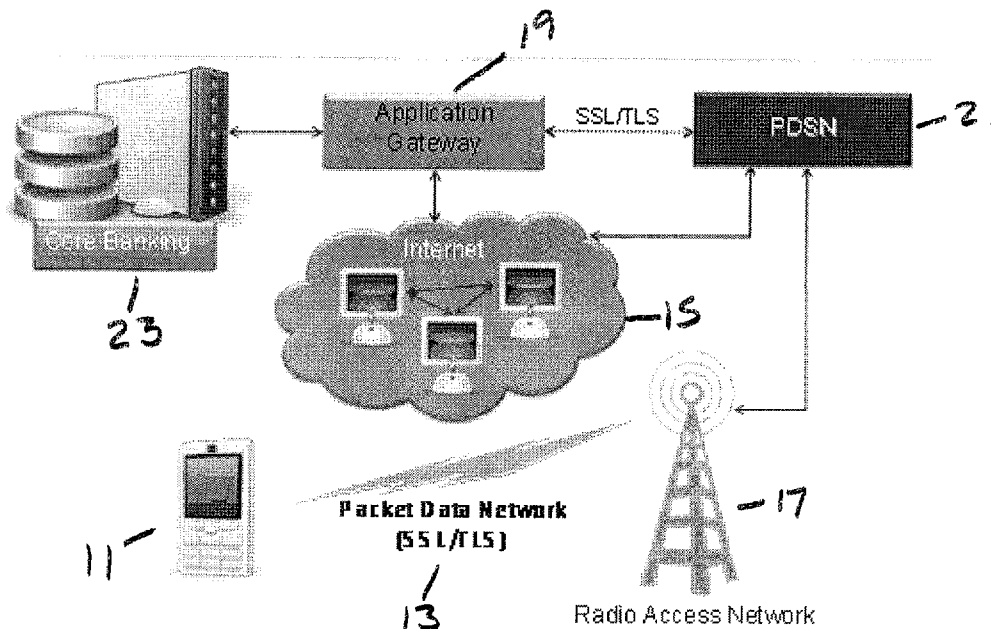




US 20120289303A1

(19) **United States**(12) **Patent Application Publication**
Jagannatha et al.(10) **Pub. No.: US 2012/0289303 A1**(43) **Pub. Date: Nov. 15, 2012**(54) **MOBILE GAMING SYSTEM**(52) **U.S. Cl. 463/17**(76) **Inventors:** **Deepa Jagannatha**, Somerset, NJ
(US); **Deepak Gonsalves**, Avenel,
NJ (US); **Dinesh Doshi**, South
Plainfield, NJ (US)(57) **ABSTRACT**(21) **Appl. No.: 13/530,813**(22) **Filed: Jun. 22, 2012****Related U.S. Application Data**(63) Continuation-in-part of application No. 12/099,901,
filed on Apr. 9, 2008, now abandoned.**Publication Classification**(51) **Int. Cl.**
A63F 9/24 (2006.01)

A system and method for providing mobile lottery gaming over a wireless mobile device/handset. The system and method permit a user of a mobile wireless device to play a lottery game electronically. The system will be designed such that various client services (banking, wallet, and gaming) can employ the existing infrastructure to perform its functions without compromising on essential features such as ease of use and security. Once a user is registered onto the system they can utilize an identification code to authorize the downloading of lottery games from a server to their mobile device. A financial account enables the user to electronically pay for the lottery games. Notification of the results of the lottery games can be made by e-mail, SMS or an IVR call.



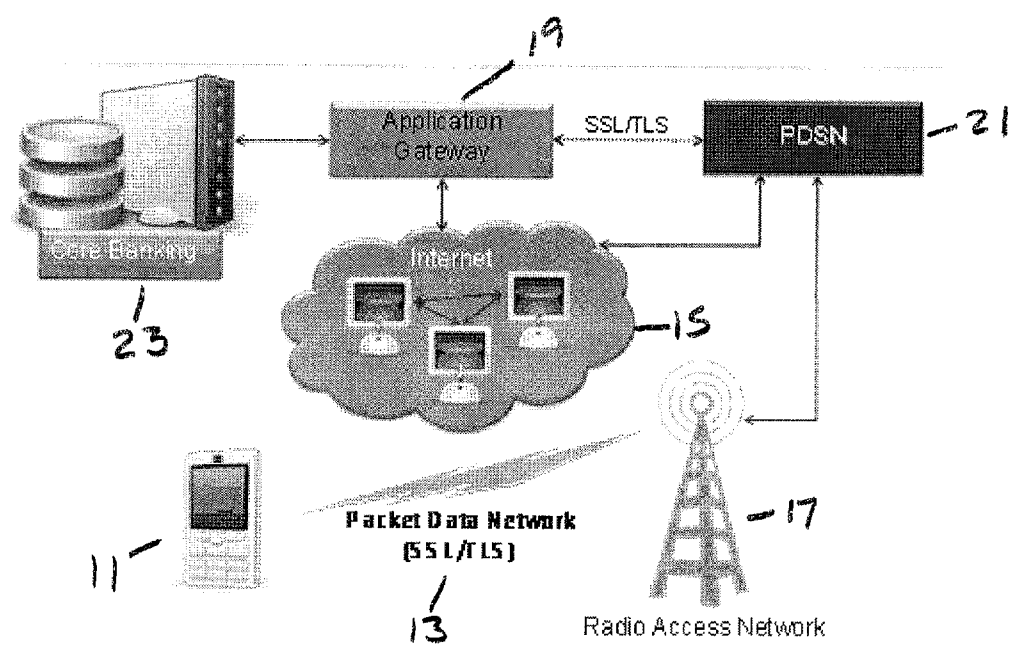


Figure 1

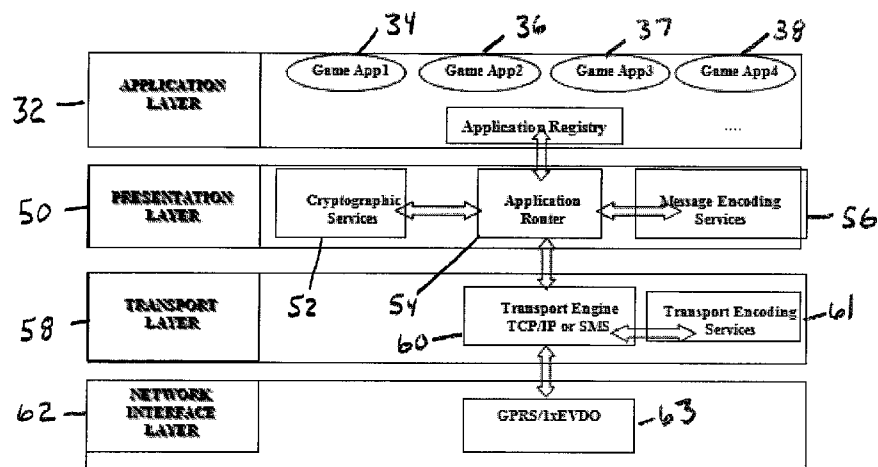


Figure 2

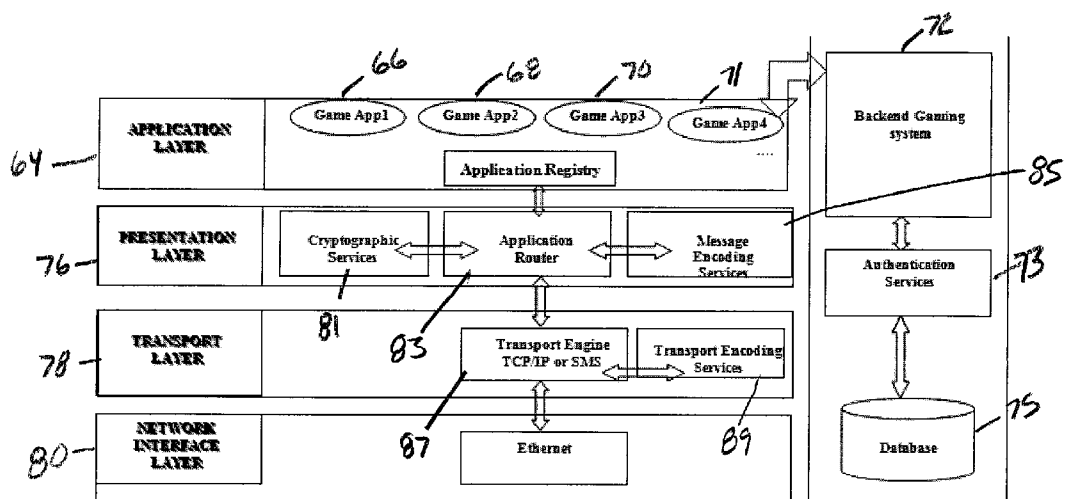


Figure 3

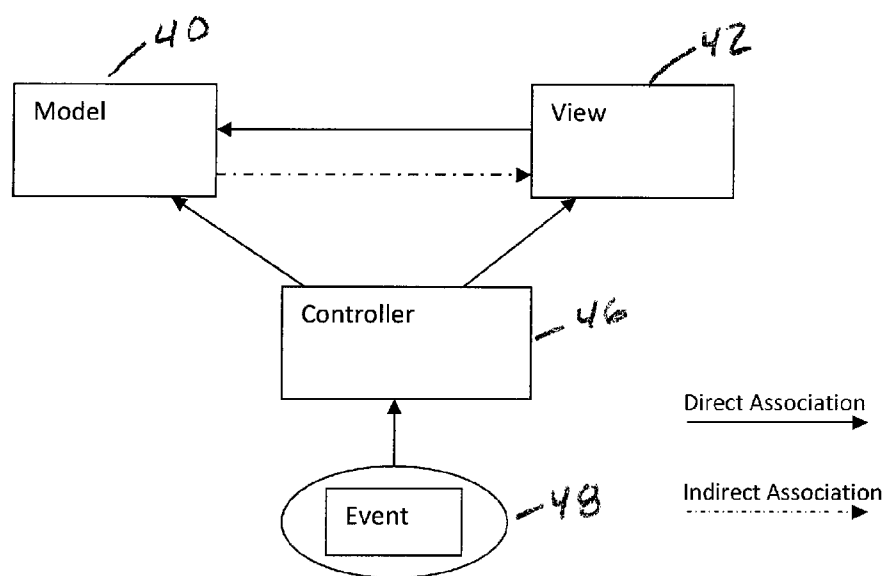


Figure 4

FIGURE 5: Registration Form

REGISTRATION FORM

NAME -82

MOBILE # -84

CREDIT CARD# -86

EXPIRY DATE -88

DOB -90

SELECT ANY ONE OF THE IMAGE BELOW:

