Multimedia Tools and Applications (2022) 81:9989–10005
https://doi.org/10.1007/s11042-022-12130-9

Secured cloud SCADA system implementation for industrial applications

Fawzy A. Osman, et al. *[full author details at the end of the article]*

Received: 14 May 2021 / Revised: 20 August 2021 / Accepted: 3 January 2022 /
Published online: 14 February 2022

Abstract
The proposed Remote SCADA System (RSS) is a smarter, faster and more reliable way to control high power machines and monitor their sensors, data, and failures. The proposed system focuses mainly on building our own complete SCADA software and not using open-source SCADA software. The proposed RSS can use unlimited number of added Remote Terminal Units (RTU) nodes and each of them can handling unlimited number of input/output and can be used under different operating systems like Windows and Android. Using RSS all machines can be monitored and controlled by a single click from anywhere at any time. By doing this a real-time response from RSS system can be achieved. It's mainly based on standard communication techniques between remote nodes and single server-side application that talk to each node with its own ID and modify its instant database. So that every time accessing this web application, a real-time access to these nodes data and a virtual control room controls each General Purpose Input/Output (GPIO) in the selected node can be gotten. When a new event is happened in server-side program, it will be broadcasted to all related. On RSS there are two main points to deal with, the request latency and security of the system. This paper studied how the system latency and security are improved to obtain the needed values. The proposed RSS is a very secure program which have 4 security levels; authentication, authorization, RSA and CBC encryption system. Also, the encryption algorithms used in RSS are RSA and CBC block cipher encryption system. It is mixed way to prevent any attacker from breaking the cipher. First of all, RSA generates the public and private keys, and then CBC generates its Initialization vector and a random encryption key. Then a special function sends all of these keys encrypted with a pre-stored token in the data base and node memory which varied from node to another. Finally, node generates its private key from loaded public key. With this combination the speed of symmetric encryption system and the security of asymmetric encryption system can be achieved. On the other hand, the level of security firewalls needed to be Brocken by the attacker to brock the cipher is increased. The proposed system achieved low cost comparing with reported work; it is lower than Arduino + WIFI method by five times and 13 times lower than Raspberry-PI method. The proposed system is applied in educational systems, where it is used for teaching unlimited number of students Online.

Keywords Remote SCADA System · RSS · RSA and CBC encryption algorithms

1 Introduction

Supervisory Control and Data Acquisition (SCADA) is a combination of software and hardware components which allow monitoring and controlling of industrial automation systems or any process targets locally or remotely [1]. The SCADA systems allow the monitoring and controlling of all the plant process easily and more precisely. By reducing the initial cost in the SCADA software and hardware system, it would be used widely in many areas of application [23]. Also, its cost can be minimized by using open-source software tools to build our private SCADA system. A low cost IoT based SCADA system is presented in many researches. IoT was implemented in an industrial automation system in [5] and conclude that the real time supervising of industrial automation systems via the use of IoT based SCADA has been more efficient for remotely controlling the industrial systems. In [28], a microcontroller (Arduino Uno board) is used with a Zigbee module to implement a wireless network of different types of sensors. In [20], cloud computing technology is implemented by using a microprocessor to connect sensor network by means of Ethernet medium. With the fast development of the wireless technology manufacturing, the microcontroller-based SCADA system is widely used in automation system applications. The microcontroller was used as remote terminal unit for data collection from the sensor network. The Open-source code can be implemented using open-source software SCADA system which is powerful for executing and monitoring the control signals in real time [27]. In [18], an open-source SCADA was implemented using OPC technique and is used for data supervision and controlling of the automation system.

Photo-Voltaic (PV) farm monitoring and controlling using IoT based open-source SCADA system was developed in [2]. In [24], a C# SCADA software was developed as an open-source SCADA system for controlling industrial systems. SCADA systems in the cloud environment are suffering from the security problems. This problem can be solved with the use of a combination of security tools [6, 8]. Also, many researches work show that many applications are migrate to the cloud SCADA system as in [12], which conclude that the cloud SCADA is more interesting for the users, cost effective and maintenance-less. To increase the system efficiency and its incoming benefits an Industrial Internet of Things (IIoT), based SCADA system was introduced in [10, 32]. There are mainly three types of Cloud systems; Public cloud infrastructure, Private cloud and Hybrid cloud infrastructure, which merge the private and public cloud infrastructures. Also, the architecture of a cloud computing system is structured as follows; the hardware layer, infrastructure layer, platform layer and the software layer. The Cloud computing systems faces an additional number of vulnerabilities more than the traditional systems which are described in [3].

Infrastructure as a Service (IaaS) can be sustainable to all of the threats which are known from the classical information and communication protocols. Platform as a Service (PaaS) is particularly differed from the IaaS in its security settings for many kinds of resources. Due to Software as a Service (SaaS) requires only the Web browser and internet medium, its security requirements are similar to the Web service security problems [4, 7, 14, 30]. There are many other problems with securing the data handling on the cloud which includes; data transferring, data management and storing, authentication and authorization etc. [14]. These problems have greeter complex in case of migration from traditional systems to the cloud-based systems. Cyber-attacks on cloud SCADA systems can be classified into three main types: Communication attacks, Software attacks and Hardware attacks as indicated in [9]. Cloud SCADA systems mainly suffer from the same cyber security problems as the other systems which are

operate through the cloud system [11, 21, 29, 34]. There are other threats than in the public cloud environment; First, more exposed to attack due to the sharing of infrastructure with unknown outside parties. Second, outside attackers due to network connections between SCADA systems and the cloud [29]. Third, some of SCADA-specific application layer protocols lack protection [15, 22, 25]. Many researches are deals with ciphertext-policy attribute-based encryption scheme and Multi level key exchange and encryption protocol IOT SCADA system [17, 26, 33].

2 Proposed system

On RSS as shown in Fig. 1, there are two main points to deal with, the latency of the request and security of the system. The proposed system is used to improve these two parameters to obtain the needed values. Nowadays all systems need to be online and remotely controlled by cross-platform applications like RSS, so let's start to get more details about it.

