Entangled Blockchains in Land Registry Management

Ashwin Sekhari
Department of CSE
National Institute of Technology,
Rourkela, Odisha, India
ashwinsekhari@gmail.com

Rishav Chatterjee
School of Computer Science Engineering
Kalinga Institute of Industrial Technology
Bhubaneswar, Odisha, India
rishavpiku@gmail.com

Ras Dwivedi
Department of CSE
Indian Institute of Technology
Kanpur, India
dwivedi@cse.iitk.ac.in

Rohit Negi
Department of CSE
Indian Institute of Technology
Kanpur, India
rohit@cse.iitk.ac.in

Sandeep K Shukla
Department of CSE
Indian Institute of Technology
Kanpur, India
sandeeps@cse.iitk.ac.in

***Abstract*—Land is an immovable property, and to prove the ownership of the land. Ideally, one must provide a legal document that should conclusively prove it. But, land record in India is a loosely defined term. Here, one is presumed to be the owner, until proven otherwise. Land record is a generic term that could mean a record of rights, *Khasra, Shajra*, etc. Moreover, different documents are maintained by different departments, and any one of them could be used to stake a claim on the land. This leads to numerous disputes, and it is evident from the number of cases, which are pending in Indian courts. In this paper, we aim to solve this problem of coordination between the various departments, and multiplicity of land records, via entanglement of blockchains .**

***Index Terms*—Blockchain, Hyperledger, Land Records, Record of Rights**

I. Introduction

Indian land record system is still inspired by the one introduced by Raja Todarmal (one of the 9 gems in the court of Akbar). This land record maintenance system is gripped with many loopholes. *Firstly* there are numerous records, each of which could potentially be produced to have the claim on the land. Many of these records could be mutated without the consent of the owner. *Secondly*, different records related to land are managed by different departments and hence they are not always on the same page. This has lead to numerous disputes for the ownership of the land. *Thirdly*, India does not have a uniform system for records related to land. Land is a state subject, and each state have different practice related to land. Lastly, much of these records are not digitized, and create huddle in coordination across the departments

With the advancement in information technology, Some data pertaining to land record is available in digital form [1] and some is in progress. These records are going to be digitized soon which will improve the existing business process like land registration, mutation, etc. But the problem of integrity and coordination still remains. Moreover, these documents would be prone to cyber attacks.

In the last few years, There is an increase in the number of cyber attacks. As per the Breach Level Index (BLI), There are approx 3 million records which have been compromised in the first half of the year 2018 [2]. With the rise of e-governance, government services will be available to citizens using information and communication technology (ICT) which might attract the adversaries and results in an increase in number of cyber attacks [3]. This technological advancement has raised information security concerns with the land records where the integrity of the ownership of the real property is the main target which results in increased number of land disputes, social restless etc. In India, it takes several years to get a dispute resolved [4] [5].

A blockchain based framework can be a solution to maintain the integrity of record of rights, and it can be useful while detecting an insider attack [6] and to incorporate the traceability. In this paper, we will present an implementation of Blockchain on Indian Land records system. We would not only increase the security and integrity of the land records, but would also enhance coordination of the department via this technology. As depicted in fig. 1, There are mainly three departments are involved in the process of maintaining the record of rights. Land records in India have been very poorly defined, and here a person is presumed to be the owner until proven otherwise. Numerous document like Khasra, Khatauni, *Shajra*, Registration document, parwana could refer to the Land record. Due to this improper definition, India has a huge amount of land dispute cases. Hence we want to create a solution that could bring all these land records under one system, and at the same time maintain its integrity.

Our contribution of this article is to develop a blockchain based solution 1). To make the land records tamper-resistant. 2). To enhance the coordination across department with blockchain 3). To expedite the business process. The rest of the paper is organized as section II

This work has been partially supported by the National Security Council Secretariat, Government of India.

