Token Model and Interpretation Function for Blockchain-Based FinTech Applications

Kanta MATSUURA\(^*\), Senior Member

SUMMARY Financial Technology (FinTech) is considered a taxonomy that describes a wide range of ICT (information and communications technology) associated with financial transactions and related operations. Improvement of service quality is the main issue addressed in this taxonomy, and there are a large number of emerging technologies including blockchain-based cryptocurrencies and smart contracts. Due to its innovative nature in accounting, blockchain can also be used in lots of other FinTech contexts where token models play an important role for financial engineering. This paper revisits some of the key concepts accumulated behind this trend, and shows a generalized understanding of the technology using an adapted stochastic process. With a focus on financial instruments using blockchain, research directions toward stable applications are identified with the help of a newly proposed stabilizer: interpretation function of token valuation. The idea of adapted stochastic process is essential for the stabilizer, too.

**key words:** FinTech, timestamp, blockchain, cryptocurrency, adapted stochastic process

1. Introduction

ICT has great impacts on the financial sector, and a wide range of such technologies form FinTech [24]. Among them, blockchain is attractive due to its innovative nature in accounting: a record of consensus with a cryptographic audit trail maintained and publicly validated by multiple nodes in a distributed autonomous system [54]. The first half of this paper (Sects. 2 and 3) overviews key concepts that contribute to this trend of blockchain. Blockchain is implemented as a protocol suite, and many important details can only be found in mailing lists, forum posts, blogs, wikis, and source codes. As a result, scientific documentation of them resorts to tutorial papers. This paper’s approach is a generalized description of secure timestamping. The second half of this paper (Sects. 4 and 5) points out research directions with a description of secure timestamping. The second half of this paper (Sects. 2 and 3) overviews key concepts that contribute to this trend of blockchain. Blockchain is implemented as a protocol suite, and many important details can only be found in mailing lists, forum posts, blogs, wikis, and source codes. As a result, scientific documentation of them resorts to tutorial papers. This paper’s approach is a generalized description of secure timestamping. The second half of this paper (Sects. 4 and 5) points out research directions with a description of secure timestamping.

2. Secure Timestamping

The mechanism of blockchain is described in the Bitcoin’s white paper in 2008 [49] as follows: “The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work.” In fact, the use of hash chains for accounting purposes was well studied in secure timestamping before the birth of blockchain.

---

\(^*\) The author is with Institute of Industrial Science, The University of Tokyo, Tokyo, 153-8505 Japan.
a) E-mail: kanta@iis.u-tokyo.ac.jp
DOI: 10.1587/transfun.E102.A.3
secure timestamping is designed to ascertain whether a digital document was created at a certain point of time. Signing on or publicizing the hash value of a document $X_n$ only implies that the document was created before $t_n$. By appending the linking information, we may try to claim that the timestamp was created after the linking information was computed. An ideal feature for this purpose can be captured by a stochastic process in finance. An adapted stochastic process, or an adapted process for short, (informally) represents observable numerical values of a system randomly changing over time whose perfect prediction is impossible but whose realized values (called occurrences) are never changed. Examples include stock prices and exchange rates of currencies.

By using a process with such features, let us consider a generalized timestamping scheme described in Fig. 1 where the root hash value and the super root hash value of the $n$-th period are denoted by $RH_n$ and $SRH_n$, respectively, and the processing time required for publicizing $SRH_n$ is denoted by $\delta t_n$. One of the schemes in [12] can be viewed as an instantiation of this generalized scheme if we define a stochastic process $S(t)$ as follows.

$$ S(t) = \begin{cases} h(S_0(t)) & (t < t_1 + \delta t_1) \\ SRH_n & (t_n + \delta t_n \leq t < t_{n+1} + \delta t_{n+1} \text{ for } n \geq 1) \end{cases} $$

where $S_0(t)$ is a popular process such as an exchange rate between particular major currencies. The workflow of this scheme can be summarized as follows.

**Generation** of a timestamp for documents submitted in the $n$-th period. Not documents but hash values of them are submitted if we need to do so due to privacy requirements or bandwidth restrictions.

Based on the above considerations, two-layered binary linking schemes were proposed in 1998 [12] where two-layered means integrate-and-then-publicize and the integrated hash value is used as $g_n$ in the linking information. We will refer to the integrated hash value as a root hash value, and the hash value of the tuple as a super root hash value, in the following. The repeated use of a hash function in a tree structure was studied well for digital signature in 1980s [44], and the optimal structure for timestamping was explored around late 1990s and early 2000s [12], [36].

