Design of A Blockchain-based Lottery System for Smart Cities Applications

Da-Yin Liao
Straight & Up Intelligent Innovations Group Co.
San Jose, CA, USA
eliao@necksoft.com

XueHong Wang
Center of Lottery Studies in China
Peking University
Beijing, China
wxh13801241001@qq.com

***Abstract*—Among the smart cities applications, optimizing lottery games is one of the urgent needs to ensure their fairness and transparency. The emerging blockchain technology shows a glimpse of solutions to fairness and transparency issues faced by lottery industries. This paper presents the design of a blockchain-based lottery system for smart cities applications. We adopt the smart contracts of blockchain technology and the cryptograph blockchain model, Hawk [8], to design the blockchain-based lottery system, FairLotto, for future smart cities applications. Fairness, transparency, and privacy of the proposed blockchain-based lottery system are discussed and ensured.**

Keywords—smart cities, blockchain, lottery, gaming, smart contract, decentralization.

I. INTRODUCTION

The vision to Smart Cities is centered at the concept of connectivity, with tight integration among citizens, devices and services, and thus introduces plethora of opportunities and challenges [16]. Smart cities aim to improve the quality of life for their citizens by infusing technology into every part of its operations and optimizing the efficiency of services for citizens to meet their changing needs for smarter living. The scenarios of Smart cities pose several challenges in the management of cities and their development. The most relevant issue of smart cities applications consists in the interoperability of heterogeneous technologies and services used in smart cities. Growing along with the advance of technologies, citizens demand more transparent and fairness services in their smart cities applications. Among the smart cities applications, lottery games are one of the most urgent segments to improve to ensure fairness of the game itself as well as transparent and fair distribution of funds, the most common problems that most citizens suspect, distrust, and complain about.

Statistics in market research reports [5] depict that the Gross Gambling Yield (GGY) from the global gambling market hit $450 billion dollars in 2016. The market will continue to grow with an impressive Compound Annual Growth Rate (CAGR) of approximately 8% by 2020. In 2015, the lottery segment dominated the global gambling market by accounting for approximately 38% of the total market share [6]. China ranks first, in terms of the total revenue generated in the market. The development of infrastructure, increasing investments and the relaxation of government regulations and policies will be some of the factors contributing to the growth of the lottery market.

Lotteries are considered different from other gambling games because the outcome of a lottery is determined solely by chance, with no skill involved [3]. Lottery players are not classic gamblers. They are not looking for the thrill in the spin of the roulette wheel nor the river card that gives the flush. Lottery players are buying a dream. They buy a chance to be an instant millionaire and are willing to allocate a small sums of money every now and then, hoping to leverage it and possibly win a life-changing prize. Behind each ticket, there is a dream waiting to be fulfilled.

Existing lottery games rely on a centralized black box of services and operations, which causes distrust on players or citizens in many aspects. Questions about the fairness and transparency of lotteries frequently arise such as

a) is the ticket/lottery real?

b) is the drawing random and secure?

c) is the winning ticket added after the drawing?

d) is the jackpot winner real?

e) is the distribution of funds fair and real? and

f) why are so many corruptions, frauds and scandals of the lottery authority reported from time to time?

The emerging blockchain technology [13] shows a glimpse of solutions to fairness and transparency issues faced by lottery industries. Rather than being controlled by a centralized authority, blockchains introduce new ways of decentralization and delegation of services with autonomous interacting pieces of code, also referred to as *smart contracts*. With blockchains in place, applications that could previous run only through a trusted intermediary, can now operate in a decentralized way, without the need for a central authority, which was imply not possible before. The absence of a central authority means more and faster reconciliation between transacting parties. A key characteristic of blockchains is the heavy use of cryptography that helps bring authoritativeness behind all the interactions in the blockchains.

This paper presents the design of a novel blockchain-based lottery system, FairLotto, that combines lottery operations models and blockchain mechanisms. In FairLotto, every interaction with all business processes is strongly cryptographically authenticated with granular authorization based on roles and privileges. We use blockchain to ensure evaluating payment, ticketing, and payouts in distribution environments where there is no central authority or administrator, and guarantee that operations will be properly enforced by all interesting entities. The decentralized and lightweight features of FairLotto can supports future smart cities applications.

0-7695-6303-1/17/31.00 ©2017 IEEE
DOI 10.1109/CIC.2017.00044

The remainder of this paper is organized as follows. Section II describes the background of this research. Section III provides a brief overview of the background of blockchain technology and fundamental definitions. Section IV overviews the lottery operations basics. In Section V we propose the the design of FairLotto, a blockchain-based lottery system. Section VI discusses the fairness, transparency, and privacy of the proposed design of blockchain-based lottery systems. Section VII concludes the paper with some future research directions.

