Technology of Secure Data Exchange in the IoT System

Hassan Mohamed Muhi-Aldeen[1], Yurii Khlaponin[2], Ibtehal Shakir Mahmoud[3], Volodymyr Vyshniakov[4], Vadym Poltorak[5], Dmytro Khlaponin[6], Muwafaq Shyaa Alwan[7]

[2, 4, 6]*Kyiv National University of Construction and Architecture, Kyiv, Ukraine*
[1],[3,7] *Al Iraqia University, Baghdad, Iraq*
[5]*NTUU "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine*

Abstract. The use of public Internet channels for managing objects in the IoT system can lead to the emergence of security threats not only for this IoT system, but also it can provide cybercriminals with resources to carry out attacks on any other objects of the global network. Therefore, you should use secure data exchange technologies that prevent unauthorized entry into the system when building IoT systems. This technology is discussed in detail in this article. The purpose of this work is to improve safety of IoT systems through the use of a perfectly secure data exchange channel.

Keywords
IoT system, safety of IoT systems, secure data exchange technologies, secure data exchange channel.

1. IoT security challenges

In 2020 the number of connected devices to the IoT exceeded 30 billion, and their annual growth increased from 3 billion in 2017 to 5 billion in 2020 as shown by the published data of researchers [1].

Forecasts up to 2025 assume that this growth will not decrease, but tends to increase. This testifies to the rapidly growing need for managing remote sites and ample opportunities for their implementation using existing tools and technologies. However, the rapid growth of needs and the broad possibilities of implementing IoT in a short time often leads to insufficiently thought out solutions from the point of view of security, which is described in [2-4], where security at the network level is attributed to the most vulnerable area. Attackers are given the opportunity to use them to implement DDoS attacks due to insufficient protection of IoT devices, the number and power of which increases with the number of IoT users. The overwhelming majority of users believe that general security rules for the IoT should be developed at the state or interstate level.

However, it is difficult to develop uniform recommendations or standards due to the difference in security requirements depending on the area of use of the IoT. The variety of areas of use is shown in Table 1.

Table 1
Gartner's analysis of the number of IoT devices in use globally, billion

Application area	2018	2019	2020
Housing	0.98	1.17	1.37
Building automation	0.23	0.31	0.44
Security systems	0.83	0.95	1.09

III International Scientific And Practical Conference "Information Security And Information Technologies", September 13–19, 2021, Odesa, Ukraine
EMAIL: muhialdeen.hassan@aliraqia.edu.iq (A. 1); y.khlaponin@gmail.com(A. 2); ibtehal.shaker@aliraqia.edu.iq (A. 3); volodymyr.vyshniakov@gmail.com (A. 4); poltorak_vp@online.ua (A. 5); dmytro.khlaponin85@gmail.com (A. 6); dr.muwafaqalwan@aliraqia.edu.iq (A. 7)
ORCID: 0000-0002-9287-0817 (A. 1); 0000-0002-9287-0817 (A. 2); 0000-0001-8333-461X (A. 3); 0000-0003-4668-712X (A. 4); 0000-0001-9231-9411(A. 5); 0000-0002-7797-4319 (A. 6); 0000-0001-7980-2716 (A. 7)

Extraction of minerals	0.33	0.4	0.49
Automotive	0.27	0.36	0.47
Medicine	0.21	0.28	0.36
Trade	0.29	0.36	0.44
Transport	0.06	0.07	0.08
Government sector	0.4	0.53	0.7

Gemalto's survey of IoT users found that 90% were unsure about security. Thus, it seems to be relevant the analysis of IoT systems from the point of view of ensuring the secure exchange of data over the Internet channels, as well as the technical solutions in this area, given in the work.

2. Analysis of data exchange options in IoT systems

To connect IoT devices to the Internet, one of two schemes can be used, shown in Fig. 1 and Fig. 2 respectively.

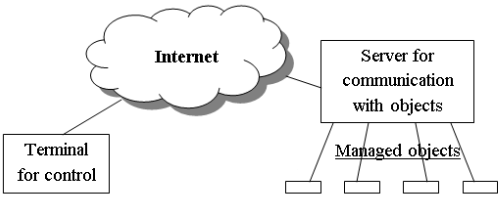

Figure 1: Direct management of objects through the public network

The scheme shown in Fig. 1 is the simplest one and can be successfully used in internal computer networks. But such solution has a number of disadvantages in the conditions of the public Internet:

- Connecting the server directly to the Internet facilitates the intervention of unpredictable external threats into management processes.