94

PASSCODE -92

96 - ☐ VERIFY AGREEMENT

ACCEPT

REJECT -100

98

FIGURE 6: Flowchart for Application based gaming

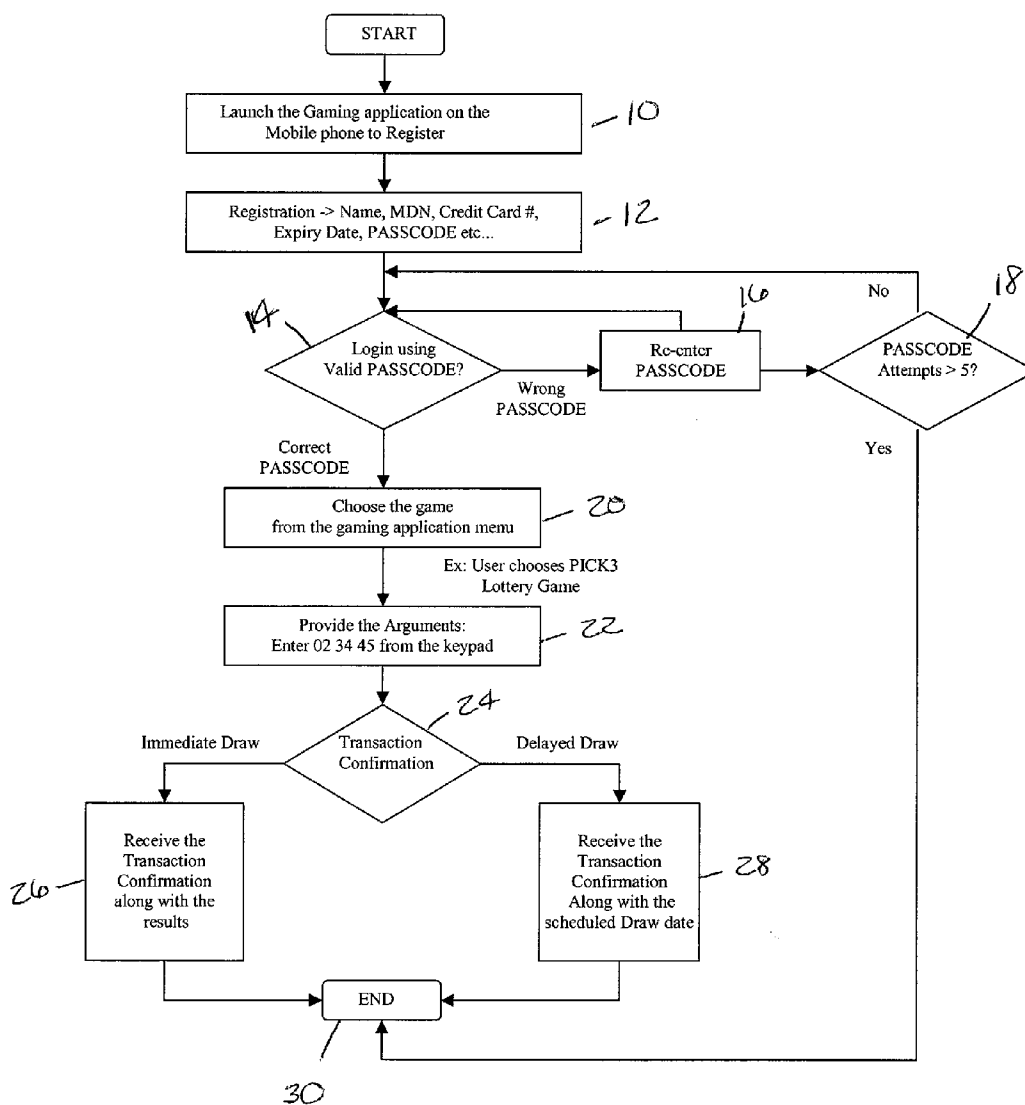


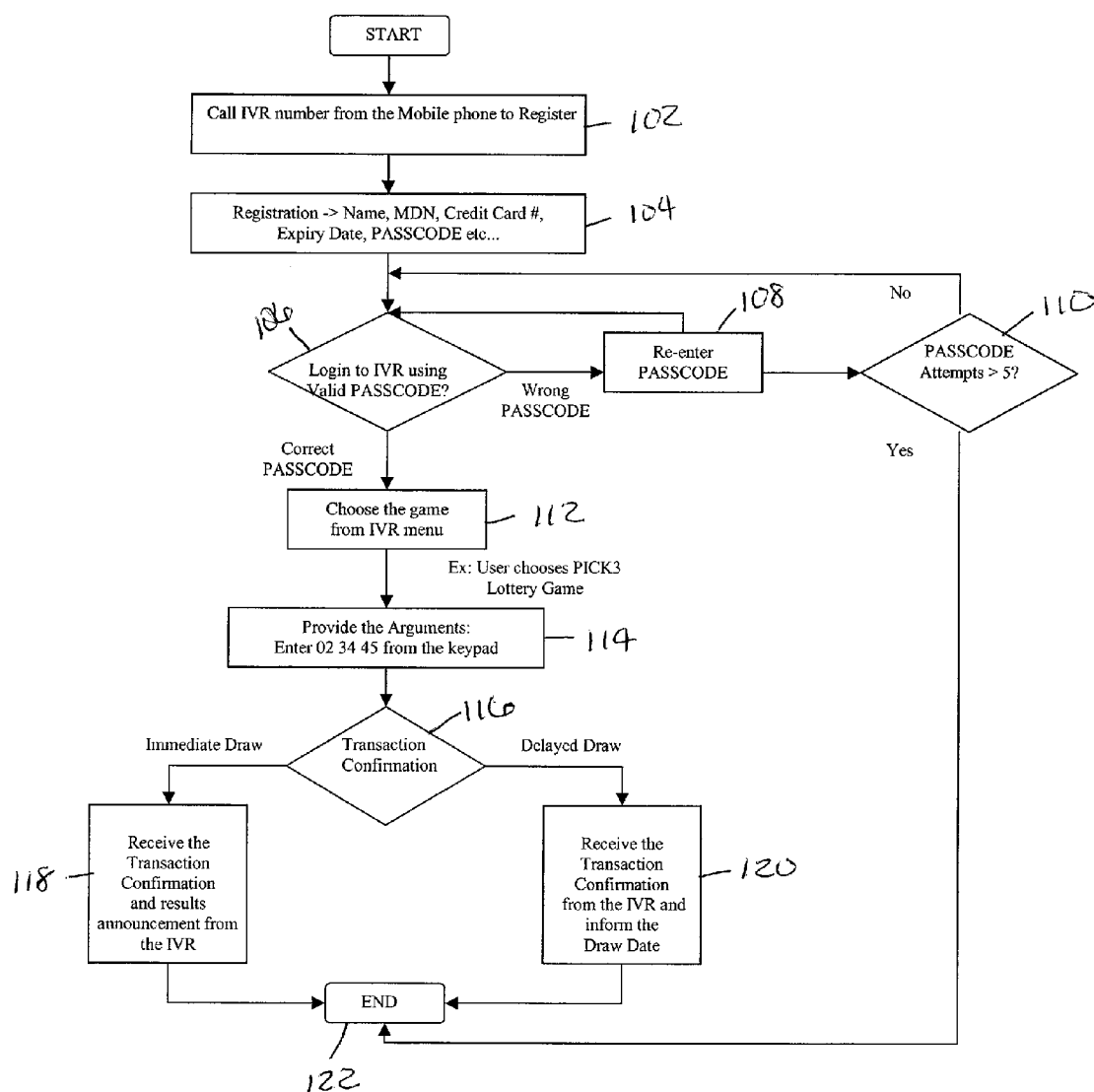
FIGURE 7: Flow chart for IVR based gaming