### 2.2 Generalization

Secure timestamping is designed to ascertain whether a digital document was created at a certain point of time. Signing on or publicizing the hash value of a document $X_n$ only implies that the document was created before $t_n$. By appending the linking information, we may try to claim that the timestamp was created after the linking information was computed. An ideal feature for this purpose can be captured by a stochastic process in finance. An adapted stochastic process, or an adapted process for short, (informally) represents observable numerical values of a system randomly changing over time whose perfect prediction is impossible but whose realized values (called occurrences) are never changed. Examples include stock prices and exchange rates of currencies.

By using a process with such features, let us consider a generalized timestamping scheme described in Fig. 1 where the root hash value and the super root hash value of the $n$-th period are denoted by $RH_n$ and $SRH_n$, respectively, and the processing time required for publicizing $SRH_n$ is denoted by $\delta t_n$. One of the schemes in [12] can be viewed as an instantiation of this generalized scheme if we define a stochastic process $S(t)$ as follows.

$$ S(t) = \begin{cases} h(S_0(t)) & (t < t_1 + \delta t_1) \\ SRH_n & (t_n + \delta t_n \leq t < t_{n+1} + \delta t_{n+1} \text{ for } n \geq 1) \end{cases} $$

where $S_0(t)$ is a popular process such as an exchange rate between particular major currencies. The workflow of this scheme can be summarized as follows.

**Generation** of a timestamp for documents submitted in the $n$-th period. Not documents but hash values of them are submitted if we need to do so due to privacy requirements or bandwidth restrictions.

---

Readers may consult financial literature (e.g. [7]) for the formal definition.
2.3 Digital Publicizing

If we use a paper-based medium, the resolution of the timestamp has a serious limitation due to the infrequency of the publicizing. In order to alleviate this problem, the use of a digital medium such as digital-TV broadcasting was proposed in 2004 [48]. The resolution realized by a paper-based medium is insufficient for many of financial transactions. Higher resolution realized by digital publicizing is much better, and blockchain deploys a witness-based approach over the Internet for this purpose as well as for the purpose of achieving a more distributed mechanism of trust.

3. Blockchain

3.1 Protocol

As a design philosophy, blockchain does not have a TSS. Therefore, we expect that someone would not only witness the submitted documents but also compute the root hash values and super root hash values. In addition, the online/offline databases of TSS are replaced with replicated and shared ledgers maintained by distributed autonomous nodes. In order to realize such a distributed trust infrastructure, we need answers at least to the following five questions:

**Question 1:** How can we distribute submitted documents among those who may play the role of witnesses?

**Question 2:** How can a verifier obtain an authentic set of data required for the verification in a timely manner?

**Question 3:** Who stores the authentic set of data?

**Question 4:** How can document submitters know that their documents are correctly processed?

**Question 5:** How can we incentivize untrusted parties to witness the submitted documents?

Let us see the answers basically in the original setup for the Bitcoin-style blockchain. In doing so, we will gradually change terminologies of timestamping schemes into those of blockchain. First, we say *transactions* instead of documents since the primary application of blockchain is a cryptocurrency. Then the answers to Questions 1–4 are the development of a peer-to-peer communication network which has the following stakeholders.

**Ledger nodes:** Distributed autonomous parties who store the authentic data regarding past transactions as a form of an append-only database. The database is called a distributed ledger, and fully replicated at the ledger nodes. The set of data appended at the same time is called a *block*. A ledger node regularly checks whether recent updates are missing by communicating with other ledger nodes, and performs missing updates if necessary.

**Transaction originators:** Those who submit transactions by broadcasting them through a ledger node in the neighborhood.

**Miners (block generators):** Those who witness and integrate submitted transactions to generate a new block as an evidence. The transactions are received from a ledger node in the neighborhood.

**Viewers:** Those who view the ledger.

The communication protocols among the above stakeholders are well summarized (though not really detailed) in survey/tutorial papers such as [2], [10], [26], and [55]. Briefly speaking, they are best-effort protocols for broadcasting, synchronization, and trouble-shooting which are similar to what we can see in the Internet protocol suite.

Finally, the answer to Question 5 is a reward. This answer raises another set of questions:

**Question 6:** What is the reward?

**Question 7:** Who will receive the reward if more than one miners do the work to receive it? That is, how can we determine the result of the competition among them?