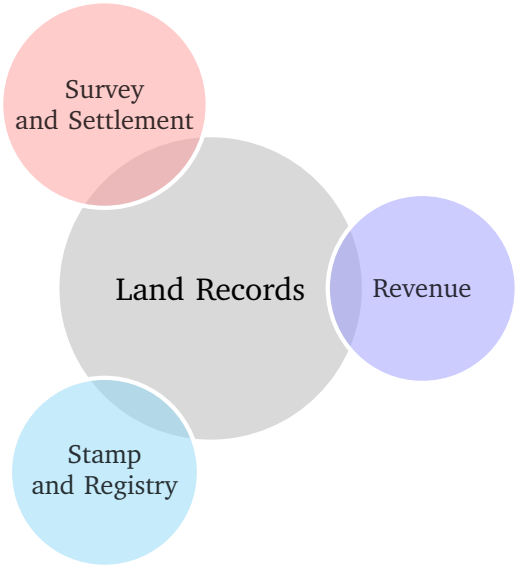

Figure 1. Departments involved in maintaining the record of rights

Table I
MODERNIZATION OF LAND RECORDS IN INDIA [7]

2008	National Land Records Modernization Programme (NLRMP).
2007	Electronic Citizen Services(ECS).
2006	JAN SEVA KENDRA.
2006	National e-Governance Plan (NeGP).
2004	Chhattisgarh Online information for Citizen Empowerment (CHOiCE).
2004	Bhulekh.
2004	Bhoomi, Nemmadi.
2003	e-Gram - Viswa Gram Project.
2000	Haryana land record information system (HALRIS).
1999	Computerisation of Registry Department(CORD).

will brief the existing posture of land records. Section III will brief the proposed solution based on blockchains. Section V will conclude our work.

II. BACKGROUND AND RELATED WORK

In previous section we showed how land records in India are vague, inefficient and prone to manipulation. To tackle this issue, India has started digitizing land records, but just digitizing them and securing them cannot be fruitful. We need to ensure integrity as well as the coherence of the land records. All of the land records must be changed at the same time and records must be updated simultaneously by every department. We propose a solution based on the entanglement of the blockchains, where every department have records on their own chain, but they are entangled so that change of record by one department, would force the change of record by another department. In this section, we list the work done by governments in India for the digitization of the land records, and then we show how blockchain technology suits our need.

"Haryana land record information system (HALRIS)", A project of Haryana government digitizing land records since 2000 [7]. Another project "Nemmadi" initiated by the government of Karnataka in 2004 has "Bhoomi" (meaning land in Kannada) program which expedites the digitization of land records and digitized around 20 million records of land ownership of 6.7 million farmers in the state [7]. Board of Revenue (Uttar Pradesh) modifies land reforms policies and implemented "BHULEKH" project in which all the Land Records of the state in each 312 tehsils of 71 districts have been computerized [7]. Chhattisgarh started "Chhattisgarh Online Information for Citizen Empowerment (CHOiCE)" project to provide service like land records. More states are replicating similar projects to digitized the land records.

The Government of India has approved the National e-Governance Plan (NeGP) in 2006 which includes "Land Records" as a state project (comes under Ministry of Rural Development) to identify and automate multiple services such as integration of textual and spatial land records, integration of registration and mutation processes, automatic updating of land records providing conclusive title to land owners, and etc. mentioning a few [1] [8]. NeGP has identified Panchayat as one of the Mission Mode Projects(MMP) to overcome the challenges at the village level. One of the main objectives of this MMP is to make land records tamper-proof, which will reduce the menace of litigation and social conflicts, associated with land disputes [7].

Blockchain is one of the latest technology that could be used to create tamper proof records. Blockchain, as the name suggests, is a chain of blocks. Blockchain is a distributed ledger, where records are stored in form of transactions. Each node have it's own copy of the ledger, and hence it is immune from single point of failure. Moreover transactions are bunched in form of a block, and each block is linked to previous block by the hash of the previous block, thus forming chain of blocks. Since it is linked with hashes, Blockchain is append only ledger, and any attempt to tamper a confirmed block, is same as attempting to break hash functions (SHA256 in case of bitcoin). This is the source of integrity of these chain. Fig 2 depicts a basic blockchain.

