TRADE: A Transparent, Decentralized Traceability System for the Supply Chain

Mourad el Maouchi
Delft University of Technology
Delft, The Netherlands
mourad@elmaouchi.com

Oğuzhan Ersoy
Delft University of Technology
Delft, The Netherlands
o.ersoy@tudelft.nl

Zekeriya Erkin
Delft University of Technology
Delft, The Netherlands
z.erkin@tudelft.nl

ABSTRACT

Traceability has become an increasingly important aspect of the supply chain in the last few years due to customer awareness as well as better planning and problem identification. Unfortunately, technological, legal, and organizational concerns limit the possibility to utilize a centralized system to achieve traceability. Trust is one of the most important factors preventing the appliance of a centralized system.

Previous works provided several approaches to create a decentralized traceability system. However, these works do not state the feasibility of their work and its appliance for the supply chain. In this paper, we propose a fully transparent and decentralized traceability system for the supply chain, namely TRADE. The system leverages the actors and supply chain structure to achieve traceability. Moreover, consumers and other parties can view all the data in the system and verify the claims of actors on the products. The latter results in positive brand reputation and auditability.

Author Keywords

Blockchain; decentralized system; traceability; supply chain; transparency; auditability.

INTRODUCTION

The supply chain has experienced several highlights in the traceability aspect throughout the last few decades. Especially in the food industry, there have been severe experiences where tracing the product life cycle is crucial such as the mad cow disease and the Asian bird influenza [11, 7]. Traceability is increasing in importance every day for the actors in the supply chain to improve the performance of the business as well as compliance with (inter)national regulations. Besides the supply chain actors, other parties such as consumers, Non-Governmental Organizations (NGOs), governments, suppliers, and buyers show an increase in demand for information regarding their products and materials.

To achieve traceability, a system is required that records and follows the trail of products [2]. The interconnected nature of the supply chain makes it difficult to introduce a centralized system in control of a third party, requiring a high level of trust. The limited amount of trust resulted in separate systems, limiting the possibility to achieve traceability throughout the entire supply chain.

Blockchain technology, first introduced with Bitcoin in 2009 [18], is rapidly increasing as a key technology to address the trust aspect by removing the necessity of having a trusted third party. Blockchain technology has been successfully applied to several industries throughout the years, such as the energy [3] and finance sector [20]. For the supply chain, approaches have been suggested in a theoretical manner without sufficient analysis [14, 22, 5].

In this paper, we address the trust aspect and propose a transparent, decentralized traceability system for the supply chain. TRADE is, to the best of our knowledge, the only system that provides a fully transparent, analyzed and feasible traceability system for the supply chain. This paper is constructed as follows. First, we discuss previous works in Section 2. We present our proposed system, TRADE, in Section 3, wherein Section 4 we discuss the validation mechanisms used in our proposed system. In Section 5, we analyze the security and performance implications of TRADE, along with experimental results from a proof-of-concept implementation. Finally, a discussion and concluding remarks are provided in Section 6.

LITERATURE REVIEW

In 2016, Kim et al. proposed an ontology-based smart contract design of a proof-of-concept traceability system using blockchain technology for the supply chain [14]. Their work shows the appliance of ontologies in their setting, rather than a focus on the blockchain appliance for the supply chain and its real-world feasibility.

Furthermore, Feng Tian combined RFID tags and blockchain technology to create a traceability system for the agri-food supply chain in China [22]. Tian discussed that a decentralized approach for traceability could solve the issues in a centralized approach, namely: trust, fraud, corruption, tampering and falsifying information. In [22], the analysis discusses the blockchain technology and traceability as separate aspects. However, the combination might introduce deficiencies concerning feasibility and performance. The proposed system has also not been implemented to validate the claims.