The encryption algorithms used in RSS are RSA and CBC block cipher encryption system. It is a mixed way to prevent any attacker from breaking the cipher. Each two transceiver sides have private key for RSA, public key for RSA, initialization vector for CBC, and encryption key for CBC.

Using these keys, the encryption can be carried out in next steps:

1. CBC encrypts the first block with Initialization vector and encryption key.
2. Then complete the blocks one over one to create a CBC encrypted block chain which can't be broken without IV and encryption key, the encryption algorithm used in CBC core is a simple XOR operation to ensure that data is completely

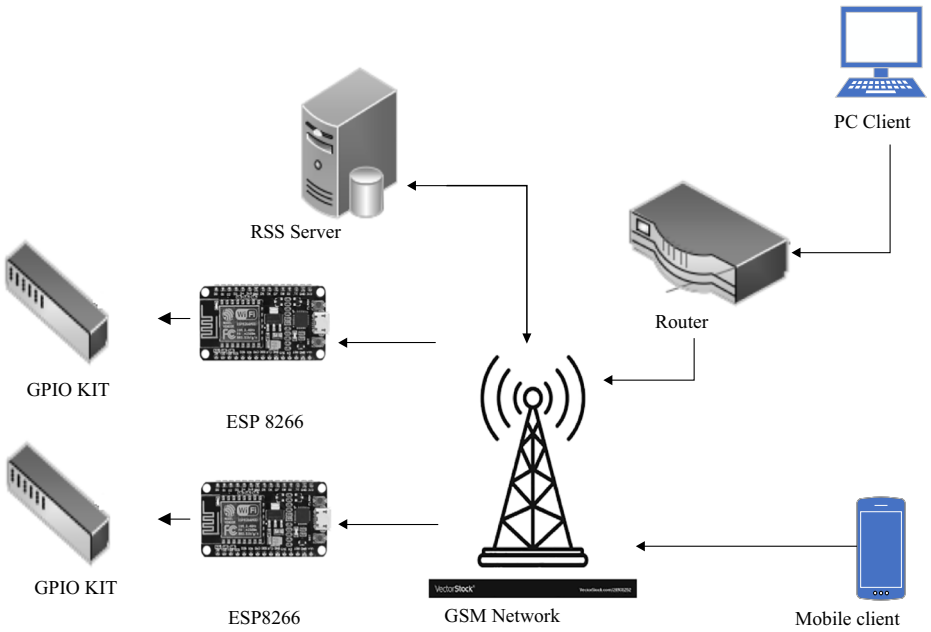

Fig. 1 Remote SCADA System (RSS) Block Diagram

defragmented to a non-readable word stream, no need here for a complex operation in order to speed up the algorithm and pay a load for encryption to the combination of two algorithms used.

3. RSA encrypt the resulted block chain from CBC and send it to the other side.
4. When other side received the encrypted blocks, it decrypts it with the reverse sequence, first it used the RSA private key to decrypt RSA sent cipher then use a CBC decryption model with pre-sent encryption key and IV for CBC.

With this combination, the speed of symmetric encryption system and the security of asymmetric encryption system have been achieved. On the other hand, the level of security firewalls needed to be Brocken by the attacker to brock the cipher has increased.

3 Features of the proposed RSS

Figure 2 shows RSS features which are:

- RSS is user friendly program that requires no pre-knowledge just go to site and start to use.
- RSS is a scalable program which can easily control larger and larger number of nodes on other hand; there is no limit for number of nodes connected to it.
- RSS is based on modular code which simplifies development in the future.
- RSS application is a cross platform application which can be open in any browser on android, IOS, PC or MAC, because it based on responsible page content to modify containers to screen width.
- RSS is a very secure program that has 4 security levels, authentication, authorization, RSA and CBC encryption system.

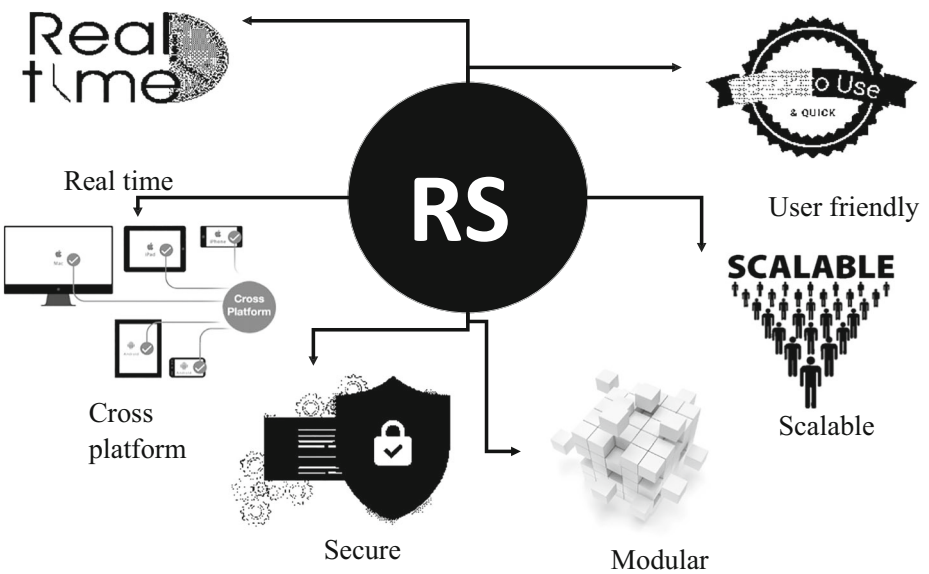

Fig. 2 Remote SCADA System (RSS) features

- RSS provides a real-time response to changes in order to get rid of timing error in emergency cases. Its latency depends only on the speed of the internet access.