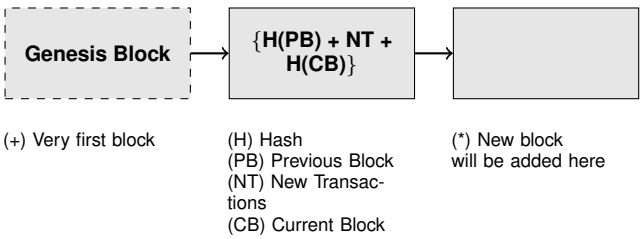

Figure 2. Blockchain

Blockchain have been classified as permissioned and permissionless blockchains. Permissionless blockchain

allows any user to join and leave the network at his will. Identities in case of permissionless blockchain need not to be known, due to which they are prone to sybil attacks [9]. Hence, in order to reach consensus, these blockchain uses methods like proof of work, proof of stake proof of space, etc. Few of the best known permissionless blockchain are Bitcoin, Ethereum, etc.

Permissioned blockchain, on other hand, requires identities of the nodes to be known beforehand. Only certified nodes can join the network. This requires establishment of central authority that generates PKI certificates. Since, identities are well established, these permissioned blockchain are not prone to Sybil attacks, and hence they could use lighter method for consensus like, solo, kafka, Byzantine consensus [10]. One of the best known permissioned blockchain is Hyperledger. Blockchain, apart from being tamper proof ledger, can also have smart contracts or chaincode. These are the programs stored on the blockchain, that are triggered by certain action. Both Hyperledger and Ethereum supports smart contracts

An open blockchain network cannot possibly serve our all needs. Maintenance of "Record of Rights" is an intricate task and only the trusted members should be able to interact with them. To keep a check on who is able to interact with the ledger it is essential to conceptualize using permissioned blockchains. Out of several permissioned blockchain frameworks like composer, sawtooth, Iroha, fabric, etc. We chose Hyperledger Fabric v1.0 for our prototype because we needed modularity and extensibility in code along with the ability to create separate channels, thus allowing us to group respective departments in order to create a separate ledger of transactions. As far as scalability aspects are concerned. The Fabric architecture uses Solo, Kafka consensus protocols and membership services for plug-and-play. The hyperledger fabric provides both modified and unmodified PKCS11 for key generation. The components of the architecture are listed below:-

1) A distributed ledger that logs all transactions between the clients and users.
2) A database (encrypted) that maintains the records of all the clients, including land contracts.
3) All the three types of nodes exist in the blockchain network: A set {E}, of endorsers that verify a transaction, set {O}, of orderers that run a consensus algorithm to create a block and a set {V} of validators that validate and store the blockchain.

III. Proposed Framework

As described above, Land records in India is a generic term for records of right, *Khatauni, Khasra, Shajra*, and other records. Since Land is a state subject, Land records are managed by the government of the state. For the purpose of this paper, we are using land records system as in the state of Uttar Pradesh. There are different ways for managing rural and urban land records in Uttar Pradesh. In this paper, we would use our design to tackle the rural land record problem

In the state of UP, land records management takes place at the office of Tehsildar. This office deals with the land registration and mutation. In this paper, we would like to keep the integrity of both registration and mutation proceedings.

