Journal of Environmental Protection and Ecology 20, No 2, 934–944 (2019)

Computer application on environmental information systems

MUTUAL AUTHENTICATION PROTOCOL FOR IoT-BASED ENVIRONMENT MONITORING SYSTEM

ZHIHUI WANG[a], JIANLI ZHAO[b], BENZHEN GUO[a], JINGJINGYANG[a], XIAO ZHANG[a]*

[a]*School of Information Science and Engineering, Hebei North University, 07 500 Zhangjiakou, Hebei, China*
[b]*State Grid Hebei Electric Power Research Institute, 050 000 Shijiazhuang, Hebei, China*
E-mail: *wangzhihui@hebeinu.edu.cn*; *3240949676@qq.com*

Abstract. In a rabbit breeding environment monitoring system designed based on the Internet of things (IoT) technology, a safety policy that can realise mutual authentication between client-side APP and IoT data acquisition as well as equipment control is required to ensure the safety of system equipment and message command. Due to the high requirement on the computing power and storage capacity of CPU used for information acquisition and equipment control under the IoT, traditional server/client authentication policy would affect the system development cost. Therefore, a mutual authentication policy was designed in this paper based on elliptic curve cryptography (ECC) with the aid of the system authorised support centre. The experiment proved that this policy could satisfy the requirement of mutual authentication raised by the breeding environment monitoring system under the IoT at a low system development cost.

Keywords: Internet of things (IoT), network security, user authentication, key negotiation, elliptic curve.

AIMS AND BACKGROUND

In a rabbit breeding environment monitoring system designed based on the internet of things technology. There are a large number of devices including the sensors and the environmental intervention controllers used in the rabbit hutches are connected to the internet[1] through the IoT gateway. The sensors will automatically collect the environmental parameters in the rabbit hutch such as the temperature, the humidity and the concentration of harmful gases, which will also be submitted to the system server for storage. The system server will automatically analyse the variation trend of the environmental parameters in the rabbit hutch and send the warning message to the users as soon as an unfavourable change, such as the temperature higher than 30°C (Refs 2 and 3), occurs to affect the normal growth of the rabbits. The users can receive the warning message from the system at any time

* For correspondence.

and anywhere through the system APP or the web pages. Also they can impose a remote intervention to the internal environment of the rabbit hutch. For example, they can turn on the cooling fan to reduce the temperature in the rabbit hutch and control the temperature between 20 and 30°C (Refs 4 and 5) that is favourable to the rabbits. Or they can turn on the ventilator fan to reduce the impact of the adverse environmental factors on the normal growth of the meat rabbits by reducing the ammonia concentration within the rabbit hutch.

Figure 1 shows the perception layer containing various data sensors and controllers that constitute this system and have been distributed in each plant within the breeding farm. The parameter early warning servers distribute at the remote internet and the user terminals construct the application layer. The network layer consists of the IoT gateway located at the entry point to the internet within the breeding farm, the remote system authentication server and the data storage server. As the devices, data and commands from the network layer are completely exposed to the internet, so it becomes extremely important to achieve a high level of equipment safety and information security.

Positioned in the network layer, the IoT gateway serves as an important intermediate to implement the conversion between the field monitoring data protocol and the internet protocol. With a high level of automation, it is transparent to the users and covers two types of core business: (1) receive the data acquired by the environmental parameter acquisition subsystem through the Zigbee network and then submit the received data via WiFi to the system server on the internet or user terminals; and (2) receive commands from the user terminals to impose interference or control on the environmental parameters via WiFi and then transmit the commands to relevant acquisition sensors or control units through the Zigbee network.

