct certain operations like trust computation. P2P communication component aims to help establish physical connection with other IDS nodes.

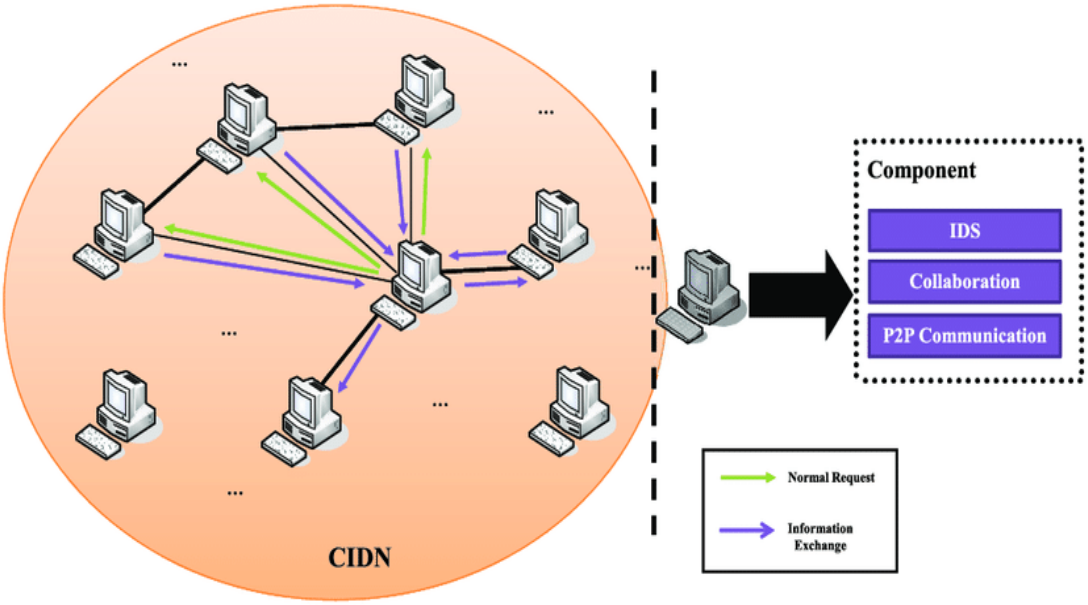


Fig. 2.2: The typical architecture of a collaborative intrusion detection network.

Building trust in collaborative intrusion detection system

Beyond the fundamental (architectural-level) research on CIDS, some work has been done lately with regards to trust management. In more detail, researchers have proposed trust management mechanisms to cope with both the insider attack problem as well as to enhance the overall quality of the collaboration. Specifically, trust management in CIDSs can be distinguished based on the overall goal of the respective (trust) mechanism. In this context, computational trust is most commonly applied to quantify the trust levels between monitoring nodes. That is, if a monitor is compromised or starts disseminating false information, its trust score will decrease and eventually some response action will take place (e.g., blacklisting). Another approach is to attempt to quantify the quality of the alert data (rather than the source of it). In such a scenario, the trust model attempts to measure the quality of the alerts themselves or assign a reputation score to certain parameters of an alert (e.g., to the IP address of an adversary). The majority of the work proposed in this area makes use of computational trust mechanisms, based on various mathematical models, to measure the trustworthiness of the monitors. In particular, the basic concept is that a monitor can utilize its own old experiences, with respect to its communication with other monitors, and via the usage of certain computational trust methods (e.g., Bayesian statistics) can infer (with a certain probability and confidence) the amount of trust it can place on others.

Requirements:

As a first step of any system design attempt, it is very important to clearly articulate the requirements of the goal system. CIDS:

- **Accountability**: Participating parties should be held accountable for their actions.
- **Integrity**: The integrity of the alert data is very important for detecting attacks over time as well as for post-mortem analysis (e.g., during forensic analysis).
- **Resilience**: The system should not have SPoFs and should not depend on small groups of participants.

- **Consensus**: The system should be able to reach consensus on the quality of individual alert data and on the trustworthiness of each participant.
- **Scalability**: The system should be able to scale to a large number of participants/monitors and also handle churn.
- **Minimum Overhead**: The communication and computation overhead should be kept as low as possible.
- **Privacy**: Participants should be able to reserve their right to privacy and selectively disclose alert data as they wish. However, at the same time, the accountability and integrity requirements should still hold.

2.2 Blockchain technology

The basic functionality provided by a block chain is a cryptographically secure mechanism for obtaining a publically verifiable and immutable sequence of records (referred to as blocks) chronogically ordered by discrete time stamps. Blockchains are typically shared and synchronized across a peer-to-peer network, and as such are typically used as a public, distributed ledger of transaction records . Every participant in the blockchain network can see the record data and reject or verify it based on a consensus protocol. Once accepted, records are appended to the blockchain in chronological order of their verification.