In UP, one of the most important document pertaining to rural land is *Khasra*. Each *Khasra* uniquely identifies a piece of land. Each *Khasra* contain the names of the owner of that land. Notice that since land could be divided into a smaller and smaller part, each *Khasra*, usually, include names of multiple owners, with their share of ownership in that *Khasra*. Any mutation in the ownership of land could be either transfer of land to single owner, or transfer of a part of the land to a sole owner, or transfer of different part of the land to a different owner. Notice that each such transfer is done via registration. We would like to maintain the integrity of this registration as well as the integrity of *Khasra*, the land records. At present, while the registration is a valid land record, there is no verification done while doing registration. All verification is done during mutation proceeding. In this paper, we do checks at the registration stage also. We generate a token, of the registration that could be later used to verify the sanctity of the original document. We propose a model based on hyperledger fabric which consists of 2 parallel blockchains to solve the issue of land registration in India. We call them *Khasra* chain and registration chain, for dealing with mutation and registration respectively. In our model, clients are authenticated, and a certificate authority distributes respective cryptography material generated by *cryptogen* and *configtxgen*. The *Khasra* chain blockchain contains all the essential information regarding that *Khasra* like *Khasra* No, CircleRate, Area, List of Owners, Their Document Hashes, etc. The Registration chain consists of Merkle root made from the hashes of the registered document. This Merkle tree is generated based on the registration done in a span of 1 minute. Based on this Merkle root, we generate tokens that could be used for the verification of the registered document later. This verification plays a major role in maintaining the integrity of the land records, and we describe this later in the paper.

A. Chaincode

Chaincode is the driving logic of the entire system and in this context, We used Chaincode to make sure that the land record mutation is genuine. The Chaincode on *Khasra* Blockchain does mutation in the *Khasra*, but that is entangled with the registration blockchain also. Any such mutation needs a valid registration. Here we would like to emphasize that registration happens via payment of stamp duties to state government. Although we have not implemented this feature in our prototype but this would act as an economical deterrent to frivolous land record mutation. Since both the blockchain are entangled, Any mutation also generates a new token that could be used for the verification of newly registered document. This entanglement is not one way. This freshly generated token is also stored as an attribute in the *Khasra* Blockchain. What this means that for a token to be valid, it should not only be present in the Registration chain, but also in *Khasra* chain, and hence

stale token cannot be used, thus preventing a double-spending attack.

We have the following chaincodes in our prototype

1) Revenue Calculation: Revenue calculation is a complex process which depends upon the type of land, circle rate, and its area. So currently our model returns area and circle rate as the outputs for the revenue function and further work is required to improve this feature.

Algorithm 1: Revenue Calculation

```
Function Revenue(KhasraNo):
    if Khasra exists on ledger then
        return Circle Rate, Area
    else
        return "Khasra Does not exist."
```

2) Updating Details: This is used to update basic values of a khasra like circle rate, area. Over time administration as well as land details change and as a result several values need to be updated. They may also undergo **chakbandi** (Consolidation).

Algorithm 2: Details Updation

```
Function Update(KhasraNo, Chakbandi(Y/N),
  CircleRate, Area):
    if Khasra exists on ledger then
        if Chakbandi == 'N' then
            KhasraNo.CircleRate = CircleRate
            KhasraNo.Area = Area
        if Chakbandi == 'Y' then
            return "Chakbandi in progress, Khasra
              cannot be updated."
    else
        return "Khasra Does not exist."
```

3) Querying the Database: This function takes key value as an argument and returns the current state of that *Khasra*.

Algorithm 3: Query for a specific object in the database based on the key value.

query();
Input : *Khasra* ID
Output: Return the details of the *Khasra* given by the *Khasra* ID.

1) Search for the key : *"KhasraID"* in the database.
2) Retrieve the JSONByte object and return it as the result of the query. If it does not exist, then return error.

4) Land Transactions: This is used to transfer the land from one person to another or several other persons. In our case, transactions refer to the transfer of ownership of a piece of land to a different owner(s). If a person has multiple lands in the same *Khasra* the document hashes (SHA-256) are appended at the end. In the pseudocode "Reciever" refers to the list of persons along with their hashes which are receiving the land.

Algorithm 4: Transaction

```
Function Transact(KhasraNo, Owner, Hash,
  Reciever):
    if Khasra exists on ledger then
        if Owner in KhasraNo.Owners then
            if Hash in KhasraNo[Owner][Hash] then
                Remove Hash
                Append Reciever.Hash to Reciever(s)
                  in ledger
                Generate new tokens
            else
                Raise "Invalid Hash"
        else
            Raise "Invalid Owner"
    else
        return Raise "Khasra Does not exist."
```

B. Viewing the Database

This is used to get all the changes which happened for a *Khasra*.