II. BACKGROUND

The terms Bitcoin [1] and blockchain are often used interchangeably, but they are not the same. Bitcoin is the name of the cryptocurrency that powers the Bitcoin network, a specific cryptocurrency, a peer-to-peer electronic cash system. The blockchain is a class of software and solutions. Bitcoin has a blockchain and, in fact, Bitcoin was the first application of blockchain, proposed in Satoshi Nakamoto's paper in 2008 [12]. Today Bitcoin has exploded and triggered a new gold rush. Garnering even more interest is blockchain, the solution underpinning Bitcoin, with many pundits predicting it has an even bigger future. While Bitcoin is well defined and tied as it is to a currency, the definition of blockchain is stretching to the point where it no longer refers to a particular technology or solution. Proposals powered by blockchains are flooding the market: from blockchain-enabled payments, through to identity management solutions, as if no problem cannot be solved by the artful application of blockchain technology. The challenges of blockchain technology include low transaction speed and verification process, data limits, and integration difficulty.

Modern state-run lotteries began in New Hampshire in 1964 to provide revenues and support education without resorting new taxes. The earliest lotteries took place in the 15th century. It has been almost 600 years since then and the point of lotteries today remains the same—lotteries are a tool used by the government to raise money without increasing taxes. Revenue is the raison d'être of lotteries. Basic lottery operations include recording the numbers and the amounts staked, drawing and determining the winning numbers, and collecting and pooling all the money placed as stakes. Computer systems are usually used for recording purchases and printing lottery tickets. Modern lotteries have been evolving different in many ways from earlier chance drawings to Lotto, Four-Number and Five-Number Game, Instant Lottery, Scratch Game, Multi-state Drawings, Quick Draw Keno, and more. For more studies on lottery gambling, please refer to the research of [2].

Liu *et al*. [11] develop a fair and efficient electronic lottery scheme that utilizes a delaying function of hash collision to generate the winning result. The anonymity of lottery purchasers are assured. A multi-level hash chain is used to reduce the computation for verifying the winning result. Kuacharoen [9] presents the design and implementation of a secure, online lottery system that can provide accuracy, privacy, transparency, and verifiability. Grumbach and Riemann [7] propose a distributed online lottery protocol that applies techniques developed for voting applications to an existing lottery protocol.

Blockchain technology has the potential to transform the gambling industry. Many entrepreneurs and others have been marching ahead into areas such as betting exchanges and provably fair games. There have been several attempts to create blockchain-based gaming or lottery platforms, but none of those initiatives have borne fruit. Their demise was brought upon by insufficient transparency, overly ambitious plans backed by zero actual products, and scandals involving the leadership behind a project, all of which are due to a young market which is still growing and learning from its mistakes. Except for the industrial pioneering efforts, to our best knowledge, no research and studies are reported on blockchain-based lottery applications.

III. OVERVIEW OF BLOCKCHAIN TECHNOLOGY

The emerging cryptocurrency systems such as Bitcoin [1, 12] build atop a novel blockchain technology where *miners* run distributed consensus whose security is ensured if no adversary wields a large fraction of the computational resource. Like Bitcoin, a blockchain reaches consensus not only on a stream of data but also on computations involving this data. In the event of contractual breaches or aborts, the decentralized blockchain ensures that honest parties obtain commensurate compensation. Existing decentralized blockchain systems lack transactional privacy with all transactions including flow of money between pseudonyms and amount transacted exposed on the blockchain. In the following, we overview the blockchain technology, transaction, structures of blockchains, and smart contracts. We then go over the newly emerging cryptocurrency system, Ethereum, which embrace the idea of running arbitrary user-defined programs on the blockchain, thus creating an expressive decentralized smart contract system. We also overview the emerging concept of decentralized autonomous organization (DAO) and a decentralized smart contract system, Hawk, which uses an efficient cryptography protocol where contractual parties interact with the blockchain and use cryptographic primitives such as zero-knowledge proofs.

A. Blockchain

A blockchain is a data structure that creates a digital ledger of data and shares it among a network of independent participants [9]. Blockchains use cryptography to allow each participant (user, node, or peer) on any given network to manage the ledger in a secure way without the need for a central authority to enforce the rules. The removal of central authority is the most important feature of the blockchain technology.

Blockchains provide every participant with a working proof of a decentralized trust. Every user or node has the exact same ledger as all of the other users or nodes in the network, which ensures a complete consensus from all users and nodes in the corresponding blockchain. In the ledger, blockchains create permanent records and history data whose permanence is based on the permanence of the network. That is, when data is recorded in a blockchain, it is extremely difficult to change or remove it.

B. Transaction

When someone wants to add a record to a blockchain, which is called a *transaction* or an entry, users in the network who have validation control of the transaction verify the proposed transaction. Transactions are generated by a sender and distributed among the peers in the network. Transactions are only valid once they have been accepted into the public

history of transactions in the blockchain. The transfer of digital tokenized assets can be achieved easily and in a cryptographically verifiable manner using a blockchain network that employs the Bitcoin transactional model.