Cryptographic hash functions

Block chains are built upon a basic cryptographic primitive: cryptographically secure hash functions[8]. Such hash functions map an arbitrary-length input to a fixed-length n-bit output and must satisfy the following security requirements:

1. Preimage resistance: Given a hash value h, it should require $O(2^n)$ effort to compute an x such that $H(x) = h$.
2. Second preimage resistance: Given an input x and its hash value $h=H(x)$, it should require $O(2^n)$ effort to compute an $x^{'}$ not equal to x such that $H(x^{'}) = h$.

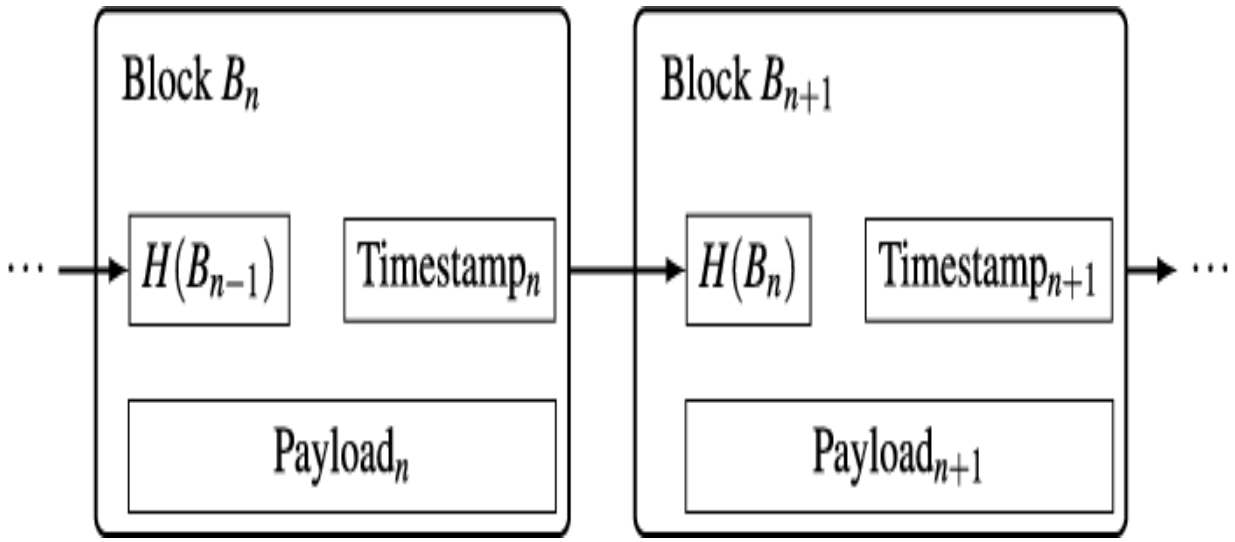


Fig. 2.3: Schematic view of a blockchain.

3. Collision resistance: It should require $O(2^{n/2})$ effort to compute any two x not equal to x' such that *H(x)*= $H(x')$

In the context of blockchains, (second) preimage resistance is of particular importance, as the ability to find second preimages with a certain midfix would enable attackers to alter existing blocks while keeping the chain intact. According to the above security requirements, such an attack should have a complexity of at least 2n for an n-bit hash function. For contemporary hash functions, we typically have n = 256 or n =512.

Fundamental properties

While the general principle of providing a secure discrete time stamping mechanism by chaining records by means of a cryptographically secure hash function was originally proposed by Haber and Stornetta in the early 1990s [7] it has only found more widespread adoption since the proposal of the Bitcoin cryptocurrency .

The exact contents of the blocks varies between different blockchain implementations. Besides the payload (containing application-specific records or transactions) a block commonly includes a timestamp and a cryptographic hash value of the entire previous block in the chain. This chaining principle is illustrated in Fig. 2.3.

The timestamp usually provides an abstract discrete notion of time in the sense that it is

monotonously increasing as the chain is extended, and does not have to be related to the time intervals between chain extensions.