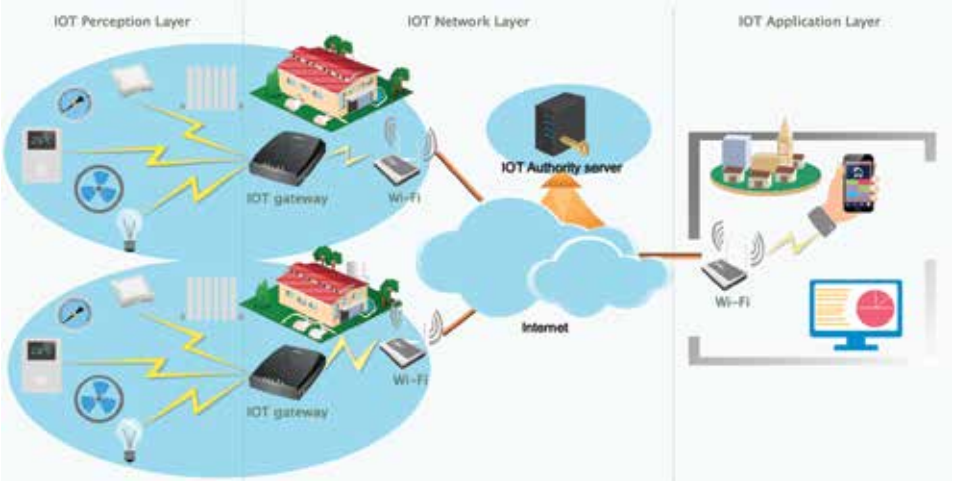

Fig. 1. Architecture diagram of rabbit breeding environment monitoring system

With low requirement on hardware computing power, the core business of gateway can be completely satisfied with a small embedded system. However, during the operation, the gateway may be required to connect directly to the client-side APP through the internet for transmitting control commands or data. At this point, the gateway must have a whole server mechanism to implement user authentication, key exchange, encryption communication and subscriber authentication.

Various authentication schemes for communicating entities that are presented so far are briefly discussed as follows:

Li et al. presented multicast scheme for authentication that also utilises one-time signature in order to overcome the memory overhead and reduce the size of signature. Although computation complexity and authentication delay of the proposed scheme is very low but still it does not resolve the key agreement problem[6]. Soohyun et al. take another step to further improve the security for communications of SG. They proposed a key agreement and mutual authentication scheme for securing communication between intelligent devices and data concentration unit (DCU) (Ref. 7). The mutual authentication is achieved using long term preshared keys (PSK) and corresponding public key certificate of DCU. However, long-term sharing becomes the bottleneck for scalability of the proposed scheme and also makes it impractical to be applicable. Gao et al. attempted to incorporate biometric features like fingerprint for achieving tenacious authentication but it proved out to be non-trivial in terms of its computational complexity[8].

In order to achieve improved security among interacting devices Nicanfar et al. introduced password based authentication using key agreement. Their scheme has the potential to provide forward and backward stealth but again non-trivial operation leads towards hard and expensive implementation[9]. Therefore, in order to reduce computational complexity, Nicanfar et al.[10] presented another scheme using Elliptic Curve Cryptography (ECC). Although use of ECC brought huge amount of reduction in computational complexity but restriction to preload the password between home area network and specific device prevents scalability and introduces overhead of maintaining a table for keeping the repository of password.

In view of this, this paper designed an improved implementation policy based on ECC with the aid of the central authentication server to provide a solution for mutual authentication, key agreement and encrypted data communication between the mobile APP and the IoT gateway.

PREPARATION WORK

Elliptic curves cryptography (*ECC*). ECC is a public key algorithm with the security based on the complexity of solving the elliptic curve discrete logarithm problem (ECDLP) (Ref. 9). Compared with DSA (Digital Signature Algorithm), RSA (Rivest Shamir Adleman) and DH (Diffie-Hellman), ECC has higher encryption

efficiency[10,11] due to the shorter key length at the same security level. The definition of elliptic curve at a prime finite field F_p must satisfy the following equation[12]:

$$E_p(a, b): y^2 \equiv x^3 + ax + b \pmod{p} \quad (1)$$

where p is the large prime number, and the larger the value of p is the higher the security and the huger the calculation amount will be; $a, b \in F_p$ is used to determine the specific elliptic curve and the following equation must be met.