C. The Structure of Blockchains

Blockchains are composed of three core parts: block, chain, and network. A block is a list of transactions recorded into a ledger over a given period of time. A chain is a hash that links one block to another. The hash in blockchain is created from the data that was in the previous block. The hash is a fingerprint of the data and locks blocks in order and time. A network is composed of nodes and each node contains a complete record of all the transactions that were ever recorded in the associated blockchains. Operating a node is expensive and time-consuming. Nodes are incentivized to operation with rewards of cryptocurrency or tokens provided by the underlying blockchain algorithm.

D. Smart Contract

A smart contract is a computerized transaction protocol that executes the terms of a contract [14]. It translates contractual clauses into code and embeds them into property that can self-enforce them, so as to minimizing the need for trusted intermediaries between transacting parties, and the occurrence of malicious or accidental exception. Smart contracts are self-executing scripts that reside on the blockchain. Every node in a smart contract-enabled blockchain is running a virtual machine (VM) and the blockchain network acts as a distributed VM. Smart contracts allow to have general purpose computations occur on the blockchain. A smart contract is deterministic; the same input will always produce the same output. Smart contracts operate as autonomous actors, whose behavior is completely predictable.

A blockchain that supports Bitcoin-style transactions enable asset transfers between counterparts that do not trust each other; however, a blockchain that supports smart contracts takes this further and allows for multi-step processes (i.e., interactions) to occur between mutually distrustful parties.

E. Ethereum

The Ethereum project [15] is one of the most developed and accessible blockchains in the ecosystem and is well recognized as an industry leader in blockchain innovation and use cases. Ethereum is an open-source, decentralized platform that run smart contracts: applications that run on a custom-built blockchain, an enormously powerful shared global infrastructure that can move value around and represent the ownership of property. It has its own Turing-complete programming language. The Ethereum protocol can do just about anything that most programming languages can do, except it is built inside a blockchain and has the added benefits and security that comes with that. Almost any software project can be built on Ethereum. The Ethereum ecosystem is currently the best place to build decentralized applications. Rapid development time, security for small applications, and the ability for applications to easily interact with one another are the key features of the Ethereum ecosystem.

F. DAO

A decentralized autonomous organization (DAO) is an organization that is run through rules encoded as smart contracts. A DAO's transaction record and program rules are maintained on a blockchain. DAOs are a type of Ethereum application that represents a virtual entity within Ethereum. DAOs aim to be open platforms where individuals control their identities and their personal data. When someone creates a DAO, he or she can invite others to participate in the governance of the organization and the participants can remain anonymous and never meet. Such shareholder participation or potentially unlimited legal liability for participants in DAOs can be problematic and could have issues such as anti-money-laundering compliance.

DAOs have a number of security vulnerabilities. Although the code is visible to all, it is hard to repair, thus leaving known security holes open to exploitation. The immutable nature of a DAO's code makes it nearly impossible to fix any bugs once the DAO is live in Ethereum.

G. Hawk

As lack of privacy is a major hindrance toward the broad adoption of decentralized smart contracts, Kosba *et al*. [8] propose the blockchain model of cryptography, Hawk, a decentralized smart contract system that stores no financial transactions in the clear on the blockchain, and thus retains transactional privacy in the public. In their definition, the blockchain refers to a decentralized set of miners who run a secure consensus protocol to agree upon the global state. The blockchain is a conceptual trusted party who is trusted for correctness and availability, but not trusted for privacy. A blockchain not only maintains a global ledger that stores the balance for every pseudonym, but also executes user-defined programs. They propose a formal model of the blockchain, including ideal specifications, as well as pieces of the protocol (i.e., programs) executed by the blockchain, the users, and the manager. Wrappers are used to implement a set of common features needed by every smart contract application, including time, public ledger, pseudonyms and adversarial reordering of messages.

The *manager* is a special party in facilitating the execution of Hawk. In Hawk, the manager can see users' inputs and is trusted not to disclose users' private data. However, the manager is not a trusted third party as those in conventional transactions and the manager cannot affect the correct execution of the Hawk program. In the context of lottery games, the role of the manager is like the lottery provider, while the users are the lottery players.

A Hawk program ϕ can be divided into two parts: the private part $\phi_{\texttt{priv}}$ and the public part $\phi_{\texttt{pub}}$. The private part $\phi_{\texttt{priv}}$ takes in parties' input data, e.g., selected numbers for lottery, as well as currency units, e.g., BTC (bitcoins). $\phi_{\texttt{pub}}$ computes to settle the payout distribution amongst the parties. In public part $\phi_{\texttt{pub}}$, no private data and money is referred, manipulated, or used.