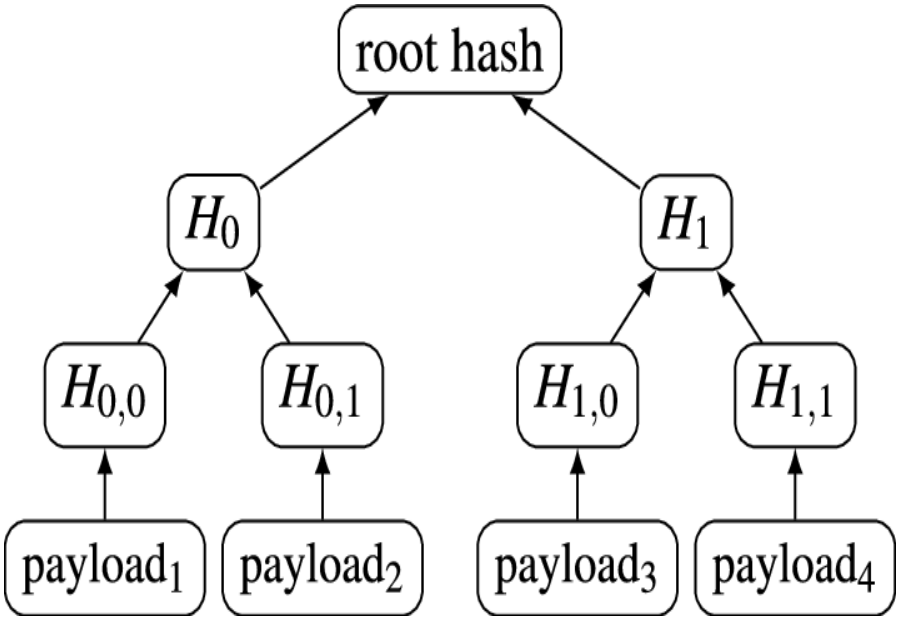


Fig. 2.4: Compact representation of four transaction payload as a Merkle tree of four hashes.

The inclusion of the hash value of block n -1 in block n makes it computationally infeasible to modify the contents of previous blocks, since it would require either finding chosen mid fix (second) preimages of the hash function or a suitable modification of all subsequent blocks. In other words, the further a blockchain has already been extended beyond block number n, the more confidence users of the blockchain can have into the integrity and immutability of this block. This chaining of hash values also makes the blockchain an append- only log of records.

To reduce the storage requirements of a blockchain, individual transactions can be hashed by means of a Merkle tree, see Fig. 2.4. The root of such a tree then serves as a compact representation of all involved payloads.

Types of blockchain

Entities can interact with a block chain either as reads or as writers. A reader is participating passively in the transaction process by reading or analysing record contents or verifying the blockchain [8]. In contrast, writers have the ability to extend the blockchain,

which typically involves participation in the consensus protocol . Block chains are then typically classified in two main categories:

Permissionless blockchains: In a permissionless or public blockchain, any entity can be free to participate as a reader or writer, and in particular also in the concensus process. Examples of permissionless blockchains include most cryptocurrencies such as Bitcoin and Zerocash , but also more general blockchains such as Ethereum .

Permissioned blockchains: For such kind of blockchains, a central entity controls the set of entities that can act as readers or writers. Consensus decisions are either taken unilaterally by this central entity, or by a pre-selected group (so-called "consortium blockchains"). Permissioned blockchains can be further categorized into public and private permissioned blockchains. They both restrict participation as writers and in the consensus process.

However, public permissioned blockchains allow anyone to read the state, while private permissioned blockchains also restrict read access. Examples of permissioned blockchains include most blockchains developed for business, in particular Hyperledger .

In the blockchain network, entities are typically identi- fied by ownership of a public/private key pair which allows them to sign transactions or any other interaction with the network. For permissioned blockchains, these keys are typically generated and certified by the central entity, which then essentially acts as a certificate authority in a public-key infrastructure (PKI).

Consensus protocol

Blockchains are intended for environments without universal trust between all participants. The availability of a reliable and universally trusted third party eliminates the need to use a blockchain [6]. Even private permissioned blockchain networks are therefore designed to incorporate a consensus process to validate transactions. We note however that even for such blockchains, a central and trusted certificate authority for the associated PKI cannot be avoided, especially given the importance of certificate revocation issues.

This lack of universal trust implies a need for a distributed consensus mechanism for block validation in blockchain networks. Such protocols can broadly be classified based on the

key property used for achieving distributed consensus:

Proof of work: In a proof of work scheme, a node in the network succeeds in having a block accepted if it can demonstrate having spent a predetermined amount of computational resources on it (hence "work"). This enforced need to spend considerable resources prevents Sybil attacks (the creation of large numbers of forged identities acting on behalf of one entity) unless a single entity controls more than half of the total computational resources in the network. A proof-of-workbased protocol based on the cryptographic hash function SHA-256 is deployed in the Bitcoin network .

Proof of stake: Consensus based on proof of stake is reached by a a combination of random selection and the wealth or influence ("stake") of the participating entities. It is based on the assumption that entities having a large stake in the blockchain have a vital interest in guaranteeing its integrity. It is used particularly in the context of cryptocurrencies such as BlackCoin or Peercoin.

Proof of elapsed time:In this variant, consensus is achieved by having every potential validator node request a secure random waiting time from a trusted execution environment which is embedded into the computing platform (such as Intel's SGX). Every node waits for the assigned time, and the first to finish claims validation leadership. Since each trusted computing environment in any node has a chance of being chosen, the probability for any entity of being in control of the validation leader is proportional to the amount of resources contributed to the overall network.