$$4a^3 + 27b^2 \neq 0 \pmod{p} \quad (2)$$

When the point G and the integer t are chosen on the elliptic curve $E_p(a, b)$ meeting the above conditions, $K = tG$, where K is also a point on the elliptic curve $E_p(a, b)$. According to the addition rule: $tQ = Q + Q + Q + ... + Q$, it is easy to obtain T, if G and t are given. However, the value of t can hardly be obtained, if K and G are known. In an ECC, G is a base point, t is a private key and T is a public key.

Encryption and decryption. Figure 2 presents the general process of encryption communication made by both of the communication parties in a typical client/server communication mode based on the elliptic curve. It consists of key exchange and encryption communication, and the specific steps are provided as below:

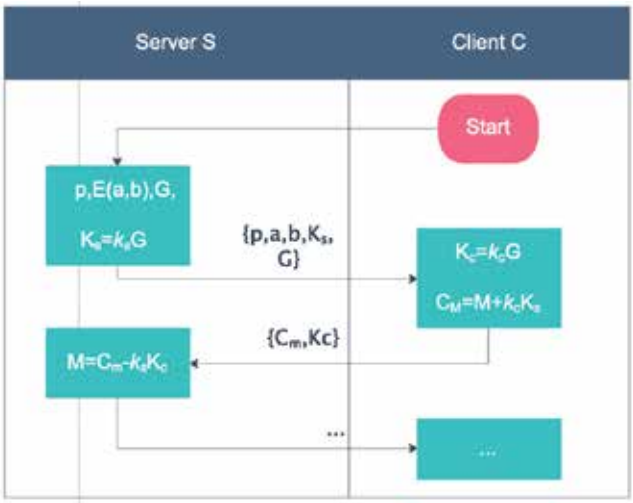

Fig. 2. Elliptic curve-based key agreement and encrypted communication process

Step 1. After receiving a request from the client C to establish a communication link, the server S will choose an elliptic curve $E_p(a, b)$ and take G, a point on the elliptic curve as the base point. Then an integer k_S ($k_S \in [1, p-1]$) will be randomly selected by the server S as the private key for the communication between the server and the client C. Calculate $K_S = k_S G$ and the result will be taken as the public key[13] for the communication between the server and the client C.

Step 2. The server S sends $\{p, a, b, K_S, G\}$ to the client C in cleartext.

Step 3. After the client C receives the message containing the public key from the server S, a random integer k_C will be generated as the private key for the communication between the client C and the server S. Then calculate $C_M = M + k_C K_S$ and $K_C = k_C G$, where M is the cleartext message encoded to a point on the elliptic curve, C_M is the ciphertext message after the encryption of the elliptic curve and K_C is the public key for the communication between the client C and the server S.

Step 4. The client C sends C_M, K_C to the server S, which will calculate $C_M - k_S K_C$ after receiving the message and the result is the cleartext message M. At this point, the key exchange between the server S and the client C is completed and a safe encrypted communication channel is now established.

Step 5. The encrypted data communication will be implemented after the public keys K_C and K_S from the counter party are stored separately by the server S and the client C, and the encrypted data are used.