The compiler in Hawk compiles the Hawk program into three types of programs which jointly define the cryptographic protocol among the users, the manager, and the blockchain. They are as the ideal program `IdealP`, the blockchain program `BlockchainP`, and the user/manager program `UserP`, respectively. Ideal programs define the correctness and security requirement by writing a specification assuming the existence of a fully trusted party. An ideal program is used by

combining with a wrapper to endow with exact with execution semantics. Hawk realizes two kinds of specifications—(1) private ledger and currency transfer, and (2) Hawk-specific primitives. Hawk relies on the existence of a private ledger that support private currency. An ideal functionality, $\mathtt{IdealP}_{\mathtt{cash}}$ is defined to describe the requirements of a private ledger. With a private ledger specified, Hawk-specific primitives are defined, including *Freeze*, *Compute,* and *Finalize* that are essential for enabling transactional privacy and programmability simultaneously.

A public ledger, denoted as `ledger`, is used in the ideal and blockchain programs. Money transfers only take place when ideal programs or blockchain programs update `ledger`. The $\mathtt{IdealP}_{\mathtt{cash}}$ specifics the requirements of a private ledger and currency transfer. Two operations, *Mint* and *Pour*, are specified in $\mathtt{IdealP}_{\mathtt{cash}}$, described as below:

- ***Mint.*** The *mint* operation allows a user P to transfer money from the public ledger `ledger` to the private pool denoted `Coins`[P]. With each transfer, a private point for user P is created, and associated with a value `val`.
- ***Pour.*** The pour operation allows a user P to spend money in its private bank privately.

In the ideal world, when an honest party P mints, the adversary A learns the pari (P, `val`) —since minting is raising coins from the public pool to the private pool. Operations on the public pool are observable to A. On the other hand, when an honest party P pours, the adversary A learns only the output pseudonyms P_1 and P_2. It does not learn which coin in the private pool `Coins` is being spent nor the name of the spender. The spent coins are anonymous with respect to the private pool `Coins`. To get strong anonymity, new pseudonyms P_1 and P_2 can be generated on the fly to receive each pour.

To enable transactional programmability and privacy simultaneously. For example, the formal specifications of the ideal program $\mathtt{IdealP}_{\mathtt{hawk}}$ include *Freeze*, *Compute* and *Finalize* modules, described as below:

- ***Freeze.*** A party P asks $\mathtt{IdealP}_{\mathtt{hawk}}$ to remove one coin from the private coin pool `Coins`, and freeze it in the blockchain by adding it to `FrozenCoins`. Party P's private input, denoted as `in`, is recorded in `FrozenCoins`. $\mathtt{IdealP}_{\mathtt{hawk}}$ checks that P has not called *Freeze* earlier, and that a coin (P, `val`) exists in `Coins` before proceeding with the *Freeze*.
- ***Compute.*** Party P calls *Compute*. Party P's private input in and the value of its frozen coin `val` are disclosed to the manager P_M.
- ***Finalize.*** The manager P_M submits a public input $\mathtt{val}_{\mathtt{M}}$ to $\mathtt{IdealP}_{\mathtt{hawk}}$ to compute the outcome of $\phi_{\mathtt{priv}}$ on all parties' inputs and frozen coin values, and redistribute the based on the outcome of $\phi_{\mathtt{priv}}$. $\mathtt{IdealP}_{\mathtt{hawk}}$ checks that the sum of `FrozenCoins` is equal to the sum of output coins to ensure the conservation of money. The functionality is parameterized by a public Hawk contract $\phi_{\mathtt{pub}}$ to check the wellformedness of the manager's input $\mathtt{in}_{\mathtt{M}}$ and redistribute public deposits.

In this research, we design and develop the blockchain-based lottery system, FairLotto, based on the smart contracts, and the Hawk model. Before introducing the FairLotto system, some basics of lottery operations are overviewed in the following section.

IV. LOTTERY OPERATIONS BASICS

Operations of a lottery game involve two kinds of basic parties: player and lottery provider. A lottery provider sells certified lottery tickets to players, verifies the winning tickets, and issues lottery prizes to the winners. A lottery game is built up through four basic stages—initialization, lottery purchase, closing time, and verifying winning numbers—described as below.

A. Initialization Stage

A lottery game starts from the announcement of the gaming rules, beginning and closing times, distribution of payouts and other information of the proposed lottery game by a lottery provider. During the initialization stage, each participant of the lottery provider and players will be associated with a pair of private and public keys, respectively.

B. Lottery Purchase Stage

The lottery purchase stage begins when a player buys a lottery. The lottery information is hashed to obtain the message digest. The obtained message digest is manipulated by the blinding scheme. The blinded message digest and the signature are encrypted using the public key of the lottery provider and sent to lottery provider to verify. The lottery provider knows nothing about the numbers of being processed.

When the lottery provider receives the message from the player, the lottery provider decrypts the message using the public key of the lottery provider and verifies the signature of the player. Upon successful verification, the lottery provider signs the blinded message digest. The signed blinded message digest is encrypted using the public key of the player and then is sent to the player.