Abeyratne et al. provided a broader view of traceability and transparency in their work [5]. In their work, transparency

ERCIM-Blockchain 2018 El Maouchi, Mourad; Ersoy, Oğuzhan and Erkin, Zekeriya (2018): TRADE: A Transparent, Decentralized Traceability System for the Supply Chain. In: W. Prinz & P. Hoschka (Eds.), Proceedings of the 1st ERCIM Blockchain Workshop 2018, Reports of the European Society for Socially Embedded Technologies (ISSN 2510-2591), DOI: 10.18420/blockchain2018_01

ACM ISBN 978-1-4503-2138-9…$15.00
DOI: 10.18420/blockchain2018_01

is argued based on the child labor scandal of Nike in 1996, whereas sustainability of products is built upon the importance of understanding the product's life-cycle [6, 8]. Abeyratne et al. discuss that the characteristics of blockchain technology can enhance trust through transparency and traceability within the supply chain. However, their work examines an example, rather than a practical appliance of blockchain technology in the supply chain.

TRADE

In this section, we design our proposed system, namely TRADE. The goal of the system is to introduce a single system for the actors to transfer product data and track products throughout the supply chain. The minimal trust between the actors makes a centralized system, in control of one party, infeasible. Therefore, we use blockchain technology as a communication network. The blockchain is an immutable record keeping system where data cannot be altered, and a product is in possession of a single actor. In the system, only authorized actors can participate and add information to the system. Nevertheless, everyone can view the stored data. The authorization to the system is handled by a central authority (CA).

This section first describes the preliminaries. Next, the system model and the accompanying actors are discussed. Lastly, the structure of transactions and the process per actor is explained.

Preliminaries

Digital signature schemes are mathematical schemes for demonstrating the authenticity of digital data or documents. Digital signatures are made possible by public-key cryptographic schemes and provide the following properties: *authentication*, *non-repudiation* and *integrity*. In our proposed system, we use the Elliptic Curve Digital Signature Algorithm (ECDSA) for the digital signatures [13], which provide smaller keys and signatures compared to RSA [12]. ECDSA is a NIST-approved digital signature algorithm [19].

System Model

TRADE consists of five actors, namely Producers, Transporters, Processors, Distributors, and Retailers. The actor types are based on the modeled supply chains in [23, 15]. We assume that the actors are distributed across different geographic locations, and they are willing to cooperate in reducing costs and improving planning algorithms by deploying a single system.

We assume that a Producer creates a product and then transports it via a Transporter to a Processor. A Processor performs internal processes on the product, which is further transported to a Distributor via a Transporter. The Distributor then distributes the end product to its final destination: a Retailer.

The actors in the system create transactions, containing product information, that is then broadcasted directly to the nodes in the network. The entire network validates the broadcasted transactions. A set of validated transactions is aggregated, by an arbitrary node, and a *block* is created, which is validated afterward. Note that we define a transaction or block as valid if it fulfills a set of requirements that are described later in Section 4. We use a public permissioned blockchain, denoted as *BC*, as a decentralized solution for TRADE, where BC_{gb} denotes the genesis block. Figure 1 depicts a schematic flow diagram of an exemplary supply chain and the data-flow in TRADE.

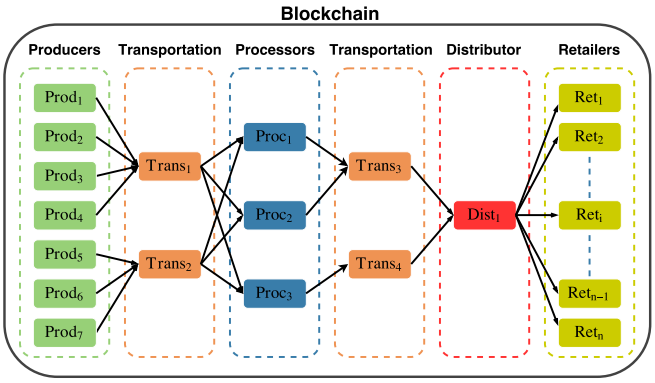

Figure 1: A schematic flow of TRADE.

System Procedures

Each transaction in TRADE is denoted as tx_h, where h denotes the hash value of the transaction. The transaction tx_h is a tuple, where $tx_h = \langle a, p_{id}, k, in, out, info, t, Sig(tx_h)\rangle$. The transaction structure and its description is shown in Table 1. We denote $x[y]$ as the element y in x and `INRF` as "if not, return false" for our proposed algorithms.