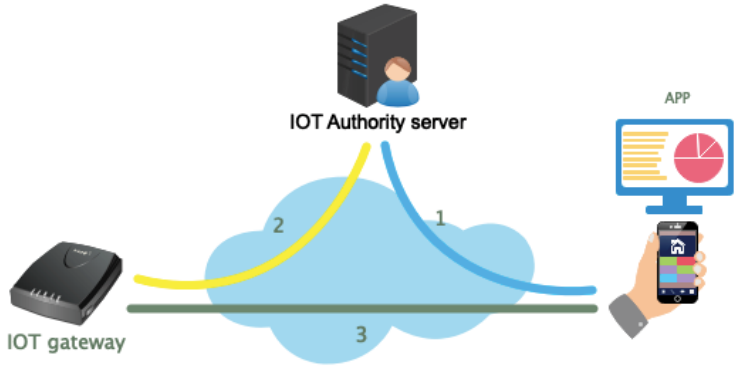

Fig. 3. Data channel of the rabbit breeding environment monitoring system

EXPERIMENTAL

As indicated in Fig. 3, the authentication policy for the IoT-based environment monitoring system consists of three parts. The first part is the authentication policy between the user and the system server via a mobile APP. The second part is the authentication policy between the environmental parameter acquisition equipment, the environmental control equipment and the IoT gateway with the system server. The third part is the mutual authentication policy between the user and the IoT gateway via a mobile App. The authentication between the user and the system server has been performed by a smart phone App developed by this system and it conforms to the traditional forms of internet applications. After the authentication completes, communications between the user terminal and the server will be encrypted with a secret key that has been agreed in the authentication process to

guarantee the information security in both of the later authentication processes involved with the gateway.

Authentication between the gateway and the server. The IoT gateway of the rabbit breeding environment monitoring system is designed to connect to the internet via WiFi. As an automatic smart device, it is transparent to users. However, the safety of this device also affects the overall security of the field subsystem in the breeding environment. Hence, the core task for the authentication between the gateway and the server system is to: (1) prevent any unauthorised device from connecting to the system; (2) prevent any unauthorised local user from initialising local devices; and (3) prevent the controlling of local gateway and devices through remote server impersonation. Considering that when the gateway is connected to the internet via WiFi, some basic information such as WiFi ssid, WiFi password, the server IP address and the domain name must be set up. Therefore, when the authentication is performed for the first time between the gateway and the server, mutual authentication must be initialised with the aid of the client-side APP, and the specific steps are provided as below:

Step 1. After the user completes the server authentication operation, a safe encrypted communication channel can be established to submit the initial request from the gateway and the basic gateway message M_{gw} to the server. M_{gw} = {Gateway ID, Wi-Fi SSID, Wi-Fi pwd, ClientAPP ID}.

Step 2. After receiving the initial gateway message from the client-side APP, the authentication server will extract the gateway ID from the message to perform the authentication. If the user has the authority to operate the target gateway, the following processes must be completed:

(1) Turn M_{gw} into M_{sg} after the server IP address, the timestamp, the validity, etc., are added. M_{sg} = {Gateway ID, SSID, Wi-Fi password, ClientAPP ID, Server IP or domain name, timestamp, validity}.

(2) The server calculates $C_{sg} = M_{sg} + k_{s\text{-}g} K_{g\text{-}s}$ according to the server-gateway private key $k_{s\text{-}g}$ and the gateway-server public key $K_{g\text{-}s}$ agreed by default for the gateway.

(3) According to the private key agreed with the APP, the server takes C_{sg} as the cleartext message for the secondary encryption and sends it to the client-side APP.

Step 3. After receiving the message from the server, the client-side APP can be connected to the gateway and send the encrypted message C_{sg} to the target gateway. According to the gateway-server private key $k_{g\text{-}s}$ and the server-gateway public key $K_{s\text{-}g}$ agreed by default, the gateway then calculates $C_{sg} - k_{g\text{-}s} K_{s\text{-}g}$ to extract M_{sg} for the verification of the validity and the gateway ID. The authentication continues if the verification is legal.

Step 4. According to the SSID and WiFi pwd contained in the message M_{sg}, the gateway can realise the configuration of the local wireless network and can be connected to the server based on the server IP address. After the initial authentica-

tion between the gateway and the server is implemented according to the key pair agreed by default, a safe encrypted communication channel can be established.