After verifying the signature, the player randomly selects a session key for the current lottery ticket and encrypts the lottery information and the certified signature of the lottery provider using the session key. The session key is then encrypted using the public key of the lottery provider. The encrypted certified lottery information, the encrypted session key and the transaction signature are encrypted using the public key of the lottery provider and are sent to the lottery provider.

The lottery provider decrypts the received information using the private key of the lottery provider and verifies the signature. The lottery provider issues a sequence number of the lottery ticket. A receipt for the lottery ticket is generated by signing the sequence number and the encrypted lottery information. The sequence number and the receipt are then sent to the player.

When the player receives the receipt from the lottery provider, then the player verifies the signature. After verifying

the signature, the player prints the lottery ticket, which completes the purchase stage.

C. Closing Time Stage

The players have certified all purchase lottery tickets and maintains a list of the corresponding sequence numbers and the encrypted session keys. Each player does not have any knowledge of the whole lottery numbers purchased, except for his or her selected lottery numbers. On the other hand, the lottery provider has encrypted purchased lottery numbers and encrypted session keys, but the lottery provider cannot obtain any information on the purchased lottery numbers. This provides a balance of power between the player and the lottery provider.

When the closing time has up, the lottery provider publishes all sequence numbers along with the corresponding receipts and signs the published information. The lottery provider also sends the list of sequence numbers, encrypted session keys, and the receipts to the players. The players stop blind signing. The signature is verified.

After all the winning numbers are drawn, the players send a list of the sequence numbers and the session keys to the lottery provider. The lottery provider verifies the signature. For each lottery ticket, the lottery information together with the certified signature can be determined using the corresponding session key. The lottery provider verifies the signature and completes the closing time stage.

D. Verifying Winning Numbers Stage

With the information of a lottery ticket such as the sequence number, the lottery information, the certified signature, the receipt information, and the session key, the player first verifies the certified signature by comparing the hash value of the lottery information with the value obtained from decrypting the signature using the public key of the player. This ensures that the lottery ticket was certified properly. To verify the receipt information, the lottery information is encrypted using the session key. The hash of the sequence number and encrypted lottery information is compared with the one obtained from decrypting the receipt using the public key of the lottery provider. The final step is to compare the purchased and determine the payouts.

V. FAIRLOTTO: A BLOCKCHAIN-BASED LOTTERY SYSTEM

The proposed blockchain-based lottery system, FairLotto, adopts blockchain technology of smart contracts and the Hawk model [7]. The organization of FairLotto can be divided into three layers—Data, Smart Contracts, and Interface layers. The Data layer lays the fundamentals of FairLotto and is composed of modules of blockchains, consensus protocols, roles, transactions, and privacy.

Above the Data layer is the Smart Contracts layer. As overview in Section II, a smart contract is a collection of code (its functions) and data (its state) that resides at a specific address on the blockchain. The smart contracts in FairLotto adopts the software stack used in a typical Ethereum node, where a fully validating node contains the entire history of the blockchain, whereas a non-validating node store only the block headers. Figure 1 depicts the software stack in a typical Ethereum node [4]. The code of smart contracts is running insider the VM (virtual machine, the runtime environment for smart contracts) and can access to network, filesystem or other processes. Smart contracts can even access to other smart contracts, as depicted in the dashed line in Figure 1.

There are four types of smart contracts defined in FairLotto, including user registration (Registration), money transfer (Transfer), lottery claiming (Claim), and payouts (Payout). On the top of FairLotto, it is the Interface layer which is made of applications and services of web, mobile apps, desktop programs, player service, lottery provider, and application services.

Figure 2 depicts the three-layered software architecture of the proposed blockchain-based lottery system, FairLotto.

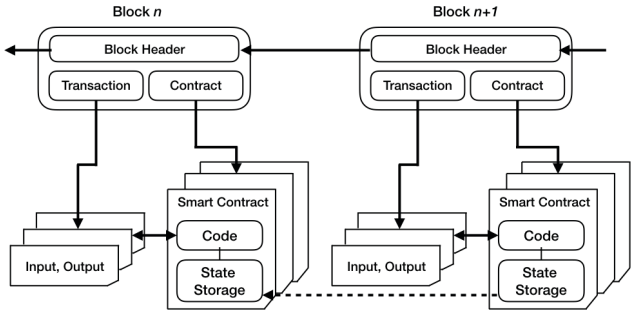

Figure 1. Blockchain Software Stacks [4]