Applications of blockchain

Cryptocurrency economy is by now the most popular application of blockchain technology and also the most controversial one since it enables a multibillion-dollar global trading market of essentially anonymous transactions without government control. At the time of writing, one Bitcoin price is around dollar 10000, and Litecoin, the number one altcoin in terms of popularity and user base, has also hit an all-time high price of dollar 100. When taking the fact that there is no trusted party into consideration, these valuations are even more

remarkable. Compared to previous digital cash constructions, the blockchain based ones ingeniously combine the distributed consensus protocol, point-to-point communication and proof of work (Bitcoin) techniques to prevent double-spend attacks and remove the need for a trusted party.

Smart contracts are contracts which are automatically enforced by computer protocols featuring the same kind of agreement to act or not act without the need for trust between parties. Smart contracts were first proposed by Szabo in 1996 and with blockchain, which can be regarded as a distributed state machine without trusted third parties, can now be brought into reality. Although the functionality is limited due to a small instruction set that is not Turing complete, Bitcoin do support a small set of smart contracts. Later on, the most notable open source project Ethereum aims at providing a Turing-complete programming language to support arbitrary code execution on its blockchain, which in turn supports any kind of smart contracts.

Goyal and Goyal investigated how to use blockchains to overcome cryptographic impossibility results, where blockchain is used as an alternative to trusted setup, such as a common reference string, assumptions in cryptography to realize non-interactive zero-knowledge systems. Also, one-time programs as introduced by Goldwasser et al. can be constructed based on proof of stake based blockchain systems without relying on trusted hardware.

Proposed system

3.1 A blockchain-based architecture for collaborative intrusion detection system

As a simple example, raw alert data generated by the monitors are stored as transactions in a blockchain, replicated among the participating nodes of the network. The nodes involved, run a consensus protocol to guarantee the validity of the transactions before adding them in a block. This process guarantees that only well-formed alerts are included in the blockchain, that alert data transactions are tamperresistant, and that each participating entity has a global view of the alerts. This way, the participants are held accountable for their actions, as the latter are transparent to the network. Furthermore, the integrity of the data is guaranteed and the system has no SPoF, as it can tolerate as many byzantine failures as the underlying consensus protocol. The communication overhead of the construct can be managed e.g. by storing hashes of the alert data in the blockchain instead of the raw data. This way, a node would be able to verify the integrity of the alerts it receives by comparing their hash value with the corresponding hashes that are stored on the chain.

Design considerations

Using blockchains as the basis of the design, inherently offers a degree of accountability, integrity and resilience to our system. Nevertheless, note that the choice of the type of blockchain and consensus algorithm to use, affects the degree to which these requirements are satisfied. Other characteristics, such as the scalability of the network, the communication overhead and the privacy of the participants, might depend on the distributed ledger type (w.r.t. the implemented consensus algorithm) or the data format via which the alerts are exchanged.

Governance of distributed ledger: In general, both public (permissionless) and corporate (permissioned) blockchain designs provide authenticity, integrity and resilience to the system, via guaranteeing a global, partially ordered view of alert transactions. However, they have their own advantages and disadvantages. Public blockchains provide an uncontrolled network, which everybody can freely join, and where every peer can read from and update the distributed ledger. Generalpurpose CIDS, especially cyber incident monitors and network telescopes, e.g., can benefit considerably from these advantages. The stored alert data are integrity-protected and available to everybody, while the participants remain accountable for their actions. These properties can provide high quality data for the scientific community to examine attacks, create statistics, or gather data to train another CIDS. Nevertheless, a major shortcoming is the possible transaction cost that a peer has to pay in order for her alert to be included in a public

In consortium blockchains, access permissions are more tightly controlled and rights to modify or even read the blockchain state are restricted to a specific set of users. For example, the consensus is controlled by a pre-selected set of nodes. In this case, the validators are known and any risk of a lot of malicious peers joining the system (e.g. due to a sybil attack) and destroying the accuracy of the CIDS is mitigated. Additionally, if a peer starts to behave maliciously, e.g., starts sending fake alerts, the organization can easily change the rules of the blockchain and revert the fake transactions. This makes consortium blockchains a better choice for institutions and groups of institutions who do not want to reveal the alert data publicly and want to keep the systems participants under control.