Mutual authentication between the user and the gateway. After the gateway is in a normal operation mode, a direct communication link can be established between the client-side APP and the gateway to send commands for data query or environmental intervention (such as remote fan control, wet pad cooling control, etc.). At this point, a safe communication channel must be established between the user and the gateway to prevent the APP control commands from being simulated, tampered or replayed. As the connection or disconnection between the user and the gateway happens frequently and the key generation algorithm requires lots of computing resources, this paper presents an improved solution by transferring some operations, including curve selection, key generation, key agreement, user authentication and authentication that should have been implemented by the gateway to the system server. It only leaves the gateway to serve for the encryption communication with an appointed gateway based on the password agreed with the server. Figure 4 describes the basic processes for implementing the mutual authentication policy designed in this paper between the user and the gateway with the aid of the authentication server of the system.

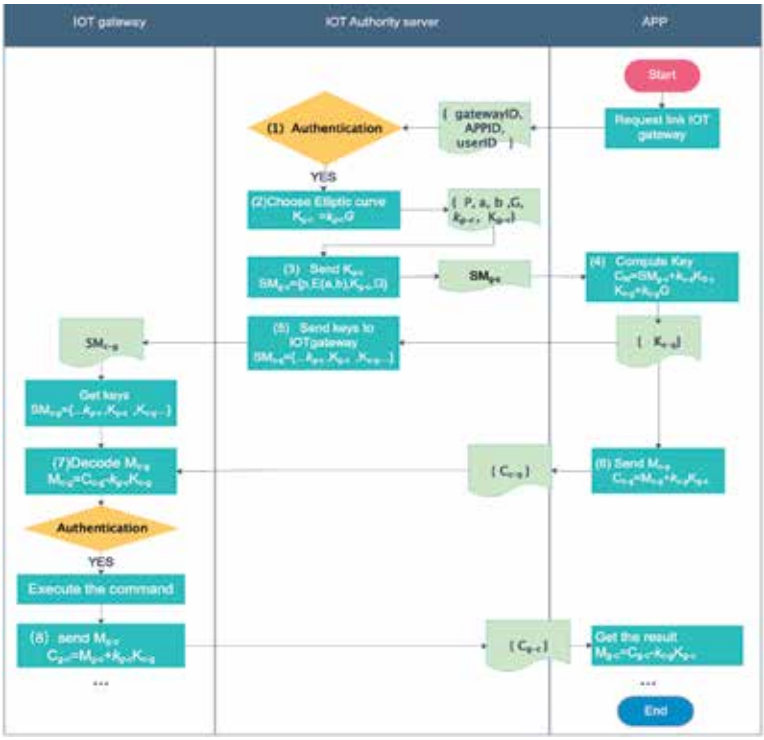

Fig. 4. Mutual authentication processes between the gateway and the user

Step 1. The user sends a request to the server for an access to the gateway equipment via the client-side APP. After the system server verifies the validity and the privilege of the user ID, it will move to the next step.

Step 2. Acting as a gateway, the system server chooses $E(a, b)$, the prime P and the base point G to generate the gateway-user private key $k_{g\text{-}c}$ and the gateway-user public key $K_{g\text{-}c}$.

Step 3. After the gateway-user key exchange message $SM_{g\text{-}c} = \{p, E(a, b), K_{g\text{-}c}, G, h$ (gateway ID, user ID, Timestamp)$\}$ is constructed by the server, it will be sent to the client-side APP through the encrypted channel between the client and the server.

Step 4. After the client-side APP receives $SM_{g\text{-}c}$, the client-gateway private key $k_{c\text{-}g}$ and the client-gateway public key $K_{c\text{-}g}$ will be generated. Then the public key $K_{c\text{-}g}$ will be sent to the authentication server.

Step 5. After the authentication server receives the client-gateway public key $K_{c\text{-}g}$, the client-gateway key exchange message $SM_{c\text{-}g} = \{$APP ID, $k_{g\text{-}c}$, $K_{g\text{-}c}$, $K_{c\text{-}g}$, timestamp, validity, h (gateway ID, user ID, Timestamp)$\}$ will be constructed and sent to the corresponding gateway, which then can be accessed by the client after $SM_{c\text{-}g}$ is received.