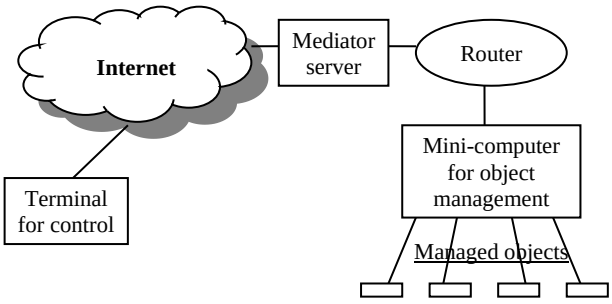

Figure 2: Object management using a mediation server

- Cybercriminals are more likely to install their botnets (malware) on your server to implement DoS and DDoS attacks.

- This server requires a dedicated IP address on the Internet, which is associated with additional material costs.

- To ensure the information security of the server, qualified service is required.

Disadvantages listed above are absent in the circuit shown in Fig. 2, where data flows between the terminal and management objects are filtered by the proxy server. This server can simultaneously serve many users, protecting their data streams from malicious attacks. Internet service providers (ISPs) can install such servers, providing customers with cloud-based access to resources. However, the user can install own separate or corporate broker server in case of high security requirements for information about objects managed. In cases when the broker's server is hit by a threat, the information about the managed objects will be kept intact. An increase in signal latency should be noted as a disadvantage of control through an intermediary server in comparison with the first scheme. But this disadvantage can be considered insignificant, since the performance in control systems cannot be high due to the presence of unpredictable network access latency using Internet channels.

3. Technical solutions to secure the IoT

Object management via the Internet does not require the transfer of large amounts of data and high-speed messaging. This allows you to use the most advanced methods of protecting data from threats of disclosure or spoofing during transmission over channels. The use of such methods makes it possible to exclude the possibility of these threats being realized, which is mathematically provable. It should be noted that no expensive technical solutions are needed for absolute protection. This protection is implemented using simple software methods. It is mathematically proven that the absolute protection of information is provided by the Vernam cipher, which is called one-time pad [5]. The use of this cipher requires the fulfillment of the conditions, the list of which is presented in Table 2.

Table 2

Conditions for ensuring absolute data protection during transmission

Condition	Condition fulfillment
Generation of random bit sequences (not pseudo-random)	A method for random bits generating is implemented, which allows you to generate random sequences on any computer, as described in [3]
Each random bit sequence can be used for encryption only once	For each communication session, random bit sequences are generated independently of each other
For the exchange of random bit sequences, an absolutely secure communication channel should be used	The exchange of random bit sequences occurs according to the Diffie-Hellman algorithm with such parameters for which there is no possibility of data disclosure in modern conditions

The work [7] substantiates the choice of the Diffie-Hellman algorithm parameters. The parameters of the algebraic group for the implementation of the algorithm are selected based on two conditions. In first, it is needed to ensure the impossibility of disclosing data. From the other hand, it is needed that the time of cryptographic transformations does not exceed the allowable value. In order to prevent data disclosure, an algebraic group in the form of a Galois field with characteristic 2 was chosen and a degree, which is a safe prime number from the series 503, 563, 587, 719, was chosen too. Since the solution to the discrete logarithm problem for such fields is unknown today, this protection cannot be hacked in modern conditions. All cryptographic transformations are implemented in the form of several dozen lines in JavaScript and can be copied and placed both in the client and server parts of the software of IoT systems. If the Node.js platform is used to write the server side, then the cryptographic transformations in the server and client sides will be identical. All fragments of the data protection program for a field of (2^503) elements are presented below.

The beginning of filling the array with N random bits looks like this:

```
var N = [504]; // Array of 503 random bits (N [0] is not used)

var T1 = new Date (); // Take the timestamp for transformations

var TN = T1.getTime (); // TN - the number of milliseconds from 01/01/1970

N [1] = TN% 2; // Fill the first bit depending on the parity of TN
```

The rest of the random bits will be formed in the cycle of filling the array MA with powers of the primitive root of the Galois field.