Consensus: Apart from the issue of who is able to view and add alert data (in the form of transactions) to the blockchain, the selection of the consensus algorithm and of the peers that

take part in it, is of great importance. Especially in corporate blockchain designs, there is the possibility to choose a subset of peers, i.e., super-peers, that will be responsible for running the consensus algorithm; hence, offering integrity guarantees to the system. Apart from that, the consensus algorithm that is selected will greatly affect the security guarantees of the system. The choice of the consensus algorithm defines the adversary model of the system, i.e., the ratio of honest and malicious peers in the CIDS. Furthermore, each approach comes with a different scalability potential, e.g., practical byzantine fault tolerant designs will generally be less scalable (in terms of the peer population) than Proof-of-Work or Proof-of-Stake based ones.

Data on/off the ledger: Another question that rises is related to the alert data and their granularity during the sharing process, in each of the two communication layers (Alert Exchange and Blockchain Consensus). For this, there are various strategies that can be considered in a CIDS; each one having its own advantages and disadvantages. For instance, exchanging raw alert data can provide the deepest level of granularity. However, this comes with a major communication overhead. In addition, considering the large amount of data that a local IDS generates, such an approach would not scale. Another approach is to instead only share compact representations of alert data. For instance, in researchers have proposed the exchange of bloom filters containing such a mapping of alerts. Such approaches fulfill a number of requirements, namely the minimal overhead, privacy and integrity. Nevertheless, when only exchanging aggregated/compact versions of the alerts the accuracy of the system might be decreased.

Data encryption: The norm in both public and corporate block chain networks is that the transactions can be observed by all participating peers, a fact which could give away information that should not be revealed, e.g., sensitive corporate information. Therefore, in some CIDS usage scenarios, it is important to provide a mechanism that protects the privacy of the participating parties with respect to alert data and confidential information (exchanged in the consensus layer). One solution can be encrypting the alert data by using symmetric key cryptography and making the keys available only to the participants who should have the right to read them . This allows every peer to stay on the same network, but be able to decrypt and

examine only the alert data which they are certified to access. However, this would produce overhead in the form of key management and distribution.

Challenges in collaborative intrusion detection:

Although intrusion detection has been studied for nearly 40 years, data sharing and trust computation in a collaborative environment are still two major challenges

Data sharing: Data sharing is a major issue for a collaborative detection system, as it is not a trivial task to let all participating parties trust each other. For example, PKI technology can help build some kind of trust, but it does not always work for intrusion detection. Moreover, due to privacy concerns, some parties are not willing to share the data. Without enough data, it is unable to optimize detection algorithms and to build a robust model for identifying suspicious events.

Trust management: It is known that CIDNs/CIDSs are vulnerable to insider attacks, where the intruders have authorized access to the network. Typically, computational trust is often used to quantify the trust levels among various nodes. In practice, a central server is deployed to collect nodes' traffic and behavioral data and to compute the trust value of each node. However, the trust management would become an issue when the organization becomes large, as it is hard to find a trusted third party, i.e., central server can be compromised.

Blockchain based solutions:

By design, blockchain technology is a decentralized and distributed [8] ledger that enables recording transactions across a set of nodes. It can be implemented in a peer-to-peer network without the need of a trusted third party. The blockchain integrity can be enforced by strong cryptography, making it nearly impossible to compromise by any individual. Due to the nature of blockchains, there is a chance of applying such emerging technology for solving the above challenges in intrusion detection.

Data sharing: The data sharing problem is mainly caused by two requirements: mutual

trust and data privacy. Mutual trust means that when sharing the data, collaborating parties have to trust others who would not disclose the data. For instance, two IT organizations would like to make an agreement that they will not share the data with others. Data privacy indicates that the shared data may contain some information linked to an actual organization, i.e., the shared traffic including IP addresses and packet payloads that can be utilized to refer the privacy of an organization.

Blockchains are one of the solutions that can be used to mitigate this challenge. More specifically, data sharing can be considered as a series of transactions. Firstly, collaborating parties should make a data-sharing agreement, which digitally signed by each party.

Then, the agreement can be kept in a blockchian box, which is public and unalterable. In this case, other parties can access the blockchain box, read the agreement, and confirm the ownership of the data. Such permanent visibility of the agreement ensures that one party cannot unilaterally repudiate it. Similar to the application of blockchains in the healthcare domain , building an open accounting system is able to offer trust among various collaborating parties

For data privacy, one solution is to share transformed data instead of raw data. For example, suppose that a collaborating party (say Party A) wants to verify the performance of their designed classifier using the data from another party (say Party B). As part of the data-sharing agreement, Party A can deposit the classifier into the blockchain box, and then Party B can retrieve the classifier, run it locally with the data and send back the result to Party A. In this case, Party B actually maintains the privacy of the raw data.

On the whole, for data sharing issue, blockchains can help build mutual trust among collaborating parties and preserve data privacy by working as a permanent public ledger of contracts between data owners and other parties.

Trust computation: Generally, a collaborative network architecture can be classified as centralized, hierarchical and distributed. In literature, distributed architecture has been widely studied, while the other two are believed to suffer from scalability and an issue of single point of failure. For a CIDN, alert exchange is extremely important among various IDS nodes, which can be used to help decide whether there is an anomaly.