Step 6. After $M_{c\text{-}g} = \{h$ (gateway ID, user ID, Timestamp), gateway ID, command$\}$ is constructed by the client-side APP, calculate $C_{c\text{-}g} = M_{c\text{-}g} + k_{c\text{-}g} K_{g\text{-}c}$ and send $C_{c\text{-}g}$ to the target gateway.

Step 7. After the gateway receives $C_{c\text{-}g}$, calculate $C_{c\text{-}g} - k_{g\text{-}c} K_{c\text{-}g}$ to obtain $M_{c\text{-}g}$, which can be used to verify the server authentication certificate, the timestamp and the validity. Command can be executed, if the verification is legal.

Step 8. Construct $M_{g\text{-}c} = \{$gateway ID, command result, timestamp, h (gatewayID, userID, Timestamp)$\}$ to calculate $C_{g\text{-}c} = M_{g\text{-}c} + k_{g\text{-}c} K_{c\text{-}g}$, and directly send $C_{g\text{-}c}$ to the client-side APP.

Through the above steps and with the aid of the authentication server, an encrypted communication channel can be established and mutual authentication can be implemented between the client and the gateway.

RESULTS AND DISCUSSION

As indicated in Fig. 1, the system architecture diagram of the rabbit breeding environment monitoring system constructs the experimental test platform of the system. The server cluster has been connected to the internet through the 10 MP optical fibre line with a fixed IP and the mobile phone installed with Android system serves as the client platform. The IoT gateway has been constructed on the low-power Cortex®-M4-based STM32F412 with 100 MHz CPU to implement the user authentication protocol designed in this paper. Three sets of IoT gateways and perception subsystems have been deployed separately in Xuanhua Yongli Rabbit

Breeding Farm, Huai'an Guoqiang Rabbit Farm and the experimental rabbit breeding base of Northern College. Both of the client-side APP and the gateway have been connected to the internet via ADSL and positioned behind the NAT device with a dynamically changed IP address.

Figure 5 presents the client authentication processes performed by the gateway with the aid of the system server. The first picture shows the list of gateways authorised by the client-side APP for the direct connection. The second picture shows a prompt that the user is not allowed to query the current data of the designated gateway. The third picture presents the current data inquired from the designated gateway through the direct connection. The fourth picture is a prompt from the system for fan startup after the user performs the fan startup operation. The fifth picture presents a prompt for the operation timeout on the client-side APP after the gateway discards the illegal request.

Fig. 5. Client authenticated processes

This section analyses the security of the improved mutual authentication protocol between the gateway and the user with the aid of the server in an intruder model assumed in Dolev-yao model[14].

Analysis of anti-eavesdropping. The communication channel between the server and the gateway or between the server and the client is a safe encrypted channel in the communication process. Therefore, eavesdroppers can hardly intercept or acquire the real content of the information. Although monitors might obtain the public key $K_{c\text{-}g}$ when the communication channel is established between the client and the gateway, they would never get access to the private key k. Even though the classified information C might be acquired, they still could never obtain the real information. Therefore, this protocol can prevent eavesdropping.

Analysis of anti-replay. Replay attack is an attack means taken by the eaves-dropper by replaying the previous information to falsely obtain trust after the eavesdropper illegally acquires the communication information through the network. However, as the information, such as the timestamp, the validity, the sequence number, etc. has been added into the system message, this protocol can prevent replay.

Impersonation attack. If the intruders want to disguise themselves as a legitimate user to communicate with the gateway, they must fabricate the valid authentication information. The analysis as above reveals that only the user designated by the server can have server authentication certificates. Therefore, this protocol can prevent the impersonation attack.

Man-in-the-middle attack. Since the intruders cannot disguise themselves as the other user to make the communication, nor could they fabricate the valid authentication information, then this protocol can withstand the man-in-the-middle attack.