Table 1: Transaction Structure

FIELD	DESCRIPTION
a	Actor issuing the transaction.
p_{id}	Unique ID for a product.
k	Number of products.
in	Hash of the previous transaction.
out	Receiver of the transaction.
$info$	List of additional information.
t	Date and time of the transaction.
$Sig_a(tx_h)$	The signature, by a, on tx_h.

Initialization

Each actor in the network performs the key-pair generation algorithm for ECDSA. We denote a key-pair as (pk_a, sk_a), where a denotes the actor. The public key of each actor is shared with the CA. The CA is consulted in case an actor does not hold pk_a of the actor that signed the transaction. Furthermore, each actor holds a list *PID* which contains all the p_{id}'s. The list is used to check if any newly registered p_{id} is unique.

Production

A Producer, denoted as $PD_i \in PD$, creates a product with a unique ID p_{id}. Afterwards, the Producer creates the additional information $info = \{dest\}$, where *dest* is the Processor $PS_i \in PS$. Since the Producer creates a new product and accompanying p_{id}, he is unable to link it to a previous transaction and

thus links it to the genesis block BC_{gb}. The final transaction is created as $tx_h = \langle PD_i, k, p_{id}, BC_{gb}, T_j, info, t, Sig_{PD_i}(tx_h)\rangle$, where *out* is set as the Transporter $T_j \in T$. Algorithm 1 shows the validation for a tx_h by a Producer.

Algorithm 1 Transaction Validation: Producer

```
1: procedure VALIDATION_PRODUCER(tx_h)
2:     Check in = BC_gb; INRF.
3:     Check p_id ∉ PID; INRF.
4:     Check info[dest] ∈ (PS_i ∈ PS); INRF.
5:     return true
6: end procedure
```

Transportation

The Transporter, denoted as $T_i \in T$, creates a transaction tx_h when he places the product in a means of transportation. He receives a product from a Producer or Processor and transfers it to a Processor or Distributor, respectively. He sets $info = \{src, dest, V_{ID}, SSCC\}$, where *src* is the actor that provided the product, *dest* is the destination actor, V_{ID} is the vehicle ID for transportation and *SSCC* is the Serial Shipping Container Code in which the product is placed, defined by *GS1* [4]. The complete transaction is denoted as $tx_h = \langle T_i, p_{id}, k, in, out, info, t, Sig_{T_i}(tx_h)\rangle$, where *out* is either a Processor or a Distributor, based on *in*. Algorithm 2 shows the validation for a tx_h by a Transporter.

Algorithm 2 Transaction Validation: Transporter

```
1: procedure VALIDATION_TRANSPORTER(tx)
2:     Check info[src] ∈ {PD, PS}; INRF.
3:     Check info[dest] ∈ {PS, D}; INRF.
4:     Check out = info[dest]; INRF.
5:     return true
6: end procedure
```

Processing

A Processor, denoted as $PS_i \in PS$, performs internal processes on p_{id}, such as combining materials, testing or sanitizing the product, denoted as *IP*. The Processor sets $info = \{dest, IP\}$, where *dest* is the final recipient, which is a Distributor. The complete transaction is denoted as $tx_h = \langle PS_i, p_{id}, k, in, T_j, info, t, Sig_{PS_i}(tx_h)\rangle$, where *out* is set to the recipient Transporter $T_j \in T$. Algorithm 3 shows the validation for a tx_h by a Processor.

Algorithm 3 Transaction Validation: Processor

```
1: procedure VALIDATION_PROCESSOR(tx)
2:     Check info[IP] ≠ ∅; INRF.
3:     Check info[dest] ∈ D; INRF.
4:     Check out = (T_j ∈ T); INRF.
5:     return true
6: end procedure
```

Distribution

A Distributor, denoted as $D_i \in D$, creates a transaction upon distribution of p_{id}. The Distributor sets $info = \{src, V_{ID}, SSCC\}$, where *src* is the Processor that sent the product to D_i and recall the definition of V_{ID} and *SSCC* as aforementioned. The complete transaction is set up as $tx_h = \langle D_i, p_{id}, k, in, out, info, t, Sig_{D_i}(tx_h)\rangle$, where *out* is set to a Retailer $R_j \in R$. Algorithm 4 shows the validation for a tx_h by a Distributor.