The block for filling an array MA with powers of A looks like this:

```
// Elements of arrays with index 0 are not used

var A = [504]; // Sequence of 503 bits for exponentiation

var B = [504]; // A sequence of 503 bits of the exponent

// Arrays for multiplying the elements of the Galois field GF (2 ^ 503)

var M1 = [504], M2 = [504], R = [504];

// M1 [], M2 [] - factors R [] - the result of multiplication

var MA = new Array (504); // Array MA [] [] of degrees A []

for (var i = 0; i<504; i ++) MA [i] = new Array (504);

for (var i = 1; i<= 503; i ++) MA [1] [i] = A [i];

// The first line of the array was filled with the value A []

for (var I = 2; I <= 503; I ++)

{// Loop filling the array MA [] [] with powers of A []

// In the next 3 lines, we continue filling the array N []

T1 = new Date (); // Take the timestamp for transformations

TN = T1.getTime (); // TN - the number of milliseconds from 01/01/1970

N [I] = TN% 2; // Fill in the next bit depending on the parity of TN

for (var J = 1; J <= 503; J ++) M1 [J] = M2 [J] = MA [I-1] [J];

MULT (); // Function for multiplying the elements of the Galois field GF (2 ^ 503)

for (var j = 1; j <= 503; j ++) MA [I] [j] = R [j];

} // Put degree 2 in MA [2] , put degree 4 in MA [3],

// put degree 8 in MA [4], put degree 16 in MA [5], etc.
```

Our task is to get the same random bit sequences C[] on both sides of the data exchange. This allows to add modulo 2 (XOR operation) bits of the C[] sequence to each bit of data being sent on the transmitting side. With such information coding, absolute protection against disclosure threats in the communication channel is provided. The recipient of the information must add modulo 2 bits of the C[] sequence to the received bits for decryption, which is exactly the same procedure as on the transmitting side.

The transformation process begins by generating a sequence of 503 random bits on each side. This is done simultaneously with filling the array MA[][] with powers of the primitive root of the Galois field. The number 2 is one of primitive roots, which should be entered into the array A[]. In our example, the least significant bits correspond to the lower array indices. Therefore, we get a primitive root like this:

```
for (var i = 1; i<= 503; i ++) A
[i] = 0; A [2] = 1; // Put the
number 2 in A []
```

For raising to a power, a well-known method of simplifying calculations was used, which consists in replacing the operation of raising to a power by a product of powers according to the next expression:

$$A^B = \prod_{i=1}^{503} A^{B_i}, \quad (1)$$

where

$$B = \sum_{i=1}^{503}$$

Since any exponent B can be represented as a sum of values selected from a range of weights 2^0, 2^1, 2^2, 2^3,…, 2^{502}, to calculate AB it is enough to multiply no more than 503 elements from the array MA.

The block for raising A to power B looks like this:

```
for (var i=1; i<=503; i++) A[i]=0;
A[1]=1; // Put a unit in A[]

for (var J=1; J<=503; J++)

if (B[J]==1) // Select the bits
equal to 1 from the binary form of
exponent

{

for (var I=1; I<= 503;
I++){M1[I]=MA[J][I]; M2[I]=A[I];}

MULT(); // Function for
multiplying the elements of the
Galois field GF(2^503)

for (var I=1; I<= 503; I++)
A[I]=R[I];

} // The elements MA[][] was
Multiplied, where B[J]=1.
```

The function of multiplying the elements of the Galois field according to the rule of polynomials looks like this:

```
function MULT()

{ // Multiplication using the
polynomial X^503=X^3+1

var i, j, r, r1, r2, r3;

for (i = 1; i<= 503; i ++) R[i] =
0;

for (i = 1; i<= 503; i ++)

if (M1[i] == 1) // Select units,
because multiplication by 0 gives
0

{

for (j = 1; j <= 503; j ++)

if (M2[j] == 1)

{

r = i + j-1;

if (r> 503)

{

r = r-503;

if (r> = 501)

{

r = r-501;

r1 = 1 + r; r2 = 4 + r; r3 = 501 +
r;

if (R[r3] == 0) R[r3] = 1; else
R[r3] = 0;

}

else {r1 = r; r2 = r + 3;}

if (R[r1] == 0) R[r1] = 1; else
R[r1] = 0;

if (R[r2] == 0) R[r2] = 1; else
R[r2] = 0;

}

else {if (R[r] == 0) R[r] = 1;
else R[r] = 0;}

}
```

```
}
} // End of function MULT ()
```

Let's imagine an algorithm for obtaining bit sequences that will be the same on both sides of the data exchange.

Step 1. The client enters a random bit into the first element of the array N, and enters the value of the primitive root of the Galois field into array A.

Step 2. The client executes the block of filling the array MA with powers of A with the simultaneous completion of filling the array with N random bits.

Step 3. The client copies array N to array B and executes the exponentiation block of A.

Step 4. The client sends to the server the result of raising A to the power of B as a sequence of 503 bits

Step 5. The server stores the sequence of bits received from the client in array C and performs actions similar to steps 1-3 of the client.

Step 6. The server sends to the client its result of raising A to power B.