The answer to Question 6 is simple: the reward is a certain amount of the cryptocurrency based on the blockchain itself. A miner originates a special transaction which will realize a reward, and integrates it together with the other transactions.

The answer to Questions 7 and 8 is the development of a consensus protocol where the Bitcoin implementation uses a *Proof-of-Work (PoW)* mechanism. In the PoW, an evidence of having done a cryptographic task is needed to achieve the reward. The idea of enforcing a carefully controlled or embedded workload on a client can be used to discourage attackers who try to flood a target resource. Security papers based on this idea were published in 1990s and early 2000s. Examples of such works include an anti-spam mechanism [21], plug-in or protocol countermeasures against Denial-of-Service (DoS) attacks [29], [30], and DoS-resistant key-agreement protocols [28], [39], [40]. The actual cryptographic task for the PoW in Bitcoin is a partial hash inversion used in Hashcash [3].

Figure 2 illustrates generation of a new block in the resultant blockchain mechanism. After receiving the submitted transactions in the n-th period, a miner verifies that each transaction is correctly formed, and integrates them to compute $RH_n$ by using a hash function in a tree structure. The miner then tries to find a random number $r_n$ called nonce which satisfies the following inequality:

$$SRH_n \equiv h(t_n||RH_n||S(t_n)||r_n) < T$$  \tag{1}

where $T$ is a target value of this task and controls its difficulty. In particular, for $n \geq 2$, the miner derives $t_{n-1}$, $RH_{n-1}$, $S(t_{n-1})$, and $r_{n-1}$ from the ledger, and computes $S(t_n) = h(t_{n-1}||RH_{n-1}||S(t_{n-1})||r_{n-1})$. Then the miner randomly generates $r_n$, and checks if the inequality (1) is satisfied. If unsatisfied, the miner randomly generates $r_n$ again, and repeats the same until the inequality (1) is satisfied.
A simple way of controlling the difficulty of the task is to change the target value. The change needs an additional protocol by which all of the participating nodes are notified. Variants of finer controls include a mechanism which allows different miners use different target values depending on their past achievements, present stakes, and so on [6]. Such variants contribute to finer control of incentives as well. If the flexible target values can be determined by the information included in the ledger, the notification protocol can be easier because stakeholders often view the ledger anyway.

Now that the nonce is successfully found, the miner broadcasts a request to achieve the reward through a ledger node in the neighborhood. The request includes a tuple \((n, \text{RH}_n, S(t_n), \delta_n)\) and the pointers which identify the transactions and their order used to compute \(\text{RH}_n\).

On receiving a new request, a ledger node verifies its validity; \(\text{RH}_n\) and \(S(t_n)\) must be consistent with the past blocks and the transactions in the current period, the internal verification items (e.g. the transaction originator’s signature) of each transaction must be valid, and the nonce \(n\) must satisfy the inequality (1). If everything is valid, the ledger node forwards the request to other ledger nodes. Once a certain quorum of the ledger nodes agree to accept the request, the new block is appended. This means that a certain amount of processing time (denoted by \(\delta_n\) in Fig. 2) is needed for finalizing the consensus. If the request comes from different miners, the winner is determined based on a rule (for example, the miner that processed the largest number of transactions will win). The change of \(S(t)\) is the same as that in the secure timestamping scheme.

### 3.2 Transaction

A transaction in the Bitcoin-style blockchain contains an array of inputs and an array of outputs, as well as other operational information such as version number and locktime. The locktime indicates the block number or timestamp at which the transaction is locked. The entire transaction is hashed, and the resultant hash value serves as its globally unique transaction ID.

Suppose that Alice is originating a transaction \(TX_A\) by which she wishes to spend her coins \(C_j\) whose amount is \(c_j\) \((j = 1, 2, \cdots, m)\). Suppose that Bob is going to receive a coin whose amount is \(c_0\) by \(TX_A\)\(^\dagger\). Those amounts must satisfy the following:

\[
\delta c \equiv \sum_{j=1}^{m} c_j - c_0 \geq 0. \tag{2}
\]

If unsatisfied, \(TX_A\) will be rejected; in other words, miners and ledger nodes check if the condition (2) is satisfied. If \(\delta c > 0\), then Alice can pay this amount to herself as a change. The change is embedded as one of the outputs in the current transaction \(TX_A\) itself.