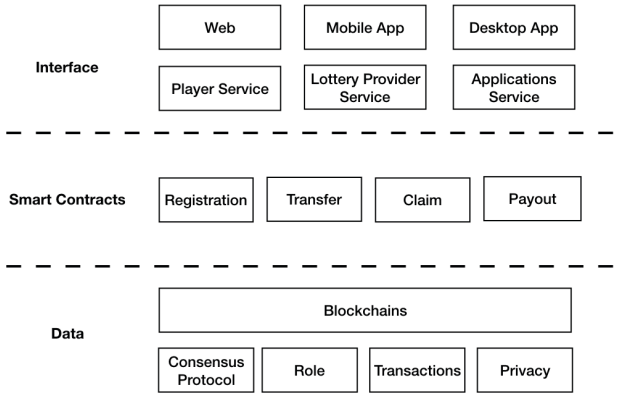

Figure 2. FairLotto Software Structure

The following notations are used in the descriptions of the proposed blockchain-based lottery system, FairLotto:

P : player;

L : lottery provider;

A : participant, $A \in \{P\} \cup \{L\}$;

κ : a pair of private and public keys of $\kappa \in \{P, L\}$, $\kappa = (\kappa.private, \kappa.public)$;

$\Xi_\kappa(m)$: signature on message m computed with the private key $\kappa.private$;

$\delta_\kappa(m, \Xi)$: verification of signature Ξ on message m with the public key $\kappa.public$;

$h(.)$: SHA-256 hash function;

$b(.)$: blinding function;

$d(.)$: decrypting function;

$e(.)$: encrypting function;

$p(.)$: payout function;

$rec(.)$: receipt generating function;

$sng(.)$: sequence number generating function;

$rnd(.)$: random number generating function;

N : set of pairs of winning numbers and payouts;

FairLotto develops several protocols executed by the blockchain, the player (P), and the lottery provider (L) respectively as programs written in pseudo code through the four basic stages—Initialization, Lottery Purchase, Closing Time, and Verifying Winning Numbers. The following depicts the protocols of `Initialization` used for the Initialization stage.

Initialization

Init: `Ids`: a set of reference string, initially empty
`Coins`: a set of party-coin pairs, each of the form (P,`$val`)
Create: Upon receiving (`create`, `$val`) from some party P:
`id` := h(`$val`)
P_{public} := h(`id`, rnd())
$P_{private}$:= h(`id`, rnd())
assert (P, `id`, P_{public}, $P_{private}$) $\notin$ `Ids`
add (P, `id`, P_{public}, $P_{private}$) to `Ids`
`ledger`|P| := 0
Mint: Upon receiving (`mint`, `$val`, `s`) from some party P:
`coin` := $Comm_s$(`$val`)
assert (P, `coin`) $\notin$ `Coins`
`ledger`|P| := `ledger`|P| + `$val`
add (P, `coin`) to `Coins`

Initialization (Continue)

Pour: On input (`pour`, $\$val_1$, $\$val_2$, P_1, P_2, $\$\underline{val}_1$, $\$\underline{val}_2$) from P.
assert $\$val_1 + \$val_2 = \$\underline{val}_1 + \$\underline{val}_2$
if P is honest:
assert (P, $\$val_i$) $\in$ `Coins` for `i` $\in \{1, 2\}$
assert $P_i \neq \emptyset$ for `i` $\in \{1, 2\}$
remove one (P, `$val`) from `Coins` for `i` $\in \{1, 2\}$
for `i` $\in \{1, 2\}$:
if P is corrupted:
send (`pour`, `i`, P_i, $\$\underline{val}_i$) to P
else
send (`pour`, `i`, P_i) to P
if P is corrupted:
assert (P, $\$val_i$) $\in$ `Coins` for `i` $\in \{1, 2\}$
remove one (P, $\$val_i$) from `Coins` for `i` $\in \{1, 2\}$
add one (P_i, $\$\underline{val}_i$) from `Coins` for `i` $\in \{1, 2\}$
if $P_i \neq \emptyset$:
send (`pour`, $\$\underline{val}_i$) to P_i for `i` $\in \{1, 2\}$

The Lottery Purchase stage uses the protocols of `LotteryPurchase`, whose pseudo code is listed as below:

LotteryPurchase

Init: L_{public}, $L_{private}$: public and private keys of L
P_{public}, $P_{private}$: public and private keys of P
`Lotto`: a set of lottery information, initially empty
`lotto`: lottery information, initially empty
`sig`: signature, initially empty
`FrozenCoins`: a set of coins and private inputs received
`PoolCoins`: a set of coins collected by the lottery
Buy: Upon receiving (`buy`, `$val`) from some party P:
`m` := h(`$val`)
`sig` := Ξ_κ(`m`)
`lotto` := b(`m`)
`encrypt` := e(`lotto`, `sig`, L_{public})
add (P, `encrypt`) to `Lotto`
send (`freeze`, `$val`) to A
send (`verify`, `msg`, `encrypt`) to L
receive δ_κ(`msg`, `encrypt`) from L
Verify: Upon receiving (`verify`, s_1, s_2) from some party P:
`decrypt` := e(s_2, L_{public})
if verified:
`string` := e(s_1, P_{public})
send (`transact`, `string`) to P

LotteryPurchase (Continue)