Algorithm 5: View all objects in the database corresponding to a partial composite key.

View();
Input : id of owner which has registered the *Khasra* ID
Output: All the details of the Owner registered by the given proxy peer and have purchased the *Khasra* given by *Khasra* ID.

1) Generate a partial composite key with Owner id and *Khasra* ID .
2) Create an iterator based on the partial composite key and use it to iterate over the database.
3) Retrieve the JSONByte object corresponding to each Owner registered by the proxy peer and who have purchased the land given by the ID value.
4) Convert the byte object into string and return the entire set as the result of the query.

Other functions like adding new owners, initializing the ledger, etc are assumed to be common knowledge.

C. Security analysis of the proposed system

Our architecture brings integrity to the land records in India. We want to emphasize that our security is multifold and far better than the present bookkeeping.

a) Tokens: Our tokens are generated from the documents registered all over the state in the span of 1 minute. We calculate the hash of the uploaded documents and then generate the Merkle tree. Our token constitutes the Merkle root, hashes of the sibling nodes in the path from leaves to the Merkle root. We here claim that the uniqueness of the token is as good as the Hash collision of SHA256. Since hash inversion is computationally hard, security of the token is as strong as SHA256. Also, since each generation of the token is linked with multiple updates in Khasra chain, security

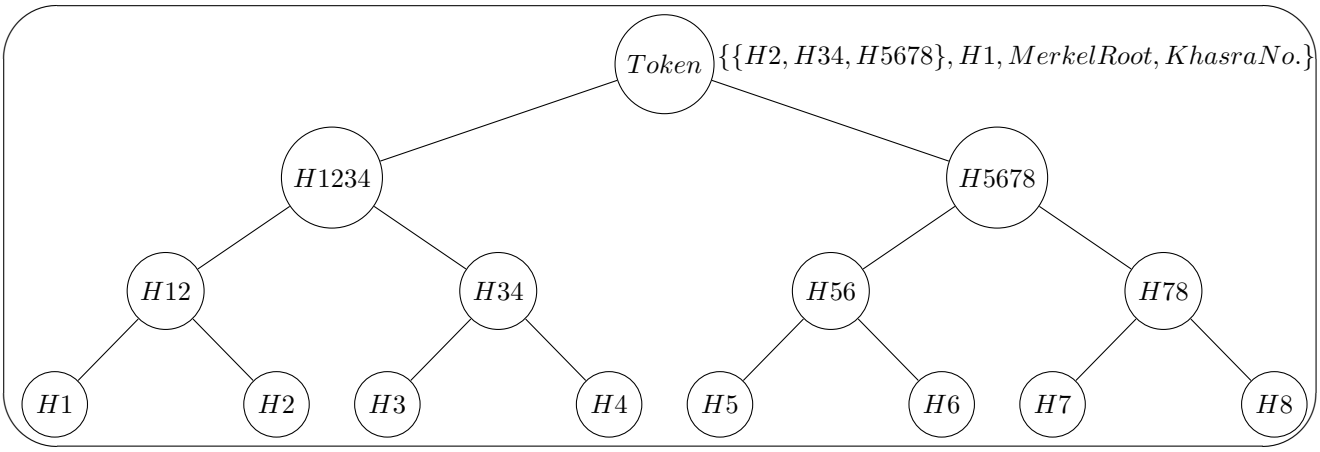

Figure 3. Token Verification

is better than in the case of single Blockchain. Since we have used tokens for the online verification of the document, we claim that this solves the problem of document forging in case of land records.

b) Security of Records: Since each Khasra is now represented in the Blockchain, and could only be changed by Chaincode. We can be sure that unauthorized mutation of a record is not possible. Even in the case of a malicious node, we want to emphasize that each variation is linked with the generation of a new token, and each token is only generated after the payment of stamp duties, which also bring security to the records. Since authorized nodes for the generation of tokens and mutation of records are different, there is no single point of failure in the system.