Each input tells which coin is going to be spent by \(TX_A\), and each coin is identified by (i) the ID (i.e. hash value) of the past transaction which brought the coin to Alice as an output, and (ii) the hash value of her public key used as one of the recipients’ addresses\(^\dagger\) in the past transaction.

In order to avoid being spent by someone else, each transaction is signed by Alice.

Each output tells who is going to receive a coin, and how much. The recipient, Bob, is identified by an address defined as the hash value of his public key. In addition, an output has some scripts which can define a wide variety of operations associated with the transaction. The scripts enable us to design a smart contract; based on the idea of forming a protocol suite including cryptographic protocols, a smart contract was proposed in 1997 as a cryptographically enforceable agreement of a formalized workflow [53], and blockchain is a realization of this concept. The components of a smart-contract transaction may include multiple numeric values with different financial implications (e.g. unit price regarding a fee paid to a miner, amount of currency transferred by the transaction, and so on).

The construction of transactions is well summarized in papers on wallets (e.g. [4]). A wallet is a module by which a stakeholder (in the above example, Bob) receives a coin, consisting at least of a public/private key pair.

### 3.3 FinTech Applications

With the help of scripts in outputs, blockchain can be used to implement a wide variety of applications as well as cryptocurrencies [57]. In fact, based on a comprehensive survey of more than 200 blockchain startups and projects, at least seven classes of applications are recognized [22]. An updated description of the seven classes is as follows.

\(^\dagger\)Although this example considers just one recipient, multiple recipients can be accommodated in a single transaction.

\(^\dagger\)\dagger\)The idea of using a public key as a privacy-preserving address or digital pseudonym can be found in a paper in 1981 [14].
Underlying infrastructure: Underlying protocols, decentralized application ecosystems, IoT architecture, and so on (e.g. Ethereum, [11], [15], and [31]).

Currency: Payments services, internal currencies, utility tok-ens, and so on (e.g. Bitcoin and [45]).

Financial services: Asset management, investment trading, crowdfunding, and so on (e.g. [47]).

Proof-as-a-service: Notaries, registers and attestation, supply-chain management, credit management, and so on (e.g. [16], [56]).

Property and ownership: Digital rights management, copy-right and ticketing services, and so on (e.g. [32]).

Identity management: Self-sovereign digital identity, au-thentication, and so on (e.g. [20]).

Distributed Governance: Voting services, distributed au-tonomous organizations, coordinated human interac-tion, and so on (e.g. [33]).

Currencies and financial services are clearly FinTech applications. However, since fees can be embedded in a transaction, the other five classes also have an aspect of FinTech.

4. Token Models

The structure of a transaction is the key for the design of blockchain applications. Although there is no rigorous definition, each transaction output can be called a token [10], [22]. A complication of blockchain is the fact that each transaction can have multiple inputs and multiple outputs which share several components such as locktime and that each input/output intuitively represents a different coin.

In this paper, in order to have a better fit for financial engineering, a token corresponding to a transaction output is modeled by extracting and interpreting some information from the block where the transaction is integrated. This idea comes from a security token model studied in early 2000s [41]–[43] which considered a cryptographically secured and timestamped token with contents.

4.1 Security Token

In order to model not only purely financial digital materials but also general digital commodities, a security token was originally defined as follows [41].

**Definition 1** (Setok): A security token or **setok** is a digital material which contains the following attributes:

- contents which may include MAC (Message Authentication Code), digital signatures, or other security-related control sequences if necessary,
- a non-negative **explicit price** (denoted by $S$) which is paid when the setok is purchased,
- a set of non-negative **explicit values** (denoted by $V_1$, $V_2$, · · · , $V_m$ where $m$ is referred to as the dimension of the explicit values) which represent some qualities of the contents in a way that larger values of each element imply better qualities regarding the feature represented by the element when the setok is purchased, and
- a **timestamp** which indicates when the setok is purchased,

and is associated with

- a non-negative **implicit price** (denoted by $S$) and
- a set of non-negative **implicit values** (denoted by $V_1$, $V_2$, · · · , $V_n$ where $n$ is referred to as the dimension of the implicit values)

in the following way.