4 Required hardware

ESP8266 It is used as a simple hardware construction of a Remote Terminal Unit, RTU, for RSS. ESP8266 is used as a microcontroller with integrated WIFI module to connect the node to global network in order to receive and transmit data to the RSS server remotely. This WIFI module has a variety of GPIOs which can be used in reading analog signal, reading and writing digital signals and PWM signals to control motors and other PWM-controlled devices. The proposed system achieved low cost comparing with reported work; it is lower than Arduino + WI-FI method by five times and 13 times lower than Raspberry-PI method as shown in Table 1 [16, 19].

Router It is used as a WIFI hotspot connected to global network in order to remotely access the node from anywhere. As a proof of concept, it can be easily replaced with a mobile hotspot with 3G open. It means that only one router is used for many nearby nodes including its WIFI range. In proposed system it is preferred to use static IP for each node in order to reach it by the server easily without any verification techniques that consume time, RSS uses both, static routing and repeated requests with current IP.

Server A Server is a computer or a device on the network that manages network resources. Servers are single-tasks device, which perform only their own server tasks. A server in this case could be a program which managing resources rather than the entire computer.

5 Required software

5.1 Block diagram of program cycle

As indicated in Fig. 3 the RTU Node start to send a handshaking request command to the server which start to verify that Node Address exists in our TRU's or not. When the server verified the Node, it starts to send its configuration and the Node start to configure its GPIO modes of operation according to the server configuration. When any changes to any Node GPIO status occurs, it sends another request to the server, if not, it sends frequently request to the server each 500 msec. also, if there any changes in the Node configuration the server send changes request to the Node.

Table 1 Comparison of client side

Method	Cost (LE)	Drawbacks
ESP8266 [proposed]	≈135 LE	Limited No. of ADC
Arduino + WI-FI[16]	≈450LE	High cost, lower reliability and need more power
Raspberry-PI[16]	≈1350LE	High cost, depends on the SD card, limited No. of GPIO, analog I/O through SPI

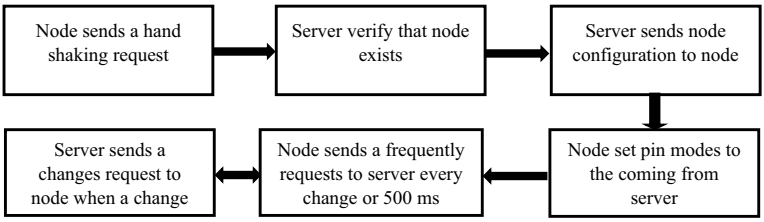

Fig. 3 Block diagram of proposed system program cycle

5.2 Arduino Code flowchart

More details for Arduino Code flowchart can be done as shown in Fig. 4.

Connect to WIFI This function is the responsible for connecting to WIFI with a pre-defined SSID and Password this process repeats itself until it is successfully connected to WIFI. It simply puts the WIFI to WIFI_STA mode and begin to connect to WIFI and get into infinity loop that only be broken by a successful connection to the WIFI, this operation is indicated in the flowchart of Fig. 4 and will be explained in next sections.

Send get request This function is mainly for sending the requests to server and gets response then gets the string body from the response then send it to concern function to deal with it and extract data from it.

```
Void connectTOWIFI (const char *ssid, const char *password)
 {
    WiFi.mode (WIFI_STA);
    delay(1000) ;
     wifi.begin (ssid, password) ;
     Serial. Print1n (" ") ;
     Serial. Print ("Connecting") ;

  while (WiFi.status () != WL_CONNECTED)    {
    de1ay(500) ;
    Seria1. Print (".") ;
     }
     }
```

Initializing pins This function is to initializing pin modes to server configuration requirements that demanded by admin of that node. This function runs only one time at booting time.

Extract-Data This function is responsible for extracting data from substrings from split-string function in order to store it in a new separate array to easily reach it by the functions that control GPIOs.

Combine-data This function is responsible for combining data in order to be sent to the server in a correct pattern to be desterilized there without any loss or error.