FIGURE 8: Flow chart for WAP based gaming

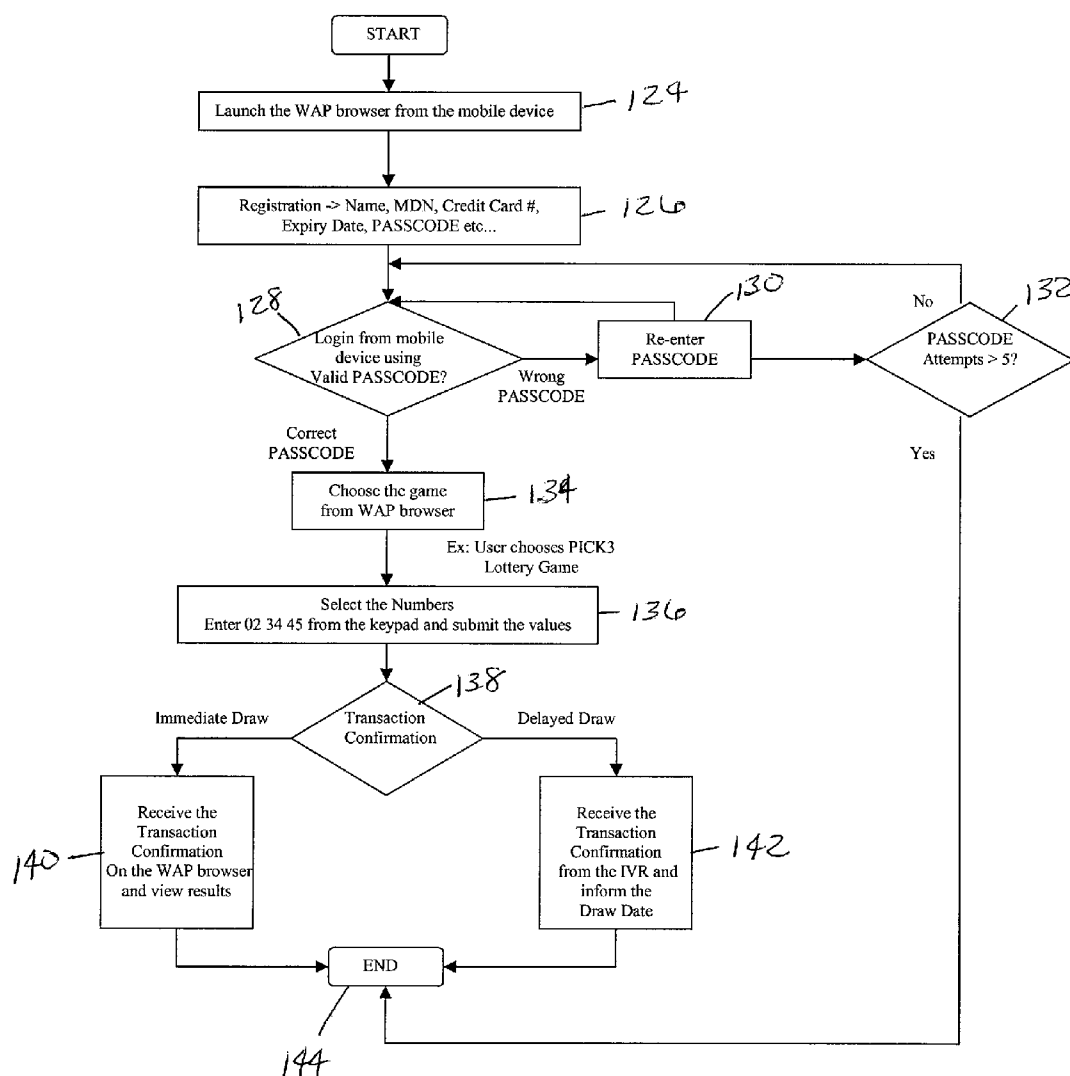
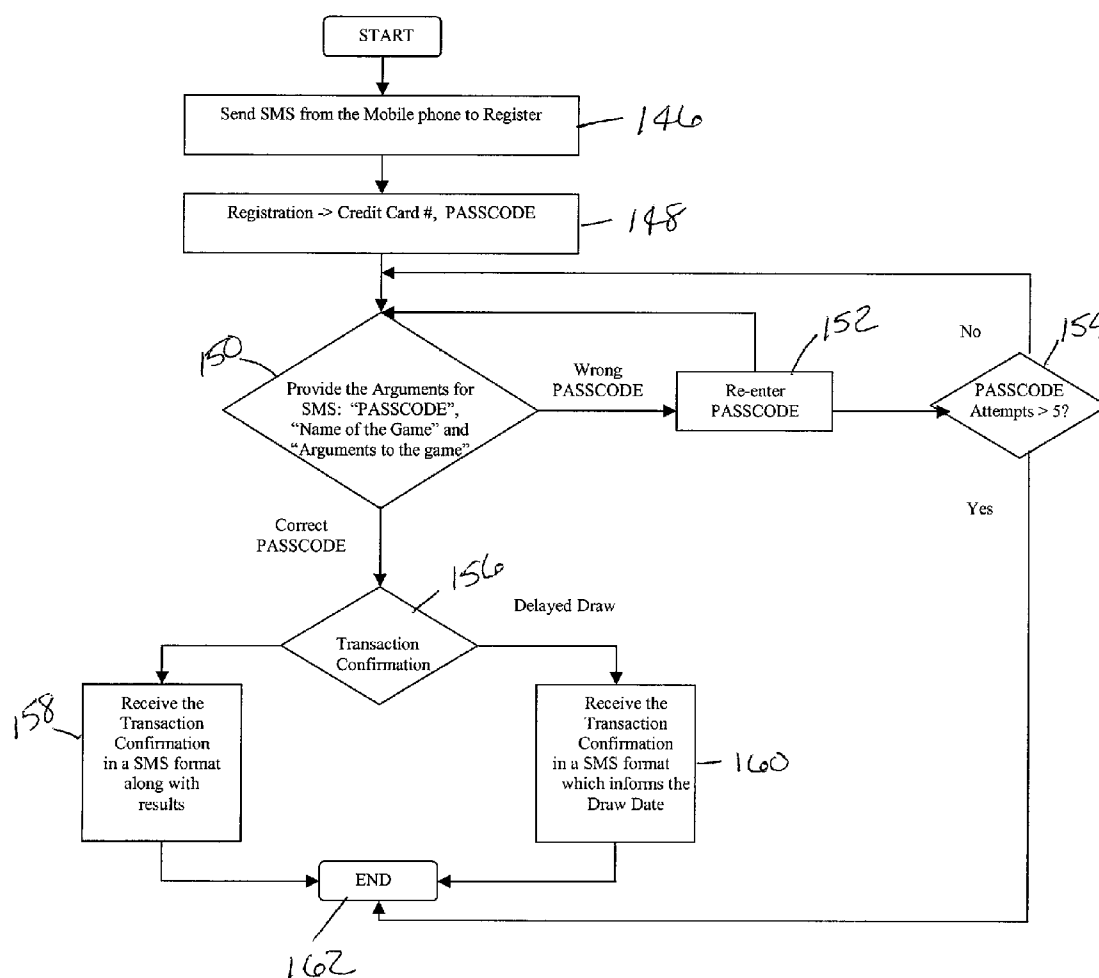