- The explicit price is specified as the occurrence of a **price-interpretation process** $Y(t) = y(t, S(t))$ at time $t = t_0$. Each occurrence of the price-interpretation process is called the up-to-date price at time $t$. The price-interpretation process is a non-negative process and also called the up-to-date price process. $y(t, s)$ is called a **price-interpretation function** and monotone increasing with respect to $s$.
- The explicit values are specified as the occurrences of **value-interpretation processes** $H_1(t) = h_1(t, V_1(t), V_2(t), · · · , V_n(t))$, $H_2(t) = h_2(t, V_1(t), V_2(t), · · · , V_n(t))$, · · · , $H_m(t) = h_m(t, V_1(t), V_2(t), · · · , V_n(t))$ at time $t = t_0$. Each occurrence $h_i(t, V_1(t), V_2(t), · · · , V_n(t))$ is called the $i$-th up-to-date value at time $t$. The value-interpretation processes are non-negative processes, and also called the up-to-date value processes. $h_1(t, V_1, V_2, · · · , V_n)$, · · · , $h_m(t, V_1, V_2, · · · , V_n)$ are called value-interpretation functions.

The price-interpretation function models the effects of taxes, transaction costs, regulation, and so on. The value-interpretation functions model the effects of security poli-cies, regulation, editorial policies, transmission delay, evalu-ation of stakeholders (e.g. firms), and so on. The “up-to-date” processes, $Y(t)$ and $H_i(t)$ $(i = 1, 2, · · · , m)$, are observable in the market and hence are adapted processes. Implicit processes often remind us of the world behind, whereas interpretation/up-to-date processes often remind us of the market.

The explicit values represent qualities of a setok, and depend on the implicit values. The bridge between them is the value-interpretation functions. Changes in the implicit values may be relaxed/exaggerated by interpretation.

The subsequent frameworks of setok include the def-initions of refundability, tradability, online divisibility, and offline divisibility. Their applications include pricing theo-ries regarding financial derivatives written on a setok. The pricing theories can help risk management because we can estimate risk parameters (e.g. probability of compromise) by solving inverse problems where the risk parameters are included in a pricing formula as independent variables.

4.2 Token Model for Blockchain

Inspired by the setok framework, I propose the following token model for blockchain-based FinTech applications. An
implementation will be suggested in 5.2.

**Definition 2 (Blockchain Token):** Consider a transaction output TXO \( A \) in a transaction TX \( A \) originated by Alice. Let BLC \(_M\) be the block where TX \( A \) is integrated by a miner M. A blockchain token corresponding to TXO \( A \) is a token composed of the following attributes:

- A non-negative explicit price (denoted by \( S \)) which is the amount of the coin written in TXO \( A \),
- A non-negative explicit value (denoted by \( V \)) which represents a quality (e.g. exchange rate to a popular fiat currency agreed over the blockchain network) of the coin, and
- A timestamp \( T_M \) which is indicated by the locktime of the youngest transaction in BLC \(_M\),

and is associated with

- A set of non-negative implicit values (denoted by \( V_1, V_2, \ldots, V_n \) where \( n \) is the number of the transactions in BLC \(_M\)) in the following way.

Each implicit value \( V_j \) is the \( j \)-th transaction itself. The explicit value is the occurrence of a non-negative value-interpretation process \[ H_M(t) = h_M(t, V_1, V_2, \ldots, V_n) \] at time \( t = T_M \) where \( h_M(t, v_1, v_2, \ldots, v_n) \) is called an interpretation function of values.

The interpretation function of values models the effects of implications of scripts (e.g. significance of the output, local exchange rate to a popular fiat currency, and so on), regulation, transmission delay, evaluation of stakeholders, valuation by stakeholders, and so on. Each implicit value reflects the local world where the corresponding transaction originator is active, whereas the interpretation process reflects a more global world realized by the blockchain network.

5. Research Directions

Academic researches of blockchain are still in their infancy, and hence there are so many immature research items. Rather than showing exhaustive list of such items, this section is focused on two research directions which are important particularly from the viewpoint of FinTech. Therefore, some major research items (e.g. scalability [37] and Sybil attacks [55]) are lacking here.

5.1 Empirical Analysis

In the economics of information security [1], it is usually hard to find rich empirical data for rigorous analysis. However, in principle, there is a full record of all public transactions in the history of Bitcoin. In addition, freely available datasets include the followings:

- Bitcoin price index (exchange rate between the US dollar and the Bitcoin),
- Total Bitcoins in circulation,
- Number of transactions excluding exchange transactions,
- Estimated output volume,
- Ratio of trade volume to transaction volume,
- Difficulty of the PoW.