Transact: Upon receiving (transact, s) from party L:
session := rnd()
k_s := e(lotto, s)
$info_{enc}$:= e(k_s, e(session), sig, L_{public})
send (append, $info_{enc}$) to L

Append: Upon receiving (append, s) from some party P:
decrypt := e(s, $L_{private}$)
if verified:
seq := sng()
receipt := rec(seq, s)
add (L, s) to Lotto
send (compute, P) to A
send (ticket, seq, receipt) to P

Ticket: Upon receiving (ticket, s, r) from party L:
decrypt := e(r, L_{public})
if verified:
output ticket
send (finalize) to A

Freeze: Upon receiving (freeze, $val, in) from some party P:
assert current time
assert P has not called freeze earlier
assert at least one copy of (P, $val) ∈ Coins
send (freeze, P) to A
add (P, $val, in) to FrozenCoins
remove (P, $val) to Coins

Compute: Upon receiving (compute) from some party P:
assert current time
if P is corrupted:
send (compute, P, $val, in) to A
else
send (compute, P) to A

Finalize: Upon receiving (finalize, $val) from party P:
assert current time
assert P has not called finalize earlier
output(finalize, $val)

The Closing Time stage uses the protocols of ClosingTime, whose pseudo code is listed as below:

ClosingTime

Init: Lotto: public and private keys of L
Seq: a list of pairs of (sequence number, session key) of P
WinNumbers: a set of pairs of (winning number, payout)

Close: Upon closing time is up:
send (publish, A) to L
stop blind signing

Publish: Upon receiving (publish, A) from some party P:
send (Lotto) to A

WrapUp: Upon complete all the drawings and $N \neq \emptyset$
add N to WinNumbers
send (Validate, Seq) to L

Validate: Upon receiving (Validate, Seq) from some party P:
for each ticket t ∈ Seq:
assert t_{sig}==δ_K(m, Ξ)

The Verifying Winning Numbers stage uses the protocols of VerifyingWinningNumbers, whose pseudo code is listed as below:

VerifyingWinningNumbers

Init: WinTicket: a set of winning ticket information

Claim: Upon receiving (claim, t) from some party P:
assert (P, t) ∉ WinTicket
decrypt := e(t, P_{public})
if verified:
payout := p(decrypt, P_{public})
send (freeze, payout, P) to A

Freeze: Upon receiving (freeze, $val, in) from some party P:
assert current time
assert P has not called freeze earlier
assert at least one copy of (P, $val) ∈ Coins
send (freeze, P) to A
add (P, $val, in) to FrozenCoins
remove (P, $val) to Coins

Compute: Upon receiving (compute) from some party P:
assert current time
if P is corrupted:
send (compute, P, $val, in) to A
else
send (compute, P) to A

Finalize: Upon receiving (finalize, $val) from party P:
assert current time
assert P has not called finalize earlier
output(finalize, $val)

In FairLotto, we define a coin as the cryptocurrency unit for the blockchain-based lottery games. FairLotto provides several useful mechanisms using the blockchain. In FairLotto, the blockchain considered as a database stores all lottery operations policies in form of transactions. It serves also as logging databases that ensures auditing functions. It prevents forgery of coins through transactions integrity checks and detects corruption of participants.

Although the random number generation function, rnd(.), is used and only used in the protocol Initialization, the generation of winning lottery numbers in FairLotto can be through traditional drawings or via a computer or a random number generator. Therefore, FairLotto can be applied as an e-lottery system. The protocol ClosingTime does not specify how long for a lottery game to close. FairLotto can also be applied as an instant game by running the ClosingTime protocol after every lottery purchase. However, such frequent closing could deteriorate the performance of the entire blockchain lottery system.

These proposed protocols in FairLotto are lightweight and can be implemented and embedded in electronic devices of everyday life, as an integral part of the Internet of Things (IoT) paradigm to support smart cities applications, especially in the Integrated Casinos and Entertainment applications in smart cities.

V. FAIRNESS, TRANSPARENCY AND PRIVACY

A. Fairness

One of the features of the blockchain technology is to achieve fairness in its protocol design. At the closing stage, the players have certified all purchase lottery tickets and maintains a list of the corresponding sequence numbers and the encrypted session keys. However, they do not know what lottery numbers were the purchased. On the other hand, the lottery provider has encrypted purchased lottery numbers and encrypted session keys, but they cannot obtain any information on the purchased lottery numbers. This eliminates the possible of frauds and cheating in FairLotto. The lottery protocol should ensure an equal probability of winning the prize for every participant.

B. Transparency

Another features of the blockchain technology is the transparency. By use of blockchain, all users know who owns every block, at any time. Changes to FairLotto are publicly viewable by all parties creating transparency, and all transactions are immutable, meaning they cannot be altered or deleted. FairLotto allows information to be transferred in a trustworthy and anonymous way. It provides a trust network that allows information to trace back the chain from current transaction or lottery ticket upwards, without revealing what parties are. Efforts to use a centralized lottery system have potentials of frauds and thus distrusted among people, and FairLotto is a far more effective way of ensuring transparency.