Step 7. The client stores the sequence of 503 bits received from the server in array A.

Step 8. The client executes the block of filling the array MA with powers of A without filling the array with N random bits.

Step 9. The client executes the block for raising A to the power B and enters the result into array C.

Step 10. The server copies array C to array A and performs the steps similar to steps 8 and 9 of the client.

The result of performing the above actions is to obtain the same random sequences of bits in the arrays C of the same name on the client and server sides, which was required for encryption using the one-time pad method.

4. Full-scale model of a secure IoT system

The main element of the IoT system that needs to be protected from false control commands and from intrusion by attackers who can create threats such as DDoS attacks is computer for object management (see Fig. 2). Connecting this computer through the Router without providing a real IP address does not provide the ability to control this computer other than through the console used to install the software or an application program that provides the protection described in the previous section. A well-known minicomputer of the Raspberry Pi 3 type, which has a 40-pin GPIO interface with wide possibilities for connecting objects for monitoring and control, was chosen as hardware. Linux version Ubuntu 20.10 was selected as the operating system, and the Node.js platform version v12.18.2 with the onoff package was used as a programming tool, which allows objects to be controlled via the GPIO interface.

The initial snippet of the CONPIN.js program installed on this computer in the / home / ubuntu / directory looks like this:

```
const HOST = '91 .198.50.144 ';

const PORT = 3000;

const Gpio = require ('onoff').
Gpio;

const fs = require ('fs');

const net = require ('net');

let SYM; // String.fromCharCode

let STREB = '////////';

let i = 0;

let TR = '';

const Gp4 = new Gpio (4, 'out');
// Pin 7 Gpio_4 # 0

const Gp17 = new Gpio (17, 'out');
// Pin 11 Gpio_17 # 1
```

This client program regularly contacts the server (Mediator server) (see Fig. 2) with a period of 20 seconds to transmit information about the state of objects and receive control signals. The duration of the period of 20 seconds is chosen from the condition of proportionality with the time of entering the Internet. The operation of this program must be protected against possible power outages. To automatically start the program after power-up, add the following three lines to the /etc/rc.local file:

```
#! / bin / sh

echo "#################### CONPIN
######################"

/ usr / bin / node / home / ubuntu
/ CONPIN &
```

The SOCKET.js program must be running on the Mediator server (see Fig. 2) located at the ISP (Internet Service Provider) site that provides services in SaaS (Software as a Service) mode. The initial snippet of this program looks like this:

```
// server / SOCKET.js //
```

```
const HOST = '91 .198.50.144 ';
const PORT = 3000;
const net = require ('net');
const fs = require ('fs');
net.createServer (function (sock)
{
```

With a single intermediary computer with a single real IP address, the provider can serve multiple IoT client systems. The number of supported systems depends only on the technical data of the computer. The operation of the SOCKET.js program must be protected from failures that can lead to an emergency shutdown. To do this, use the process manager pm2 automatic program restart tool, which must be downloaded using the npm install pm2 –g command. After that, the SOCKET.js program should be launched with the pm2 start SOCKET.js command. In this case, in case of any failures, the program will automatically restart [8,13].

The main task of the Mediator server is to protect the resources of IoT systems from the penetration of intruders who have as their goal the implementation of DoS and DDoS attacks. This requires unauthorized entry into the Mediator server, which is unlikely, provided the provider follows standard instructions. Usually this situation arises due to the fault of the provider's staff. In any case of failures on this server, the provider always has the ability to switch to a backup server or restore the operation of the same server using copies, which is the norm in the work of providers [9,12].

The exchange of data between users of the IoT system and their objects is carried out via a web interface through intermediate data files. These files are created anew at each data exchange session. Each individual user on the Mediator server is allocated his own directory, where, in addition to the SOCKET.js program with a unique value for the PORT parameter, the vybir.js program is located, the initial fragment of which looks like this:

```
// vybir.js - HTTP Server Ver. 18 February 2021
var http = require ('http');
var url = require ('url');
var fs = require ('fs');
var static = require ('node-static');
var querystring = require ('querystring');
var file = new static.Server ('.');
http.createServer (function (req, res)
{
```

In the vybir.js program, a separate TCP port number is allocated for each user. The CONPIN.html file with images of object state indicators and control buttons is also located in the user directory. The user can download this file through the link given to him like http://91.198.50.144:8000/CONPIN.html. All communication processes, including the authorization procedure, are protected using the means described in the previous section. The above link is unprotected as it is only intended to demonstrate the control process using eight binary objects as an example. Authorization data is stored in the same directory in an encrypted file.