FIGURE 9: Flow chart for SMS based gaming

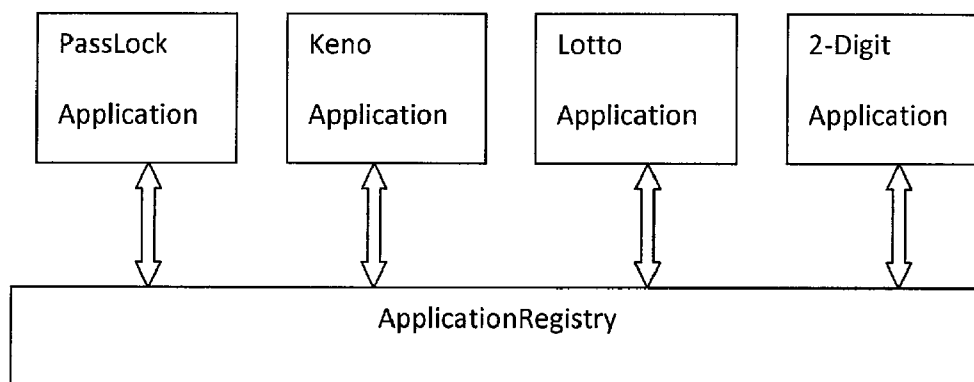


Figure 10

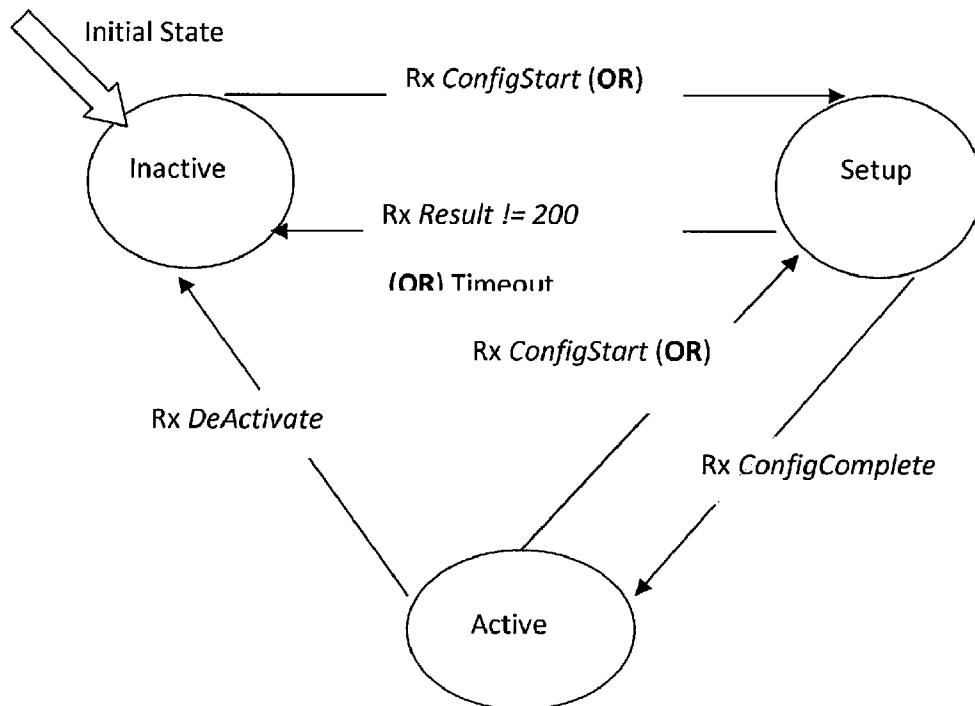


Figure 11

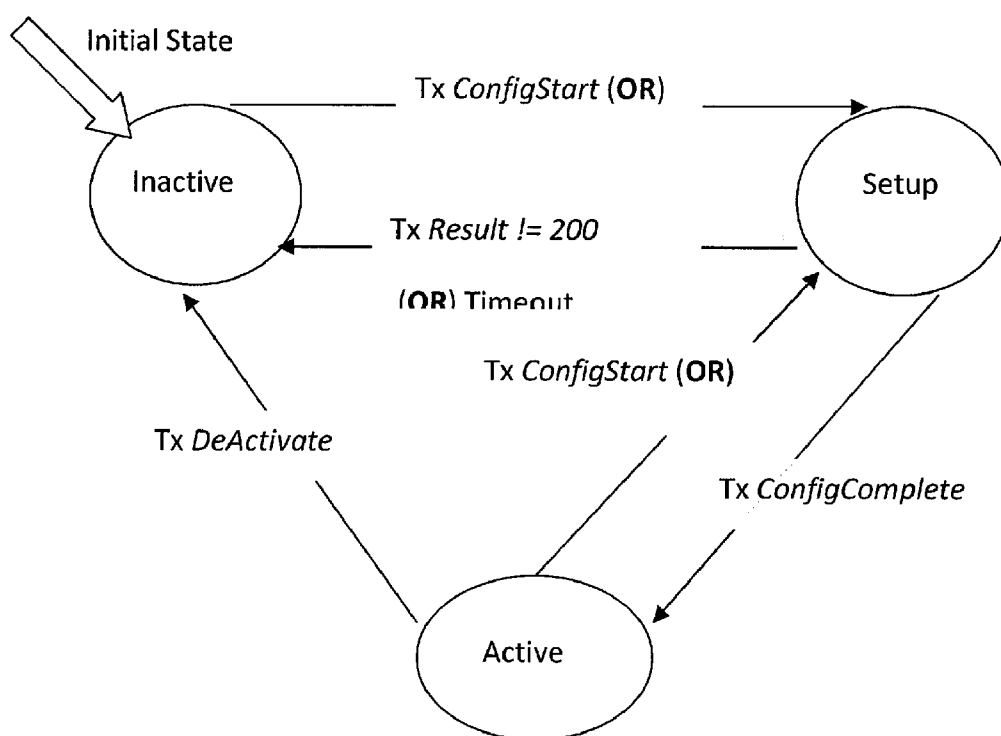


Figure 12

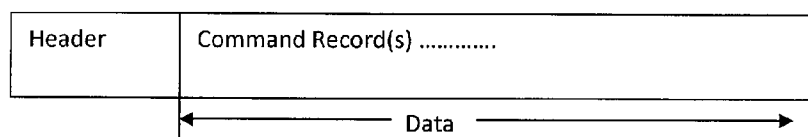


Figure 13

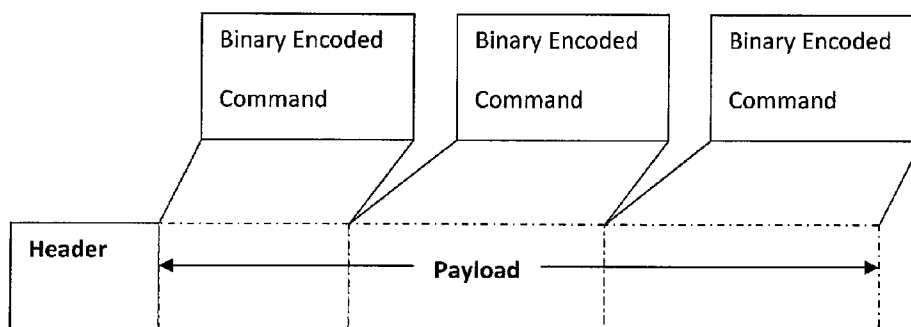


Figure 14

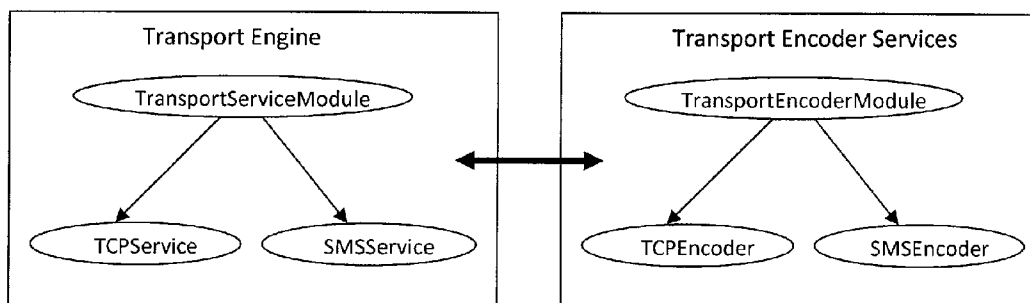


Figure 15

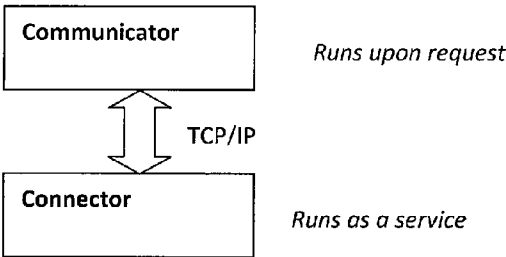


Figure 16

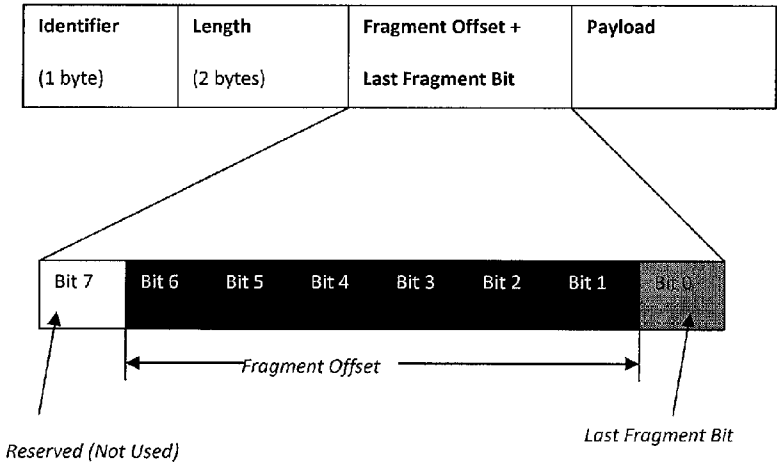


Figure 17

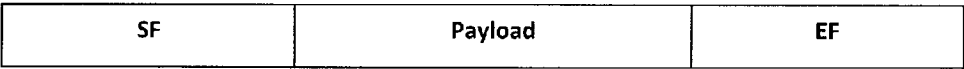


Figure 18

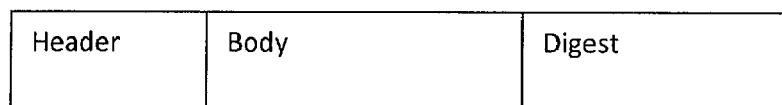


Figure 19

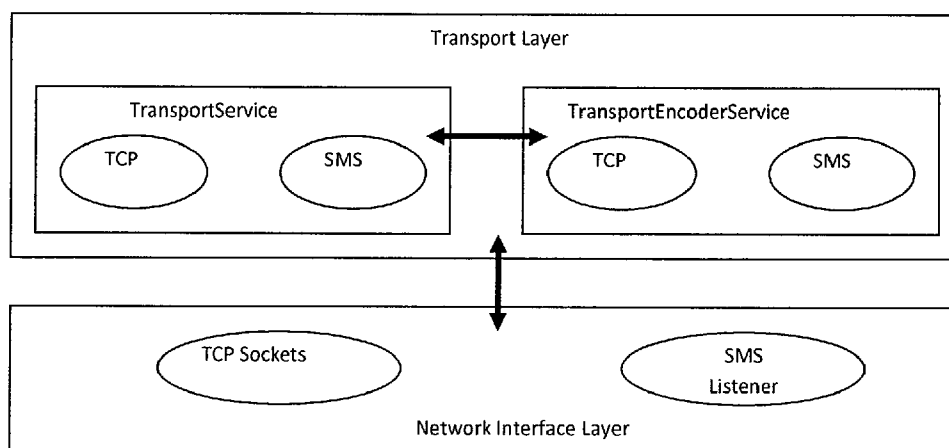


Figure 20

MOBILE GAMING SYSTEM

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application is a continuation-in-part of U.S. patent application Ser. No. 12/099,901 filed Apr. 9, 2008, entitled Mobile Gaming System, the contents of which is incorporated herein by reference.

FIELD OF THE INVENTION

[0002] The current invention relates to a mobile gaming system on wireless networks and a method associated therewith. More particularly the present invention relates to lottery gaming on wireless communication devices and methods of implementing these games.

BACKGROUND OF THE INVENTION

[0003] There are many ways of playing lottery games. The most popular is purchasing a lottery ticket from a vendor at a local store or other establishment. The results of the lottery can be tracked by the purchaser, after the actual lottery, by watch in TV, listing to the radio or seeing the results posted at the establishments which sell the lottery tickets.

[0004] This conventional method of playing lottery games is burdensome to today's highly mobile individual. This method requires the individual to go out of their way to find a vendor or establishment that sells lottery tickets. The purchaser must normally wait in line to make his or her purchase of one or more tickets. In addition, if an individual is resting at home or is at work and decides to play the lottery he or she must complete the task at hand and travel to a location which sells lottery tickets, stand in line and purchase the lottery tickets.

[0005] The reduction in size of the personal computer has resulted in handheld Personal Digital Assistants (PDA) and cellular telephone with the capability to download games and other types of software. These devices have enabled a user to download many popular games such as solitaire, poker, video slots and other lottery games. At first these games were embedded in the devices by the Original Equipment Manufacturer (OEM). Later the games could be purchased as software and loaded into personal computers. From these PCs they could be downloaded through a cable connection to the PDAs. Eventually wireless communications replaced hard wire connections and the games could be wirelessly downloaded into the handheld devices and cell phones.

[0006] There are currently many different handheld devices and cellular phones which employ many different platforms on which these devices operate. At this time there are no standards between the platforms and therefore the games must be specifically designed for the individual platforms. This is very costly. The different platforms may not allow a game designed for a certain platform to operate correctly on another platform. In addition, if an individual decides to utilize a different PDA or cellular phone then they must repurchase all the games they had in their previous PDA or cell phone. This becomes prohibitively expensive and may sway the individual's decision to purchase the game in the first place.

[0007] There are also differences between some devices and the resources available for the devices such as Windows® for PCs and OSX® for Apple computers. The games must include software which is compatible with these different

operating systems. These differences further add to the incompatibility between different handheld devices and games which can be downloaded onto these devices.

[0008] When cellular phones are employed as a platform there are a further set of game distribution challenges. The cellular phone companies normally employ different transmission standards for their wireless networks which are only compatible with phones specifically designed to operate with these standards. Therefore, to be compatible with the different cell phones the games must be designed to operate with each of these different standards.

[0009] Mobile lottery gaming on a Wireless Mobile device/handset are played using the technologies supported by the device. The games may be installed over the air, they may be loaded onto the device with a cable, or they may be embedded on the handheld devices by the OEM or by the mobile operator. There are many challenges with mobile lottery game development including nonstandard platforms with too many platform specific incompatibilities, multiple platforms with inter-operability issues, differences in the devices and their resources, and distribution challenges to deploy with cell phone networks.