In addition, alert exchange can be used to compute the trustworthiness of a node within

the network. For example, Fung et al [5]. designed a type of challenge-based CIDNs, in which the trustworthiness of a node could be computed based on the satisfaction of received alert-related information. Their proposed architecture can be robust against some insider attacks like newcomer attack and Betrayal attack, but is still vulnerable to advanced collusion attack where a group of malicious peers cooperate together by providing false alarm rankings in order to compromise the network, e.g., passive message fingerprint attacks .

Therefore, how to perform trust computation in a robust way remains a challenge.

Blockchain technology provides a potential way to mitigate this issue. For instance, Alexopoulos et al. introduced a blockchain-based CIDS, which applied blockchains for enhancing trust among IDS nodes. In particular, they considered the raw alerts generated by each IDS node as transactions in a blockchain, which could be replicated among the collaborating nodes of a CIDN. Then, all collaborating nodes adopted a consensus protocol to guarantee the validity of the transactions before putting them in a block. This operation can guarantee that alerts stored in the blockchain are tamper resistant.

3.2 Scope of application for blockchain

When considering to use a blockchain in a particular application scenario, it is important to keep in mind that it might not be the most technically suitable solution. Even if an application can benefit from a blockchain, its exact type has to be appropriate.

The decision process is based upon the presence (or non-presence) of a number of elementary properties, and is outlined in Fig. 3.1.

As a first criterion, if an application does not need to store state (or data records of any kind), it clearly does not have any use for a block chain. Also, if only one entity ever writes or changes state, there is no need for record validation through consensus mechanisms, and any traditional database will offer superior performance compared to the use of blockchains.
On the other hand, the existence of multiple writers motivates the need for moderation of state updates.

This moderation can either be achieved by a universally TTP, which should ideally be always online and available for a network of many entities with multiple writers to work

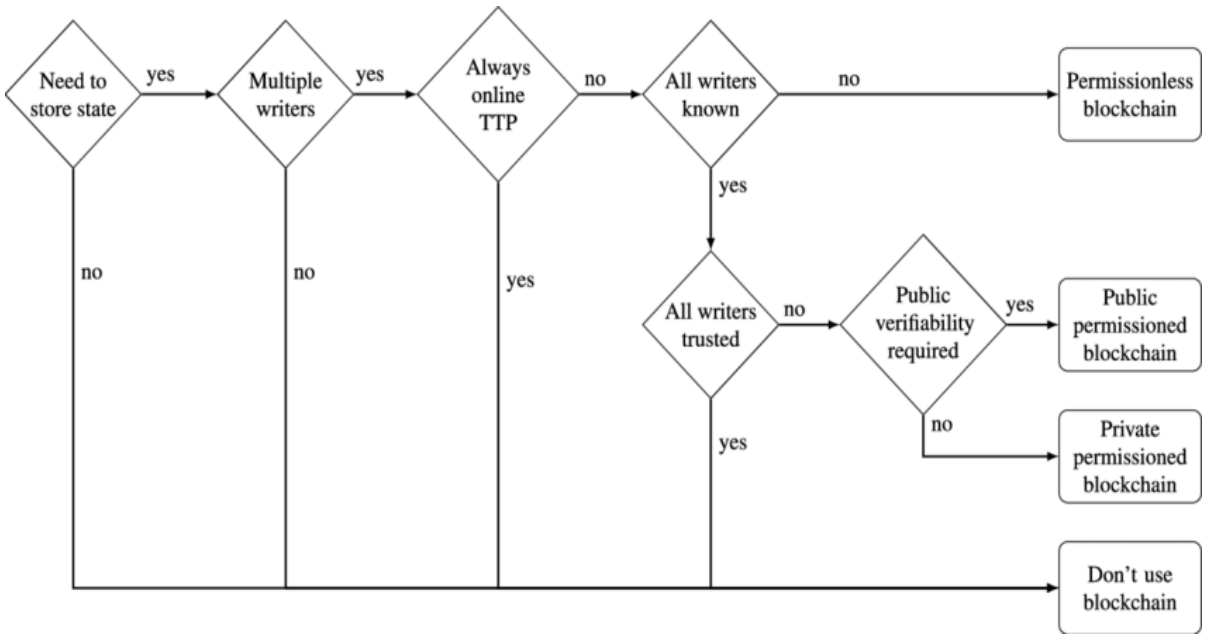


Fig. 3.1: Schematic decision diagram according to to determine whether a blockchain (and if yes, which type of blockchain) to use in which application scenario.

seamlessly. If the introduction of such a TTP is not feasible for the application scenario under consideration, a block chain might still not be required in case all writers are identified in advance and are trusted. If all writers are known but not necessarily trusted, a permissioned blockchain can be used. Whether a private or public permissioned block chain should be chosen then depends on the question whether public verifiability of the records is required. If this is the case, anyone should be admitted as readers, implying a public permissioned blockchain. Otherwise, a private permissioned blockchain is appropriate. In this context, it is important to note that even in a public permissioned or permissionless block chain, the option to use encryption or hashing to protect record contents always exists

Finally, in case the set of writers is not known in advance or can fluctuate greatly, a permissionless block chain can offer a suitable solution. This is for instance the case for crypto currencies.