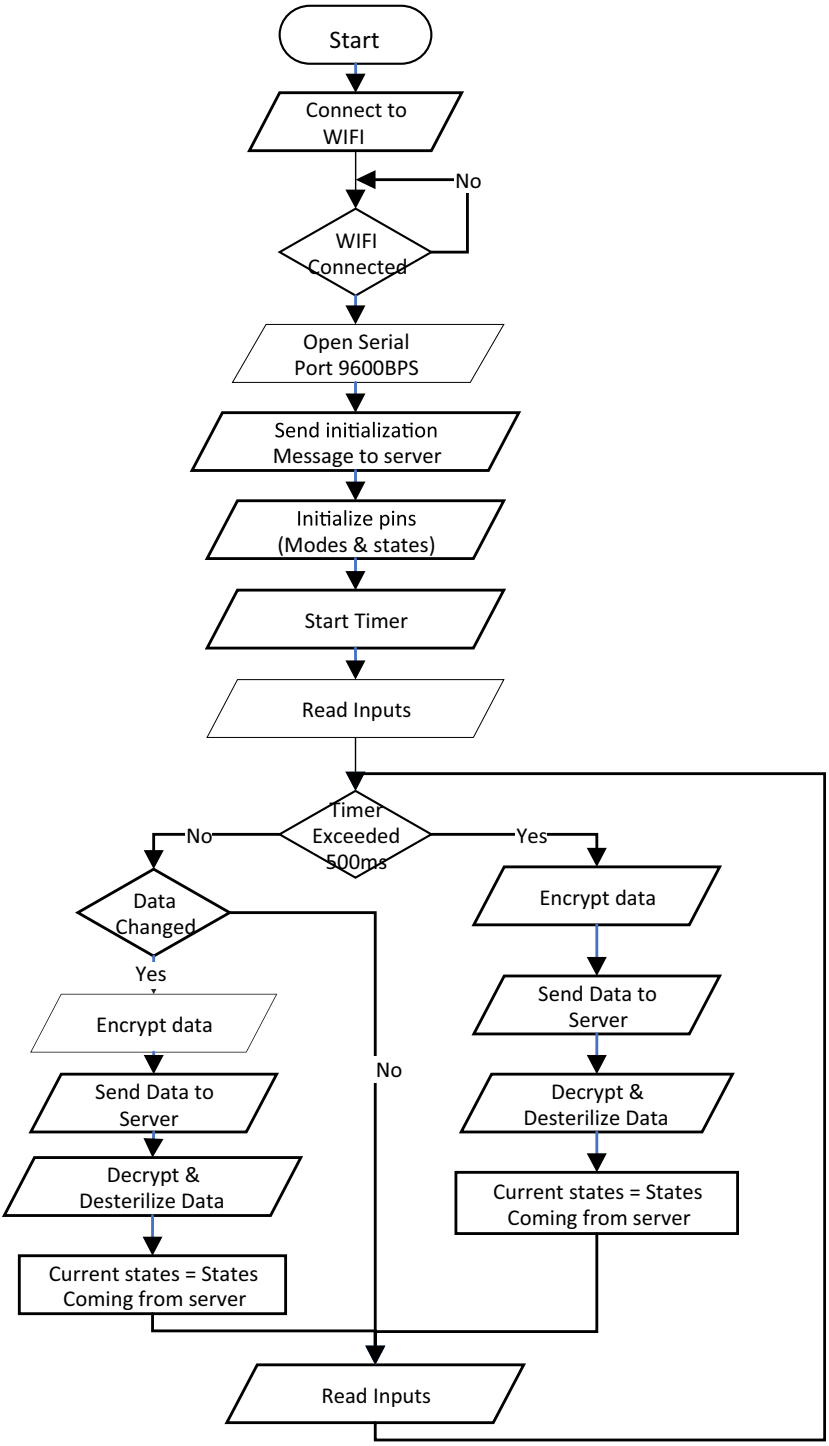

Fig. 4 Arduino Code flowchart

Print-Buff This function is for testing purposes only to print data values coming from server and sent to server in order to be seen by developer when testing or modifying code.

valToStr & StrToVal These two functions are used to string if numeric data to be sent to server and convert strings back to numeric values. This helps in dealing with data in both server and node side in the right way.

Applying changes This function is mainly applying state changes from the server by admin of the nodes. These changes are gotten from the extract data function which stores the data into a new array that can be used.

```
 Void applyChanges (string state[ ] , string value[ ])
{
      For (int  i = 0; i   < DATA_SIZE-2;  i++)
     {
         Serial . println(i);
         If (state [i] == " ") break ;
        else
       {
       }
     }
```

5.3 Web server flowchart

The flowchart of the presented Web-Server illustrated in Fig. 5 and more details for this flowchart can be can be done as:

- Include third-party dependencies.
- Configure Express.js app settings.
- Define middle-wares.
- Define routes.
- Connect to databases (MongoDB).
- Start the application.

5.4 Web application flowchart

A complete flowchart that indicates the steps of the proposed Web Application page are presented in flowchart of Fig. 6 and an illustration for some of the required tools and written code are given below.

EJS: Simply stands for Embedded JavaScript. It is a simple template language/engine that allows the user to generate its own HTML with plain JavaScript. EJS is mostly useful whenever you have to output HTML with a lot of JavaScript if you're dealing with generating dynamic contents or offering something that has to do with real-time updates. EJS is preferred to use due to its simplicity in syntax and very easy to set up.