5. Conclusions

The reasons for the emergence of security problems in IoT systems are described. Potential security threats have been identified, both for the IoT itself and for the use of its resources by intruders in the implementation of attacks on other objects of the Internet.

Variants of data exchange schemes in IoT systems have been analyzed and the choice of the most secure scheme has been substantiated.

The technical solutions that make it possible to secure data exchange in IoT systems by building an ideally secure data exchange channel are considered in detail. These solutions are presented in the form of text programs in the JavaScript language and can be embedded in any user software.

Using the example of the current model of the IoT system, it is shown that it is possible to eliminate problems with emergencies in IoT systems that arise for various reasons, including malfunctions of programs, temporary power outages or attempts to unauthorized entry into the system. A link to a resource on the Internet is provided to demonstrate the process of managing objects.

The technical solutions proposed in this work make it possible to fully secure IoT systems from information threats.

6. References

[1] Orlov S. (2020) Pochemu problem bezopasnosti interneta veshhej okazalos' tak trudno reshit'? https://safe.cnews.ru/articles/2020-05-1_pochemu_problemu_bezopasnosti_interneta

[2] Frustaci, M., Pace, P., Aloi, G., & Fortino, G. (2018). Evaluating critical security issues of the iot world: Present and future challenges. IEEE Internet of Things Journal, 5(4), 2483-2495.

[3] Conti, M., Dehghantanha, A., Franke, K., & Watson, S. (2018). Internet of Things security and forensics: Challenges and opportunities. Future Generation Computer Systems, 78(2), 544-546.

[4] Giray, G., Tekinerdogan, B., & Tüzün, E. (2018). IoT system development methods. In Internet of Things (pp. 141-159). CRC Press/Taylor & Francis.

[5] Shannon C. Communication Theory of Secrecy Systems. Bell System Technical Journal. 1949. 28 (4). Pp. 656–715.

[6] Chupryn V.M. Generuvannja vypadkovyh chisel shtatnymy zasobamy hostiv merezhi Internet./ V.M. Chupryn, V.M.Vyshnjakov, M.P. Prygara // Zahyst informacii'. – 2016. – T. 18, №4. – C. 323-335.

[7] Chupryn V.M., Vyshnjakov V.M., Prygara M.P. Metod protydii' nezakonnomu vplyvu na vyborciv u systemi Internet golosuvannja. Bezpeka informacii'. – 2017. – Tom 23, №1. – C. 7–14.

[8] V.M. Chupryn, V.M.Vyshnjakov, O.O. Komarnyc'kyj, Metod protydii' atakam poserednyka u transparentnij systemi internet golosuvannja, Zahyst informacii', Ukrainian Information Security Research Journal. - K.: NAU, 2018. – T.20. -№3. – C.180-187. http://jrnl.nau.edu.ua/index.php/ZI/article/view/13079

[9] Valentyn Sobchuk, Volodymyr Pichkur, Oleg Barabash, Oleksandr Laptiev, Kovalchuk Igor, Amina Zidan. Algorithm of control of functionally stable manufacturing processes of enterprises. 2020 IEEE 2nd International Conference on Advanced Trends in Information Theory (IEEE ATIT 2020) Conference Proceedings Kyiv, Ukraine, November 25-27. pp.206 –211.

[10] Oleksandr Laptiev, Savchenko Vitalii, Serhii Yevseiev, Halyna Haidur, Sergii Gakhov, Spartak Hohoniants. The new method for detecting signals of means of covert obtaining information. 2020 IEEE 2nd International Conference on Advanced Trends in Information Theory (IEEE ATIT 2020) Conference Proceedings Kyiv, Ukraine, November 25-27. pp.176 –181.

[11] O.Svynchuk, O. Barabash, J.Nikodem, R. Kochan, O. Laptiev. Image compression using fractal functions.Fractal and Fractional, 2021, 5(2), 31.pp.1-14. DOI:10.3390/fractalfract5020031 - 14 Apr 2021.

[12] Androshchuk, A., Yevseiev, S., Melenchuk, V., Lemeshko, O., Lemeshko, V. Improvement of project risk assessment methods of implementation of automated information components of non-commercial organizational and technical systems. EUREKA, Physics and Engineeringthis link is disabled, 2020, 2020(1), pp. 48–55

[13] V. Khoroshko, Y. Khokhlacheva, Y. Khlaponin, E. Gavrilko. Parametric monitoring of computing processes in information and computing systems. Workshop Proceedings (http://ceurws.org) Vol-2067 urn:nbn:de:0074-2067-8-0 P. 45 – 53. – ISSN 1613-0073