D. Workflow of the GUI Components

The first thing that happens while making one transaction is registering our admin user with our network's CA. If successful, the CA will send enrollment certificates that the SDK will store for us in our local file system. When the admin wishes to make a transaction from the user interface, the SDK will create an invocation transaction. The operations (AddPeople, Updation, Revenue) get built as a proposal to invoke the Chaincode function. Transactions (via the SDK) will send this proposal to a peer for endorsement. The peer will simulate the transaction by running the Go function and record any changes it attempted to write to the ledger. If the function returns successfully the peer will endorse the proposal and send it back to the Client. Errors will also be sent back, but the proposal will not be endorsed. Client (via the SDK), will then send the endorsed proposal to the orderer. The Orderer will organize a sequence of proposals from the whole network. It will check the sequence of transactions is valid by looking for transactions that conflict with each other. Any transactions that cannot be added to the block because of conflicts will be marked as errors. The orderer will broadcast the new block to the peers of the network. Our peer will receive the new block and validate it by looking at various signatures and hashes. It is then finally committed to the peer's ledger. At this point, the new transaction exists in our ledger and should soon exist as a record in all peer's ledgers.

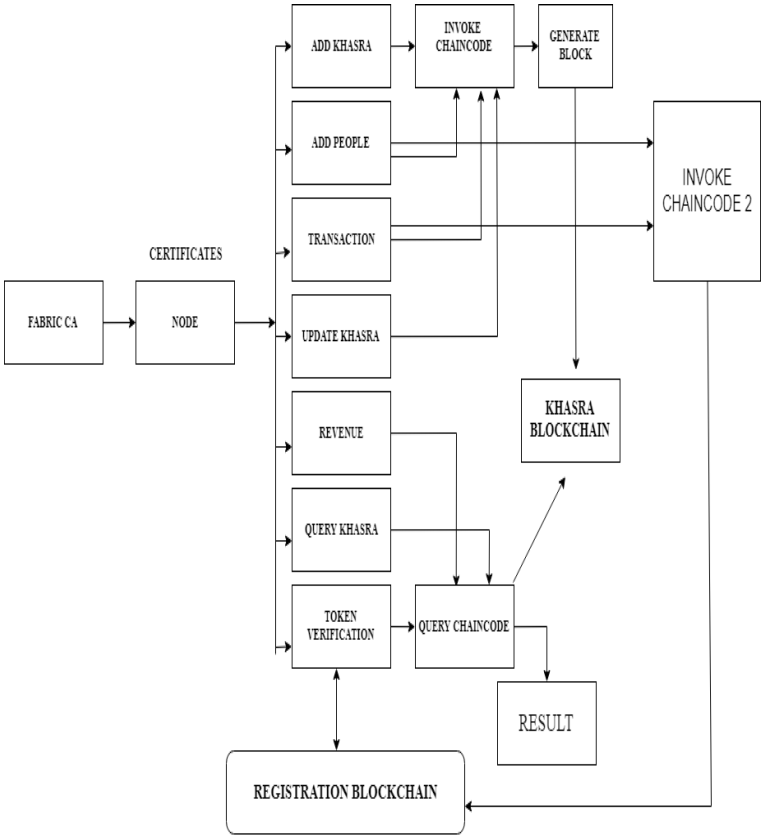

Figure 4. GUI Layout

IV. USER SCALABILITY EXPERIMENTS FOR ROBUST ENVIRONMENTS

We have successfully deployed our application with a working GUI on multiple peers by using physical machines which were assigned different roles in the network.