This situation suggests that there is a good chance of realizing deep empirical economic analyses regarding Bitcoin and possibly many of other alternative currencies (altcoins) which resemble Bitcoin; in fact, there have been published a lot of literature that show empirical analyses by using such rich datasets (e.g. [25], [34], [46]).

In blockchain, any stakeholder may misbehave. Double-spending attacks are by transaction originators, and malleability attacks are by recipients [18]. In addition, miners should be carefully marked since they may exploit the competition settlement mechanism in the consensus protocol and eventually replace an existing candidate of a new block (possibly created by themselves) with their selfish request [5]. Such malicious behaviors of miners imply that there is something fundamentally broken in the protocol’s incentive structure [51], and empirical data can help us evaluate the effects of countermeasures.

5.2 Financial Engineering

Bitcoin’s dollar value has been showing an extremely high volatility since its early years [59], which is considered one of the weakness origins of an ecosystem [58]. Volatility is a fluctuation measure of financial securities, and traditionally, it is shown that introducing financial derivatives can reduce the volatility of their underlying asset by promoting information dissemination and collection [13], [17], [19], [35]. In this regard, it is worth considering derivatives written on cryptocurrencies.

However, recent research trends include blockchain-based implementation of common derivatives [8], and the current state of the art is to tackle technical developments. We will next face financial risk management problems even after solving such development problems. In fact, a recent paper of financial derivatives using blockchain [23] mentions the followings:

- “In a derivative where settlement is fully automated, either the underlying security (or a token representing it) needs to be on the blockchain already or the blockchain needs to be able to assign a value to the security.”
- “For derivatives on cryptocurrencies or more esoteric securities, models simply do not exist yet and are an open area of research.”

The first sentence points out the necessity of a comprehensive marketplace based on blockchain, and there is a
view that its possibility is very likely [52]. Thus research efforts can be shared by those who have concerns about instability of cryptocurrencies and those who are interested in a decentralized e-marketplace of complex financial contracts.

The second sentence is included in a paragraph which states that the Nobel-awarded Black-Scholes model [9] had been influential in traditional markets with respect to financial risk management. The Black-Scholes model is based on stochastic theories using adapted processes. The se-
tok framework [41]–[43] can be considered a corresponding model for an e-marketplace of digital commodities where we can assume adapted processes. The token model proposed in 4.2 may help thus upcoming research trends of financial engineering if the interpretation function is implemented in a way that the value-interpretation process can be regarded as an adapted process. Here is an example of starting points toward this direction.

Suppose that underlying cryptocurrencies may have very different exchange rates at different places. This situation may make it hard to use validation by a single stake-
holder for stability and reliability reasons. In order to solve this problem, let us consider the following implementation of the interpretation function.

1. Transaction originators append their valuation of the token value to their transactions. Let $u_j$ be the valuation result by the $j$-th originator ($j = 1, 2, \cdots, n$).

2. A miner uses $u_1, u_2, \cdots, u_n$ as main inputs to the interpretation function of values. It may simply take the average. It may consider a weighted average based on other information such as scripts and locktime. The resultant explicit value is appended to the block. In this implementation, we use an interpretation function which does not depend on the miner. Therefore, in the following, the value-interpretation process is denoted not by $H_M(t)$ but by $H(t)$.

3. In the consensus protocol, the ledger nodes agree to the explicit value as well. The explicit value is recorded at the ledger.

$H(t)$ can be used in the design of financial derivatives. If the consensus algorithm is nicely implemented and $H(t)$ can be regarded as an adapted process, then theoretical studies on financial derivatives may become easier.

6. Conclusion

This paper overviewed an academic pedigree of blockchain where most of its technical components including secure timestamping schemes originated from the academic literature well before the Bitcoin’s white paper. Nevertheless, as pointed out in [50], the system design of the Bitcoin’s blockchain is more innovative than one might think. By using an adapted stochastic process to generalize the timestamping schemes, the potential of blockchain as a digital evidence technology can be more clearly recognized. This recognition helps us consider research directions toward more stable FinTech applications, and this paper proposed a blockchain token model as an example of starting points.

Acknowledgments

This work was supported by JSPS KAKENHI (Grant Num-
ber JP17KT0081) and JSPS/DST Japan-India Science Co-
operative Program.

References


[8] A. Biryukov, D. Khovratovich, and S. Tikhomirov, “FinFeld: Secure derivative contracts for Ethereum,” Revised Selected Papers of Fi-


[21] C. Dwork and M. Naor, “Pricing via processing or combatting