SUMMARY OF THE INVENTION

[0010] The disclosed embodiments provide a system and method for providing lottery like games over a wireless network. The system and method permit a user of a mobile wireless device to play a lottery game electronically with the feel of a real game. Once a user is registered onto the system they can utilize an identification code to authorize the downloading of lottery games from a server to their mobile device. A financial account enables the user to electronically pay for the lottery games. Notification of the results of the lottery games can be made by e-mail, SMS or an IVR call.

[0011] Accordingly, it is an objective of the instant invention to provide a system for playing lottery like games over a wireless network utilizing a mobile wireless device which the user currently owns.

[0012] It is a further objective of the instant invention to provide a system for playing lottery like games over a wireless network wherein a proprietary mobile device is not required for playing mobile lottery games or different games that are available over the wireless network.

[0013] It is another objective of the instant invention to allow the user the easiest mode of playing the lottery game that the user is familiar with.

[0014] It is yet another objective of the instant invention to provide a system for playing lottery like games over a wireless network utilizing an application based system.

[0015] It is a still further objective of the invention to provide a system for playing lottery games over a wireless network utilizing an interactive voice response system.

[0016] It is still yet another objective of the instant invention to provide a system for playing lottery like games over a wireless network utilizing a WAP (Wireless Application Protocols) browser to allow a mobile phone to access the Internet.

[0017] Other objects and advantages of this invention will become apparent from the following description taken in conjunction with any accompanying drawings wherein are set forth, by way of illustration and example, certain embodiments of this invention. Any drawings contained herein con-

stitute a part of this specification and include exemplary embodiments of the present invention and illustrate various objects and features thereof.

BRIEF DESCRIPTION OF THE FIGURES

- [0018] FIG. 1 is an overall view of the architecture for a mobile gaming system;
- [0019] FIG. 2 is a diagram of a the architecture for a client application gaming system;
- [0020] FIG. 3 is a diagram of the architecture for a gaming server application gaming system;
- [0021] FIG. 4 is a diagram of a model view controller;
- [0022] FIG. 5 is a registration form on a website;
- [0023] FIG. 6 is a flowchart for a application based gaming system;
- [0024] FIG. 7 is a flowchart for an IVR based gaming system;
- [0025] FIG. 8 is a flowchart for a WAP based gaming system;
- [0026] FIG. 9 is a flowchart for a SMS based gaming system;
- [0027] FIG. 10 is an illustrative diagram of the Application-Registry interface;
- [0028] FIG. 11 is a client state transition diagram;
- [0029] FIG. 12 is a server state transition diagram;
- [0030] FIG. 13 is a structure of binary encoding;
- [0031] FIG. 14 is an illustration of a header+payload;
- [0032] FIG. 15 is an illustration of the Transport Engine and Transport Encoding Services;
- [0033] FIG. 16 is an illustration of two layers communicating via TCP/IP;
- [0034] FIG. 17 is an illustration of SMS datagrams;
- [0035] FIG. 18 is a format for the payload;
- [0036] FIG. 19 depicts the client/server message structure; and
- [0037] FIG. 20 depicts the transport layer.

DETAILED DESCRIPTION OF THE INVENTION

[0038] A presently preferred embodiment of the instant invention is an application based mobile lottery gaming system. This system permits the player or user to have a GUI (graphic user interface) based interface with a mobile device to play games and contact a server. The mobile device can be, but are not limited to, a PDA (personal digital assistant) with a wireless connection, a laptop computer with a wireless connection or a cell (mobile) phone. The GUI interface permits the user/player to establish a more secured, easy to learn and interruptible connection with a server. An example of a mobile lottery gaming system is illustrated in FIG. 1 depicting an overall architecture of the system. The system will be designed such that various client services (banking, wallet, and gaming) can employ the existing infrastructure to perform its functions without compromising on essential features such as ease of use and security. A mobile device 11 communicates with an access network 13. The access network communicates over the Internet 15 by coupling through a radio access network to an application gateway 19 via a PDSN 21. The Application gateway 19 coupled to a core banking module 23.

[0039] Referring to FIG. 6, the gaming application is launched 10 on a mobile device which allows the user to register. The registration process 12 allows the user establish an account with a server. The user can then login using their

passcode and/or selected image 14. If the incorrect passcode and/or image is entered 16 the user is instructed to enter this information again. After a number of unsuccessful login attempts, for example 5, the user is sent to the end of the program and not permitted to play the game 18. If the user wants to continue they must repeat the registration process 12 again. Once the user has logged in they can choose the game they wish to play for the gaming application menu 20. For example the user may wish to play a PICK 3 Lottery game. Next at 22 the user will provide the arguments or select the number they wish to play on the keypad of the mobile device or other data entering device. For example the user may enter "02", "34" and "45" on the keypad. The transaction is confirmed at step 24. Depending on the game which is selected there will either be an immediate draw of the winning numbers 26 or a date will be given on which the winning numbers are to be drawn 28. If the drawing is at a later date the user can check back on their mobile device for the winning number or can check other sources such as the television, a lottery terminal, the newspaper, etc. The gaming application is then ended at 30.

[0040] The architecture for application based lottery gaming is designed to accommodate various application modules. The preferred embodiment is divided into two models. A Client Application model and a Gaming Server model.

[0041] The Application model architecture is designed as a stack of protocols. FIG. 2 is an example of the Client Application model architecture. There are four layers in this model. Each layer has a set of functions to perform. The application layer 32 consists of multiple game applications 34, 36, 37, 38, etc. The applications are connected to the lower layers via two way communication for their function. Each of these game applications will be implemented based on a Model-View-Controller (MVC) paradigm. The MVC diagram is illustrated in FIG. 4. The Model 40 holds the data for the application. The View 42 is responsible to display the data to the user. The Controller 46 registers and receives events from the application router. Upon receipt of the event, the controller 46 will modify the data in the model 40 and notifies the view 42. The view 42 will update its display of data to the user. The applications will avail of the pre-negotiated layer services that it wishes to employ, such as security services (NULL encryption, Custom or SSL Encryption), Message Encoding (ComML or Binary Encoding) and transport mechanism (TCP/IP or SMS). An instance of each of these services will be created upon request.

[0042] The presentation layer 50 consists of essentially 3 components; Cryptographic Services 52, an Application Router 54 and Message Encoding Services 56.

[0043] Cryptographic services 52. The Security Services comprise a group of security modules which provide the required security during a data communication. It includes the most commonly used cryptographic services. Application router 54 receives data from both the application layer 32 and the transport layer 58. An application registers for events from the application router 54. An application will send a message via the application router. The application router will then invoke the Message Encoder module 56 for appropriate encoding, such as ComML. It then invokes the appropriate security module for encrypting the ComML message. The encrypted payload is then passed on to the transport layer 58. The application encoder facilitates encoding and decoding of messages which are invoked by the application router. In a preferred embodiment the encoder supports ComML encod-

ing. Alternative encoding schemes can also be supported in this framework. Client/Server communication can take place by means of protocol. However, the protocol also needs to follow the pre-agreed data encoding in order to properly communicate. Supports two such data encoding schemes: Binary Encoding and ComML Encoding. The Presentation Layer maintains a State Machine in order to maintain shared State between the Client and the Server.

[0044] The Application Router is responsible for routing Commands to a specific module, as well as ensuring that Commands sent out from the modules are correctly encapsulated and the proper encryption procedures are applied. An application registers for events from the application router. An application will send a message via the application router. The application router will then invoke the Message Encoder module for appropriate encoding (such as ComML). The application router then invokes the appropriate security module for encrypting the ComML message. The encrypted payload is then passed on to the transport layer.

[0045] The Transport layer **58** utilizes a transport engine **60** for communication between the application router **54** a network interface **62**, and a transport encoding services **61** and a gaming server **63** using GPRS/1xEVDO. The Network interface layer **62** uses the underlying Air Interface technology for the mobile device to communicate with the gaming server. It can be either a GPRS network or a 1xEV-DO network.

[0046] Referring to FIGS. **11** and **12**, a state machine stores the status of client and server at a given time and can operate on input to change the status and/or cause an action or output to take place for any given change. Client and Server maintain shared states between themselves. The State Machine for the Client and Server will reside at the Presentation Layer. At any given instance of time the client will be in the following three states:

[0047] Inactive State: This state is the initial state of the Client upon installation. The server instance will also be in the Inactive State. In this state, the device can either send a RequestConfig Command to the server or wait for a ConfigurationStart Command.

[0048] Setup State: In this state, the device and the network are in the processing of negotiating various modules and their parameters. The configuration can only be done by the server and the client can accept it, reject it or propose an alternative. If the Setup state fails, it will transition back to Inactive State.

[0049] Active State: In this state, the device and network are ready to exchange application data. The server is ready to accept Commands on behalf of applications in order to process it via 3rd party application servers.

[0050] A Server architecture is illustrated in FIG. **3**. The server architecture is designed as a stack of protocols. There are four layers in the game server model. Each layer has a set of functions to perform. The application layer **64** consists of multiple application services, each serving a specific application or game **66**, **68**, **70**, **71** etc. Each application service communicates with a backend gaming system **72** designed for a specific game, which is part of the business layer. The backend gaming system **72** employs authentication service **73** based upon a database **65**.

[0051] The presentation layer **76** also consists of 3 components: Cryptographic Services **81**, an Application Router **83** and Message Encoding Services **85**. Cryptographic services **81** comprise a group of security modules which provide the required security during a data communication. Application router **83** receives data from both the application layer **64** and

the transport layer **78**. An application will send a message via the application router. The application router will then invoke the Message Encoder module **85** for appropriate encoding, such as ComML. It then invokes the appropriate security module for encrypting the ComML message. The encrypted payload is then passed on to the transport layer **78**.

[0052] Message encoding services **85** facilitates encoding and decoding of messages which are invoked by the application router. In a preferred embodiment the encoder supports ComML encoding. Alternative encoding schemes can also be supported in this framework.

[0053] The Transport layer **78** utilizes a transport engine **87** for communication between the application router **85** a network interface **80**, and a transport encoding services **89**, in the server architecture the interface is preferably by Ethernet **91**.