Challenges and future trends

In this section, some challenges regarding the intersection of block chain technology and intrusion detection are discussed , and future directions are identified.

Challenges and limitations

Basically Block chains and intrusion detection can complement each other. On one hand, as mentioned above, block chain technology can be used to improve the performance of an IDS, especially a CIDS in the aspects of data sharing and trust computation. On the other hand, intrusion detection can help detect anomalies during block chain transactions. Pham and Lee conducted a study to apply anomaly detection as a proxy for suspicious user or event detection, which is similar to the fraud detection in credit card systems. However, each of them has some challenges and limitations remain unsolved at current stages.

Intrusion detection has been studied for a long time, but there are still many issues remained unsolved in real-world applications, which may significantly degrade the detection performance.

- **Overhead traffic with limited handling capability**: In a heavy network traffic environment, overhead packets can greatly degrade the performance of a detection system. If the traffic exceeds the maximum processing capability of an IDS, a large amount of network packets have to be discarded. For example, the computational burden is at least linear in the size of the packet payload.

- **Limited signature coverage**: The detection capability of signature-based detection depends heavily on the available signatures. In other words, the detection performance is limited to the number and quality of the deployed signatures. However, signatures are usually limited and unable to cover all known attacks and exploits.

- **Inaccurate profile establishment**: For anomaly-based detection, it is difficult to build an accurate normal pro- file due to the dynamic nature of traffic. More specifically, an

anomaly-based IDS often leverages machine learning techniques to build a profile. However, training data, especially labelled attack data, is very limited in practice, resulting in an inaccurate machine learning classifier.

- **Massive false alerts**: It is very important for an IDS to generate accurate alerts to notify security administrators about network anomalies. However, false alarms are a big challenge during detection because of immature signatures and inaccurate profiles, which may signifi- cantly degrade the detection performance and increase the workload of security analysts. For instance, a large company may generate more than 10,000 false alarms each day.

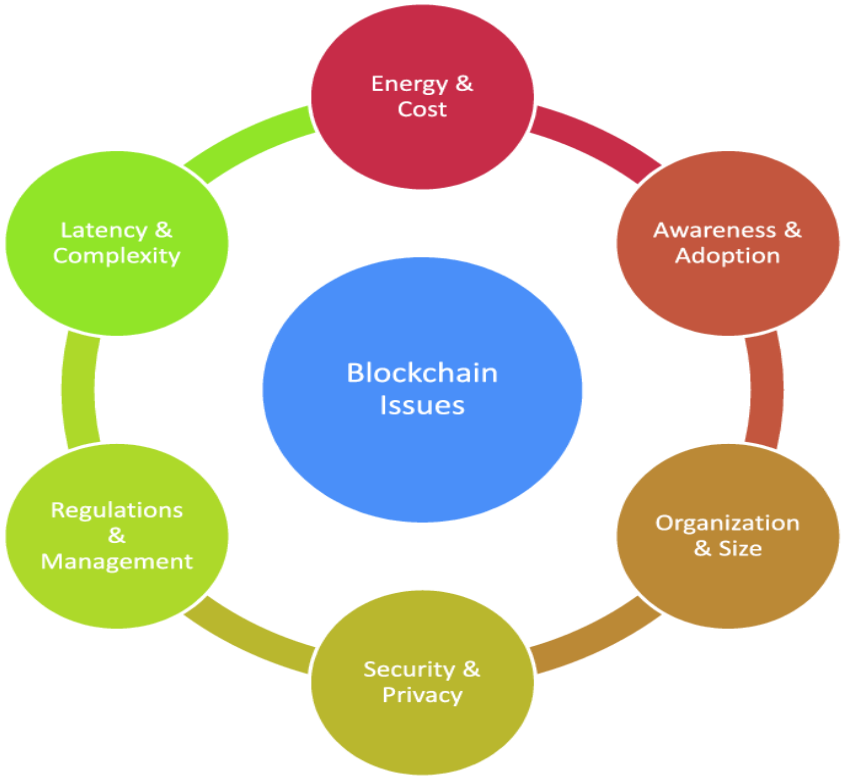


Fig. 3.2: Blockchain technology: challenges and limitations.