Discrete roles and their permmissions:

1) Orderer: Ordering service provides a shared communication channel to clients and peer which offers an atomic-broadcast service for transactions contained in messages. In short it implements a

delivery guarantee and maintains consistency of the ledger state.
2) Certificate Authority: Certificate Authority provides digitally signed certificates to nodes according to their roles and provide them with a digital identity.
3) Tehsildar: Tehsildars are the ones which are responsible for directly interacting with the ledger and randomly endorsing the transactions submitted by the other tehsildars.
4) Test_User: Users refer to common public and can view the ledger using a state database. They can request for new tokens to the tehsildars.

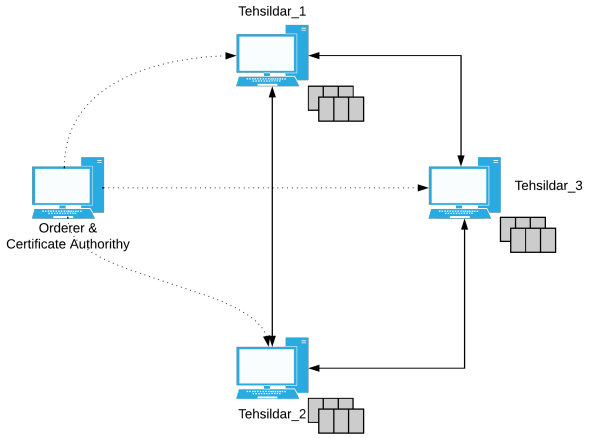

Figure 5. Scalablity Test Network

A test environment was created and real world transactions were simulated in the network.

1) Machine_1: Orderer + Certificate Authority
2) Machine_2: Tehsildar_1
3) Machine_3: Tehsildar_2
4) Machine_4: Tehsildar_3
5) Machine_5: Test_User

The stakeholders pertaining to that of our model could be defined as follows:-

1) Validators: They are responsible for validating the transactions and are a subset of tehsildars.
2) Orderers: They run the consensus algorithm and are responsible for block formation.
3) Endorsers: Endorsers follow the endorsement policies and verify the transactions.

V. CONCLUSION AND FUTURE SCOPE

The blockchain is currently in an evolving stage, as far as use cases like Land Registrations are concerned since the deployment levels have been increasing gradually. It is going to revolutionize the current system by eradicating numerous flaws at the same time. We have presented an Architecture in which entangled chain could be used to secure land records. In this model, we have only considered Registration Document and Khasra, and we hope that the other documents could be linked in similar ways thus create a mesh, which would be better in terms of security.

In this implementation, we have not done optimization in terms of space, which is one of the major factors in terms of the scaling of blockchains. We hope that future work would also consider this aspect. We hope that methods for geospatial sharding could be easily used, as here land data is segregated along the Tehsils.

REFERENCES

[1] R. Chauhan, "National e-governance plan in india," *United Nations University–International Institute for Software Technology*, 2009.
[2] "Breach level index." https://breachlevelindex.com/, 2018.
[3] C. Czosseck, R. Ottis, and A.-M. Talihärm, "Estonia after the 2007 cyber attacks: Legal, strategic and organisational changes in cyber security," *IJCWT*, vol. 1, pp. 24–34, 2011.
[4] B. Debroy and S. Jain, "Strengthening arbitration and its enforcement in india–resolve in india," 2016.
[5] "Millions of cases stuck in courts show need for 'urgent' land reform - advocacy,"
[6] S. Sharma, R. Gupta, S. S. Srivastava, and S. K. Shukla, "Detecting insider attacks on databases using blockchains," in *ACM-SIGSAC Conference on Computer and Communications Security*, 2017.
[7] R. K. Bagga and P. Gupta, *Transforming Government: e-Governance Initiatives in India*. Icfai University Press, 2009.
[8] D. Mathur, P. Gupta, and A. Sridevi, "e-governance approach in india the national e-governance plan (negp)," *Transforming Government*, p. 3, 2009.
[9] J. R. Douceur, "The sybil attack," in *International workshop on peer-to-peer systems*, pp. 251–260, Springer, 2002.
[10] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 4, no. 3, pp. 382–401, 1982.