[0054] Below is an example of the ComML encoding used between the client architecture and the server architecture.

```
<!-- Root or Element -->
<!ELEMENT ComML (ComHdr, ComBody)>
<!ELEMENT ComHdr (ProtoVersion, Random, SessionID, MsgSeq,
Destination)>
<!ELEMENT ComBody ((ConfigurationStart | RequestConfig |
Configure | Commit | ConfigurationComplete | Page | Get |
Put | Alert | Results)+, Final?)>
<!--Protocol Version. The protocol version supported by the
communicating parties-->
<!ELEMENT ProtoVersion (#PCDATA)>
<!--Session ID. Each session can comprise of a multiple client/server
interactions.
They will be represented by this unique Session ID-->
<!ELEMENT SessionID (#PCDATA)>
<!--Message Sequence Number-->
<!ELEMENT MsgSeq (#PCDATA)>
<!--Message Sequence Number-->
<!ELEMENT Random (#PCDATA)>
<!--The Destination of the Message-->
<!ELEMENT Destination (Type, SubType)>
<!ELEMENT Type (#PCDATA)>
<!ELEMENT SubType (#PCDATA)>
<!--ConfigurationStart is sent by the Server to the client.
ConfigurationStart will contain a MessageID, and Entity List-->
<!ELEMENT ConfigurationStart ((CmdID, Entity+))>
<!--RequestConfig is sent by the Client to the Server.
RequestConfig will contain a MessageID, and Entity List-->
<!ELEMENT RequestConfig ((CmdID, Entity+))>
<!--ConfigurationComplete is sent by the Server to the client.
ConfigurationComplete will contain a MessageID, and Entity List-->
<!ELEMENT ConfigurationComplete ((CmdID, Entity+))>
<!--Configure is sent by the Server to the client.
Configure will contain a MessageID, and Entity List-->
<!ELEMENT Configure ((CmdID, Entity+))>
<!--Commit is sent by the Server to the client.
Commit will contain a MessageID, and Entity List-->
<!ELEMENT Commit ((CmdID, Entity+))>
<!--Alerts are sent by the Server to the client upon registration by the
Client. Alerts will contain a CommandID, and Entity List-->
<!ELEMENT Alert ((CmdID, Entity+))>
<!--Page is sent by the Server, using the SMS mode to request the Client
for a Connection Setup. Page will contain a CommandID, and
Entity List-->
<!ELEMENT Page (CmdID, Entity+)>
<!--Get Command can be used either by the Client or the Server.
Get will contain a CommandID, Optional Credentials and ItemList-->
<!ELEMENT Get ((CmdID, Entity+))>
<!--Put Command can be used either by the Client or the Server.
Put will contain a CommandID, Optional Credentials and ItemList-->
<!ELEMENT Put ((CmdID, Entity+))>
<!--Results will contain the execution result of Get and Put Command .
The CmdID MUST match that of command relates to.
Results will contain a CommandID, Optional Credentials and
ItemList-->
```

-continued

```

<!ELEMENT Results (CmdID, MsgRef, Status)>
<!-- Entity element type -->
<!ELEMENT Entity (ItemEntity?, CustomEntity? )>
<!ELEMENT ItemEntity (Name, Meta, Data)>
<!ELEMENT CustomEntity (Meta, Data)>
<!-- Contains the Status Code-->
<!-- Represents the CommandID for the Data -->
<!ELEMENT CmdID (#PCDATA)>
<!-- Represents the MessageReference for the Data -->
<!ELEMENT MsgRef (#PCDATA)>
<!ELEMENT Status (Meta, Data)>
<!-- Represents the Meta Type for the Data -->
<!ELEMENT Meta (#PCDATA)>
<!-- Represents the Data -->
<!ELEMENT Data (#PCDATA)>
<!--The Final Node denotes the Final Transaction for this session. Any
new communication MUST begin as a new Session-->
<!ELEMENT Final (EMPTY)>
<!-- End of Body Definitions- -->
<!-- 2nd Level Definitions for Hdr - -->
<!-- Application Target. Used by the Application Router - -->
<!ELEMENT AppTarget (AppID, AppName?)>
<!-- Application Source. Used by the Application Router - -->
<!ELEMENT AppSource (AppID, AppName?)>
<!-- A Unique Application ID is used to distinguish each Application - -->
<!ELEMENT AppID (#PCDATA)>
<!-- Optional Application Name - -->
<!ELEMENT AppName (#PCDATA)>
<!-- Data represents the actual Data for an element - -->
<!ELEMENT Data (#PCDATA)>
<!-- Item element type - -->
<!ELEMENT Item (Meta?, Data)>
<!-- Contains the Status Code - -->
<!ELEMENT Status (#PCDATA)>
<!-- End of 2nd Level Definitions - -->

```

[0055] The Protocol Layers for both Client and Server architecture is further described as follows. Referring to FIG. 10, the Application Layer consists of various modules (gaming modules, banking modules, etc) which provide user interaction on the Client side. The Server Application Layer modules will provide services to the modules on the Client.

[0056] The ApplicationRegistry is the interface for all the Application Modules to the Router. All application modules MUST register with the Registry. The Application Layer, which comprises of the AppRegistry along with all the application modules, is collectively called as “The Communicator”. The Presentation Layer modules, the Transport Layer modules are collectively called as “The Connector”.

[0057] The Application Registry also controls the invocation of various application modules. When the user invokes the application, an instance of the Application Registry is created. The Application Registry in-turn invokes the default application, the Default application module for the Application registry is the GI_Gaming.Console. This module, upon instantiation, challenges the user for a 4 digit PASSCODE.

[0058] The application layer will consist of multiple applications. The application will avail of the lower layers for its functions. Each application will be implemented based on a Model-View-Controller (MVC) paradigm.

[0059] For the Server Side: The Application Layer consists of multiple application services, each servicing a specific application or game. Each of the application service communicates with a backend gaming system designed for a specific game, which is part of the business layer.

[0060] The Application Encoder facilitates encoding and decoding of messages which are invoked by the Application router. Currently the encoder supports ComML encoding.

Alternative encoding schemes may also be supported in this framework including Binary Encoding and ComML Encoding. The structure of binary encoding is shown in FIG. 13. Referring to FIG. 14, set forth is the header+payload.

[0061] ComML Encoding: The ComML consists of ComHdr and ComBody

[0062] ComHdr: Header Encoding Protocol Version: Presently in Use: 1

[0063] Random: (1 byte) Each entity (client or server) must generate a 1 byte random number using a cryptographically secure pseudo-random number generator for every transaction.

[0064] Session ID (4 bytes) Each communication between the client and server comprises of a unique session ID. The Session ID SHALL be of 4 bytes in length. A general convention to construct a unique session ID is as follows: Each session ID generated by the client MUST be an odd number; Each session ID generated by the Server MUST be an even number; Server and Client MUST remember the last session ID generated; All Session IDs transmitted must be in network byte order

[0065] Message Sequence (1 byte) Each message within a session MUST have a unique Message sequence. The originator of the session will begin with a message sequence 1. Subsequently, all messages will have an incremental message sequence number. For Eg: (if client is the originator) Client→Server Message Sequence: 1 (for message 1); Server→Client Message Sequence: 2 (for message 2).

[0066] Destination Type (1 byte) The Destination Module Type refers to the Type of the Module to which the Command (s) is destined to. Refer to Table 8.1 for a list of Destination Element.

[0067] Destination Subtype (2 bytes) This represents the Destination Subtype Module. Refer to Table 8.2 for a list of Destination Instances. Both the Destination Type and the Destination Subtype are used by the application router on either side to route messages to the destination module.

[0068] ComBody—Body of the information/message sent. The ComBody consists of commands destined to a specific module, based on the Destination Type and Destination Subtype.

[0069] The Transport Layer provides transparent transfer of data between end systems and move PDUs of data between the two communicating systems. The transport service is said to perform “peer to peer” communication, with the remote (peer) transport entity. Referring to FIG. 15, the Transport Layer consists of 2 components: Transport Engine & Transport Encoding Services.

[0070] The Transport Engine takes care of transmission and reception of the payload. Currently TCP/IP and SMS transport engine instances are supported. TCP/IP transport engine will make use of either TCP or UDP sockets for communication. Below is the Transport Layer Module Description.

[0071] TransportServiceModule Type: It is a Module Type that provides transport related services to the upper layer. Module Types are either defined as interfaces or as abstract classes.

[0072] TCPService Module Subtype: It is a class inherited from the TransportServiceModule Type. Objects are instantiated from this Subtype whose reference will be of TransportServiceModule Type. This instance will be instantiated when the client server communication uses TCP transport Service.

[0073] SMSService Module Subtype: It is a class inherited from the TransportServiceModule Type. Objects are instantiated from this Subtype whose reference will be of TransportServiceModule Type. This instance will be instantiated when the client server communication uses SMS transport Service.

[0074] The Transport Encoding Services is responsible for encoding and decoding of the transport layer payload. An instance of the required encoder is created by the respective engine.

[0075] TransportEncoderModule Type: It is a Module Type that provides transport related services to the upper layer. Module Types are either defined as interfaces or as abstract classes.

[0076] TCPEncoder Module Subtype: It is a class inherited from the TransportServiceModule Type. Objects are instantiated from this Subtype whose reference will be of TransportServiceModule Type. This instance will be instantiated when the client server communication uses TCP transport Service.

[0077] SMSEncoder Module Subtype: It is a class inherited from the TransportServiceModule Type. Objects are instantiated from this Subtype whose reference will be of TransportServiceModule Type. This instance will be instantiated when the client server communication uses SMS transport Service.

[0078] Client Inter-Process Communication. The application on the client is typically broken down into two distinct components: The Connector, which runs as a service and The Communicator which runs upon request, either by the user (opening the application) or by the Connector (when an asynchronous message or event is received). The Connector implements the Presentation Layer and the transport layer, whereas The Communicator implements the application layer. The two layers communicate via TCP/IP as shown in FIG. 16.

[0079] Common Client and Server Components: The Transport Layer consists of the following components: SMSListener, which listens to incoming SMS messages; and TCP/UDP Sockets, which makes TCP/UDP Connection.

[0080] Referring to FIG. 17, the SMS Listener will listen for incoming SMS datagrams and will also be responsible in sending out an SMS datagrams to the destination. It takes care of fragmentation of SMS Packet during transmission and de-fragmentation to create an SMS Packet during receipt.