Block chain technology is an emerging solution, which still suffers from some inherent challenges and limitations, as summarized in fig. 3.2.

- **Energy and cost**: The computational power is a concern for block chain usage . Taking Bitcoin mining as an example, it requires a high energy level to calculate and verify

transactions. Wang and Liu found that the computational power was added on single miners at first, but could be greatly increased when the network evolved.

- **Security and privacy**: Many existing blockchain-related applications require smart transactions and contracts to be linked to known identities, which raise the privacy and security concerns of the data stored on the shared ledger. Moreover, block chain technology itself could be an attractive target for cyber-criminals, and thus suffer from various attacks like distributed denial-of-service attacks (DDoS).

- **Latency and complexity**: Due to the distributed nature, block chain-based transactions may spend several hours to finish until all parties update their corresponding ledgers. This latency would create much uncertainty for transaction participants and open a hole for cyber -criminals.

- **Awareness and adoption**: One of the major challenges regarding block chain technology is the lack of awareness and adoption. For example, many people are short of understanding of how it works. The future development of block chain depends upon how many parties adopting the technology, but now it is still a question.

- **Organization and size:** It is very likely that many different organizations would develop their own block chains and standards. With the increased size of distributed ledgers, this may greatly degrade the performance and make the block chains less efficient than current frameworks.

- **Regulations and management**: Regulations are often far behind the advanced technology. Due to the lack of common standards for completing transactions on a block chain, Bitcoin block chain has bypassed existing regulations for better efficiency. However, block chain applications are expected to work within regulations.

Future directions

As an emerging technology, block chains definitely will keep evolving, because of its disruptive capability across various industries and domains. The technology is expected to validate itself with more proof-of-concept implementations. In the field of intrusion detection, block chain technology can make positive impacts, but its major applications are more focused on the following aspects, in terms of a trade-off between benefit and cost.

- **Data sharing**: By design nature, block chains are suitable for handling the recording of events, medical records, and transaction processing. As data management is a big issue for a large distributed detection system or network, block chains have a great potential to improve the performance through enforcing trust and data privacy among collaborating parties.

- **Alert exchange**: Alexopoulos et al. already introduced how to use block chains to secure the alerts generated by various nodes and ensure only truthful alerts would be exchanged. Due to the lack of real system applications, it is an interesting and important direction for future research studies.

- **Trust computation:** As mentioned above, some collaborative detection approaches (e.g., challenge-based CIDN) utilize alerts to evaluate the trustiness of others, block chains can thus provide a solution to enhance the process of trust computation. For instance, designing block chain-based approaches to verify whether the received alert-information is unaltered or not.

Conclusion

Block chain technology is an emerging solution for decentralized transactions and data management without the need of a TTP. It is an open and distributed ledger, enabling the recording of transactions among various parties in a verifiable way. To date, block chains have been studied in several domains like health care and supply chain management, but there has been little work investigating its potential application in the field of intrusion detection.
Motivated by this observation, the paper includes the applicability of block chain technology to mitigate the challenges of data sharing and trust computation in a collaborative detection environment. It has been identified that block chains have a potential impact on the improvement of an IDS, whereas not all IDS issues can be solved with this technology.

References

[1] F. Gong, “Next generation intrusion detection systems (IDS),” McAfee Netw. Secur. Technol. Group, Santa Clara, CA, USA, White Paper, 2003

[2] M. Roesch, “Snort: Lightweight intrusion detection for networks,” in Proc. USENIX Lisa Conf., 1999, pp. 229-238

[3] A. K. Ghosh, J. Wanken, and F. Charron, “Detecting anomalous and unknown intrusions against programs,” in Proc. Annu. Comput. Secur. Appl. Conf. (ACSAC), 1998, pp. 259-267

[4] K. A. Scarfone and P. M. Mell, “Guide to Intrusion Detection and Prevention Systems (IDPS),” NIST, Gaithersburg, MD, USA, Tech. Rep. NIST SP 800-94, 2007.

[5] C. Duma, M. Karresand, N. Shahmehri, and G. Caronni, “A trust-aware, P2P-based overlay for intrusion detection,” in Proc. DEXA Workshop, 2006, pp. 697.

[6] K. Wst and A. Gervais, “Do you need a blockchain?” IACR Cryptol. ePrint Arch., 2017, p. 375

[7] I. Damgrd, “Collision free hash functions and public key signature schemes,” in Advances in Cryptology EUROCRYPT (Lecture Notes in Computer Science), vol. 304, D. Chaum andW. L. Price, Eds. Heidelberg, Germany: Springer, 1987, pp. 216

[8] R. C. Merkle, “Protocols for public key cryptosystems,” in Proc. IEEE Symp. Secur. Privacy, Oakland, CA, USA, Apr. 1980, pp. 122-134.