C. Privacy

While solutions exist, including blockchains and strong encryption, there are still cyber security concerns that need to be addressed before the general public will entrust their personal data to a blockchain solution. FairLotto extends the Hawk model [8], that realizes parties interacting with a trustworthy virtual computer that executes programs involving money and data. As a decentralized smart contract system, FairLotto does not store financial transactions in the clear on the blockchain, thus retaining transactional privacy from view of the public.

VI. CONCLUDING REMARKS

In this paper, we develop a blockchain-based lottery system, FairLotto, for future smart cities applications. The proposed lottery systems can ensure their fairness and transparency. We adopt the smart contracts of blockchain technologies and the cryptograph blockchain model, Hawk [8], in designing the blockchain-based lottery system, FairLotto. Four protocols of `Initialization`, `LotteryPurchase`, `ClosingTime` and `VerifyWinningNumbers` are developed for the four lottery stages, respectively. These protocols in FairLotto are lightweight and can be an integral part of the IoT paradigm to support smart cities applications, especially in the Integrated Casinos and Entertainment applications in smart cities. Fairness, transparency, and privacy of the proposed blockchain-based lottery system are discussed and ensured in our system design. The lottery protocol should ensure an equal probability of winning the prize for every participant. FairLotto is a far more effective way of ensuring transparency. As a decentralized smart contract system, FairLotto stores no financial transactions in the clear on the blockchain, thus preserving transactional privacy.

The proposed design of FairLotto can be applied as an e-lottery system. The winning numbers in FairLotto can be generated through traditional drawings or via a computer or a random number generator. FairLotto can also be applied as an instant game by running the `ClosingTime` protocol after every lottery purchase. However, frequent closings could demand more operations and may deteriorate the performance of the entire blockchain lottery system.

We are implementing the FairLotto system in Ethereum. Further analysis and numerical testing results on the performance of FairLotto deployed in the Ethereum platform will be provided in the future publications or presentations. Future research directions may include the jackpot function into the lottery game design. Both the calculation of given percentage of the amount of each purchased ticket and the increments of the total amount in the jackpot pool are trivial but take time to update the blockchain of each node. The resulting performance and solutions could be an interesting topic to explore further. Another topic may consider the formal descriptions and analysis of state transitions in a blockchain system of smart contracts.

REFERENCES

1. A. Antonopoulos, Mastering Bitcoin: Unlocking Digital Cryptocurrencies, O'Reilly Media, 1st ed., Sebastopol, California, 2015.
2. V. Ariyabuddhiphongs, "Lottery Gambling: A Review," Journal of Gambling Studies, vol. 27, no. 1, 2011, pp. 15-33.
3. T. Barker and M. Britz, Jokers Wild: Legalized Gambling in the Twenty-first Century, Praeger, 2000.
4. T. Dinh, J. Wang, G. Chen, R. Liu, B. Ooi, and K. Tan, "BLOCKBENCH: A Framework for Analyzing Private Blockchains," Proc. of the 2017 ACM International Conference on Management of Data, May 2017, pp. 1085-1100.
5. Global Gambling Market Gross Gaming Yield (GGY) from 2001 to 2019. https://www.statista.com/statistics/253416/global-gambling-market-gross-win/, access on September 14, 2017.
6. Global Gambling Market 2016-2020, Sep 2016, technavio. https://www.technavio.com/report/global-gaming-global-gambling-market-2016-2020, access available on September 14, 2017.
7. S. Grumbach and R. Riemann, "Distributed Random Process for A Large-scale Peer-to-Peer Lottery," IFIP International Conference on Distributed Applications and Interoperable Systems, Springer, Cham, June 2017, pp. 34-48.
8. A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts," 2016 IEEE Symposium on Security and Privacy, May 2016, pp. 839-858.
9. P. Kuacharoen, "Design and Implementation of A Secure Online Lottery System," IAIT, 2012, pp. 94-105.
10. T. Laurence, Blockchain for Dummies, John Wiley & Sons, Hoboken, New Jersey, 2017.
11. Y. Liu, H. Liu, L. Hu, and J. Tian, "A New Efficient E-Lottery Scheme Using Multi-Level Hash Chain," Proc. IEEE International Conference on Communication Technology, November 2006, pp. 1-4.
12. S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.
13. M. Swan, Blockchain: Blueprint for A New Economy, O'Reilly Media, Sebastopol, California, 2015.
14. N. Szabo, "Formalizing and Securing Relationships on Public Network," First Monday, vol. 2, no. 9, 1997.
15. The Ethereum Project. https://www.ethereum.org. access available on September 14, 2017.
16. A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for Smart Cities," IEEE Internet of Things Journal, vol.1, no. 1, February 2014, pp. 22-32.