CONCLUSIONS

This paper analysed the requirement of safety policy implemented to the IoT gateway devices and the equipment on the network layer in the rabbit breeding environment monitoring system. Taking the computing power of the gateway hardware into full account and based on the system design cost, an improved solution for the internet authentication policy was proposed in this paper based on the traditional public key mechanism. With the aid of the authentication server, this solution was designed as a strategy for the mutual authentication between the IoT gateway and the user, the key exchange and the encrypted communication. Finally, an experimental environment was built to prove that this authentication policy is applicable to achieving the design indicators of the IoT gateway on a hardware platform, where the computing power of gateway is limited.

Acknowledgements. The work was supported by the Population Health Information in Hebei Province Engineering Research Centre, Zhangjiakou Programs for Science and Technology Development by Zhangjiakou Municipal Science and Technology and Seismological Bureau (Grant:1711038C) Zhangjiakou Municipal Science and Technology and Seismological Bureau by the development and research of monitoring system of the host computer in the local area network (Grant:0801113B).

REFERENCES

1. L. ATZORI, A. IERA, G. MORABITO: The Internet of Things: a Survey. Comput Netw, **54**, 2787 (2010).
2. J. K. SEREM, M. M. WANYOIKE, C. K. GACHUIRI, S. K. MAILU, P. K. GATHUMBI et al.: Characterization of Rabbit Production Systems in Kenya. Journal of Agricultural Science and Applications, **2**, 5 (2013).
3. Ye. JINGZHONG, W. YIHUAN, L. NORMAN: Farmer Initiatives and Livelihood Diversification: from the Collective to a Market Economy in Rural China. J Agrar Change, **9** (2), 175 (2009).
4. L. CARTUCHE, M. PASCUAL, E. A. GÓMEZ, A. BLASCO: Economic Weights in Rabbit Meat Production. World Rabbit Sci, **22** (3), 165 (2014).
5. S. O. APORI, J. K. HAGAN, D. OSEI: The Growth and Reproductive Performance of Different Breeds of Rabbits Kept under Warm and Humid Environments in Ghana. Online J Anim Feed Res, **4** (3), 51 (2014).

6. LI QINGHUA, C. GUOHONG: Multicast Authentication in the Smart Grid with One-time Signature. IEEE T Smart Grid, **2** (4), 686 (2011).
7. S. OH, J. KWAK: Mutual Authentication and Key Establishment Mechanism Using Dcu Certificate in Smart Grid. Appl Math Inform Sci, **6** (1S), 257S (2012).
8. Q. GAO: Biometric authentication in Smart Grid. In: Proceedings of the 2012 International Energy and Sustainability Conference (IESC), 2012, p. 1.
9. H. NICANFAR, V. C. M.LEUNG: Password-authenticated Cluster-based Group Key Agreement for Smart Grid Communication. Secur Commun Netw, **7** (1), 221 (2014).
10. H. NICANFAR, V. C. M. LEUNG: Multilayer Consensus ECC-based Password Authenticated Key-exchange (MCEPAK) Protocol for Smart Grid System. IEEE T Smart Grid, **4** (1), 253 (2013).
11. D. LI, Z. AUNG, J. R.WILLIAMS, A. SANCHEZ: Efficient and Fault-diagnosable Authentication Architecture for AMI in Smart Grid. Secur Commun Netw, **8** (4), 598 (2015).
12. N. KOBLITZ, A. MENEZES, S. VANSTONE: The State of Elliptic Curve Cryptography. Designs, Codes and Cryptography, **19** (2–3), 173 (2000).
13. M. BURROWS, R. M. NEEDHAM: A Logic of Authentication. Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences, **426** (1871), 233 (1989).
14. K. MAHMOOD, S. A. CHAUDHRY, H. NAQVI, S. KUMARI, X. LI, A. K. SANGAIAH: An Elliptic Curve Cryptography Based Lightweight Authentication Scheme for Smart Grid Communication. Future Gener Comp Sy, **81**, 557 (2018).

Received 26 April 2019
Revised 26 May 2019