[0081] Identifier: A unique identifier is given by the sender to an SMS Packet. Each SMS Datagram, which comprises a part of the SMS Packet, will contain the same identifier. This will be used by the receiver for de-fragmentation. Length: The Length of the SMS Datagram payload. Fragment Offset: Represents the 6 bits (Bit 1 through Bit 6) that represents the offset into the fragmented SMS packet. The sender starts the fragment with count=0 for the 1st fragment. Last Fragment Bit: This flag is set to 1 if the SMS message represents the last fragment. All other fragments will have the value=0. Reserved—Currently not used.

[0082] SMS Packet: SMS Packet is defined as a complete entity which can be possibly fragmented. The upper layers will provide an SMS packet to be transmitted to the SMSListener. SMS Datagram: SMS Datagram is defined as a payload of 1 SMS message. It could comprise of a fragmented SMS Packet or a complete SMS packet, depending upon the size of the SMS Packet. SMS Message: SMS Message comprises of header and payload. The payload will be and SMS Datagram.

[0083] The TCP/UDP Sockets takes care of the TCP/IP communication between the client and the server. It makes use of TCP reliable stream communication. Since TCP is employed, fragmentation, retransmission, etc will be taken care by the TCP stack implementation on a given platform. The Data format for the payload is shown in FIG. 18 will be as follows: Start Flag (SF): this flag indicates the start of the TCP frame; End Flag (EF): this flag indicates end of TCP frame; Payload: Information/Data.

[0084] The Client and Server will communicate using either ComML or Binary encoding scheme, the client/server messages will contain the structure shown in FIG. 19. The data representation contains a Header section and a Body Section.

[0085] The Business Layer (server only) consists of entities that contain the business logic of various games and applications. The application layer will interact with business layer modules to provide a comprehensive service to clients.

[0086] Referring to FIG. 20, the Network Interface layer is the most basic network layer, providing only the means of transmitting raw bits rather than packets over a physical data link connecting network nodes. Network Interface Layer gets the data from the upper layer (Transport Layer) and makes a peer to peer connection between the client and server using TCP (Sockets) or SMS (Binary Format) to send and receive data.

[0087] An example of the above applications will now be described. The user registers on the gaming server registration website for the first time. During registration the user will be asked to select a Passcode or Password that will be used for all future transactions. FIG. 5 is an example of a registration form template. The user will enter their name at 82. The user will then enter their mobile device identification number, for example their cell phone number, at 84. They can enter any other mobile identification number which the system can recognize. Next the user will enter their credit card number at 86. Next the expiration date of the credit card will be entered at 88. The user will next enter his/her date of birth at 90. The user will then select a passcode or password at 92. An image to be associated with their passcode/password can then be selected at 94. An agreement stating the rules and conditions of the game can then be presented to the user. If the user agrees to these conditions they will check the “Verify Agreement” box 96. The agreement is optional and may not be employed. If the user is satisfied with all the information they entered, they will then click on the “Accept” button 98. If they need to change some of the information they will click the “Reject” button 100 and enter new information. As part of the registration process the client and the server will follow the Asymmetric key encryption procedure of Passcode/Password verification. The server will send the encrypted XML configuration file to the client immediately after successful registration. This method will remove the need for the client application to communicate with the server each time the user logs in. The user will enter the Passcode/Password. The server will receive the Passcode/Password and encrypt it using the Asymmetric encryption algorithm. The server will then send the digitally signed XML configuration file with all the information including its Public Key, Digital signature and he encrypted Passcode/Password to the client. The client will store this information in their mobile device. Every time the user logs in, the client application prompts for a Passcode/Password. The Passcode/Password verification will happen at

the client side since the XML configuration (encryption details) file is already present in the client's device.

[0088] The client application can be deployed in the following ways. The user can receive a SMS with a HTTP/WAP link. With a HTTP link the user can directly click on the link and download the application to the mobile device and install the application. With a WAP link the user can open a WAP browser and download the application and install it. The user can also insert a smart card into the mobile device with the client application loaded into it.

[0089] After the client application has been installed the user can launch the application to start playing the games. In the application menu on the client device the user can choose from the different games which are available. Once the user chooses the game from the menu the user then logs in by using his/her Passcode/Password each time before playing. Once the user starts playing the lottery game every state of the game will be stored in the client application and then communicated to the server. The communication language used for this communication was described in the application layer description above.

[0090] During the registration process the mode of payment which the user wishes to employ will be stored along with all relevant details into the database. The user's credit card will be charged based on the number of lottery games that are played. If the user does not provide the correct details for payment the application will not allow the user to launch the game. The criteria for validation will be done by a 3rd party system verifying the authenticity of the credit card payment mode.

[0091] Confirmation of transactions. For every transaction made the user will be notified with following details. Transaction ID, date of transaction, transaction amount and date of lottery draw for games where the winner is not announced immediately.

[0092] All the winners will be notified utilizing the following modes of communication. E-mail, SMS, an IVR (interactive voice response).

[0093] The following steps illustrate the process for application based gaming.

[0094] Step 1. The user will register on the gaming server website for the first time, see FIG. 5.

[0095] Step 2. The user will provide all the required information. USER NAME, MOBILE NUMBER, CREDIT CARD NUMBER, EXPIRY DATE, DATE OF BIRTH, IMAGE and PASSCODE/PASSWORD.

[0096] Step 3. After successful registration the user will login using the registered Passcode/Password. Step 4. After successful login the application will provide a list of games supported from which the user can select one from the menu.

[0097] Step 5. The user can start playing the game using the mobile device keypad or stylus.

[0098] Step 6. Once the game is complete the user will receive a notification of the transaction.

[0099] Step 7. The user will receive a confirmation of the results via e-mail, SMS or an IVR call.

[0100] Another embodiment of the present invention is IVR based mobile lottery gaming. IVR (interactive voice response) is a general term for mobile phone-based voice value-added services. By dialing a designated number, mobile phone users are able to listen to or send information or even participate in interactive activities like lottery games.

The following process explains the details of how IVR can be used to play mobile lottery games. FIG. 7 is a flowchart illustrating this process.

[0101] An IVR number is called from a mobile device which allows the user to register **102**. The registration process **104** allows the user establish an account with a server. The user can then login using their passcode and/or selected image **106**. If the incorrect passcode and/or image is entered **108** the user is instructed to enter this information again. After a number of unsuccessful login attempts, for example 5, the user is sent to the end of the program and not permitted to play the game **110**. If the user wants to continue they must repeat the registration process **104** again. Once the user has logged in they can choose the game they wish to play from the IVR menu **112**. For example the user may wish to play a PICK 3 Lottery game. Next at **114** the user will provide the arguments or select the numbers they wish to play on the keypad of the mobile device or other data entering device. For example the user may enter "02", "34" and "45" on the keypad. The transaction is confirmed at step **116**. Depending on the game which is selected there will either be an immediate draw of the winning numbers **118** or a date will be given on which the winning numbers are to be drawn **120**. If the drawing is at a later date the user can check back on their mobile device for the winning number or can check other sources such as the television, a lottery terminal, the newspaper, etc. The gaming application is then ended at **122**.

[0102] A detailed description of the IVR based gaming process is as follows. The user registers of the gaming server registration website for the first time. During registration the user will be asked to select a Passcode/Password that will be used for all future transactions. FIG. 5 illustrates the registration website. The following details are required to be provided by the user for registration. User name, mobile device number, credit card number, expiry date, date of birth and passcode/password.

[0103] After successful registration the user can start playing the lottery game. The user will call the toll free IVR number for playing the lottery games. For example, when the user calls the toll free number (1-800-555-5555) the user will be connected to the IVR server. The user will be greeted by voice prompts. The user will follow the voice prompts and enter the required details accordingly. The IVR will prompt the user to choose the game from the available menu. The prompts can be as follows: 1—PICK 3; 2—PICK 6; 3—LOTTOPLUS; 4—KENOPLUS; 5—SCRATCH & WIN; etc. If the user selects 1 the game will be PICK 3. Every time the user wants to play the Passcode/Password must be entered. The Passcode/Password will be received by the IVR in the form of DTMF signals or Voice recognition.

[0104] To start the game the user provides the relevant inputs for the game selected and completes the game. For example, if the user wants to play PICK 3, the user will pick three numbers as inputs to the game. The user then enters the three numbers or says the three numbers (IVR with voice recognition) after the prompts. During the registration process the mode of payment that the user wishes to use will be stored along with all relevant details into the database. The user's credit card will be charged based on the number of lottery games that are played. If the user does not provide the correct details for payment the application will not permit the user to play the game. The criteria for validation will be done by a 3rd party system verifying the authenticity of the credit card payment mode. For every transaction made the user will

be notified with the following details: Transaction ID, date of transaction, transaction amount and date of lottery draw (for games where the winner is not announced immediately). All of the winners will be notified in the following modes of communication: e-mail, SMS, an IVR call, etc.

[0105] The following steps illustrate IVR based gaming.

[0106] Step 1. The user calls the IVR to register for mobile lottery gaming.

[0107] Step 2. IVR provides all the details about the mobile lottery gaming and information required from the user to register like name, mobile number, credit card number, expiry date, age and passcode/password.

[0108] Step 3. After successful registration the user logs in using the registered Passcode/Password.

[0109] Step 4. After successful login the IVR system will provide the list of games supported from which the user can select on to play.

[0110] Step 5. Based on the lottery game selected, the user can provide the arguments of numbers as described in FIG. 7. The user should enter the arguments or numbers from the mobile device keypad and the IVR receives the arguments through DTMF signals. The user and the IVR can also exchange information by interactive voice recognition and response.

[0111] Step 6. Once the entry is recorded in the server the user will receive a notification of the transaction.

[0112] Another embodiment of the present invention is WAP based lottery gaming. WAP is more than just browser software for a phone. It is a complete suite of protocols that tries to provide its capabilities on a multitude of data-based wireless protocols. The WAP browser provides media to access the Internet on the mobile device by taking into account the limited resources available in the mobile device. The following process explains the details of how WAP based lottery gaming can be used to play mobile lottery games. FIG. 8 is a flowchart illustrating this process.