Algorithm 4 Transaction Validation: Distributor

```
1: procedure VALIDATION_DISTRIBUTOR(tx)
2:     Check info[src] ∈ PS; INRF.
3:     Check out ∈ (R_j inR); INRF.
4:     return true
5: end procedure
```

Retailer

The Retailer, denoted as $R_i \in R$, is the end-actor that eventually sets the products for sale. This actor does not create a transaction. Therefore, retailers do not actively participate in the system, but rather function as an end-station for the products throughout the supply chain.

VALIDATION

Validation of Transaction Authenticity

Digital signatures are applied to prevent forgery and to proof the integrity of a transaction in TRADE. Each transaction holds a signature, made by the creator *a* using his private key sk_a. Anyone with the public key pk_a of *a* can validate the signature. The creator *a* is the only party capable of signing tx_h since he is the only one in possession of sk_a. The integrity of a transaction is held since an altered transaction results in an invalid digital signature. An invalid signature results in an invalid transaction.

Validation of Transactions

The validation of a transaction is dependent on the actor that created the transaction. Recall that a transaction is a tuple containing multiple fields, as shown in Table 1. Actors, upon receiving a transaction, need to check each field of the transaction. In Algorithm 5, we combine our previous proposed algorithms in a single algorithm to validate a transaction. Note that if one of the Check calls returns false, then the execution aborts and returns false. Therefore, in order to return true at the end, all of the Check calls must return true. The same procedure applies for the other validation processes.

Algorithm 5 Transaction Validation

```
1: procedure VALIDATION_TX(tx_h)
2:     ∀x ∈ tx_h, x ≠ null; INRF.
3:     Timestamp of tx_h < current timestamp; INRF.
4:     Validate digital signature of tx_h.
5:     Check Validation_Producer(tx_h) = true; INRF.
6:     Check Validation_Transporter(tx_h)) = true; INRF.
7:     Check Validation_Processor(tx_h) = true; INRF.
8:     Check Validation_Distributor(tx_h) = true; INRF.
9:     return true.
10: end procedure
```

Validation of Blocks

A number of transactions are collected and aggregated in a block, which is broadcasted to the network and requires validation. Note that the validation of a block is different than the validation of a transaction. The block structure is similar to the one described in Bitcoin[1]. Let b be a block and $b[TX]$ be the transaction list in b. We propose an algorithm, described in Algorithm 6, that validates a block.

Algorithm 6 Validation of a Block

```
1:  procedure BLOCK_VALIDATION(b)
2:      Check the syntactic correctness of b.
3:      Check that no duplicate of b exists.
4:      Check length of b[TX] > 1; INRF.
5:      Validate Merkle root.
6:      for each tx_i ∈ b do
7:          Check Validation_TX(tx_i) = true; INRF.
8:      end for
9:      Relay block all actors.
10:     return true.
11: end procedure
```

ANALYSES

In this section, TRADE is analyzed in three dimensions: security, performance and experimental results. Firstly, we discuss the security imposed by the system. Secondly, we provide a theoretical analysis of the performance of the computational and communication complexities. Finally, we discuss the measurements obtained from a proof-of-concept implementation to show the practical performance of the system.

Security Analysis

TRADE does not allow any unauthorized participation since it uses a public permissioned blockchain. The consensus model provides the integrity of the block structure, and the signature algorithm secures the transactions.

For TRADE, the consensus model preserves the integrity of a propagated block. TRADE does not enforce a specific consensus model. There are several models available that can be used for our system [16, 24]. The security of the blocks is thus dependent on the chosen consensus model.

TRADE uses digital signatures to provide authenticity and integrity of each transaction, where ECDSA is used as the digital signature scheme. The security of ECDSA relies on the elliptic curve discrete logarithm problem (ECDLP), which is considered to be computationally hard [13]. Therefore, the security of a digital signature, and thus the transaction, is kept under the ECDLP assumption.

The source and destination of a transaction are viewable to everyone. Also, the throughput of an actor can be derived by using the timestamp in combination with the amount in a transaction. Our proposed system does not take privacy concerns into account and thus is not envisioned to be preserved.