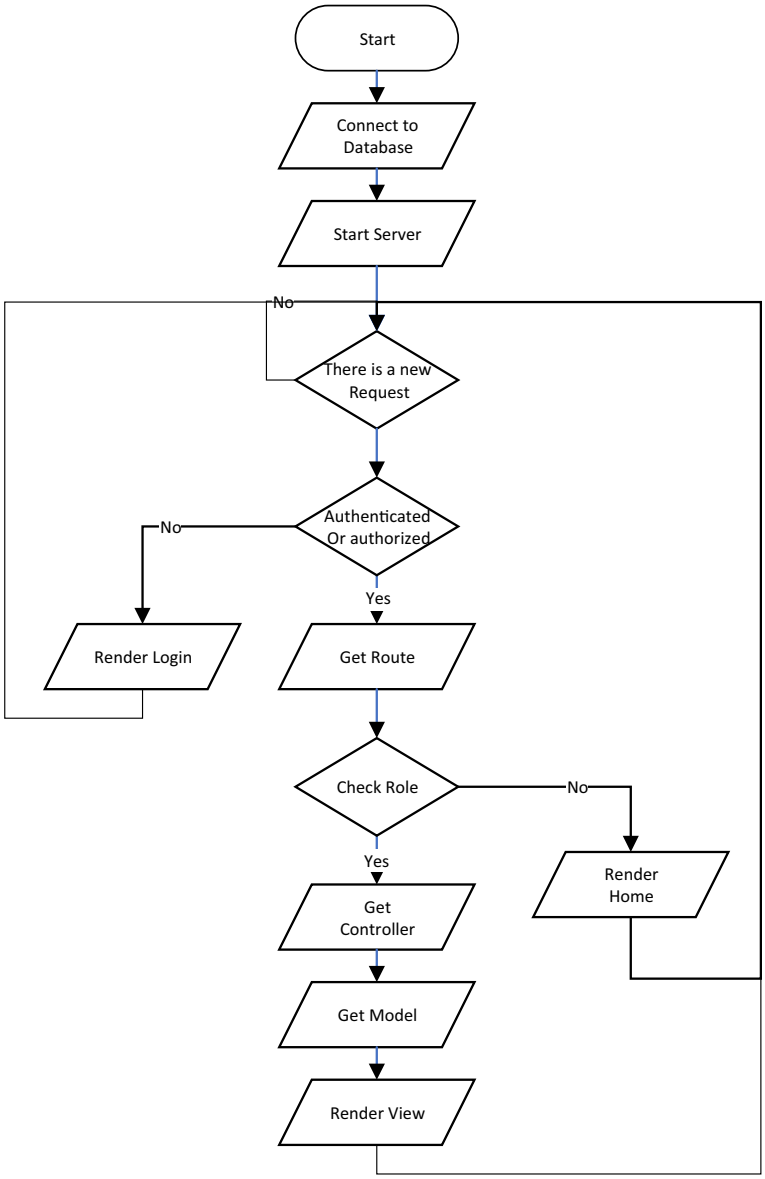

Fig. 5 Web-Server flowchart

CSS: Cascading Style Sheets, CSS, is a simple design language used for making web pages presentable. CSS handles the appearance and performance of the web page. Using CSS, page text color can be controlled, the fonts style, the spacing, columns dimension, what background appearance or colors to be used and other different effects. CSS is easy to use and understand and it provides a helpful control tools for creating an HTML document. CSS is commonly merged with the HTML or XHTML languages.

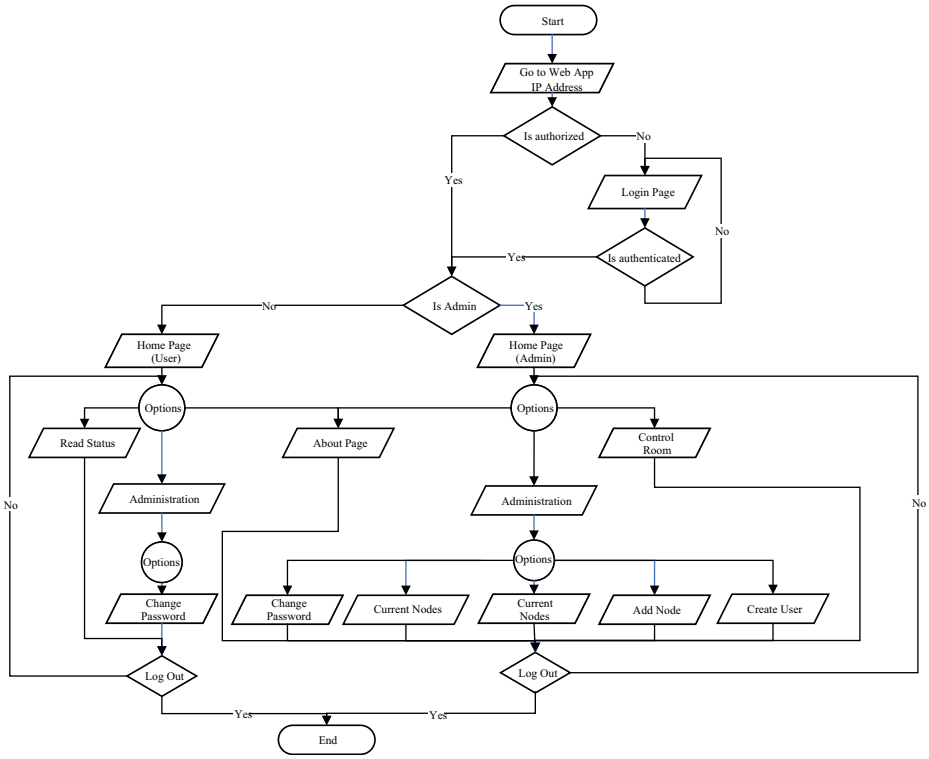

Fig. 6 Flowchart for web Application

```
<!DOCTYPE html>
<html lang="en">
    <%- include('./parts/head') %>
    <body>
        <div class="container-scroller">
            <%- include('./parts/topBar') %>
            <div class="container-fluid page-body-wrapper">
                <%- include('./parts/sideBar') %>
                <div class="main-panel">
                    <div class="content-wrapper">
                        <%#- include('./parts/pageHeader') %>
                        <!-- start page content-->
                        <!-- end of page content-->
                    </div>
                    <div class="col-md-6 d-flex align-items-stretch">
                    <div >
                    <%- include('./parts/footer') %>
                </div>
            </div>
        </div>
        <%- include('./parts/scripts') %>
```

5.5 Database flowchart

MongoDB MongoDB is a document-oriented database that provides high performance, high availability, and easy scalability. Each database creates its own group of files on the system files. A single MongoDB server typically has multiple databases and a complete flowchart for the presented Data-Base files is shown in Fig. 7.

Security:

1. Authentication: In RSS a login name and password for user authentication are used, user name is a symbol-free string notates to the user name and it saved as it is in database, but password is saved as hash in database. Hash function is one-way function which is a hard-loss encryption technique.
2. Authorization: Authorization is granted to the valid authenticate of the users according to their stored information in the Access Management System (AMS). Authorization is more powerful than authentication, especially for widely distributed digital content providers.
3. RSA: RSA encryption function seemed to be the only known way for a trap-door one-way permutation, but now, there are many others certainly existed [13]. The average length size of n must increase with time as more efficient effected factoring algorithms are made and as computers are getting faster. RSA is slower than certain other symmetric cryptosystems [31]. RSA is commonly used as a tool to securely transmit the security keys for another approaches of less secure, but faster algorithm [35].