[0113] A WAP browser is launched from the mobile device **124**. The registration process **126** allows the user establish an account with a server. The user can then login using their passcode and/or selected image **128**. If the incorrect passcode and/or image is entered **130** the user is instructed to enter this information again. After a number of unsuccessful login attempts, for example 5, the user is sent to the end of the program and not permitted to play the game **132**. If the user wants to continue they must repeat the registration process **126** again. Once the user has logged in they can choose the game they wish to play from the WAP menu **134**. For example the user may wish to play a PICK 3 Lottery game. Next at **136** the user will provide the arguments or select the numbers they wish to play on the keypad of the mobile device or other data entering device. For example the user may enter "02", "34" and "45" on the keypad. The transaction is confirmed at step **138**. Depending on the game which is selected there will either be an immediate draw of the winning numbers **140** or a date will be given on which the winning numbers are to be drawn **142**. If the drawing is at a later date the user can check back on their mobile device for the winning number or can check other sources such as the television, a lottery terminal, the newspaper, etc. The gaming application is then ended at **144**.

[0114] A detailed description of WAP based lottery gaming is as follows. The user registers of the gaming server registration website for the first time. During registration the user will be asked to select a Passcode/Password that will be used

for all future transactions. FIG. 5 is an example of the registration website. The user will be required to provide the following details. The user's name, the mobile device number, a credit card number, the expiry date of the credit card, the user's date of birth and a passcode/password. After successful registration the user will login to the gaming server with the unique Passcode/Password before playing the game. The user will start playing the lottery game through the WAP browser on the mobile device. All the states of the game will be stored in the server until the user logs off the game. During the registration process the mode of payment that the user wishes to utilize will be stored along with all relevant details into the database. The user's credit card will be charged based on the number of lottery games which are played. If the user does not provide the correct details for payment the application will not permit the user to play the game. The criteria for validation will be done by a 3rd party system reifying the authenticity of the credit card payment mode. For every transaction made the user will be notified with the following details: transaction ID, date of transaction, transaction amount and date of lottery draw (for games where the winner is not announced immediately). Once the payment transaction is complete the user will be immediately notified. The winners will be notified by e-mail, SMS, an IVR call, etc.

[0115] The following steps illustrate WAP based gaming.

[0116] Step 1. The user launches the WAP browser to register for the mobile lottery gaming.

[0117] Step 2. The user enters all the information required from the user to register.

[0118] Step 3. After successful registration the user logs in using the registered Passcode/Password.

[0119] Step 4. After successful login the system will provide a list of the games supported from which the user can select a game to play.

[0120] Step 5. Based on the lottery game the user can enter the arguments or numbers in the browser as described above.

[0121] Step 6. Once the entry is recorded in the server the user will receive a notification of the transaction.

[0122] Another embodiment of the present invention is SMS based lottery gaming. Short Message Service (SMS) is a communication protocol allowing the interchange of short text messages between mobile telephony devices. The following process illustrates how mobile phone users can use SMS to play lottery games on the mobile devices. FIG. 9 is a flowchart illustrating SMS based gaming.

[0123] A SMS is sent from the mobile device to register **146**. The registration process **148** allows the user establish an account. The user can then login using their passcode and/or selected image **150**. If the incorrect passcode and/or image is entered **152** the user is instructed to enter this information again. After a number of unsuccessful login attempts, for example 5, the user is sent to the end of the program and not permitted to play the game **154**. If the user wants to continue they must repeat the registration process **148** again. Once the user has logged in they can choose the game they wish to play from the menu at **150**. For example the user may wish to play a PICK 3 Lottery game. Next at **150** the user will provide the arguments or select the numbers they wish to play on the keypad of the mobile device or other data entering device. For example the user may enter "02", "34" and "45" on the keypad. The transaction is confirmed at step **156**. Depending on the game which is selected there will either be an immediate draw of the winning numbers **158** or a date will be given on which the winning numbers are to be drawn **160**. If the

drawing is at a later date the user can check back on their mobile device for the winning number or can check other sources such as the television, a lottery terminal, the newspaper, etc. The gaming application is then ended at 162.

[0124] A detailed description of SMS based gaming is as follows. The user will register of the gaming server registration website for the first time. During registration the user will be asked to select a Passcode/Password that will be used for all future transactions. FIG. 5 illustrates a registration website. The user will be asked to provide the following details. The user's name, the mobile device number, a credit card number, the expiry date of the credit card, the user's date of birth and a passcode/password. The user will send a SMS to the server with the required details for a game. The user's passcode/password, the name of the game and the arguments or selected numbers. For example, if the user wants to play a PICK 3 game, the user will select three numbers and send a SMS in the following text format: "PWD PICK3 23 45 12". During the registration process the mode of payment that the user wishes to utilize will be stored along with all the relevant details into the database. The user's credit card will be charged based on the number of lottery games that are played. If the user does not provide the correct details for payment the application will not permit the user to play the game. The criteria for validation will be done by a 3rd party system verifying the authenticity of the credit card payment mode. For every transaction made the user will be notified with the following details. The transaction ID, the date of transaction, the transaction amount, and the date of the lottery draw (for games where the winner is not announced immediately). The winners will be notified by an e-mail, a SMS an IVR call, etc.

[0125] The following steps illustrate SMS based gaming.

[0126] Step 1. The user sends a SMS to the SMSC gateway register for mobile lottery gaming.

[0127] Step 2. The user registers using a credit card number and PIN number.

[0128] Step 3. After successful registration the user sends a SMS to the SMSC gateway in the following format "PASS-CODE/PASSWORD", "Name of the Game" and "Arguments to the game".

[0129] Step 4. Once the entry is recorded in the server the user will receive a notification of the transaction as a SMS confirmation.

[0130] Step 5. If the lottery game provides immediate results, like a scratch and win game the user will be notified immediately. In the case of a delayed drawing, the user will be notified of the drawing date.

[0131] All patents and publications mentioned in this specification are indicative of the levels of those skilled in the art to which the invention pertains. All patents and publications are herein incorporated by reference to the same extent as if each individual publication was specifically and individually indicated to be incorporated by reference.

[0132] It is to be understood that while a certain form of the invention is illustrated, it is not to be limited to the specific form or arrangement herein described and shown. It will be apparent to those skilled in the art that various changes may be made without departing from the scope of the invention and the invention is not to be considered limited to what is shown and described in the specification and any drawings/figures included herein.

[0133] One skilled in the art will readily appreciate that the present invention is well adapted to carry out the objectives and obtain the ends and advantages mentioned, as well as

those inherent therein. The embodiments, methods, procedures and techniques described herein are presently representative of the preferred embodiments, are intended to be exemplary and are not intended as limitations on the scope. Changes therein and other uses will occur to those skilled in the art which are encompassed within the spirit of the invention and are defined by the scope of the appended claims. Although the invention has been described in connection with specific preferred embodiments, it should be understood that the invention as claimed should not be unduly limited to such specific embodiments. Indeed, various modifications of the described modes for carrying out the invention which are obvious to those skilled in the art are intended to be within the scope of the following claims.

What is claimed is:

1. Mobile lottery gaming on a wireless mobile device comprising:

a mobile device for use in wireless communicating with a remote game server;

said game server having a stack of wireless application protocols including a memory for storing information to identify a mobile device and for routing and parsing of messages to said mobile device;

an application layer residing on each said mobile device containing a number of lottery game applications to be played;

a presentation layer residing on said mobile device, said presentation layer having a cryptographic module to provide security services during data communication and a message encoder for encrypting of data passed through an application router in communication with said application layer; and

a transport layer residing on said mobile device for receipt of data communication from said presentation layer and directing to a network layer permitting two-way communication between said mobile device and said game server.

2. The system of claim 1 wherein said application layer is further defined as a model-view-controller for communicating data between an event and a display on said mobile device.

3. The system of claim 1 wherein said message encoder provides ComML protocol data encoding prior to routing of a message to said transport layer.

4. The system of claim 1 wherein said transport utilizes TCP/IP protocols for communication between an application of said mobile device and said game server.

5. The system of claim 1 wherein said transport utilizes SMS protocols for communication between an application of said mobile device and said game server.

6. The system of claim 1 wherein said application layer communicates with a backend gaming system having authentication services before providing data to said application layer.

7. The system of claim 1 wherein payment authorization can be transmitted from said mobile device to said server.

8. The system of claim 1 including an interactive voice response system utilized to communicate between said mobile device and said server.

9. The system of claim 1 wherein said network layer communicates by a GPRS network or a 1 xEV-DO network.

10. The system of claim 1 wherein a WAP is utilized to communicate between said mobile device and said server to transmit information.

11. The method of claim 1 wherein a short message service is utilized to communicate between said mobile device and said server to play said games.

12. Mobile lottery gaming on a wireless mobile device comprising:

a mobile device for use in wireless communicating with a remote game server;

said game server having a stack of wireless application protocols including a memory for storing information to identify a mobile device and for routing and parsing of messages to said mobile device;

an encrypted file residing on each said mobile device, said encrypted file containing information which identifies each said mobile device to said game server permitting expedited client side login/password credential verification;

an application layer residing on each said mobile device containing a number of lottery game applications to be played, said application layer having a model-view-controller for communicating data between an event and a display on said mobile device;

a presentation layer residing on said mobile device, said presentation layer having a cryptographic module to provide security services during data communication and a message encoder for encrypting of data passed through an application router in communication with said application layer, said message encoder employing a ComML protocol data encoding prior to routing of a message to said transport layer; and

a transport layer residing on said mobile device for receipt of data communication from said presentation layer and directing to a network layer permitting two-way communication between said mobile device and said game server.

13. The system of claim 1 wherein said transport utilizes TCP/IP protocols for communication between an application of said mobile device and said game server.

14. The system of claim 1 wherein said transport utilizes SMS protocols for communication between an application of said mobile device and said game server.

15. The system of claim 1 wherein said application layer communicates with a backend gaming system having authentication services before providing data to said application layer.

16. The system of claim 1 wherein payment authorization can be transmitted from said mobile device to said server.

17. The system of claim 1 including an interactive voice response system utilized to communicate between said mobile device and said server.

18. The system of claim 1 wherein said network layer communicates by a GPRS network or a 1xEV-DO network.

19. The system of claim 1 wherein a WAP is utilized to communicate between said mobile device and said server to transmit information.

20. The method of claim 1 wherein a short message service is utilized to communicate between said mobile device and said server to play said games.

* * * * *