Note that TRADE do not tackle the problem of proving physical delivery of the products in the supply chain. We assume that the delivery can be verified by a tracking item such as RFID tags. Therefore, an malicious actor cannot claim that a false delivery or missing delivery of products because of the proof of the physical delivery mechanism.

[1]Bitcoin block structure: https://en.bitcoin.it/wiki/Block

Computational Complexity

For the analysis of the computational complexity, we list the number of operations performed by each actor in three aspects: (i) the creation of transactions, (ii) validation of transactions and (iii) the validation of blocks. The amount of performed operations depends on a number of variables, listed in Table 2.

Recall that a transaction consists of a set of values. The only computed value is the digital signature. Therefore, we focus on the computation complexity of the digital signature scheme. In Table 3, the amortized number of operations for the aforementioned aspects are listed.

Table 2: Parameters used in the computational analysis.

SYMBOL	DESCRIPTION
$\mathcal{N}$	Number of actors in the network.
γ	Number of transactions per minute, by an actor.
ℓ	Number of transactions in a block.
s	Key-size in bits for the elliptic curve.

Transaction Creation

For the creation of a transaction, a digital signature is created. The digital signature procedure is dependent on the key-size s for the chosen elliptic curve. The computational complexity, per transaction, is thus linear in s.

Transaction Validation

The computational complexity of the validation of a transaction depends on the digital signature. The validation procedure of a digital signature is, equal to the creation, dependent on the key-size s.

Block Validation

The validation of a block has the highest computational complexity. Firstly, the Merkle root is required to be validated, which requires multiple hashing operations and is computed in $\log(\ell)$ [21]. Then, each transaction is validated inside the block. The verification of ℓ digital signatures requires $s\ell$ verifications per block. Since $s\ell \gg \log(\ell)$ for $\ell > 1$, the block validation procedure is dominated by the validation procedure of digital signatures. Consequently, the block validation has a computational complexity of $\mathcal{O}(s\ell)$.

Communication Complexity

To analyze the communication complexity of TRADE, we list the number of communications required on the network for the broadcast of a transaction and a block. The required communication depends on a number of variables in Table 2.

In the initialization phase, each actor sends their public key to the CA and requires $\mathcal{N}$ communication rounds. This procedure only re-occurs if an actor updates their key-pair. The

Table 3: Computational complexity of the operations in TRADE.

PROTOCOL	ACTOR
Transaction Creation	$\mathcal{O}(s)$
Transaction Validation	$\mathcal{O}(s)$
Block Validation	$\mathcal{O}(s\ell)$

public keys of the actors are stored locally by each actor to reduce the communication rounds necessary. Next, each transaction is broadcasted to the network, which requires $\mathcal{N}-1$ rounds with the assumption that each actor knows each other and their addresses on the network allowing a direct connection. The same applies to the broadcast of blocks. Since the initialization phase only occurs at the beginning of the system, the communication complexity is dominated by the broadcast procedure for transactions and blocks. Therefore, the communication complexity of TRADE is $\mathcal{O}(\mathcal{N})$.

Experimental Results

To measure the runtime of TRADE, we created a proof-of-concept implementation of the system in `Python` 2.7 by creating a simple blockchain implementation based on the work of Daniel van Flymen[2] and the `fastecdsa` package[3]. The p_{id} values are represented as 32-bit fixed-point numbers.

The measures of the runtime were executed on our commodity hardware, running macOS 10.13 on a dual-core 3^{rd} generation 2.9GHz Intel® Core i7 processor with 16GB RAM. We measured the runtime for the transaction and validation of a transaction. For accurate measurements, we executed 1000 iterations for each procedure. We use the NIST P-curves for our measurements. Figure 2 shows the impact of s on the runtime for transaction creation and validation. It is clear that the procedures grow quadratically based on s. Using $s = 256$ for an elliptic curve, each actor is able to create approximately $\frac{1}{2.84 \cdot 10^{-3}} = 351$ transactions per second and validate transactions at a speed of $\frac{1}{2.28 \cdot 10^{-3}} = 437$ transactions per second. For the latter, an actor can validate $437/\ell$ blocks per second, depending on ℓ.