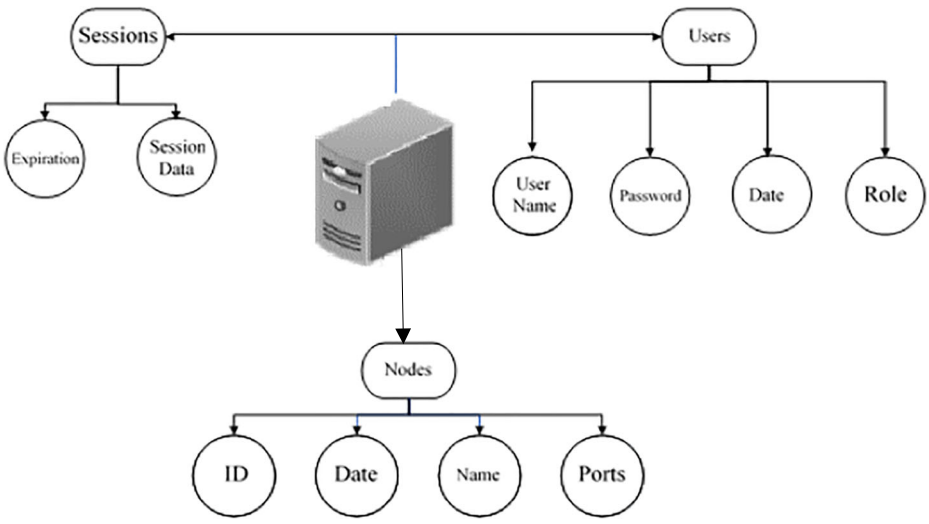

Fig. 7 Data-Base flowchart

Output: public key: *kpud = (n, e) and private key: kpr = (d)*

1. *Choose tow large primes p and q.*
2. *Compute n = p . q.*
3. *Compute* $\Phi(n) = (p-1)(q-1)$.
4. *Select the public exponent e ∈ {1, 2… Φ (n) - 1} such that gcd (e, Φ (n) =1.*
5. *Compute the private key d such that d.e ≡ 1mod Φ (n).*

4. CBC: Cipher block chaining, is a block cipher mode that provides confidentiality but not message integrity in cryptography.

Web server code

```
const express = require("express");
const session = require("express-session");
const SessionStore = require("connect-mongodb-session")(session);
const path = require("path");
const homeRouter = require("./routes/homeRouter");
const bodyParserMW = require("body-parser").urlencoded({
});
app.set("view engine", "ejs");
app.set("views", "views");
app.use(…
);
app.use(express.static(path.join(__dirname, "assets")));
app.use(bodyParserMW);
app.use("/", homeRouter);
app.listen(3000, (err) => {…
});
<exports.addNode = (req, res, next) => {…
  };
<exports.getCurrentNodes = (req, res, next) => {…
};
<exports.getNodeData = (req, res, next) => {…
};
<exports.getNodeJson = (req, res, next) => {…
};
<exports.controlNodeData = (req, res, next) => {…
};
<exports.nodeRequest = (nodeId, data, res) => {…
};
<exports.extractNodeData = (msg) => {…
};
<exports.toggleNodePort = (req, res, next) => {…
};
<exports.updatePWMNode = (req, res, next) => {…
};
```

6 Proposed RSS GUI interface

Figure 8 show a screen shot for administrator user in proposed RSS GUI interface. This administration options are create new user, create new admin, monitoring current users, delet user, create nodes, monitoring and control current nodes, delet node and change password.

Figure 9 shows Create New User by the adminstrator. Admin can be Create New User with a user name and password and User can Change his own user name and password.

Figure 10 shows how to add a new node and how to classify its Input/Output options with the proposed RSS runing under Android Cell Phone.

7 Comparison of server options

A comparison between different implemented Cloud SCADA systems and the proposed Remote SCADA System, RSS, is introduced in Table 2 which shows that the proposed one is more secured, easier to use and more scalable to add unlimited number of RTU nodes with unlimited number of GPIO.

8 Conclusions

The proposed Remote SCADA System (RSS) is a smarter, faster and more reliable way to control high power machines and monitor their sensors, data, and failures. It's mainly based on standard communication techniques between remote nodes and single server-side application that talk to each node with its own ID and modify its instant database so that every time to access this web app you get a real-time access to these nodes data and a virtual control room control each GPIO in the selected node. When a new event happened in server-side program it will be broadcasted to all related nodes to this event. On RSS there are two main points to deal