Even though our blockchain framework does not rely on a specific consensus protocol, for the experimental results, we implemented a naive Proof-of-work consensus protocol [18]. At the same time, there have been several consensus proposals which achieve 10-100x throughput of the Bitcoin's proof-of-work protocol, such as Bitcoin-NG [9], Honey Badger [17] and Algorand [10]. Therefore, it is important to note that there are consensus protocols which can securely handle the number of transactions required in our TRADE framework.

There are approximately 32.9 million shipping containers globally as of 2013 [1]. Based on the assumption that a container changes possessor up to 100 times per year, and for each time a transaction is made, approximately 317 transactions are made per second. The supply chain requires fewer transactions per second than all containers globally. Given our experimental results, it is clear that our system achieves the required performance to be applied in a real-world setting.

[2]A simple Blockchain Implementation, https://github.com/dvf/blockchain

[3]fastecds: https://pypi.python.org/pypi/fastecdsa

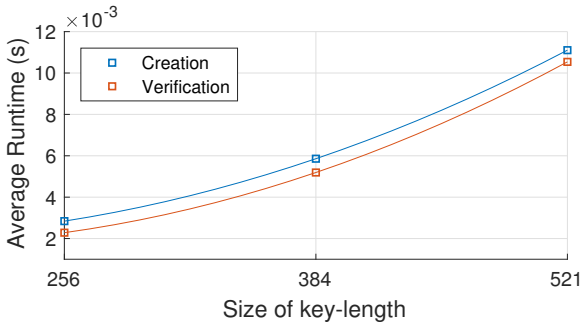

Figure 2: Average computation time for the creation and validation of a single transaction, based on s.

DISCUSSION AND CONCLUSIONS

In previous works [14, 22, 5], researchers proposed several frameworks to achieve a decentralized, traceability system. The previous works are purely theoretical, and thus do not provide any implementation. Furthermore, no complexity of the approaches is given. Therefore, the feasibility of the previous works is missing.

In this paper, we proposed TRADE, a fully transparent, decentralized traceability system for the supply chain. Each actor creates a transaction regarding a product p_{id} containing the full information on the product. The stored data inside a transaction is fully transparent allowing each actor in the network to view the data. Each transaction is signed by the issuing actor using a digital signature, providing a proof of authenticity, integrity, and non-repudiation. The valid transactions are aggregated in a block and broadcasted to the network. Each transaction regarding a product p_{id} is linked throughout the supply chain on the blockchain, providing full traceability and insight for each actor. The insight on the data can be used to improve planning and scheduling, and faster recalls for the supply chain. Also, consumers can also view this data and gain insight into the full life-cycle of products. Standardization is enforced in TRADE since each transaction, depending on the issuing actor, has a corresponding validation procedure.

TRADE achieves a significant performance to create and validate transactions, as well as the validation of blocks. We show that it is feasible to apply blockchain technology for the supply chain to achieve traceability. Moreover, consumers and other parties can view the data to gain knowledge on the procedures performed on their product as well as information on the sustainability, if the actors provide it. Actors are in control to share such information, which is recommended since it aids the company brand and increases the trust of consumers in the company. In case actors are willing to share data in a single system and achieve full traceability, blockchain technology is shown feasible to accomplish this in a real-world setting for the supply chain.

Future Work

TRADE is a generic blockchain framework for traceability systems. As being said, there are open research questions for specific use cases. An important open research question is about privacy: for some supply chain mechanisms, the actors might be competitors of each other and do not enjoy the transparent system. For these cases, privacy-preserving traceability system should be designed. Another research question is about the performance of TRADE: the naive implementation given in the paper does not provide the upper limit of the throughput of TRADE. It needs more research to investigate the performance impact of the parameters like the scripting language, consensus protocol or the blockchain platform itself. Also, performance analysis of TRADE in the existing blockchain platforms can be explored.