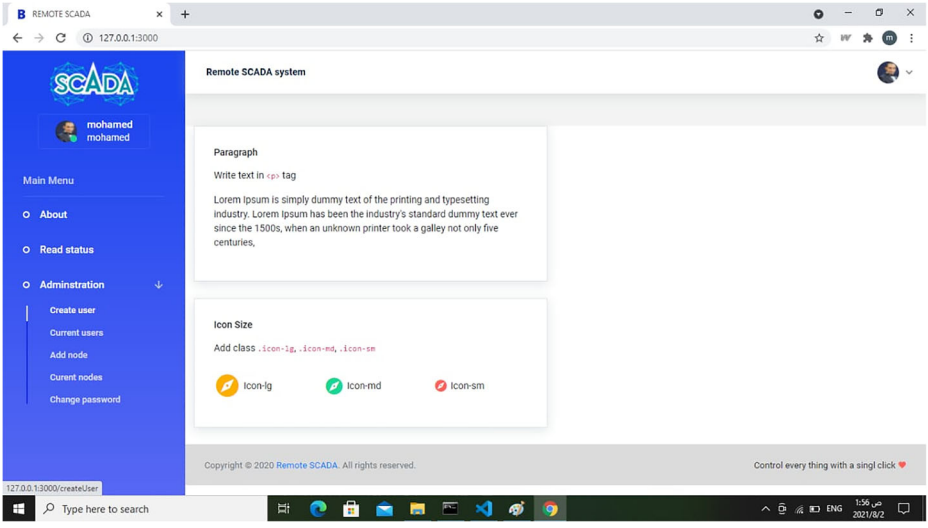

Fig. 8 Adminstration options

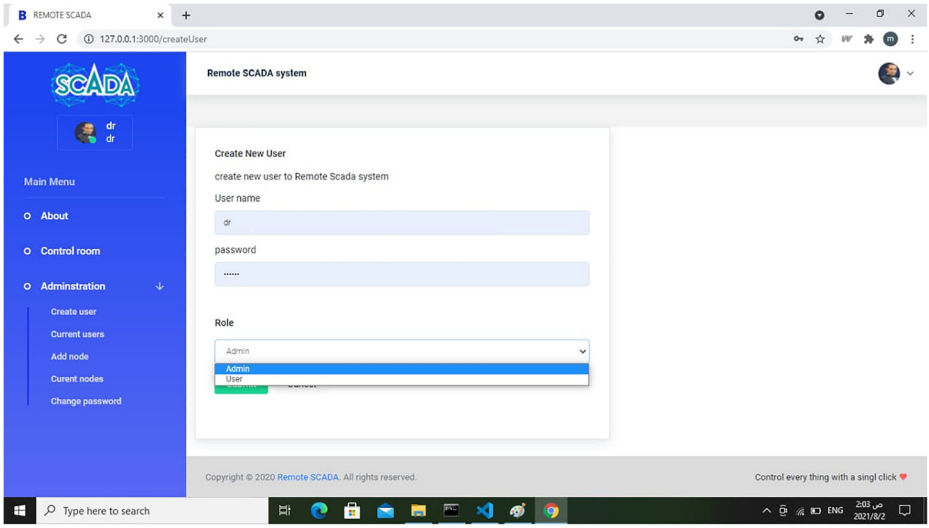

Fig. 9 Create New User

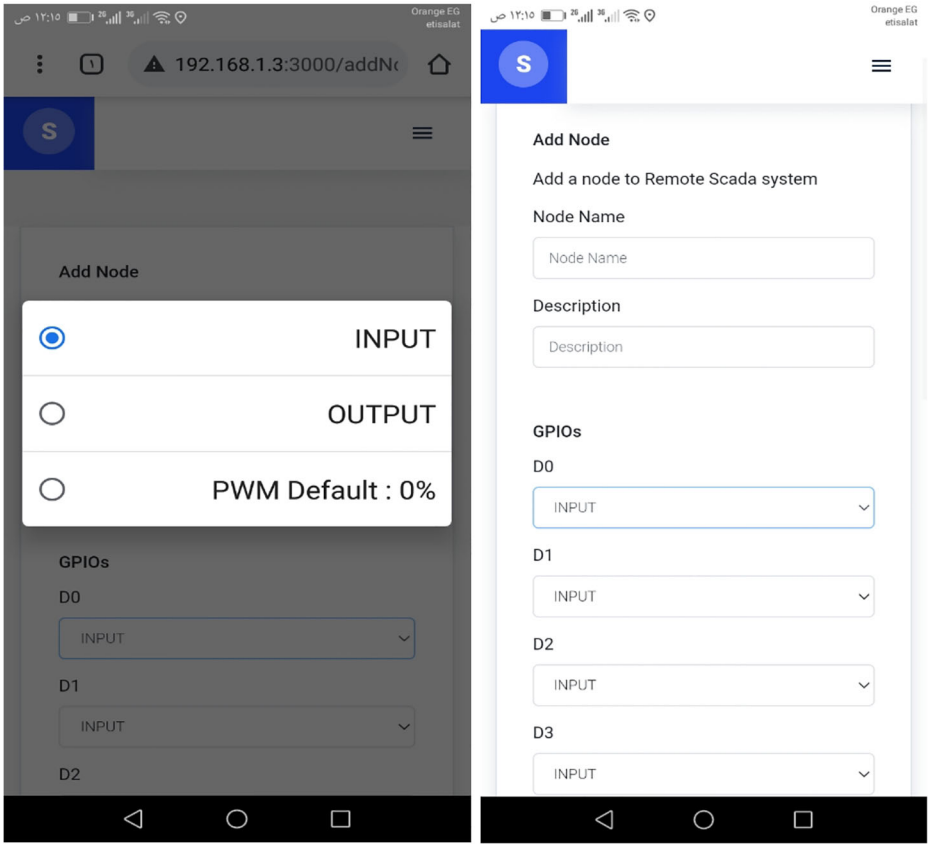

Fig. 10 Add New Node and choose its Options windows under Android

Table 2 Comparison between different implemented Cloud SCADA systems

SCADA server	Security	Installation	Easy of using	Adding new components
Remote desktop Connection with Local data storage [16]	Lower security	Easy to install	Easy to use	Need a special method
Open SCADA software [16]	High security	Medium	Support some protocols	Easy to add new components
IOT service provider	Users have to pay after certain data limit			
Private IOT serve	Secured due to the use of private server	Difficult to install	Easy to use	Easy to add new components.
Proposed Remote SCADA System, (RSS)	High level of security consists of 4 internal layers	No need for installation as it a web-based application	Very easy to use no pre-knowledge needed as it just a web site	Scalable enough to add unlimited number of nodes without any changes to server.

with, the latency of the request and security of the system. These two pointes are covered also here to ensure a higher-level security system. The proposed Remote SCADA System, RSS, is a very secure program that has 4 security levels, authentication, authorization, RSA and CBC encryption system and achieved low cost comparing with reported work. Also, the encryption algorithms used in RSS are RSA and CBC block cipher encryption system. It is mixed way to prevent any attacker from breaking the cipher. First of all, RSA generates the public and private keys; this CBC generates its Initialization vector and a random encryption key. Then a special function sends all these keys encrypted with a presorted token in the data base and node memory which varied from node to another. Finally, node generates its private key from loaded public key. With this combination speed of symmetric encryption system and the security of asymmetric encryption system are achieved. Also, the level of security firewalls needed to be Brocken by the attacker to brock the cipher is increased. The time of user's entry are recorded also, the malfunction node is detected and recorded. The system was verified and tested for many conditions of the security which indicate its features and robustness.

Funding Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB).

References

1. Aghenta LO, Iqbal MT (2019) Low-cost, open source IoT-based SCADA system design using Thinger. IO and ESP32 Thing. Electronics 8(8):822
2. Aghenta LO, Iqbal MT (2019) Development of an IoT Based Open Source SCADA System for PV System Monitoring. IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), pp 1-4
3. Aghenta LO, Iqbal MT (2019) Low-Cost, open source IoT-Based SCADA system design using Thinger.IO and ESP32 Thing. Electronics 8(8):822
4. Aniruddha S, Rumale DN, Chaudhari (2017) Cloud computing: software as a service. 2nd IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)
5. Carlsson O (2015) Enabling large IoT platforms in industrial process automation. Luleå tekniska universitet
6. Chang V, Ramachandran M (2015) Towards achieving Data Security with the Cloud Computing Adoption Framework. IEEE Trans Service Comput 9(1):1–15
7. Chen Y, Chen J, Gan J (2015) Experimental study on cloud computing based electric power SCADA system. ZTE Commun 13(3):33–41
8. Cherdantseva Y, Burnap P, Blyth A (2016) A review of cyber security risk assessment methods for SCADA systems. J Comput Secur 56:1–27
9. Cherdantseva Y, Burnap P, Blyth A, Eden P et. al (2016) A review of cyber security risk assessment methods for SCADA systems. Comput Secur 56:1–27
10. Church P, Mueller H, Ryan C, Gogouvitis SV et al (2017) Migration of a SCADA System to IaaS Clouds – A Case Study. J Cloud Comput Adv Syst Appl 6(11):1–12
11. El Mrabet Z, Kaabouch N, El Ghazi H, El Ghazi H (2018) Cyber-security in smart grid: survey and challenges. Comput Electr Eng 67:469–482
12. Elezia M, Raufia B (2015) Conception of Virtual Private Networks using IPsec suite of protocols, comparative analysis of distributed database queries using different IPsec modes of encryption. Procedia - Soc Behav Sci 195:1938-1948
13. Gupta SC, Sanghi M (2020) On an efficient RSA public key encryption scheme. Malaya J Matematik 8(3): 1138–1141
14. Hari Krishna B, Kiran S, Murali G, Pradeep Kumar R, Reddy (2016) Security issues in service model of cloud computing environment. Procedia Comput Sci 87:246–251
15. Howard PD (2015) A security checklist for SCADA Systems in the Cloud. GCN
16. Jayasinghe S, Iqbal LT, Mann G (2017) Low-cost and open-source SCADA options for remote control and monitoring of inverters. IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)
17. Long M, Kong M, Long S, Zhang X (2020) An improved differential fault analysis on block cipher klein-64. Comput Mater Continua (CMC) 65(2):1425-1436
18. Merchán DF, Peralta JA, Vazquez-Rodas A, Minchala LI, Astudillo-Salinas D (2017) Open source SCADA system for advanced monitoring of industrial processes. International Conference on Information Systems and Computer Science (INCISCOS), pp 160-165
19. Hashem MYM, Osman FA, Eltokhy MAR, Gab Allah AS (2020) Low-cost design and implementation of cloud SCADA system. Eng Res J 166:ELE15-ELE29
20. Mononen T, Mattila J (2017) A low-cost cloud-extended sensor network for supervisory control. IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM), pp 502-507
21. Moustapha Fall C, Chuvalas N, Warning M, Rabiee, Purdy C. IEEE Enhancing SCADA System Security, 978-1-7281-8058-8/20/202
22. Nazir S, Patel S, Patel D (2017) Assessing and augmenting SCADA cyber security: a survey of techniques. Comput Secur 70:436–454
23. Phuyal S, Izykowski J, Bista D, Bista R (2019) Internet of Things in power industry: current scenario of Nepal. International Symposium on Current Research in Hydropower Technologies (CRHT'IX), vol 9
24. Phuyal S, Bista D, Izykowski J, Bista R (2020) Design and Implementation of Cost Efficient SCADA System for Industrial Automation. Int J Eng Manuf 2:15–28
25. Poletykin A (2018) Cyber security risk assessment method for SCADA of industrial control systems. IEEE International Russian Automation Conference (RusAutoCon), pp 1-5
26. Poomagal CT, Sathish Kumar GA, Mehta D (2020) Multi level key exchange and encryption protocol for internet of things (IoT). Comput Syst Sci Eng 35(1):51–63
27. Prokhorov AS, Chudinov MA, Bondarev SE (2018) Control systems software implementation using open source SCADA-system OpenSCADA. IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), pp 220-222

28. Rajkumar RI, Alexander TJ, Devi P (2016) ZigBee based design of low cost SCADA system for industrial process applications. IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), pp 1-4
29. Sajid A, Abbas H, Saleem K (2016) Cloud-assisted IoT-Based SCADA systems security: a review of the state of the art and future challenges. IEEE Access 4:1375–1384
30. Soufiane S, Halima B (2017) SaaS Cloud Security: attacks and proposed solutions. Trans Mach Learn Artif Intell 5(4):291–301
31. Obaid TAS (2020) Study a public key in RSA algorithm. EJERS, Eur J Eng Res Sci 5(4)
32. Tsochev G, Yoshinov R, Zhukova N (2020) Some security issues with the industrial Internet of Things and comparison to SCADA Systems. Труды СПИИРАН. (Том 19 № 2. ISSN 2078-9181)https://doi.org/10.15622/sp.2020.19.2.5
33. Wang C, Yuan Y (2020) An efficient ciphertext-policy attribute-based encryption scheme with policy update. Comput Mater Continua (CMC) 63(2):1031–1041
34. Yadav G, Paul K (2001) Architecture and security of SCADA systems: a review. arXiv: 02925v1 [cs.CR] 9 Jan 2020
35. Yu H, Kim Y (2020) New RSA encryption mechanism using one-time encryption keys and unpredictable bio-signal for wireless communication devices. Electronics 9:246

Affiliations

Fawzy A. Osman [1] · Mohamed Y. M. Hashem [2] · Mostafa A. R. Eltokhy [2]

✉ Mostafa A. R. Eltokhy
mostafaeltokhy2717@yahoo.com

Fawzy A. Osman
fawzi.osman@bhit.bu.edu.eg

Mohamed Y. M. Hashem
m_yusuf7@yahoo.com

[1] Electrical Engineering Department, Faculty of Engineering, Benha University, Cairo, Egypt

[2] Electronics Technology Department, Faculty of Technology and Education, Helwan University, Cairo, Egypt