REFERENCES

1. 2013. World Shipping Council. (2013). http://www.worldshipping.org/about-the-industry/containers/global-container-fleet
2. 2014. ISO 9000 2015 Definitions. (Nov 2014).
3. 2017. *Blockchain X Energy, A Natural Match*. Standard. Blocklab.
4. 2017. Serial Shipping Container Code (SSCC). (Jan 2017). https://www.gs1.org/serial-shipping-container-code-sscc
5. Saveen A Abeyratne and Radmehr P Monfared. 2016. Blockchain ready manufacturing supply chain using distributed ledger. (2016).
6. Ann Baier. 2005. Organic certification process. *National Sustainable Agriculture Information Service.[Accessed 21 March 2012]* (2005).
7. W Chansud, J Wisanmongkol, and U Ketprom. 2008. RFID for poultry traceability system at animal checkpoint. In *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2008. ECTI-CON 2008. 5th International Conference on*, Vol. 2. IEEE, 753–756.
8. Sara D Elder, Hisham Zerriffi, and Philippe Le Billon. 2013. Is Fairtrade certification greening agricultural practices? An analysis of Fairtrade environmental standards in Rwanda. *Journal of Rural Studies* 32 (2013), 264–274.
9. Ittay Eyal, Adem Efe Gencer, Emin Gün Sirer, and Robbert van Renesse. 2016. Bitcoin-NG: A Scalable Blockchain Protocol. In *13th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2016, Santa Clara, CA, USA, March 16-18, 2016*. 45–59. https://www.usenix.org/conference/nsdi16/technical-sessions/presentation/eyal
10. Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. 2017. Algorand: Scaling Byzantine Agreements for Cryptocurrencies. In *Proceedings of the 26th Symposium on Operating Systems Principles, Shanghai, China, October 28-31, 2017*. 51–68. DOI:http://dx.doi.org/10.1145/3132747.3132757
11. Elise Golan, Barry Krissoff, and Fred Kuchler. 2004. Food traceability. *Amber Waves* 2, 2 (2004), 14.
12. Nicholas Jansma and Brandon Arrendondo. 2004. Performance comparison of elliptic curve and rsa digital signatures. *nicj. net/files* (2004).
13. Don Johnson, Alfred Menezes, and Scott Vanstone. 2001. The Elliptic Curve Digital Signature Algorithm (ECDSA). *International Journal of Information Security* 1, 1 (2001), 36–63.
14. Henry M Kim and Marek Laskowski. 2016. Towards an ontology-driven blockchain design for supply chain provenance. (2016).
15. Bhuwan Maharjan, Bikash Bhagat, Laxmina Shrestha, Madhu Sudan Koirala, Saroj Shrestha, and Supriya Tamrakar. Supply Chain Analysis for Bread. (????). https://www.scribd.com/doc/187747813/Supply-Chain-Management-of-Bread
16. Juri Mattila. 2013. The blockchain phenomenon. *Reuters* (2013), 1–7. http://blogs.reuters.com/felix-salmon/2013/04/09/the-disruptive-potential-of-native-advertising/
17. Andrew Miller, Yu Xia, Kyle Croman, Elaine Shi, and Dawn Song. 2016. The Honey Badger of BFT Protocols. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*. 31–42. DOI: http://dx.doi.org/10.1145/2976749.2978399
18. Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. (2008).
19. National Institute of Standards and Technology. 2013. FIPS PUB 186-4 FEDERAL: Digital Signature Standard (DSS). *Processing Standards Publication* July (2013), 1–119. http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf
20. Paweł Szewczyk. 2016. The Potential Impact of the Blockchain Technology on the Financial Sector. *Finance Today and Tomorrow: Opportunities, Threats, and Challenges* (2016), 63.
21. Michael Szydlo. 2004. Merkle tree traversal in log space and time. In *Eurocrypt*, Vol. 3027. Springer, 541–554.
22. Feng Tian. 2016. An agri-food supply chain traceability system for China based on RFID & blockchain technology. In *Service Systems and Service Management (ICSSSM), 2016 13th International Conference on*. IEEE, 1–6.
23. Sean P Willems. 2008. Data set - Real-world multiechelon supply chains used for inventory optimization. *Manufacturing & Service Operations Management* 10, 1 (2008), 19–23.
24. Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang. 2017. An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. In *Proceedings - 2017 IEEE 6th International Congress on Big Data, BigData Congress 2017*. 557–564.